НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А.Н. Тихонова

На правах рукописи

НЕСТЕРЕНКО АЛЕКСЕЙ ЮРЬЕВИЧ

Математические методы обеспечения защищенного взаимодействия средств защиты информации

Диссертация на соискание ученой степени доктора физико-математических наук

специальность 2.3.6
«Методы и системы защиты информации, информационная безопасность»

Научный консультант
д.ф.-м.н.
Чирский Владимир Григорьевич

Москва — 2023 г.

Оглавление

Основные обозначения

В тексте диссертационной работы будут использованы следующие основные обозначения.

$\mathbb{N}$ - множество натуральных чисел,
$\mathbb{N}_0$ - множество целых неотрицательных чисел,
$\mathbb{Z}$ - кольцо целых рациональных чисел,
$\mathbb{Z}_m$ - кольцо вычетов по модулю $m \in \mathbb{N}$, $m > 1$,
$\mathbb{Z}_m^*$ - группа обратимых элементов кольца $\mathbb{Z}_m$,
$\mathbb{Q}$ - поле рациональных чисел,
$\mathbb{Q}(\sqrt{-d})$ - поле мнимых квадратичных иррациональностей, где d – натуральное число, свободное от квадратов,
$\mathrm{sign}(x)$ - функция, возвращающая знак действительного числа x,
$\mathbb{C}$ - поле комплексных чисел,
$\mathbb{C}_+$ - верхняя комплексная полуплоскость,
$\mathbb{Z}_{\mathbb{K}}$ - кольцо целых алгебраических чисел поля $\mathbb{K}$,
$[\mathbb{H} : \mathbb{Q}]$ - степень расширения поля алгебраических чисел $\mathbb{H}$ над полем рациональных чисел $\mathbb{Q}$,
Λ_τ - решетка, образованная базисом $\{1, \tau\}$, где $\tau \in \mathbb{Q}(\sqrt{-d})$,
$\mathbb{F}_p$ - конечное простое поле характеристики p, p – нечетное простое число,
$\mathbb{F}_p^*$ - мультипликативная группа поля $\mathbb{F}_p$,
$|\mathbb{K}|$ - мощность (количество элементов) конечного множества $\mathbb{K}$,
$SL_2(\mathbb{Z})$ - группа квадратных матриц размера 2×2 с целыми коэффициентами и определителем, равным 1,
$GL_m(\mathbb{K})$ - группа квадратных матриц размера $m \times m$ с коэффициентами из поля $\mathbb{K}$,
S_m - симметрическая группа перестановок на конечном множестве из m элементов,
$\mathcal{E}_{a,b}(\mathbb{K})$ - эллиптическая кривая, заданная коэффициентами $a, b \in \mathbb{K}$ над произвольным полем $\mathbb{K}$,
$\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$ - эллиптическая кривая, заданная решеткой Λ_τ над полем комплексных чисел,
$\mathcal{O}$ - бесконечно удаленная точка эллиптической кривой,
$\mathtt{ord}\, P$ - порядок точки P эллиптической кривой,
$\mathbb{V}_n(\mathbb{K})$ - векторное пространство над полем $\mathbb{K}$ векторов длины n,
$\mathbb{V}_n$ - векторное пространство двоичных векторов длины n,
$\mathbb{V}_\infty(\mathbb{K})$ - векторное пространство над полем $\mathbb{K}$ векторов произвольной, конечной длины,

$\mathbb{V}_\infty$ - векторное пространство двоичных векторов произвольной, конечной длины,

$\mathbb{B}$ - булево множество, элементы которого принимают значения «истина» (***true***) или «ложь» (***false***),

$||$ - операция конкатенации двух векторов,

$\mathrm{len}_2(x)$ - длина двоичного вектора $x \in \mathbb{V}_\infty$,

$\mathrm{len}_{\mathbb{K}}(x)$ - длина вектора $x \in \mathbb{V}_\infty(\mathbb{K})$,

$\mathrm{msb}_w(x)$ - старшие w разрядов вектора $x \in \mathbb{V}_\infty(\mathbb{K})$,

$\mathrm{lsb}_w(x)$ - младшие w разрядов вектора $x \in \mathbb{V}_\infty(\mathbb{K})$,

0^n - двоичная последовательность длины n, состоящая из одних нулей,

$\{x, y, ..\}$ - множество, состоящее из элементов $x, y, \ldots$,

$a \in_R \mathbb{K}$ - символ, обозначающий, что элемент a выбирается случайно, равновероятно из конечного множества $\mathbb{K}$,

$x =^? y$ - функция, возвращающая «истину» в случае, если значения x и y совпадают, и «ложь» – в противном случае,

$\xleftarrow{\xi_1}$

$\xrightarrow{\xi_2}$ - процесс передачи величин ξ_1, ξ_2 по каналу связи,

$\square$ - символ завершения доказательства.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Диссертация посвящена решению проблемы построения и математического обоснования безопасности криптографических протоколов, применяемых для обеспечения защищенного обмена информацией по открытым каналам связи.

Решение данной проблемы является важным в теоретическом и практическом отношении для разработки отечественных средств, применяемых для защиты информационных систем, информационно-телекоммуникационных сетей связи, автоматизированных систем управления, а также, для защиты критической информационной инфраструктуры Российской Федерации.

В диссертационной работе разработан математический аппарат, позволяющий строить криптографические протоколы и их формализованные модели на основе предъявляемых требований по безопасности.

Использование формализованных моделей позволяет свести задачу оценки безопасности криптографического протокола к определению трудоемкости решения ряда сложных математических задач, в частности, задачи дискретного логарифмирования, задаче определения начального заполнения генератора псевдослучайных последовательностей, задаче построения коллизии для сжимающего отображения и т.п. В диссертационной работе рассматриваются способы решения указанных задач, а также методы выбора параметров криптографических протоколов при которых рассматриваемые задачи оказываются трудноразрешимыми.

Диссертация представляет результаты исследований в области математических проблем информационной безопасности. Тема, объект и предмет исследований диссертации соответствуют паспорту научной специальности 2.3.6 «Методы и системы защиты информации, информационная безопасность» (физико-математические науки) по следующим областям исследований:

- теория и методология обеспечения информационной безопасности и защиты информации;
- методы, аппаратно-программные средства и организационные меры защиты систем (объектов) формирования и предоставления пользователям информационных ресурсов различного вида;
- модели и методы оценки защищенности информации и информационной безопасности объекта;
- технологии идентификации и аутентификации пользователей и субъектов информационных процессов;

- принципы и решения (технические, математические, организационные и др.) по созданию новых и совершенствованию существующих средств защиты информации и обеспечения информационной безопасности;

- исследования в области безопасности криптографических алгоритмов, криптографических примитивов и криптографических протоколов.

Актуальность темы. Необходимость проведения научных исследований в целях создания перспективных средств обеспечения информационной безопасности определяется Доктриной информационной безопасности Российской Федерации. Области применения результатов таких исследований определяются Федеральным законом «О безопасности критической информационной инфраструктуры Российской Федерации», постановлением Правительства РФ о реализации государственной программы «Информационное общество», а также рядом приказов отраслевых министерств и ведомств.

В подавляющем большинстве случаев использование средств обеспечения информационной безопасности возможно только после проведения процедуры их сертификации и проверки выполнения требований, предъявляемых ФСТЭК и ФСБ России. При проведении указанной процедуры важным фактором является получение обоснованных оценок показателей мер защиты, реализуемых средствами защиты информации и, в частности, входящими в их состав криптографическими протоколами. Высокий уровень защиты информации, обеспечиваемый криптографическими протоколами, является необходимым фактором для обоснования безопасности передаваемой информации.

При передаче информации ее безопасность обеспечивается, как правило, совокупностью нескольких криптографических протоколов:

- протоколом односторонней, взаимной или многосторонней аутентификации участников информационного взаимодействия,

- протоколом выработки общей для участников взаимодействия ключевой информации, действующей в рамках одной сессии информационного взаимодействия,

- транспортным протоколом, предназначенным для передачи защищенной информации по каналам связи,

- процедурами выработки производной ключевой информации, контроля за временем и объемом используемой ключевой информации,

- вспомогательными протоколами, предназначенными для передачи ошибок информационного взаимодействия, квитирования абонентов, инициализации процедуры выработки нового сессионного ключа и т.п.

Указанной совокупностью протоколов ограничивается область диссертационных исследований. Выбор конкретного криптографического протокола, применяемого в средстве защиты информации, проводится с учетом большого числа факторов, к которым могут быть отнесены — число участников информационного взаимодействия, объем передаваемой информации, срок действия информационной системы или телекоммуникационной сети связи, свойства открытого канала связи, по которому передается информация, тип используемой ключевой информации и т.д.

Различные эксплуатационные требования к применяемым криптографическим протоколам приводят к необходимости разработки как универсальных решений и включения таких решений в действующую в Российской Федерации систему стандартизации, так и разработки частных решений, специализированных для конкретных условий эксплуатации. Однако во всех случаях необходимо обеспечение одинокого высокого уровня защиты информации.

Цель диссертационной работы заключается в совершенствовании методов построения криптографических протоколов, применяемых для обеспечения защищенного обмена информацией по открытым каналам связи, а также методов получения обоснованных оценок безопасности криптографических протоколов.

Для достижения поставленной цели были решены следующие актуальные и трудные математические задачи:

- уточнения трудоемкости нахождения дискретных логарифмов в группах точек эллиптических кривых и построения параметров эллиптических кривых, обладающих заданной трудоемкостью решения задачи дискретного логарифмирования;

- выработки псевдослучайных последовательностей, удовлетворяющих предъявляемым требованиям по безопасности;

- построения режима работы блочных шифров, реализующего аутентифицированное шифрование;

- оценки численных значений показателей эффективности мер защиты для криптографических протоколов, используемых в средствах защиты информации.

Степень разработанности темы. В диссертации решены актуальные и трудные математические задачи, представляющие исключительную важность для обеспечения защищенного взаимодействия средств защиты информации. В рамках вводных разделов диссертации к каждой главе диссертации представлен исчерпывающий обзор предшествующих результатов по теме исследования, наиболее важные из которых указаны далее в разделе «Краткое содержание работы».

Научная новизна. В диссертационной работе получены следующие новые результаты.

1. Получена верхняя оценка числа шагов алгоритма Госпера, используемого для поиска двух совпадающих элементов числовых последовательностей. Автором предложен метод дискретного логарифмирования в группе точек эллиптической кривой, в основе которого лежит алгоритм Госпера, и получена асимптотическая оценка трудоемкости предложенного метода. Соответствие асимптотической оценки получаемым на практике значениям подтверждено результатами практических экспериментов на ЭВМ.

2. Доказана теорема о существовании алгоритма дискретного логарифмирования в группе точек эллиптической кривой, использующего информацию о мультипликативном порядке неизвестного. Получены точные оценки трудоемкости такого алгоритма и объема используемой им памяти. Предложено два различных способа реализации рассматриваемого алгоритма, позволившие снизить объем используемой памяти и практически реализовать алгоритм на ЭВМ. Теоретические оценки трудоемкости алгоритма подтверждены результатами практических экспериментов на ЭВМ.

3. Введено понятие «слабого» ключа и определено значение средней трудоемкости алгоритма дискретного логарифмирования в группе точек эллиптической кривой. Получено точное количество «слабых» ключей для эллиптических кривых, параметры которых рекомендованы Р 1323565.1.024-2019 для использования в средствах защиты информации.

4. Предложен алгоритм вычисления явного представления эндоморфизмов эллиптических кривых. Предъявлены ранее не известные эндоморфизмы для всех эллиптических кривых, чье кольцо эндоморфизмов изоморфно порядку мнимого квадратичного поля с числом классов равным единице.

5. Построены формы эллиптических кривых, обеспечивающие минимальную трудоемкость вычисления предъявленных эндоморфизмов. Доказана теорема о представлении натуральных чисел значениями многочленов в точках мнимого квадратичного поля. Предложен способ применения доказанной теоремы для реализации нового алгоритма вычисления кратной точки на эллиптической кривой.

6. Предъявлены усиленные, по сравнению с ГОСТ Р 34.10-2012, требования к параметрам эллиптических кривых, рекомендуемых к применению в средствах защиты информации. Предложен алгоритм построения таких эллиптических кривых. Приведены явные значения параметров построенных эллиптических кривых, доказывающие возможность достижения предъявленных требований.

7. Доказана теорема об иррациональности значений действительных чисел, определяемых рядами специального вида. Предложены новые алгоритмы представления действительных чисел специального вида в виде систематической дроби по произвольному основанию и способ применения предложенных алгоритмов для выработки псевдослучайных последовательностей. Получены верхние оценки объема памяти, необходимого для реализации предложенных алгоритмов.

8. Доказана теорема об оценках неизвестных коэффициентов действительных чисел специального вида. Автором работы предложены алгоритмы восстановления неизвестных коэффициентов по известному рациональному приближению числа специального вида. Доказаны утверждения о невозможности применения предложенных алгоритмов для построения более точных рациональных приближений.

9. Предложен метод локальной аутентификации пользователей средств защиты информации, основанный на алгоритме представления действительных чисел специального вида в виде систематической дроби по произвольному основанию.

10. Определен новый класс ключевых функций хэширования, представляющих собой линейные формы от перестановок на множестве кодов аутентификации. Доказаны теоремы о том, что функции из данного класса являются равновероятными функциями относительно сжимаемых сообщений и строго равновероятными функциями относительно множества ключей.

11. Предложен режим аутентифицированного шифрования, в основе которого лежит построенный класс равновероятных ключевых функ-

ций хэширования. Доказана теорема о выполнении свойства равновероятности для сжимающего отображения предложенного режима при фиксированных ключах шифрования и аутентификации. Приведены результаты практической реализации предложенного режима, показывающие его преимущество в скорости при программной реализации над регламентированными в Российской Федерации алгоритмами аутентифицированного шифрования.

12. Построена гибридная схема и ряд ее модификаций, реализующих процесс шифрования с помощью полиномиального преобразования. Определена модель возможностей нарушителя и, в этой модели, доказана теорема о стойкости предложенной схемы шифрования относительно задач определения секретного ключа аутентификации, дешифрования и навязывания сообщений. Предложен протокол передачи ключевой информации, основанный на использовании рассматриваемой гибридной схемы шифрования.

13. Предложен новый протокол выработки общего ключа со взаимной аутентификацией субъектов взаимодействия. Доказана теорема о стойкости предложенного протокола относительно задач определения общего ключа, дешифрования и навязывания передаваемой в ходе выполнения протокола информации. Предложено семейство криптографических протоколов, предназначенное для обеспечения защищенного взаимодействия в сетях «Интернета вещей».

14. Автором предъявлена формальная модель, имитирующая криптографический протокол в виде дискретной динамической системы. В рамках данной модели формализован перечень свойств безопасности и определены показатели эффективности мер защиты, обеспечиваемых криптографическим протоколом. Для получения численных значений показателей эффективности мер защиты предложен метод, использующий оценки трудоемкости компрометации криптографических преобразований, изменяющих состояния дискретной динамической системы.

15. Предложена методика проведения исследования безопасности криптографических протоколов.

Теоретическая значимость работы. Результаты исследования развивают методы оценки безопасности средств защиты информации, использующих математический аппарат эллиптических кривых.

Автором найден нетривиальный алгоритм решения задачи дискретного логарифмирования в группе точек эллиптической кривой, трудоем-

кость которого зависит от разыскиваемого неизвестного значения. Это привело не только к корректировке методов выбора параметров эллиптических кривых, но и к необходимости выработки секретных ключей криптографических схем и протоколов из множества значений, на которых найденный алгоритм имеет максимальную трудоемкость.

Предложенный автором способ выработки псевдослучайных последовательностей удовлетворил требованиям, традиционно накладываемым на криптографические датчики случайных чисел, а также обеспечил ряд дополнительных эксплуатационных характеристик, например, существенно затруднил реализацию датчиков на программируемых логических интегральных схемах.

Разработанный в диссертации новый класс ключевых сжимающих отображений, а также свойства, которыми данный класс обладает, позволили не только разработать несколько новых режимов аутентификационного шифрования для блочных шифров, но и обеспечить возможность высокоэффективной реализации данных режимов на вычислительных средствах с различной архитектурой.

Разработанная в работе методика проведения исследований безопасности криптографических протоколов является на настоящий момент единственным математически обоснованным документом, позволяющим не только проводить комплексные исследования всех факторов, влияющих на безопасность криптографического протокола, но и получать численные значения показателей мер защиты информации.

Практическая значимость работы. Результаты диссертационных исследований автора были использованы при подготовке положительных заключений о возможности применения ряда государственных стандартов и рекомендаций по стандартизации в области криптографической защиты информации, в частности, ГОСТ Р 34.10-2012 «Процессы формирования и проверки электронной цифровой подписи», Р 1323565.1.004-2017 «Схемы выработки общего ключа с аутентификацией на основе открытого ключа», Р 1323565.1.018-2018 «Криптографические механизмы аутентификации в контрольных устройствах для автотранспорта», Р 1323565.1.024–2019 «Параметры эллиптических кривых для криптографических алгоритмов и протоколов», Р 1323565.1.028–2019 «Криптографические механизмы защищенного взаимодействия контрольных и измерительных устройств».

Результаты выносимые на защиту.

- Доказательство теоремы 1.2 об оценке числа шагов алгоритма Госпера, используемого для поиска двух совпадающих элементов чис-

ловых последовательностей.

- Алгоритм решения задачи дискретного логарифмирования в группе точек эллиптической кривой, основанный на методе Госпера и асимптотическая оценка сложности данного алгоритма.

- Доказательство теоремы 1.3 об алгоритме дискретного логарифмирования в группе точек эллиптической кривой, использующего информацию о мультипликативном порядке неизвестного, а также точные оценки трудоемкости такого алгоритма и объема используемой им памяти.

- Два варианта (однопоточный и параллельный) алгоритма решения задачи дискретного логарифмирования в группе точек эллиптической кривой, использующего информацию о мультипликативном порядке неизвестного.

- Алгоритм вычисления явного представления эндоморфизмов эллиптических кривых, а также явный вид эндоморфизмов для всех эллиптических кривых, чье кольцо эндоморфизмов изоморфно порядку мнимого квадратичного поля с числом классов равным единице.

- Доказательство теоремы 1.6 о представлении натуральных чисел значениями многочленов в точках мнимого квадратичного поля, и алгоритм вычисления кратной точки эллиптической кривой, основанный на утверждении доказанной теоремы.

- Алгоритм построения эллиптических кривых, удовлетворяющих усиленным, по сравнению с ГОСТ Р 34.10-2012, требованиям к параметрам эллиптических кривых, а также явные значения построенных параметров.

- Доказательство теоремы 2.1 об иррациональности действительных чисел, определяемых рядом $\alpha = \sum_{k=0}^{\infty} \frac{x_k}{k!}$ для периодической последовательности рациональных чисел $(x_k)_{k=0}^{\infty}$.

- Доказательство теоремы 2.3 об оценке неизвестных натуральных значений $x_1, \ldots, x_m$, участвующих в определении действительного числа $\alpha = \sum_{n=0}^{\infty} \sum_{i=1}^{m} \frac{u_i b^{-n}}{dn+x_i}$, при известных значениях $b, d, u_1, \ldots, u_m$ и известном рациональном приближении к α.

- Алгоритм вычисления неизвестных элементов периодической последовательности $(x_k)_{k=0}^{\infty}$, определяющих действительное число $\alpha = \sum_{k=0}^{\infty} \frac{x_k}{k!}$, при известном рациональном приближении к α.

- Доказательство утверждений (см. теоремы 2.4 и 2.5) о совпадении разложений в систематическую дробь, а также доказательство критерия (см. теорему 2.6) нормальности действительных чисел из рассматриваемых классов.

- Алгоритм преобразования парольной информации, используемый для локальной аутентификации пользователей средств защиты информации.

- Новый класс ключевых функций хэширования, представляющих собой линейные формы от перестановок на множестве кодов аутентификации. Доказательство теорем 3.1, 3.2 и 3.3 о том, что функции из данного класса являются равновероятными функциями относительно сжимаемых сообщений и строго равновероятными функциями относительно множества ключей.

- Режим аутентифицированного шифрования и доказательство теоремы 3.4 о выполнении свойства равновероятности для сжимающего отображения предложенного режима при фиксированных ключах шифрования и аутентификации.

- Гибридная схема, реализующая процесс шифрования с помощью полиномиального преобразования, а также доказательство теоремы 4.1 о стойкости предложенной схемы шифрования относительно задач определения секретного ключа аутентификации, дешифрования и навязывания сообщений.

- Протокол выработки общего ключа со взаимной аутентификацией субъектов взаимодействия, а также доказательство теоремы 4.2 о стойкости предложенного протокола относительно задач определения общего ключа, дешифрования и навязывания передаваемой информации.

- Формальная модель, имитирующая криптографический протокол в виде дискретной динамической системы, а также метод получения численных значений показателей эффективности мер защиты, использующий оценки трудоемкости компрометации криптографических преобразований, изменяющих состояния дискретной динамической системы.

- Методика проведения исследования безопасности криптографических протоколов.

Методы исследования. В рамках диссертационного исследования применяются математические методы алгебры, теории чисел, алгебраической геометрии и теории функций комплексного переменного, теории вероятностей и математической статистики, а также теории автоматов.

Достоверность результатов. Достоверность полученных результатов обеспечивается строгими математическими выкладками и доказательствами, апробацией на конференциях и семинарах, а также публикациями в рецензируемых научных журналах. Результаты других авторов, упомянутые в тексте диссертации, отмечены ссылками на соответствующие публикации.

Апробация работы. Результаты, полученные в диссертации, докладывались на международных и всероссийских конференциях и научно-исследовательских семинарах.

- Семинар научного руководителя Московского института электроники и математики им. А.Н. Тихонова федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский университет «Высшая школа экономики», г. Москва, 2022 г.
- Международный симпозиум XVII International Symposium Problems of Redundancy in Information and Control Systems, г. Москва, 2021 г.
- Научно-практическая конференция «РусКрипто», г. Солнечногорск, 2021 г., 2015 г., 2012 г.
- Международная конференция «Компьютерная безопасность и криптография SIBECRYPT-18», г. Абакан, 2018 г.
- Международный симпозиум «Современные тенденции в криптографии CTCrypt», г. Суздаль, 2018, г. Казань, 2015 г.
- Семинар «Математические методы криптографического анализа» кафедры информационной безопасности факультета вычислительной математики и кибернетики МГУ им. В.В. Ломоносова», г. Москва, 2023 г., 2018 г.
- Всероссийский симпозиум по прикладной и промышленной математике ВСППМ, г. Сочи, 2016 г.
- Международная конференция Indo-Russian conference on Algebra, Number Theory, Discrete Mathematics and their Applications, г. Москва, 2014 г.

- Международународный симпозиум «The 7th International Computer Science Symposium in Russia», г. Нижний Новгород, 2014 г.
- Международная конференция «Алгебра и теория чисел: современные проблемы и приложения», г. Саратов, 2013 г., 2012 г.
- XXXIII-я дальневосточная математическая школа-семинар им. академика Е.В. Золотова, г. Владивосток, 2008 г.
- Третья международная научная конференция по проблемам безопасности и противодействия терроризму, г. Москва, 2007 г.
- Седьмая международная научно-техническая конференция «Новые информационные технологии и системы», г. Пенза, 2006 г.

Публикации по теме исследования. Результаты работы изложены в 29 публикациях; в том числе, в 21 публикации в изданиях, индексируемых в Web of Science, Scopus, RSCI и входящих в списки ВАК Минобрнауки России; из них 15 – в изданиях, индексируемых в Web of Science, Scopus, RSCI. Также автором получены 4 свидетельства о государственной регистрации программ для ЭВМ.

Структура и объем работы. Диссертация состоит из введения (общей характеристики работы), четырех глав, заключения, списка литературы, включающего 395 источников, и приложения с программами для ЭВМ. Общий объем диссертации составляет 426 (без приложения — 397) страниц.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обоснована актуальность темы диссертации, сформулированы цели и задачи исследований, отражены научная новизна, теоретическая и практическая значимость полученных результатов, представлены основные результаты, которые выносятся на защиту.

В **первой** главе диссертационной работы развивается математический аппарат, необходимый для уточнения трудоемкости решения задачи дискретного логарифмирования в группе точек эллиптических кривых и построения параметров эллиптических кривых, обладающих заданной трудоемкостью решения задачи дискретного логарифмирования.

Основным преобразованием, используемым в средствах защиты информации, является операция вычисления «кратной» точки эллиптической кривой. Пусть $p > 3$ простое число и $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ произвольная точка

эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p) : y^2 \equiv x^3 + ax + b \pmod p$, определенной над конечным простым полем $\mathbb{F}_p$. Точка $Q \in E_{a,b}(\mathbb{F}_p)$ называется точкой кратности $k \in \mathbb{N}$, если

$$Q = [k]P = \underbrace{P + \cdots + P}_{k \text{ раз}}.$$

Задача определения неизвестного значения k по известным точкам P, Q называется задачей дискретного логарифмирования в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$. Первый метод решения задачи дискретного логарифмирования, имеющий сложность, меньшую чем сложность тотального опробования, предложил в 1962 году А.О. Гельфонд[1]. Метод Гельфонда в отечественной литературе принято называть методом «согласования», в зарубежной – методом «больших и малых шагов» Д. Шенкса[2]. Данный метод применим к любой абелевой группе, а его трудоемкость оценивается величиной $O(\sqrt{q})$ групповых операций, где q порядок группы. Метод требует хранения $O(\sqrt{q})$ элементов группы, что делает его неприменимым при больших значениях q.

Большую роль в решении задачи дискретного логарифмирования сыграли методы поиска циклов в последовательностях. Самый известный метод решения этой задачи предложен в 1968 году Р. Флойдом[3]. Позднее появились методы Р. Брента, Б. Госпера[4], Р. Седжвика и Т. Сжимански[5], Г. Ниваша[6] и др.

Основываясь на методе Р. Флойда в 1975 году Дж. Поллард[7], предложил вероятностный алгоритм решения задачи дискретного логарифмирования (ρ-метод Полларда). Математическое ожидание трудоемкости его работы оценивается величиной $O(\sqrt{q})$, однако, в отличие от метода согласования, алгоритм Полларда-Флойда использует константный объем памяти. Также Поллард[8] предложил еще один вариант данного алгоритма, называемый λ-методом. В 2001 годе Э. Теске[9] предложила модификацию алгоритма Полларда-Флойда, позволившую незначительно снизить его

[1]см. гл.6, § 3 в книге Нечаев В.И. Элементы криптографии (Основы теории защиты информации). — М. : Высшая школа, 1999. — С. 109.

[2]Shanks D. Class number, a theory of factorization and genera // Proceedings Of Symposium Pure Mathematics. — Vol. 20. — Providence, R. I. : AMS, 1971. — P. 415–440.

[3]см. п.3.1, задача 6b, в книге Кнут Д.Э. Искусство программирования для ЭВМ. Получисленные алгоритмы. — 3 изд. — М. : Вильямс, 2000. — Т. 2. — С. 788.

[4]Beeler M., Gosper R.W., Schroeppel R. HACKMEM. — 1972.

[5]Sedgewick R., Szymansky T.G., Yao A.C. The Complexity of Finding Cycles In Pediodic Functions // Siam Journal Of Computing. — 1982. — Vol. 11, no. 2. — P. 376–390.

[6]Nivash G. Cycle Detecting Using a Stack // Journal Information Processing Letters. — 2004.

[7]Pollard J.M. A Monte Carlo Method for Factorisation // BIT. — 1975. — no. 15. — P. 331–334.

[8]Pollard J.M. Monte Carlo methods for index computation (mod p) // Mathematics Of Computation. — 1978. — Vol. 32, no. 143. — P. 918– 924.

[9]Teske E. On Random Walks For Pollard's Rho Method // Mathematics of Computation. — 2000. — Vol. 70. — P. 809–825.

сложность за счет использования памяти, хранящей $O(\log_2 q)$ элементов группы.

В 1998 году П. ван Ооршотом и М. Винером[10] был предложен универсальный метод поиска коллизий. Математическое ожидание трудоемкости его работы также оценивается величиной $O(\sqrt{q})$, однако он допускает эффективную параллельную реализацию. Вопросы применения метода Ооршота-Винера к группе точек эллиптических кривых рассматривались в ряде работ[11,12], также известны[13,14] результаты практического применения данного метода для простых значений p порядка 2^{112} и 2^{114}.

Развивая указанные алгоритмы в 2010 году автор диссертации предложил алгоритм дискретного логарифмирования, основанный на идеях Б. Госпера.

Рассмотрим задачу поиска двух совпадающих элементов последовательности $(a_n)_{n=0}^{\infty}$, определяемой равенством $a_{n+1} = f(a_n)$ для некоторого фиксированного отображения f. Фиксируем значение $n > 0$ и поместим во множество $M(n)$ элементы $a_{n_0},\ a_{n_1},\ \ldots$ рассматриваемой последовательности, с условием

$$n_i = \max_{r<n}\{r|\nu_2(r+1) = i\},$$

для всех возможных значений $i = 0,\ 1,\ \ldots$, где функция $\nu_2(r+1)$ возвращает наибольшую степень двойки, делящую величину $r+1$. Из определения следует, что для фиксированного значения n множество $M(n)$ конечно, содержит не более $\lfloor \log_2 n \rfloor + 1$ чисел и отличается от множества $M(n+1)$ лишь одним элементом. В § 1.2.3 диссертационной работы доказана следующая теорема.

Теорема 1.2. *Пусть заданы параметры λ и τ, определяющие длину подхода к циклу и длину цикла последовательности $(a_n)_{n=0}^{\infty}$. Тогда найдутся натуральные индексы r и $n = r + \tau$ такие, что*

1. *элемент a_r принадлежит множеству $M(n)$ и выполнено равенство $a_n = a_r$,*

[10]Oorschot P.C., Wiener M.J. Parallel Collision Search with Cryptanalytic Applications // Journal of Cryptology. — 1999. — Vol. 12. — P. 1–28.

[11]Bos J.W., Costello C., Miele A. Elliptic and Hyperelliptic Curves: A Practical Security Analysis // Public-Key Cryptography (PKC 2014). — Berlin, Heidelberg : Springer Berlin Heidelberg, 2014. — P. 203–220.

[12]Wiener M.J., Zuccherato R.J. Faster Attacks on Elliptic Curve Cryptosystems // Selected Areas in Cryptography (SAC-98). — Berlin, Heidelberg : Springer Berlin Heidelberg, 1999. — P. 190–200.

[13]Solving a 112-Bit Prime Elliptic Curve Discrete Logarithm Problem on Game Consoles Using Sloppy Reduction / J.W. Bos, M.E. Kaihara, T. Kleinjung et al. // Int. J. Appl. Cryptol. — 2012. — Feb. — Vol. 2, no. 3. — P. 212–228.

[14]Solving a 114-Bit ECDLP for a Barreto-Naehrig Curve / T. Kusaka, S. Joichi, K. Ikuta et al. // Information Security and Cryptology – ICISC 2017. — 2018. — P. 231–244.

2. $\lambda + \tau \leqslant n < \lambda + 2\tau$.

Доказательство теоремы является конструктивным и позволяет предложить алгоритм явного определения величины τ. Сравнение такого алгоритма с известными ранее приводится в следующей таблице.

Алгоритм	*Трудомкость*	*Объем памяти*
Флойд	$\tau\left(3\left\lceil\frac{\lambda}{\tau}\right\rceil + 1\right)$	3
Брент	не менее $\tau + \frac{3}{2}\max\{\lambda+1, \tau\}$	4
Теорема 1.2 (Госпер)	не более $\lambda + 2\tau$	$\lceil\log_2(\lambda + 2\tau)\rceil + 4$

Таблица 1.2: Оценки трудоемкости и объема памяти для алгоритмов поиска длин циклов в последовательностях

Из приведенных значений следует, что алгоритм Госпера имеет наименьшую трудоемкость среди рассматриваемых алгоритмов используя при этом несколько больший объем памяти.

Алгоритм поиска длины цикла может быть использован для решения задачи дискретного логарифмирования в группе точек эллиптической кривой, т.е. отыскания натуральной величины k по известным параметрам эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и точкам $P, Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ таким, что

$$Q = [k]P, \quad P, Q \in \mathcal{E}_{a,b}(\mathbb{F}_p), \quad k \in \mathbb{Z}_q^*, \quad q = \mathtt{ord}(P).$$

Определим в качестве множества $\mathcal{M}$ — подгруппу порядка q, порожденную точкой P, т.е.

$$\mathcal{M} = \{P,\ [2]P,\ [3]P,\ \ldots,\ [q]P = \mathcal{O}\}, \quad \mathcal{M} \subseteq \mathcal{E}_{a,b}(\mathbb{F}_p)$$

где $\mathcal{O}$ – бесконечно удаленная точка кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$. Зафиксируем натуральное число $s = \lceil\log_2 q\rceil$ и разобьем множество $\mathcal{M}$ на s не пересекающихся подмножеств

$$\mathcal{M} = \bigcup_{l=0}^{s-1} J_l$$

следующим образом – будем относить к подмножеству J_l те точки множества $\mathcal{M}$, у которых x-координата сравнима с l по модулю s.

Построим случайное отображение f множества $\mathcal{M}$ в себя. Для этого выберем случайным образом вычеты

$$\gamma_i, \omega_i \in_R \mathbb{Z}_q^*, \quad i = 1, \ldots, s,$$

и для произвольной точки $R \in \mathcal{M}$ такой, что $R = (x, y)$ и $x \equiv l \pmod{s}$, определим

$$f(R) \;=\; R + [\gamma_l]P + [\omega_l]Q, \quad \text{если} \quad R \in J_l.$$

Используем построенное отображение для выработки последовательности точек $\{R_n\}_{n=0}^{\infty}$ множества $\mathcal{M}$. Выберем в качестве начальной точки

$$R_0 = [\alpha_0]P + [\beta_0]Q, \quad \alpha_0, \beta_0 \in_R \mathbb{Z}_q^*,$$

где α_0, β_0 – случайные вычеты из $\mathbb{Z}_q^*$, и определим

$$R_{n+1} = f(R_n) = R_n + [\gamma_l]P + [\omega_l]Q, \quad \text{если} \quad R_n \in J_l.$$

Для каждой точки R_n найдется номер $l_n \in \mathbb{Z}_s$ такой, что $R_n \in J_{l_n}$. Тогда для любого индекса $n = 0, 1, \ldots$получаем равенство

$$\begin{aligned} R_{n+1} &= R_n + [\gamma_{l_n}]P + [\omega_{l_n}]Q = \\ &= R_{n-1} + [\gamma_{l_{n-1}}]P + [\omega_{l_{n-1}}]Q + [\gamma_{l_n}]P + [\omega_{l_n}]Q = \ldots \end{aligned}$$

$$\begin{aligned} \cdots &= R_0 + \left[\sum_{j=0}^{n} \gamma_{l_j} \pmod{q}\right] P + \left[\sum_{j=0}^{n} \omega_{l_j} \pmod{q}\right] Q = \\ &= \left[\alpha_0 + \sum_{j=0}^{n} \gamma_{l_j} \pmod{q}\right] P + \left[\beta_0 + \sum_{j=0}^{n} \omega_{l_j} \pmod{q}\right] Q = \\ &= [\alpha_{n+1}]P + [\beta_{n+1}]Q, \end{aligned}$$

где $\alpha_{n+1}, \beta_{n+1} \in \mathbb{Z}_q$. Используя утверждение теоремы 1.2 можно найти два элемента R_t и R_l такие, что $R_t = R_l$, тогда

$$\begin{aligned} &[\alpha_t + \beta_t k \pmod{q}]P = \\ &\quad = [\alpha_t]P + [\beta_t k \pmod{q}]P = [\alpha_t]P + [\beta_t]Q = R_t = \\ &\quad = R_l = [\alpha_l]P + [\beta_l]Q = [\alpha_l]P + [\beta_l k \pmod{q}]P = \\ &\quad = [\alpha_l + \beta_l k \pmod{q}]P. \end{aligned}$$

Последнее равенство позволяет выразить неизвестное k через значения величин $\alpha_t, \alpha_l, \beta_t, \beta_l \in \mathbb{Z}_q$

$$k \equiv \frac{\alpha_t - \alpha_l}{\beta_l - \beta_t} \pmod{q}.$$

Для практической реализации описанного алгоритма необходимо использование двух массивов. Первый массив – S, будет хранить точки

$$S[l] = [\gamma_l]P + [\omega_l]Q, \quad l = 0, 1, \ldots, s-1,$$

а также выбранные ранее случайным образом вычеты $\gamma_l, \omega_l \in \mathbb{Z}_q^*$.

Во втором массиве будут храниться элементы множества $M(n)$, определяемого при вычислении n-го элемента последовательности $\{R_n\}_{n=0}^{\infty}$. Каждый элемент массива должен хранить в себе точку R, находящуюся во множестве $M(n)$, а также коэффициенты α, β, выражающие точку R через исходные точки P и Q.

Определение размера второго массива тесно связано с трудоемкостью рассматриваемого алгоритма. Рассматривая отображение $f : \mathcal{M} \to \mathcal{M}$ как случайную величину на множестве всех возможных отображений множества $\mathcal{M}$ в себя, можно записать[15,16], что

$$\lim_{q\to\infty} \frac{E_q(\lambda, f)}{\sqrt{q}} = \lim_{q\to\infty} \frac{E_q(\tau, f)}{\sqrt{q}} = \sqrt{\frac{\pi}{8}}.$$

Теперь, делая предположение о независимости величин λ и τ, можно считать, что асимптотическая оценка числа шагов n в рассматриваемом алгоритме следует из равенства

$$\lim_{q\to\infty} \frac{E_q(\lambda, f) + 2E_q(\tau, f)}{\sqrt{q}} = 3\sqrt{\frac{\pi}{8}}.$$

Определим $h = 2 + \lfloor \log_2 \sqrt{q} \rfloor$. Поскольку множество $M(n)$ содержит не более $\lfloor \log_2 n \rfloor + 1$ элементов последовательности $\{R_n\}_{n=0}^{\infty}$, получим неравенство

$$\lfloor \log_2 n \rfloor + 1 < \left\lfloor \log_2 3\sqrt{\frac{\pi q}{8}} \right\rfloor + 1 = \\ = \left\lfloor \frac{1}{2}\Big(\log_2 9\pi + \log_2 q - 3 \Big) \right\rfloor + 1 < \left\lfloor \frac{\log_2 q}{2} \right\rfloor + 2 = h,$$

из которого следует, что ожидаемый размер множества $M(n)$ не превысит величины h. Алгоритм был реализован автором на ЭВМ. Получаемые на практике значения величины n хорошо согласуются с ожидаемым теоретическим значением $3\sqrt{\frac{\pi q}{8}}$.

[15]Flajolet P., Odlyzko A.M. Random mapping statistics // Advances in Cryptology: Proc. Eurocrypt'89. — Vol. 434. — NY. : Springer, 1990. — P. 329–354.

[16]Колчин В.Ф. Случайные графы. — 2 изд. — М. : Физматлит, 2004. — С. 206.

Рассмотренные выше методы являются универсальными и применимы к любой абелевой группе. Для решения задачи дискретного логарифмирования в мультипликативной группе $\mathbb{F}_p^*$ известны алгоритмы[17,18], имеющие трудоемкость, меньшую чем у перечисленных ранее методов. Однако для группы точек эллиптической кривой, определенной над конечным простым полем $\mathbb{F}_p$, вопрос о построении алгоритма, имеющего алгоритмическую сложность меньшую, чем $O(\sqrt{q})$, остается открытым.

Вопросы решения задачи дискретного логарифмирования для эллиптических кривых частного вида рассматривались в работах Т. Сато и К. Араки[19], И.А. Семаева[20], Н. Смарта[21], Р. Шипси и К. Суорт[22]. Методы сведения задачи дискретного логарифмирования в группе точек эллиптической кривой к другим трудноразрешимым математическим задачам рассматривались в работах А. Менезеса, С. Ванстоуна, Т. Окамото[23], а также К. Пети, М. Костерса и А. Мессенга[24]. Следует также отметить носящие теоретический характер работы Дж. Сильвермена[25] и И.А. Семаева[26].

Алгоритмы дискретного логарифмирования, рассматриваемые в § 1.3 диссертационной работы, также представляют собой методы решения задачи в частном случае.

Одним из важных вопросов при анализе средств защиты информации является вопрос о существовании так называемых «слабых» ключей, то есть секретных значений, использование которых приводит к снижению алгоритмической сложности решения задачи, обосновывающей стойкость средства защиты информации. Примеры «слабых» ключей, применительно к задачам анализа алгоритмов блочного шифрования, можно найти в

[17]Gordon D. Discrete Logarithms in Fp Using the Number Field Sieve // SIAM J. Discrete Math. — 1993. — Vol. 6. — P. 124–138.

[18]Joux A., Lercier R. Improvements to the general Number Field Sieve for Discrete Logarithms in Prime Fields // Mathematics Of Computation. — 2003. — Vol. 72, no. 242. — P. 953–967.

[19]Satoh T., Araki K. Fermat quotients and the polynomial time discrete log algorithm for anomalous curves // Comm. Math. Univ. Sancti Pauli. — 1998. — Vol. 47. — P. 81–92.

[20]Semaev I. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p // Mathematics of Computation. — 1998. — Vol. 67, no. 221. — P. 353–356.

[21]Smart N. The discrete logarithm problem on elliptic curves of trace one // Journal of Cryptology. — 1999. — Vol. 12. — P. 193–196.

[22]Shipsey R., Swart C. Elliptic divisibility sequences and the elliptic curve discrete logarithm problem. — 2008.

[23]Menezes A., Vanstone S., Okamoto T. Reducing elliptic curve logarithms to logarithms in a finite field // Proc. 23rd ACM Symp. Theory of Computing. — 1991. — P. 80–89.

[24]Petit C., Kosters M., Messeng A. Algebraic Approaches for the Elliptic Curve Discrete Logarithm Problem over Prime Fields // Public-Key Cryptography (PKC 2016). — Berlin Heidelberg : Springer, 2016. — P. 3–18.

[25]Silverman J.H. The Xedni Calculus and the Elliptic Curve Discrete Logarithm Problem // Designs, Codes and Cryptography. — 2000. — no. 20. — P. 5–40.

[26]Semaev I. Summation Polynomials and the Discrete Logarithm Problem on Elliptic Curves. — 2004. – preprint.

работах Н. Фергюссона[27], Дж. Ким[28], а также рекомендациях к TDEA[29]. Для задачи дискретного логарифмирования вопрос о «слабых» ключах был решен автором диссертационной работы. В § 1.3 доказана следующая теорема.

Теорема 1.3. *Пусть $p > 3$ – простое число, $\mathcal{E}_{a,b}(\mathbb{F}_p)$ эллиптическая кривая и $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ – точка кривой, порождающая циклическую подгруппу $\langle P\rangle \subseteq E_{a,b}(\mathbb{F}_p)$ простого порядка q.*

Пусть $Q = [k]P$, $k \in \mathbb{Z}_q$ и $\mathtt{ord}_q\, k = r$. Если $q > 6$ и $r \geqslant 6$, то алгоритмическая сложность нахождения величины k не превосходит $8\sqrt{r}\log_2 q$ групповых операций.

Доказательство теоремы конструктивно и позволяет предъявить алгоритм поиска неизвестной величины k, основанный принципах «согласования» А.О. Гельфонда. Поскольку при практически важных значениях параметров эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ такой метод не может быть реализован на практике, автором были разработаны две его модификации, основанные на упомянутом выше λ-методе Полларда.

Зафиксируем произвольный первообразный корень g по модулю q, определим $\alpha \equiv g^{\frac{q-1}{r}} \pmod q$ и будем искать неизвестное значение k в виде $k \equiv \alpha^x \pmod q$. Рассмотрим подмножество точек на эллиптической кривой $\mathcal{M}$, определяемое равенством

$$\mathcal{M} = \{[\alpha]P,\ [\alpha^2 \pmod q]P,\ \ldots,\ [\alpha^r \pmod p]P = P\}.$$

Точка Q принадлежит рассматриваемому подмножеству $\mathcal{M}$, поскольку $Q = [k]P = [\alpha^x]P \in \mathcal{M}$ для некоторого значения $x \in \mathbb{Z}_r$.

Определим случайное отображение $f : \mathcal{M} \to \mathcal{M}$. Для этого зафиксируем $s = \lceil \log_2 r \rceil$, выберем случайным образом вычеты

$$\xi_0,\ \ldots,\ \xi_{s-1} \in_R \mathbb{Z}_r^*.$$

и для любой точки $R \in \mathcal{M}$, заданной в аффинной форме координатами (x_R, y_R), определим

$$f(R) = [\alpha^{\xi_l} \pmod q]R, \quad \text{где} \quad l \equiv x_R \pmod s$$

для некоторого $l \in \mathbb{Z}_s$.

[27]Ferguson N. Authentication weaknesses in GCM. — 2005.

[28]Kim J. On the security of the block cipher GOST suitable for the protection in U-business services // Personal and Ubiquitous Computing volume. — 2013. — P. 1429–1435.

[29]Barker E., Mouha N. Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher. — 2012. — NIST Special Publication 800-67, Revision 2.

Первая модификация алгоритма поиска неизвестного значения k использует факт пересечения двух различных последовательностей, образуемых при помощи отображения f. Выберем случайные вычеты $\gamma_0, \omega_0 \in_R \mathbb{Z}_r^*$ и определим начальные точки

$$R_0 = [\alpha^{\gamma_0} \pmod q]P, \quad U_0 = [\alpha^{\omega_0} \pmod q]Q.$$

Остальные элементы последовательностей $\{R_n\}_{n=0}^{\infty}$ и $\{U_i\}_{i=0}^{\infty}$ определим равенствами

$$R_{n+1} = f(R_n), \quad U_{n+1} = f(U_n), \quad n = 0, 1, \ldots$$

Легко видеть, что выполнены равенства

$$\begin{aligned} R_{n+1} = [\alpha^{\xi_{l_n}} \pmod q]R_n = [\alpha^{\xi_{l_n}}\alpha^{\xi_{l_{n-1}}} \pmod q]R_{n-1} = \cdots \\ = [\alpha^{\gamma_0+\sum_{i=0}^{n}\xi_{l_i}} \pmod q]P = [\alpha^{\gamma_n} \pmod q]P \in \mathcal{M}, \end{aligned}$$

для некоторого $\gamma_n \equiv \gamma_0 + \sum_{i=0}^{n} \xi_{l_i} \pmod r$, и

$$\begin{aligned} U_{i+1} = [\alpha^{\xi_{l_i}} \pmod q]U_n = [\alpha^{\xi_{l_i}}\alpha^{\xi_{l_{i-1}}} \pmod q]U_{i-1} = \cdots \\ = [\alpha^{\omega_0+\sum_{j=0}^{n}\xi_{l_j}} \pmod q]Q = [\alpha^{\omega_i} \pmod q]Q = [k\alpha^{\omega_i} \pmod q]P \in \mathcal{M}, \end{aligned}$$

для некоторого $\omega_i \equiv \omega_0 + \sum_{j=0}^{n} \xi_{l_j} \pmod r$.

Выберем индекс n_0, удовлетворяющий неравенству

$$n_0 \geqslant \left\lfloor \sqrt{\frac{\pi r}{2}} \right\rfloor$$

и зафиксируем точку R_{n_0}. Поскольку n_0 достаточно велико, можно ожидать, что точка R_0 лежит на цикле последовательности $\{R_n\}_{n=0}^{\infty}$. Если найдется индекс i такой, что $R_{n_0} = U_i$, то выполнено равенство

$$[\alpha^{\gamma_{n_0}} \pmod q]P = [\alpha^{\omega_i} \pmod q]Q = [k\alpha^{\omega_i} \pmod q]P.$$

и неизвестное k удовлетворяет сравнению

$$k \equiv \alpha^{\gamma_{n_0}-\omega_i} \pmod q.$$

При случайном выборе точек R_0 и U_0 может случиться так, что последовательности $\{R_n\}_{n=0}^{\infty}$ и $\{U_i\}_{i=0}^{\infty}$ не будут иметь общих точек. Отображение f разбивает[30] множество $\mathcal{M}$ на m не пересекающихся областей. При

[30] Flajolet P., Odlyzko A.M. Random mapping statistics // Advances in Cryptology: Proc. Eurocrypt'89. — Vol. 434. — NY. : Springer, 1990. — P. 329–354.

этом, для математического ожидания $E_r(m, f)$ величины m выполнено равенство

$$\lim_{r\to\infty} \frac{E_r(m,f)}{\log_2 r} = 1.$$

Следовательно, можно ожидать, что для нахождения совпадающих точек $R_{n_0} = U_i$ и, как следствие, определения неизвестного k, потребуется выбрать $\log_2 r$ случайных пар R_0 и U_0.

Поскольку для каждой из последовательностей $\{R_n\}_{n=0}^{\infty}$ и $\{U_i\}_{i=0}^{\infty}$ вычисляется не более n_0 элементов с трудоемкостью не более, чем $2\lceil\log_2 q\rceil$ групповых операций, а общее количество вырабатываемых пар R_0 и U_0 не превышает $\lfloor\log_2 r\rfloor$, то алгоритмическая сложность первой модификации алгоритма поиска неизвестного значения k не превышает

$$4n_0\lfloor\log_2 r\rfloor\lceil\log_2 q\rceil < 2\sqrt{2\pi r}\log_2 r(\log_2 q + 1).$$

операций в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$. Данная оценка несколько хуже, чем утверждение теоремы 1.3. Вместе с тем в алгоритме требуется хранение лишь трех точек эллиптической кривой, что делает возможной его реализацию на ЭВМ при любых значениях r.

Вторая модификация алгоритма поиска неизвестного значения k комбинирует идеи λ-метода Полларда и метода поиска коллизий Ооршота-Винера. Данная модификация описывается в § 1.3.2 диссертационной работы и предназначена для реализации на ЭВМ, допускающих распараллеливание вычислений.

Для того, чтобы исключить частные случаи из рассмотрения и обеспечить высокую трудоемкость решения задачи дискретного логарифмирования необходимо выбирать параметры эллиптических кривых специальным образом. Перечни требований к параметрам эллиптических кривых могут быть найдены в стандартах ГОСТ Р 34.10-2012, BSI TR-03111 и др., в работах Д.Бернштейна и Т.Ланге[31], П.Баррето, Г.Перейры и Дж.Рикардини[32], а также автора настоящей диссертационной работы.

Напомним, что нечетное простое число называют *безопасным*, если число $\frac{p-1}{2}$ также является простым. В этом случае простое число $\frac{p-1}{2}$ принято называть простым числом Софи Жермен[33].

Пусть $0 < \alpha < \beta$ натуральные числа, $p > 3$ простое число. Мы будем называть эллиптическую кривую $\mathcal{E}_{a,b}(\mathbb{F}_p)$, определенную сравнением

$$y^2 \equiv x^3 + ax + b \pmod p,$$

[31]Bernstein D., Lange T. Faster addition and doubling on elliptic curves // Advances in Cryptology: ASIACRYPT 2007. — Vol. 4833. — NY. : Springer, 2007. — P. 29–50.

[32]Barreto P., Pereira G., Ricardini J. A note on high-security general-purpose elliptic curves // Cryptology ePrint Archive, Report 2013/647. — 2013.

[33]Shoup V. A Computational Introduction to Number Theory and Algebra. — 2nd edition. — Cambridge University Press, 2009. — P. 590.

безопасной, если найдется точка $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ такая, что $\mathtt{ord}\, P = q$ и выполняются следующие условия:

1. $m = |\mathcal{E}_{a,b}(\mathbb{F}_p)|$ и $m \neq p$;
2. p безопасное простое, т.е. $\frac{p-1}{2}$ также простое число;
3. $j(\mathcal{E}_{a,b}) \not\equiv 0$ или $1728 \pmod p$, где величина $j(\mathcal{E}_{a,b})$ определена сравнением $j(\mathcal{E}_{a,b}) \equiv 1728 \cdot \frac{4a^3}{4a^3-27b^2} \pmod p$;
4. $2^\alpha < q < 2^\beta$;
5. q безопасное простое, т.е. $\frac{q-1}{2}$ также простое число;
6. для фиксированного значения B условие $p^t \not\equiv 1 \pmod q$ выполняется для всех $t = 1, 2, \ldots, B$.

Легко видеть, что безопасная эллиптическая кривая удовлетворяет требованиям из ГОСТ Р 34.10-2012 при $\alpha = 254$ и $\beta = 256$, дополненным требованиями простоты чисел $\frac{p-1}{2}$ и $\frac{q-1}{2}$.

Первое условие из данного выше определения делает нецелесообразным применение упомянутых ранее методов Т. Сато и К. Араки, И.А. Семаева и Н. Смарта. Условие безопасности простого числа p делает нецелесообразным применения метода К. Пети, М. Костерса и А. Мессенга. Условие безопасности простого числа q минимизирует мощность множества «слабых» ключей. Последнее, шестое условие делает нецелесообразным применение метода А. Менезеса, С. Ванстоуна, Т. Окамото.

Для построения безопасных кривых автором применялся следующий подход. В начале, с использованием поиска простых чисел в арифметических прогрессиях, определяются простое число p и порядок группы точек эллиптической кривой q, удовлетворяющие перечисленным выше требованиям, после чего, с использованием математического аппарата теории комплексного умножения[34], определяются коэффициенты безопасной эллиптической кривой (впервые такой подход к построению эллиптических кривых был реализован на практике Ф. Морейном[35,36]).

Детальное описание алгоритма построения эллиптической кривой и доказательство вспомогательных утверждений, позволяющих снизить трудоемкость поиска безопасных простых p и q, содержится в § 1.5.2.

[34]Семинар по комплексному умножению / Ж.-П. Серр, А. Борель, К. Ивасава, Чоула // Математика. — 1968. — Т. 12. — С. 55–95.

[35]Atkin A.O.L., Morain F. Elliptic curves and primality proving // Mathematics Of Computation. — 1993. — Vol. 61. — P. 29–68.

[36]Morain F. Implementation of the Atkin-Goldwasser-Kilian primality testing algorithm. — 1988. — RR-0911, INRIA.

В приложении к диссертационной работе содержится более 60-ти эллиптических кривых, построенных с помощью разработанной автором программной реализации предложенного алгоритма, в частности, эллиптическая кривая

$$y^2 \equiv x^3 - 3x + \\ + 3024818943147551221418867269063791031023404613954261875826530956434811262719 9 \\ \pmod{2^{256} - 188069}.$$

Еще одной задачей, возникающей при применении эллиптических кривых в средствах защиты информации, является задача снижения трудоемкости реализации операции вычисления кратной точки для значений p в интервале $2^{160} < p < 2^{640}$. Для ее решения принято использовать комбинации из одного или нескольких подходов:

1) оптимизация элементарных операций сложения, умножения, а также взятия обратного элемента в поле $\mathbb{F}_p$;

2) использование проективных координат для реализации операций в группе точек эллиптической кривой;

3) использование различных представлений (форм) эллиптических кривых, позволяющих минимизировать количество элементарных операций в поле $\mathbb{F}_p$, необходимых для реализации операций сложения и удвоения точек эллиптической кривой (среди таких форм следует отметить формы Вейерштрасса, Якоби, Гессе, Монтгомери, Эдвардса, и др.);

4) использование алгоритмов вычисления кратной точки, минимизирующих количество операций сложения и удвоения в группе точек эллиптической кривой (к таким алгоритмам относятся «оконные» методы, методы с несколькими основаниями и т.п.);

5) использование оптимизаторов программного кода, минимизирующих число используемых переменных и операций пересылки данных между регистрами вычислительного средства;

6) использование эндоморфизмов эллиптической кривой.

Остановимся на последнем подходе подробнее и предположим, что нам известен комплексный эндоморфизм $\phi : \mathcal{E}_{a,b}(\mathbb{F}_p) \to \mathcal{E}_{a,b}(\mathbb{F}_p)$. В 2001 году Р. Галант, Р. Ламберт и С. Ванстоун[37] предложили использовать для вычисления кратной точки равенство

$$Q = [k]P = [k_1]P + [k_2]\phi(P),$$

где k_1, k_2 целые, зависящие от числа k и эндоморфизма ϕ коэффициенты, удовлетворяющие неравенствам $0 \leqslant k_1, k_2 \leqslant c_0\sqrt{\mathtt{ord}(P)}$ для некоторой эф-

[37]Gallant R.P., Lambert R.J., Vanstone S.A. Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms // Advances in Cryptology – CRYPTO 2001. — 2001. — P. 190–200.

фективно вычислимой[38] константы c_0. Независимо, этот подход рассматривался в работах А.Г. Ростовцева[39,40]. В дальнейшем подход развивался в работах зарубежных авторов[41,42,43].

До недавнего времени было известно только четыре эллиптических кривых с явно заданным эндоморфизмом и лишь две из них могли применяться в средствах защиты информации. В 2014 году автор диссертационной работы предложил эффективно реализуемый на практике алгоритм вычисления явного вида комплексного эндоморфизма эллиптической кривой, основанный на теории комплексного умножения. Это позволило разработать программное обеспечение и вычислить явный вид эндоморфизмов для большого числа эллиптических кривых.

Зафиксируем $\tau \in \mathbb{C}_+$ – мнимую квадратичную иррациональность такую, что $\mathrm{Im}(\tau) > 0$, тогда найдется эллиптическая кривая $\mathcal{E}_{a,b}(\mathbb{F}_p)$ такая, что решетка

$$\Lambda_\tau = \{n + m\tau,\ n, m \in \mathbb{Z}\} \subseteq \mathbb{Q}(\sqrt{-d}),$$

определяемая для некоторого свободного от квадратов числа $d \in \mathbb{N}$, изоморфна кольцу эндоморфизмов кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$.

Пусть $\mathcal{E}(\Lambda_\tau)$ комплексная эллиптическая кривая с решеткой периодов $\{1, \tau\}$. Тогда, каждая точка на кривой $\mathcal{E}(\Lambda_\tau)$ может быть параметризована значениями эллиптической функции Вейерштрасса $\wp$, а эндоморфизм ϕ_α, соответствующий элементу $\alpha \in \Lambda_\tau$, задается[44] отображением

$$(\wp(z) : \wp'(z) : 1) \to (\wp(\alpha z) : \wp'(\alpha z) : 1).$$

Поскольку эллиптические функции образуют поле, а функция $\wp(z)$ – четна, то $\wp(\alpha z)$ и $\wp'(\alpha z)$ как функции переменной z могут быть рацио-

[38]Sica F., Ciet M., Quisquater J.J. Analysis of the Gallant-Lambert- Vanstone Method Based on Efficient Endomorphisms: Elliptic and Hyperelliptic Curves // Selected Areas in Cryptography. SAC 2002. — 2003. — P. 21–36.

[39]Ростовцев А.Г. О выборе эллиптической кривой над простым полем для построения криптографических алгоритмов // Проблемы информационной безопасности. Компьютерные системы. — 1999. — Т. 3. — С. 37–40.

[40]Ростовцев А.Г. Арифметика эллиптических кривых над простыми полями без удвоения точек // Проблемы информационной безопасности. Компьютерные системы. — 2000. — Т. 4.

[41]An Alternate Decomposition of an Integer for Faster Point Multiplication on Certain Elliptic Curves / Y.-H. Park, S. Jeong, C. Kim, J. Lim // Public Key Cryptography. PKC 2002. — 2002. — P. 323–334.

[42]Galbraith S.D., Lin X., Scott M. Endomorphisms for faster elliptic curve cryptography on general curves // Journal Of Cryptology. — 2011. — Vol. 24. — P. 446–469.

[43]Improved Algorithms for Efficient Arithmetic on Elliptic Curves Using Fast Endomorphisms / M. Ciet, T. Lange, F. Sica, J.J. Quisquater // Advances in Cryptology — EUROCRYPT 2003. — 2003. — P. 388–400.

[44]Husemöller D. Elliptic Curves. — 2 edition. — New-York : Springer- Verlag, 2004.

нально выражены[45,46] через $\wp(z)$ и $\wp'(z)$. Обозначим символом $j()$ модулярную функцию. Тогда найдется рациональная функция $R(x) = \dfrac{P(x)}{Q(x)}$, где $P(x), Q(x) \in \mathbb{H}[x]$ и $\mathbb{H} = \mathbb{Q}(\sqrt{-d}, j(\tau))$ такая, что

$$\wp(\alpha z) = R(\wp(z)), \quad \wp'(\alpha z) = \frac{R'(\wp(z))\wp'(z)}{\alpha}.$$

Степени многочленов $P(x), Q(x)$, как функции от величины α, определяются[47] равенствами $\deg P(x) = N(\alpha), \quad \deg Q(x) = N(\alpha) - 1$, где $N(\alpha)$ – норма алгебраического числа $\alpha \in \Lambda_\tau$. Таким образом, задача явного определения эндоморфизма ϕ_α сводится к построению рациональной функции $R(x)$.

В § 1.4.2 автором описывается следующий алгоритм. Для произвольного целого неотрицательного k рассмотрим четную эллиптическую функцию $f_k(z)$, имеющую в нуле полюс второго порядка и определяемую рядом

$$f_k(z) = \sum_{n=0}^{\infty} d_{k,n} z^{2n-2} = \frac{d_{k,0}}{z^2} + d_{k,1} + d_{k,2} z^2 + \cdots, \quad d_{k,n} \in \mathbb{H}.$$

Примером такой функции могут служить $\wp(z)$ или $\wp(\alpha z)$. Воспользовавшись формулой для разложения функции Вейерштрасса $\wp(z)$ в ряд Лорана запишем равенство

$$f_k(z) = d_{k,0} \underbrace{\left(\frac{1}{z^2} + \sum_{n=2}^{\infty} c_n z^{2n-2} \right)}_{\wp(z)} + d_{k,1} + \sum_{n=2}^{\infty} \left(d_{k,n} - d_{k,0} c_n\right) z^{2n-2} =$$

$$= l_k(\wp(z)) + f'_{k+1}(z),$$

где $l_k(x) = d_{k,0}x + d_{k,1} \in \mathbb{H}[x]$ и $\deg l_k(x) = 1$. Функция $f_{k+1}(z)$ также является четной эллиптической функцией и имеет в нуле полюс второго порядка, следовательно, полагая $f_0(z) = \wp(\alpha z)$, для любого m мы можем записать равенство

---

[45]Гурвиц А. Теория аналитических и эллиптических функций. — М. : ГТТИ, 1933. — С. 344.

[46]Cox D. Primes of the form $x^2 + ny^2$: Fermat, Class Field Theory and Complex Multiplication. – NY. : J.Wiles and Sons, 1989. – P. 363.

[47]Stark H. Class numbers of complex quadratic fields // Modular Functions of one variable I. — Vol. 320 of Lecture Notes in Math. — Springer-Verlag, 1973. — P. 153–174.

$$\wp(\alpha z) = l_0(\wp(z)) + \frac{1}{f_1(z)} = \cdots$$
$$\cdots = l_0(\wp(z)) + \cfrac{1}{l_1(\wp(z)) + \cfrac{1}{\cdots + \cfrac{1}{l_{m-1}(\wp(z)) + \cfrac{1}{f_m(z)}}}}.$$

Поскольку представление $\wp(\alpha z)$ в виде рациональной функции от $\wp(z)$ единственно и $\deg l_k(x) = 1$ для всех $k \in \mathbb{N}_0$, то при $m = N(\alpha) - 1$ будет выполнено равенство $f_m(z) = l_m(\wp(z))$ и мы получим разложение функции $\wp(\alpha z)$ в непрерывную дробь. Приводя полученное представление к виду рациональной дроби мы получаем искомое равенство.

Для иллюстрации разработанного алгоритма автором диссертационной работы были построены эндоморфизмы для всех эллиптических кривых $\mathcal{E}(\Lambda_\tau)$ таких, что $j(\tau) \in \mathbb{Z}$. В частности были получены следующие новые эндоморфизмы:

- кривая $y^2 = 4x^3 - 15x - 11$, $\alpha = \sqrt{-3}$ и $N(\alpha) = 3$:

$$\phi_\alpha : (x, y) \to \left(-\frac{4x^3 + 12x^2 + 33x + 28}{3(2x+3)^2},\ \frac{-8x^3 - 36x^2 - 6x + 13}{3\alpha(2x+3)^3} y \right),$$

- кривая: $y^2 = 4x^3 - 264x - 847$, $\alpha = \frac{1}{2}(1 + \sqrt{-11})$ и $N(\alpha) = 3$:

$$\phi_\alpha : (x, y) \to \left(-\frac{(\alpha+2)x^3 + 6(\alpha+5)x^2 - 33(4\alpha-13)x - 11(59\alpha-134)}{9(x-\alpha+6)^2}, \right.$$
$$\left. -\frac{(\alpha+2)x^3 + 9(\alpha+5)x^2 + 33(4\alpha-1)x + 11(19\alpha-70)}{9\alpha(x-\alpha+6)^3} y \right).$$

Полный перечень построенных эндоморфизмов приводится в приложении к диссертационной работе. Для некоторых эллиптических кривых из рассматриваемого класса, с использованием соотношений Дж.Тейта, были найдены формы, минимизирующие количество элементарных операций, необходимых для вычисления построенных эндоморфизмов. Перечень таких кривых содержится в таблице 1.5.

При этом некоторые из построенных эндоморфизмов принимают следующий вид:

1. кривая $v^2 = u^3 - 6u^2 + u$, $\alpha = 2\sqrt{-1}$ и $N(\alpha) = 4$,

$$\hat{\phi}_\alpha : (u, v) \to \left(-\frac{(u-1)^2 v^2}{4u^2(u+1)^2}, -\frac{(u^5 + 3u^4 - 30u^3 + 30u^2 - 3u - 1)}{4\alpha u^2(u+1)^3} v \right).$$

№	τ	θ	γ	$\mathcal{H}(\mathbb{C})$
1	$2\sqrt{-1}$	-1	2	$v^2 = u^3 - 6u^2 + u$
2	$\sqrt{-2}$	-2	$\frac{2}{3}$	$v^2 = u^3 - 4u^2 + 2u$
3	$\sqrt{-3}$	-1	2	$v^2 = u^3 - 6u^2 - 3u$
4	$\frac{3}{2}\left(1+\sqrt{-3}\right)$	-3	1	$v^2 = u^3 - 9u^2 - 3u - \frac{1}{4}$
5	$\frac{1}{2}\left(1+\sqrt{-7}\right)$	$\frac{\alpha-4}{2}$	$-\frac{(\alpha+10)}{112}$	$v^2 = u^3 - \frac{3}{32}(\alpha-6)u^2 - \frac{1}{64}(3\alpha-2)u$
5	$\frac{1}{2}\left(1+\sqrt{-7}\right)$	$-\frac{1}{2}$	$\frac{1}{2}$	$v^2 = u^3 - \frac{3}{4}u^2 - 2u - 1$

Таблица 1.5: Эллиптические кривые с целым j-инвариантом.

2. кривая $v^2 = u^3 - \frac{3}{32}(\alpha-6)u^2 - \frac{1}{64}(3\alpha-2)u$, $\alpha = \frac{1}{2}(1+\sqrt{-7})$ и $N(\alpha) = 2$:

$$\hat{\phi}_\alpha : (u, v) \to \left(-\frac{(\alpha+1)u^2 + u - \mu}{4u},\ -\frac{(\alpha+1)u^2 + \mu}{4\alpha u^2} v, \right),$$

где $\mu = \frac{\alpha-2}{16} = \left(\frac{\alpha}{4}\right)^2 = \frac{1}{4(\alpha+1)}$.

Для применения построенных эндоморфизмов в средствах защиты информации автором был разработан способ вычисления кратной точки эллиптической кривой, описываемый в § 1.4.5 диссертационной работы. Автором доказана следующая теорема.

Теорема 1.6. *Пусть $d > 1$ – свободное от квадратов, целое число и задан элемент $\alpha \in \Lambda_\tau \subseteq \mathbb{Z}_\mathbb{K} \subset \mathbb{Q}(\sqrt{-d})$ такой, что $N(\alpha) \geqslant 2$. Определим натуральное число*

$$n_\alpha = \frac{N(\alpha) - \delta_\alpha}{2}, \quad \textit{где} \quad \delta_\alpha \equiv N(\alpha) \pmod 2,$$

и множество $\mathcal{N} = [-n_\alpha, -n_\alpha + 1, \ldots, n_\alpha - 1, n_\alpha]$. Тогда, если α удовлетворяет неравенству $|\operatorname{tr}(\alpha) - 1| \leqslant n_\alpha$, то для любого натурального k найдется многочлен $g(x) \in \mathcal{N}[x]$ такой, что

$$k = g(\alpha) = \sum_{i=0}^{w+c_1} g_i \alpha^i, \quad g_i \in \mathcal{N},$$

где $\deg g(x) \leqslant w_1 = c_1 + \left\lceil 2 \log_{N(\alpha)} k \right\rceil$, где

$$c_1 = \begin{cases} 4, & \textit{если} \quad \alpha = 1 \pm \sqrt{-2}, \\ 3, & \textit{иначе.} \end{cases}$$

Отметим, что вопросы представления целых чисел в системах счисления с произвольным действительным основанием α ведут начало от работ

А. Реньи[48] и В. Перри[49]. Используемый в данной работе метод представления натурального числа k в системе счисления с комплексным основанием α базируется на результатах работ В. Мюллера[50] и Н. Смарта[51]. Также, в более слабой форме, доказанная теорема формулировалась в упомянутой ранее работе А.Г. Ростовцева.

Доказательство теоремы 1.6 конструктивно и позволяет предъявить алгоритм построения коэффициентов многочлена $g(x)$ по заданным значениям k и α. После чего, вычисление кратной точки сводится к вычислению равенства

$$[k]P = [g_0]P + [g_1]\phi_\alpha(P) + [g_2]\phi_\alpha^2(P) + \cdots + [g_{w_1}]\phi_\alpha^{w_1}(P) = \\ = [g_0]P + \phi_\alpha\Big([g_1]P + \cdots \phi_\alpha\big([g_{w_1-1}]P + \phi_\alpha([g_{w_1}]P)\big)\Big).$$

Отметим, что для фиксированной точки P можно заранее вычислить точки $[g_0]P, \ldots, [g_{w_1}]P$, понизив тем самым трудоемкость вычисления кратной точки до w_1 операций вычисления эндоморфизма ϕ_α и w_1 операций сложения точек на эллиптической кривой.

Во **второй** главе диссертационной работы приводятся результаты исследований, позволяющие обосновать целесообразность применения в средствах защиты информации генераторов псевдослучайных чисел, основанных на представлении действительных иррациональных чисел в виде систематических дробей по заданному основанию.

В § 2.2 диссертационной работы рассматриваются два класса действительных чисел. Пусть $b > 1$, $d > 1$ – целые числа, m – натуральное и $x_1, \ldots, x_m \in \mathbb{N}$ попарно различные числа, удовлетворяющие неравенствам $0 < x_k \leqslant d$ для всех $k = 1, \ldots, m$. Пусть $u_1, \ldots, u_m \in \mathbb{Q}$ — не все одновременно равные нулю рациональные числа. К первому классу относятся действительные числа вида

$$\alpha = \sum_{n=0}^{\infty} \left(\frac{u_1}{dn + x_1} + \cdots + \frac{u_m}{dn + x_m} \right) b^{-n} = \sum_{n=0}^{\infty} \sum_{k=1}^{m} \frac{u_k}{dn + x_k} b^{-n}.$$

[48] Rényi A. Representations for real numbers and their ergodic properties // Acta Mathematica Academiae Scientiarum Hungaricae. — 1957. — Vol. 8. — P. 477–493.

[49] Parry W. On the β-expansions of real numbers // Acta Mathematica Academiae Scientiarum Hungaricae. — 1960. — Vol. 11. — P. 401–416.

[50] Müller V. Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two // Journal of Cryptology. — 1998. — Vol. 11. — P. 219–234.

[51] Smart N. Elliptic Curve Cryptosystems over Small Fields of Odd Characteristic // Journal of Cryptology. — 1999. — Vol. 12. — P. 141– 151.

Из работ Р. Тайдемана и его соавторов[52,53] следует, что число α – иррационально. Ко второму классу чисел относятся числа вида

$$\alpha = \sum_{n=0}^{\infty} \frac{x_n}{n!}.$$

где $(x_n)_{n=0}^{\infty}$ — чисто периодическая последовательность рациональных чисел с периодом длины m. Иррациональность таких чисел следует из доказанной автором, совместно с В.Г. Чирским, теоремы, см. § 2.2.2.

Теорема 2.1. *Пусть m – натуральное число и $(x_n)_{n=0}^{\infty}$ — чисто периодическая последовательность рациональных чисел с периодом длины m такая, что существует индекс $k \geqslant 0$ для которого $x_k \neq 0$. Пусть α число из второго класса и $\alpha \neq 0$, тогда α — иррационально.*

Отметим, что иррациональность чисел из второго класса для случая непериодической последовательности коэффициентов $(x_n)_{n=0}^{\infty}$ изучалась в работе Дж. Ханцля и Р. Тайдемана[54].

Напомним, что любое действительное число α может быть представлено в виде систематической дроби

$$\alpha = \sum_{k=0}^{\infty} a_k b^{-k}, \quad a_k \in \mathbb{Z}, \quad \text{и} \quad 0 \leqslant a_k < b \quad \text{при} \quad k > 0.$$

для произвольного натурального $b > 1$. Разработанные автором алгоритмы, позволяющие представить действительные числа из рассматриваемых классов в виде систематической дроби рассматриваются автором в § 2.3 диссертационной работы.

Наличие таких алгоритмов позволяет рассмотреть вопрос о целесообразности применения в средствах защиты информации генераторов псевдослучайных последовательностей, образованных коэффициентами представления числа α из рассматриваемых классов в виде систематической дроби. Хорошо известно[55,56], что такие последовательности не являются периодическими.

К генераторам, применяемым в средствах защиты информации, предъявляется ряд требований, среди которых отметим следующие.

---

[52]Transcendental infinite sums / S.D. Adhikari, N. Saradha, T.N. Shorey, R. Tijdeman // Indag. Math. — 2001. — Vol. 12. — P. 1–14.

[53]Tijdeman R. On irrationality and transcendency of infinite sums of rational numbers // Diophantine Equations / Ed. by N. Saradha. — New Delhi, India : Narosa Publisher, 2008. — P. 279–296.

[54]Hancl J., Tijdeman R. On the irrationality of factorial series II // Journal of Number Theory. — 2010. — Vol. 130. — P. 595– 607.

[55]Remmert R., Ullrich P. Elementare Zahlentheorie. — Berlin : Birkhäuser, 1995. — P. 276.

[56]Нестеренко Ю.В. Теория чисел. — М. : Академия, 2008. — С. 272.

1. Необходимо[57,58], чтобы для любых значений входных параметров n, $x_1, \ldots, x_m$ вырабатываемая генератором последовательность $(a_k)_{k=1}^n$ представляла собой реализацию случайной величины, равномерно распределенной на интервале $[0, b-1]$. Для некоторых из используемых в средствах защиты информации генераторов псевдослучайных последовательностей сформулированное требование доказать не удается. Поэтому, на практике, оно заменяется на более слабое – требование статистической неотличимости выработанной последовательности от равномерно распределенной.

2. Необходимо, чтобы задача определения любого подмножества элементов последовательности $(a_k)_{k=1}^n$ по известному другому подмножеству элементов той же последовательности имела высокую алгоритмическую сложность. Частными случаями данного свойства являются высокая трудоемкость определения начальных значений генератора $x_1, \ldots, x_m$ по элементам последовательности $(a_k)_{k=1}^n$ и отсутствие у последовательности $(a_k)_{k=1}^n$ периода длиной τ при $\tau < n$.

Впервые вопрос о нормальности произвольных действительных иррациональных чисел, т.е. о равномерном распределении коэффициентов систематических дробей в произвольной системе счисления, по-видимому, был поставлен Э. Борелем[59]. Отдельно стоит выделить случай числа π, для которого исследования начались существенно ранее. Исторический обзор указанных вычислений можно найти в монографии А.В. Жукова[60], а также в статье Д. Борвейна и соавторов[61]. Один из последних результатов о вычислении мантиссы числа π можно найти в работе А. Йи[62].

Гипотеза о нормальности числа π формулировалась в работе Д. Бейли и Р. Кренделла[63] и экспериментально проверялась в ряде работ[64,65,66].

---

[57]Бабаш А.В., Шанкин Г.П. Криптография / Под ред. В.П. Шерстюка, Э.А. Применко. — М. : Солон-Пресс, 2007. — С. 512.

[58]Основы криптографии / А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин, А.В. Черемушкин. — 2 изд. — М. : Гелиос АРВ, 2002. — С. 480.

[59]Borel E. Lesons sur la theorie des functions. — Paris, 1914.

[60]Жуков А.В. Вездесущее число π. — 5-е изд. — М. : Либроком, 2012. — С. 240.

[61]The Quest for π / D.H. Bailey, J.M. Borwein, P.B. Borwein, S. Plouffe // Mathematical Intelligencer. — 1997. — Vol. 19, no. 1. — P. 50–57.

[62]Yee A.J. World π record for both desktop and supercomputer. — 2012.

[63]Bailey D.H., Crandall R.E. On the random character of fundamental constant expansions // Experimental Mathematics. — 2001. — Vol. 10, no. 2. — P. 175–190.

[64]Lagarias J.C. On the normality of arithmetical constants // Experimental Mathematics. — 2001. — Vol. 10, no. 3. — P. 355– 368.

[65]Tu S.J., Fischbach E. A study on the randomness of the digits of π // International Journal of Modern Physics C. — 2005. — Vol. 16, no. 2. — P. 281–294.

[66]Нестеренко А.Ю. О статистических свойствах некоторых трансцендентных чисел // Ученые записки Орловского государственного университета. — 2012. — № 6 (часть 2). — С. 170–176.

В настоящее время доказана нормальность только нескольких иррациональных чисел специального вида[67,68], однако для произвольного иррационального числа α, не известен критерий, позволяющий определить, является ли число α нормальным или нет.

В начале 20-го века Г. Вейль, Г.Харди и Дж.Литллвуд доказали[69], что для почти всех иррациональных чисел α последовательность действительных чисел $\{\alpha b^n\}$, где $b > 1$ натуральное число, равномерно распределена на отрезке $[0, 1)$. Позднее Н.М. Коробов показал[70], что из равномерного распределения элементов указанной последовательности следует равномерное распределение коэффициентов разложения числа α в системе счисления по основанию b. В § 1.2.3 автором доказана теорема, являющаяся прямым следствием результатов Н.М. Коробова.

Представим число α в виде быстро сходящегося ряда

$$\alpha = \sum_{n=0}^{\infty} \omega_n$$

где $\omega_n \in \mathbb{Q}$ и найдется индекс $n_0 \in \mathbb{N}$ такой, что для любого индекса $n \geqslant n_0$ будет выполнено неравенство

$$0 < |\omega_n| < f(n)b^{-n}, \quad 0 < f(n) < 1, \quad \lim_{n\to\infty} f(n) = 0,$$

для некоторой функции $f(n)$ натурального аргумента n.

Определим начальные значения $\alpha_0 = \alpha$, $\delta_{-1} = 0$ и последовательность рациональных величин

$$\alpha_n = b\delta_{n-1} + \omega_n b^n, \quad a_n = \lfloor \alpha_n \rfloor, \quad \delta_n = \alpha_n - a_n, \quad n = 0,\ 1,\ \ldots,$$

где $\delta_n, \omega_n, \alpha_n \in \mathbb{Q}$, $\delta_{-1}, a_n \in \mathbb{Z}$.

Теорема 2.6. *Пусть $\alpha > 0$ иррациональное число из рассматриваемых классов чисел. Тогда коэффициенты систематической дроби числа α по основанию b равномерно распределены на интервале $[0, b-1]$, если последовательность величин $(\delta_k)_{k=0}^{\infty}$ является реализацией равновероятно распределенной на интервале $[0, 1)$ случайной величины.*

[67] Champernowne D.G. The Construction of the Decimals Normal in the Scale of Ten // Journal Of London Mathematical Society. — 1933. — Vol. 8. — P. 254–260.

[68] Copeland A.H., Erdos P. Note on Normal Numbers // Bulletin Of American Mathematical Society. — 1946. — Vol. 52. — P. 857–860.

[69] Кейперс Л., Нидеррайтер Г. Равномерно распределенные последовательности. — М. : Наука, 1985. — С. 408.

[70] Коробов Н.М. О некоторых вопросах равномерного распределения // Известия Академии наук СССР. Серия математическая. — 1950. — Т. 14. — С. 215–231.

Утверждение доказанной теоремы позволяет свести проверку гипотезы о нормальности числа α к проверке статистической гипотезы о равномерном распределении на интервале $[0,1)$ последовательности значений $(\delta_k)_{k=1}^{\infty}$, вырабатываемых в ходе представления числа α в виде систематической дроби.

Задача о восстановлении начального состояния генератора псевдослучайных чисел является хорошо известной[71,72]. Вместе с тем, применительно к систематическим дробям действительных иррациональных чисел данная задача, по видимому, впервые рассматривалась автором[73]. Для восстановления неизвестных параметров чисел из первого класса может быть использована доказанная в § 2.4.1 теорема.

Теорема 2.3. *Определим последовательность действительных чисел* α_k

$$\alpha_1 = s_r(\alpha) = \sum_{n=0}^{r} a_n b^{-n}, \quad \alpha_k = \alpha_{k-1} - u_{k-1}\xi_{k-1} \quad \textit{для} \quad k = 2, \ldots, m,$$

где величины $\xi_1, \ldots, \xi_{m-1}$ *удовлетворяют равенствам*

$$\alpha = \sum_{i=1}^{m} u_i \xi_i, \quad \xi_i = \sum_{n=0}^{\infty} \frac{b^{-n}}{(dn + x_i)^s}, \quad i = 1, \ldots, m.$$

Если для r *выполнены условия:*

1. *величина* $s_r(\alpha)$ *отлична от нуля,*
2. *выполнено неравенство* $u_m > \frac{2(b-1)(dr+x_m)^s}{b^r(1-b^{-r})}$,
3. *выполнено неравенство* $\sum_{i=1}^{m} u_i < (b-1)(dr)^s b^r$,

то для всех индексов $k = 1, \ldots, m$ *выполнены неравенства*

$$\left(\frac{u_k}{\alpha_k}\right)^{\frac{1}{s}} < x_k < \left(\frac{b}{\alpha_k(b-1)} \sum_{i=k}^{m} u_i\right)^{\frac{1}{s}}.$$

Утверждение теоремы 2.3 определяет верхние и нижние оценки неизвестных $x_1, \ldots, x_m$ Это позволяет предъявить алгоритм, перебирающий

[71]Иванов М.А., Чугунков И.В. Теория, практика и оценка качества генераторов псевдослучайных последовательностей. — М. : Кудиц-Образ, 2003. — С. 240.

[72]Поточные шифры. Результаты зарубежной открытой криптологии. — 1997.

[73]Нестеренко А.Ю. Алгоритм восстановления параметров одного класса иррациональных чисел // Известия Саратовского университета. Серия: Математика. Механика. Информатика. — 2013. — Т. 13, № 4 (часть 2). — С. 89–93.

неизвестные значения в интервалах, зависящих от величины r, определяющей точность приближения $s_r(\alpha)$ к числу α. Практическая работоспособность данного алгоритма, при небольших значениях m, была подтверждена автором диссертации экспериментально.

Еще один алгоритм восстановления неизвестных параметров чисел из рассматриваемых классов приводится автором в § 2.4.2. Числа из первого класса могут быть записаны в виде

$$\alpha = \sum_{n=0}^{\infty} \sum_{l=1}^{d} \frac{w_l}{dn + l} b^{-n},$$

где $w_l = u_i$ если найдется индекс $i \in \{1, \ldots, m\}$ такой, что $x_i = l$. В противном случае, $w_l = 0$. Для чисел из второго класса выполнено, полученное в ходе доказательства теоремы 2.1, равенство

$$\alpha = \sum_{i=1}^{m} x_i \xi_i, \quad \text{и} \quad \sum_{i=1}^{m} \xi_i = e,$$

где e основание натурального логарифма. В обоих случаях исходное число α образует целочисленное соотношение

$$c_1\xi_1 + \cdots + c_m\xi_m + c_{m+1}\alpha = 0, \quad c_1, \ldots, c_{m+1} \in \mathbb{Z},$$

в котором величины $\xi_1, \ldots, \xi_m$ и приближение $s_r(\alpha)$ к числу α известны нарушителю.

Задача поиска неизвестных значений $c_1, \ldots, c_{m+1}$ в указанном целочисленном соотношении ведет отсчет от расширенного алгоритма Эвклида. Случай произвольного натурального значения m исследовался большим числом авторов, включая Якоби, Пуанкаре, Минковского, Перрона и т.д. Первым алгоритмом, для которого получена оценка трудоемкости, полиномиальная от нормы разыскиваемого соотношения, является алгоритм Фергюссона и Фуркада[74]. Далее, появился целый ряд алгоритмов – LLL[75], HJLS[76] и PSLQ[77]. Последний из них наиболее пригоден для поиска неизвестных значений $c_1, \ldots, c_{m+1}$, а количество итераций алгоритма PSLQ

[74]Ferguson H.R.P., Forcade R.W. Generalization of the Euclidean algorithm for real numbers to all dimensions higher than two // Bulletin (New Series) of the American Mathematical Society. — 1979. — no. 1. — P. 912–914.

[75]Lenstra A.K., H.W. Lenstra H. W., Lovasz L. Factoring polynomials with rational coefficients // Mathematische Annalen. — 1981. — Vol. 4, no. 261. — P. 515–534.

[76]Polynomial Time Algorithms for Finding Integer Relations Among Real Numbers / J. Hastad, B. Just, J.C. Lagarias, C.P. Schnorr // SIAM Journal of Computing. — 1989. — Vol. 18. — P. 859–881.

[77]Ferguson H.R.P., Bailey D.H. A Polynomial Time, Numerically Stable Integer Relation Algorithm. — 1992.

может быть оценено[78] величиной $\frac{m(m+1)\ln(2^m N(c))}{2\ln\sqrt{2}}$, где $N(c) = \sqrt{\sum_{i=1}^{m+1} c_i^2}$ – норма неизвестного целочисленного соотношения.

Алгоритм поиска неизвестных значений был реализован автором на ЭВМ. Согласно экспериментальным исследованиям, алгоритм PSLQ успешно завершает свою работу в случае, когда задано рациональное приближение $s_r(\alpha)$ к числу α с точностью $r \geqslant \lceil (m+1)\log_r(N(c))\rceil$. Конкретные результаты вычислений приведены в § 2.4.2.

Естественной защитой генератора псевдослучайных последовательностей от приведенной атаки является отбрасывание первых $\lceil (m+1)\log_r(N(c))\rceil$ элементов последовательности коэффициентов $(a_n)_{n=1}^{\infty}$. В этом случае, нарушитель не может получить точное приближение к числу α и, воспользовавшись описанным выше методом, восстановить неизвестные параметры числа α.

С другой стороны, нарушитель может перехватить произвольный фрагмент последовательности коэффициентов $(a_n)_{n=k+1}^{k+r}$ и, рассматривая его как приближение к некоторому числу β, предпринять попытку найти неизвестные параметры числа β. Безуспешность такой попытки следует из доказанных автором в § 2.4.3 утверждений.

Теорема 2.4. *Разложение в систематическую дробь действительного иррационального числа* $\alpha = \sum_{n=0}^{\infty}\sum_{i=1}^{m}\frac{w_l}{dn+x_i}b^{-n}$ *совпадает с разложением в систематическую дробь действительного иррационального числа* β *того же вида тогда и только тогда, когда* $\beta = \alpha b^s$ *для некоторого целого числа* s.

Теорема 2.5 содержит аналогичное утверждение относительно действительных чисел вида $\alpha = \sum_{n=0}^{\infty}\frac{x_n}{n!}$. Таким образом, по известному фрагменту последовательности коэффициентов $(a_n)_{n=k+1}^{k+r}$ действительного числа α из рассматриваемых классов нарушитель не может восстановить неизвестные параметры другого числа β и, тем самым, выработать другие фрагменты последовательности коэффициентов $(a_n)_{n=k+r}^{\infty}$. Вопрос о возможности восстановления параметров некоторого числа β из других классов действительных чисел, в настоящее время является открытым.

В качестве примера практического применения рассмотренного метода генерации псевдослучайных последовательностей в § 2.6 описывается метод локальной аутентификации пользователей средства защиты информации, удовлетворяющий предъявляемым требованиям к эксплуатационным характеристикам и безопасности[79].

[78] Ferguson H.R.P., Bailey D.H., Arno S. Analysis of PSLQ, an integer relation finding algorithm // Mathematics Of Computation. — 1999. — Vol. 68, no. 225. — P. 351–369.

[79] Password Hashing Competition. — 2015.

В **третьей** главе диссертационной работы приводятся результаты исследований, позволившие обосновать целесообразность применения в средствах защиты информации равновероятных сжимающих отображений, представляющих собой линейные формы от значений взаимно-однозначных функций.

Введем в рассмотрение следующий класс сжимающих отображений. Пусть множество кодов аутентичности $\mathbb{A}$ есть конечная аддитивная абелева группа. Зафиксируем натуральные числа l, s, u и рассмотрим конечное множество $\mathbb{B}$ такое, что $|\mathbb{A}| = |\mathbb{B}|^u$. Рассмотрим множество отображений

$$\pi_n : \mathbb{V}_u(\mathbb{B}) \to \mathbb{A},\ n = 1, \ldots, l,$$

задающее взаимно-однозначное соответствие между векторным пространством $\mathbb{V}_u(\mathbb{B})$ и конечным множеством $\mathbb{A}$.

Определим множество сообщений $\mathbb{S}$ и множество ключей $\mathbb{K}$ равенствами

$$\mathbb{S} = \mathbb{V}_{lu}(\mathbb{B}) = \{(x_1, \ldots, x_{lu})\}, \quad \mathbb{K} = \mathbb{V}_{slu}(\mathbb{B}) = \{(k_1, \ldots, k_{slu})\},$$

где координаты $x_1, \ldots, x_{lu},\ k_1, \ldots, k_{slu} \in \mathbb{B}$, а также рассмотрим сжимающее отображение

$$g(k_1, \ldots, k_s, x) : \mathbb{V}_{s+1}(\mathbb{B}) \to \mathbb{B},$$

удовлетворяющее следующим свойствам.

1. При фиксированном наборе значений $k_1, \ldots, k_s \in \mathbb{B}$ отображение

   $$g(k_1, \ldots, k_s, x) = g(x) : \mathbb{B} \to \mathbb{B},$$

   является взаимно-однозначным отображением множества $\mathbb{B}$ в себя.

2. При фиксированном значении $x \in \mathbb{B}$ и любом значении $z \in \mathbb{B}$ уравнение $g(k_1, \ldots, k_s, x) = z$ имеет ровно $|\mathbb{B}|^{(s-1)}$ различных решений, относительно неизвестных $k_1, \ldots, k_s$.

3. Зафиксируем произвольные элементы $x, y \in \mathbb{B}$ и будем считать, что вычеты $z, t \in \mathbb{B}$ пробегают множество всех возможных значений. Тогда суммарное число решений системы уравнений

   $$\begin{cases} g(k_1, \ldots, k_s, x) = z, \\ g(k_1, \ldots, k_s, y) = t, \end{cases}$$

   относительно неизвестных $k_1, \ldots, k_s$, в точности равно $|\mathbb{B}|^s$.

Пусть $x = (x_1, \ldots, x_{lu}) \in \mathbb{S}$, $k = (k_1, \ldots, k_{slu}) \in \mathbb{K}$. Определим сжимающее отображение $h(x, k) : \mathbb{S} \times \mathbb{K} \to \mathbb{A}$

$$h(x, k) = \sum_{n=1}^{l} \pi_n(z_{(n-1)u+1}, \ldots, z_{nu-1}, z_{nu}),$$

где

$$z_i = g(k_{s(i-1)+1}, \ldots, k_{si}, x_i),$$

для всех $i = 1, \ldots, lu$. Определенное отображение $h(x, k)$ представляет собой класс ключевых функций хэширования, параметризованный отображениями $\pi_1, \ldots, \pi_l$ и функцией g. В § 3.2.1 диссертационной работы доказаны следующие утверждения, описывающие свойства введенного отображения $h(k, x)$.

Теорема 3.1. *Для любого $a \in \mathbb{A}$ и любого $k = (k_1, \ldots, k_{slu}) \in \mathbb{K}$ найдется ровно $|\mathbb{A}|^{(l-1)}$ элементов $x = (x_1, \ldots, x_{lu}) \in \mathbb{S}$ таких, что $h(x, k) = a$.*

Из утверждения теоремы 3.1 следует, что отображение $h(x, k)$ является равновероятной ключевой функцией хэширования относительно сжимаемых сообщений, поскольку

$$|\mathbb{A}|^{(l-1)} = \frac{|\mathbb{A}|^l}{|\mathbb{A}|} = \frac{|\mathbb{S}|}{|\mathbb{A}|},$$

и вероятность выбора случайного сообщения $x \in \mathbb{S}$ с заданным значением функции $h(x, k) = a$ не зависит от выбора ключа $k \in \mathbb{K}$, значения кода аутентичности $a \in \mathbb{A}$ и равна $|\mathbb{A}|^{-1}$.

Теорема 3.2. *Для любого $a \in \mathbb{A}$ и любого $x = (x_1, \ldots, x_{lu}) \in \mathbb{S}$ найдется ровно $|\mathbb{B}|^{u(sl-1)}$ элементов $k = (k_1, \ldots, k_{slu}) \in \mathbb{K}$ таких, что $h(x, k) = a$.*

Из теоремы 3.2 следует, что число ключей $k \in \mathbb{K}$ таких, что выполнено равенство $h(x, k) = a$ в точности равно

$$|\mathbb{B}|^{u(sl-1)} = \frac{|\mathbb{B}|^{slu}}{|\mathbb{B}|^u} = \frac{|\mathbb{K}|}{|\mathbb{A}|}$$

и вероятность выбора случайного ключа $k \in \mathbb{K}$ такого, что $h(x, k) = a$, не зависит от выбора сообщения $x \in \mathbb{S}$, значения кода аутентичности $a \in \mathbb{A}$ и равна $|\mathbb{A}|^{-1}$.

Теорема 3.3. *Для любых элементов $a, b \in \mathbb{A}$ и любых $x = (x_1, \ldots, k_{lu})$, $y = (y_1, \ldots, y_{lu})$ из множества $\mathbb{S}$ найдется не более $|\mathbb{B}|^{u(sl-1)}$ элементов $k = (k_1, \ldots, k_{slu})$ таких, что*

$$\begin{cases} h(x, k) = a, \\ h(y, k) = b. \end{cases}$$

Утверждение теоремы 3.3 позволяет получить оценку условной вероятности выбора ключа k такого, что $h(x, k) = a$ при условии, что

$h(y,k) = b$. Приведенный в § 3.2 пример показывает, что полученная оценка является достижимой.

Для построения равновероятного отображения $h(x,k)$ использовался подход, ведущий свое начало от работ Дж. Картера и М. Вегмана[80,81], а также Д. Стинсона[82]. Позднее подход развивался при построении ключевых функций хэширования в ряде работ[83,84,85,86]. Существенным развитием указанных работ является предъявленная автором диссертационной работы возможность применения построенного отображения для реализации аутентифицированного шифрования, т.е. режима работы произвольного блочного шифра, реализующего одновременный процесс шифрования и имитозащиты данных[87].

Формально, такой режим может быть определен следующим образом. Пусть $v \in \mathbb{N}$, $x, y, c \in \mathbb{S}$, $k_1, k_2 \in \mathbb{K}$ и $a \in \mathbb{A}$. Определим отображения

$$\textbf{\textit{authenc}}(k_1, k_2, iv, y, x) = \{c, a\} : \quad \mathbb{K} \times \mathbb{K} \times \mathbb{V}_v \times \mathbb{S} \times \mathbb{S} \to \mathbb{S} \times \mathbb{A},$$

$$\textbf{\textit{authdec}}(k_1, k_2, iv, y, c, a) = \{x, b\} : \quad \mathbb{K} \times \mathbb{K} \times \mathbb{V}_v \times \mathbb{S} \times \mathbb{S} \times \mathbb{A} \to \mathbb{S} \times \mathbb{V}_1$$

такие, что для любых $k_1, k_2 \in \mathbb{K}$, $iv \in \mathbb{V}_v$ и $y \in \mathbb{S}$ выполнено равенство

$$\textbf{\textit{authdec}}(k_1, k_2, iv, y, \textbf{\textit{authenc}}(k_1, k_2, iv, y, x)) = \{x, \textbf{\textit{true}}\}.$$

Будем говорить, что отображение ***authenc*** зашифровывает сообщение x и вычисляет код аутентичности (имитовставку) сообщений x, y, а отображение ***authdec*** расшифровывает шифртекст c и проверяет код аутентичности (имитовставку) сообщений x, y.

Примером режима аутентифицированного шифрования служит режим, регламентируемый Р 1323565.1.026-2019. Другой пример предложен автором в § 3.3.1. Для его описания необходимо определить следующие элементарные преобразования.

---

[80]Carter J.L., Wegman M.N. Universal Classes of Hash Functions // Journal Of Computer and System Sciences. — 1979. — Vol. 18. — P. 143–154.

[81]Wegman M.N., Carter J.L. New Hash Functions and their Use in Authentication and Set Equality // Journal of Computer and System Sciences. — 1981. — Vol. 22, no. 3. — P. 265–279.

[82]Stinson D.R. Universal hashing and message authentication codes // Designs, Codes, and Cryptography. — 1994. — Vol. 4, no. 4. — P. 369– 380.

[83]Etzel M., Patel S., Ramzan Z. Square Hash: Fast Message Authentication via Optimized Universal Hash Functions // Advances in Cryptology – Crypto 99. — Springer, 1999. — P. 234–251.

[84]Halevi S., Krawczyk H. MMH: Software Message Authentication in the Gbit/second Rates // Proceedings Of Fast Software Encryption. — Springer, 1997. — P. 172–189.

[85]Nandi M. On the Minimum Number of Multiplications Necessary for Universal Hash Constructions. — 2013.

[86]UMAC: Fast and Secure Message Authentication / J. Black, Halevi S., H. Krawczyk et al. // Advances in Cryptology – Crypto 99. — Springer, 1999. — P. 216–233.

[87]Bellare M., Namprempre C. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm // Advances in Cryptology — ASIACRYPT 2000 / Ed. by T. Okamoto. — Berlin : Springer, 2000. — P. 531–545.

1. Алгоритм блочного шифрования $E_k(x) : \mathbb{V}_{256} \times \mathbb{V}_w \to \mathbb{V}_w$, где w это длина блока алгоритма шифрования, $w \in \{64, 128\}$.

2. Отображение $\phi : \mathbb{V}_8 \to \mathbb{V}_8$, представляющее собой нелинейную перестановку на двоичных векторах длины 8, определяемую в стандарте ГОСТ Р 34.11-2012.

3. Отображение $S : \mathbb{V}_w \to \mathbb{V}_w$, определяемое равенством

$$S(x) = (\phi(x_0)|| \dots ||\phi(x_{(w/8)-1})).$$

4. Линейный оператор $L(x) : \mathbb{V}_w \to \mathbb{V}_w$, который представляет собой умножение двоичного вектора $x \in \mathbb{V}_w$ на фиксированную, обратимую матрицу $L \in GL_w(\mathbb{F}_2)$.

5. Отображение $\pi : \mathbb{V}_w \times \mathbb{V}_w \to \mathbb{V}_{2w}$, зависящее от четырех констант $\zeta_0, \dots, \zeta_3 \in \mathbb{V}_w$ и представляющее собой 4-х раундовую сеть Фейстеля с раундовой функцией $L(S(x \oplus \zeta))$. Аналитически, отображение π может быть записано следующим образом.

$$\pi(x_0||x_1) = (\underbrace{x_2 \oplus L(S(x_3 \oplus \zeta_2))}_{x_4} || \underbrace{x_3 \oplus L(S(x_4 \oplus \zeta_3))}_{x_5}) = (x_4||x_5),$$

где $x_2 = x_0 \oplus L(S(x_1 \oplus \zeta_0), \quad x_3 = x_1 \oplus L(S(x_2 \oplus \zeta_1)$.

6. Отображение $G_{\gamma_{-1},\alpha}(n) = \gamma_{-1}\alpha^{n+1}$, в котором α является примитивным элементом конечного поля $\mathbb{F}_{2^{2w}}$, порожденного следующим примитивным многочленом

$2w$	$p(x)$
128	$x^{128} + x^7 + x^2 + x + 1$
256	$x^{256} + x^{10} + x^5 + x^2 + 1$

а также произвольным элементом $\gamma_{-1} \in \mathbb{F}_{2^{2w}}$. В дальнейшем мы будем использовать обозначение $\gamma_n = (\gamma_{n,0}||\gamma_{n,1}) = G_{\gamma_{-1},\alpha}(n)$, $n = 0, 1, \dots$.

Мы будем представлять открытые данные, подлежащие зашифрованию, а также ассоциированные данные, как конкатенацию фрагментов фиксированной длины

$$x = x_0|| \dots ||x_{l-1}||x_l, \quad y = y_0|| \dots ||y_{r-1}||y_r,$$

где $\mathrm{len}_2(x_0) = \dots = \mathrm{len}_2(x_{l-1}) = \mathrm{len}_2(y_0) = \dots = \mathrm{len}_2(y_{r-1}) = w$, а также $\mathrm{len}_2 x_l \leqslant w$, $\mathrm{len}_2 y_r \leqslant w$ и $l, r \in \mathbb{N}_0$. Кроме того, введем ограничение на общую длину открытых данных и будем считать, что $\mathrm{len}_2 x \geqslant 2w$.

Разобьем входные данные на пары и определим число пар равенствами

$$l_0 = l \pmod 2 + \left\lfloor \frac{l}{2} \right\rfloor, \quad r_0 = r \pmod 2 + \left\lfloor \frac{r}{2} \right\rfloor.$$

Вместе с каждой парой блоков будет преобразовываться элемент последовательности γ_n, при этом, элементы $\gamma_0, \ldots, \gamma_{r_0-1}$ будут соответствовать парам ассоциированных данных, а элементы $\gamma_{r_0}, \ldots, \gamma_{r_0+l_0-1}$ – парам открытого текста.

Определим процедуру зашифрования пары блоков открытых данных равенствами

$$\begin{aligned} c_{2n} &= E_{k_1}(x_{2n} \oplus \gamma_{n+r_0,0}) \oplus \gamma_{n+r_0,0}, \\ c_{2n+1} &= E_{k_1}(x_{2n+1} \oplus \gamma_{n+r_0,1}) \oplus \gamma_{n+r_0,1}, \end{aligned}$$

для $n = 0, 1, \ldots, l_0 - 1$ и $\gamma_n \in \mathbb{F}_{2^{2w}}$ определенного выше. Тогда, зашифрованный текст определяется равенством

$$c = (c_0 || \ \ldots \ || c_{l-1} || \operatorname{lsb}_{\operatorname{len}_2(x_l)}(c_l)).$$

Данный способ зашифрования иногда называют «гамма-коммутатор-гамма» или, в англоязычной литературе, «xor-encryption-xor»[88].

Теперь рассмотрим натуральное число m, удовлетворяющее неравенствам $1 \leqslant m \leqslant 2w$, и определим код аутентификации длины m следующими равенствами

$$\begin{aligned} s = (s_0||s_1) = \sum_{n=0}^{r_0-1} \pi\big(E_{k_1}(y_{2n} \oplus \gamma_{n,0})||E_{k_1}(y_{2n+1} \oplus \gamma_{n,1})\big) \oplus \\ \sum_{n=r_0}^{r_0+l_0-1} \pi\big(E_{k_1}(x_{2(n-r_0)} \oplus \gamma_{n,0})||E_{k_1}(x_{2(n-r_0)+1} \oplus \gamma_{n,1})\big) \oplus \\ \pi\big(E_{k_1}(\operatorname{len}_2(y) \oplus \gamma_{r_0+l_0,0})||E_{k_1}(\operatorname{len}_2(x) \oplus \gamma_{r_0+l_0,1})\big), \end{aligned}$$

и $a = \operatorname{msb}_m\Big(E_{k_2}(s_0)||E_{k_2}(s_1 \oplus E_{k_2}(s_0))\Big)$.

Как видно из приведенных равенств, значение суммы s зашифровывается на ключе k_2 в режиме простой замены с зацеплением и с использованием нулевого инициализационного вектора. Окончательным кодом аутентификации служат старшие m бит вектора, полученного в результате шифрования.

Для построенного отображения автором доказана следующая теорема.

Теорема 3.4. *Пусть блочный шифр $E_k(x) : \mathbb{V}_w \to \mathbb{V}_w$ является перестановкой множества $\mathbb{V}_w$ для любого фиксированного значения ключа $k \in \mathbb{K}$. Тогда, для любых ключей шифрования и имитозащиты $k_1, k_2 \in \mathbb{K}$, а также инициализационного вектора $iv \in \mathbb{V}_{6w}$ определенное выше сжимающее отображение* $\texttt{authenc}(k_1, k_2, iv, x, y)$*, вычисляющее пару значений*

[88] Lyskov M., Rivest R., Wagner D. Tweakable Block Ciphers // Journal Of Cryptology. — 2011. — Vol. 24. — P. 588–613.

$\{c, a\}$, обладает свойством равновероятности, т.е. для любого $a \in \mathbb{V}_{2w}$ найдется в точности

$$2^{\mathrm{len}_2(x)+\mathrm{len}_2(y)-2w}$$

пар x, y, для которых код аутентичности, вырабатываемый отображением $\texttt{authenc}(k_1, k_2, iv, x, y)$, *совпадает с a.*

Свойство равновероятности позволяет обеспечить защиту предложенного режима аутентифицированного шифрования от атак, направленных на подделку и навязывание передаваемой информации. Детальное рассмотрение подходов к построению коллизий для построенного отображения содержится в § 3.3.4.

В рамках разработанного автором программного СКЗИ с открытыми исходными текстами[89], были получены следующие показатели скорости работы различных алгоритмов аутентифицированного шифрования для блочного шифра «Магма» (вычисления производились на персональной ЭВМ с процессором Intel (i5-8250U) и тактовой частотой 1.60GHz), см. таблицу 3.1.

Режим	Скорость, МБс	%
ecb-magma	49,411111	100
mgm-magma	23,424500	47
ctr-cmac-magma	24,101876	48
ctr-hmac-magma-streebog256	35,713519	72
ctr-hmac-magma-streebog512	35,713049	72
xtsmac-magma	45,877878	92

Таблица 1.5: Режимы аутентифицированного шифрования для шифра «Магма».

Аналогичные результаты были получены и для блочного шифра «Кузнечик»[90]. Из приведенных значений следует, что предложенный автором алгоритм аутентифицированного шифрования позволяет достичь наибольшей скорости при программной реализации на универсальных процессорах.

В последней, **четвертой**, главе диссертационной работы рассматриваются вопросы разработки и обоснования безопасности криптографических протоколов защищенного взаимодействия. К таким протоколам от-

[89] Libakrypt: software crypto module for user space. – 2022. – (in accordance with R 1323565.1.012-2017) – https://git.miem.hse.ru/axelkenzo/libakrypt-0.x.

[90] Блочные шифры «Магма» и «Кузнечик» регламентируются стандартом Российской Федерации ГОСТ Р 34.12-2015.

носятся транспортные протоколы такие, как MACSec[91], L2TP[92], DTLS[93] и т.п., а также криптографические схемы асимметричного и гибридного шифрования, не предполагающие интерактивного обмена в процессе зашифрования сообщения. Примерами таких схем служат применяемые на практике варианты классических схем RSA[94], NTRU[95] или МакЭлиса[96], стандартизированные[97,98] или предлагаемые к стандартизации решения[99]. Вторым классом рассматриваемых протоколов являются протоколы аутентификации и выработки общего ключа. Примерами таких протоколов служат TLS[100,101], семейство протоколов SIGMA[102], протоколы IKEv2[103,104], а также схемы выработки общего ключа с аутентификацией[105] и протоколы промышленного «Интернета вещей»[106,107].

[91]IEEE 802.1AE-2018 – IEEE Standard for Local and metropolitan area networks-Media Access Control (MAC) Security. — 2018.

[92]Layer Two Tunneling Protocol «L2TP» / W. Townsley, A. Valencia, A. Rubens et al. — 1999. — RFC 2661.

[93]Rescorla E., Tschofenig H., Modadugu N. The Datagram Transport Layer Security (DTLS) Protocol Version 1.3. — 2022. — RFC 9147.

[94]PKCS #1: RSA Cryptography Specifications Version 2.2 / K. Moriarty, B. Kaliski, J. Jonsson, A. Rush. — 2016. — RFC 8017.

[95]Chen C., Danba O., Hoffstein J. et al. NTRU: Algorithm Specifications And Supporting Documentation. — 2019.

[96]Albrecht M., Bernstein D., Chou T. et al. Classic McEliece: conservative code-based cryptography. — 2020.

[97]ISO/IEC 18033-2:2006. Information technology. Security techniques. Encryption algorithms — Part 2: Asymmetric ciphers. — 2006.

[98]Р 1323565.1.025.–2019 Информационная технология. Криптографическая защита информации. Форматы сообщений, защищенных криптографическими методами. — М. : Стандартинформ, 2019.

[99]Aragon N., Barreto P., Bettaieb S. et al. BIKE: Bit Flipping Key Encapsulation. — 2021.

[100]Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.3. — 2018. — RFC 8446.

[101]Р 1323565.1.030.–2020 Информационная технология. Криптографическая защита информации. Использование криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.3). — М. : Стандартинформ, 2020.

[102]Krawczyk H. SIGMA: The «SIGn-and-MAc» Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols // Advances in Cryptology - CRYPTO 2003. — 2003. — P. 400–425.

[103]Internet Key Exchange Protocol Version 2 (IKEv2) / C. Kaufman, P. Hoffman, Y. Nir et al. — 2014. — RFC 7296.

[104]МР 26.2.001.–2022 Информационная технология. Криптографическая защита информации. Использование российских криптографических алгоритмов в протоколе обмена ключами в сети Интернет версии 2 (IKEv2). — М.:ТК26, 2022.

[105]Р 1323565.1.004.–2017 Информационная технология. Криптографическая защита информации. Схемы выработки общего ключа с аутентификацией на основе открытого ключа. — М. : Стандартинформ, 2017.

[106]Р 1323565.1.028.–2019 Информационная технология. Криптографическая защита информации. Криптографические механизмы защищенного взаимодействия контрольных и измерительных устройств. — М. : Стандартинформ, 2019.

[107]Р 1323565.1.032.–2020 Информационная технология. Криптографическая защита информации. Использование российских криптографических механизмов для реализации обмена данными по протоколу DLMS. — М.:Стандартинформ, 2020.

В § 4.1 уточняются общие модели угроз и нарушителя, используемые далее при оценке безопасности криптографических протоколов. Целью такой оценки является определение численных значений одного или нескольких показателей эффективности мер защиты, реализуемых криптографическим протоколом. При этом, система считается защищенной (безопасной), если полученные в ходе исследования значения показателей попадают в заданную область, установленную нормативными, правовыми документами или требованиями по безопасности.

В § 4.2 рассматривается предложенная автором, совместно с А.В. Пугачевым[108], базовая гибридная схема шифрования ECISPE[109], использующая в качестве шифрующего преобразования полином малой степени. Данная схема использует как асимметричные, так и предварительно распределенные ключи. Доказывается теорема 4.1 о сведении стойкости предложенной гибридной схемы к решению задач дискретного логарифмирования и Диффи-Хеллмана в группе точек эллиптической кривой, задаче определения элементов двоичных последовательностей, вырабатываемых ДСЧ, а также задачи построения коллизии для функции выработки имитовставки. Результаты, полученные в предыдущих главах диссертации, позволяют обеспечить высокую трудоемкость решения указанных задач.

Также рассматриваются несколько модификаций базовой схемы ECISPE, позволяющих изменить её эксплуатационные особенности без изменения стойкости, в частности, предлагается протокол передачи ключевой информации. Данный протокол представляет собой вариант гибридной схемы, обеспечивающий возможность зашифровывать информацию на ключах выработанных заранее, после чего передавать данные ключи вместе с зашифрованной информацией. Также, необходимость передачи ключевой информации от одного субъекта к другому возникает в автоматизированных системах с централизованным изготовлением ключевой информации.

В § 4.3 рассматривается предложенный автором протокол «Крокус». Целью выполнения данного протокола является взаимная аутентификация двух субъектов взаимодействия и последующее однократное выполнение операции типа «запрос-ответ». Данная операция востребована при обеспечении криптографической защиты запросов в удаленные базы данных, а также Интернет-протоколов HTTP и Gemini.

Протокол «Крокус» представляет собой модификацию схемы Диффи-Хеллмана выработки общего ключа, реализуемую в группе точек эллиптической кривой, и состоит из четырех последовательно выполняемых

[108]Нестеренко А.Ю., Пугачев А.В. Об одной схеме гибридного шифрования // Прикладная дискретная математика. — 2015. — № 4. — С. 56–71.

[109]англ. Elliptic Curve based Integrated Scheme with Polynomial Encryption.

фаз: фазы взаимной аутентификации, фазы выработки общего ключа, фазы подтверждения выработанного общего ключа и фазы обмена зашифрованными сообщениями типа «запрос-ответ». Все перечисленные фазы реализуются в ходе обмена шестью сообщениями.

Автором доказывается теорема 4.2 о сведении стойкости протокола «Крокус» к решению следующих задач: задач дискретного логарифмирования и Диффи-Хеллмана в группе точек эллиптической кривой, задаче подделки электронной подписи, задаче определения элементов двоичных последовательностей, вырабатываемых ДСЧ, а также задачи построения прообраза для функции выработки производного ключа.

Также приводится разработанная автором, совместно с А.М. Семеновым и П.А. Лебедевым, модификация протокола «Крокус», направленная на снижение числа передаваемых в ходе выполнения протокола сообщений и сохранение уровня стойкости протокола. Данная модификация была успешно применена для защиты каналов управления контрольными и измерительными устройствами, и стандартизирована к качестве рекомендаций Р 1323565.1.028.–2019 (протокол SP FIOT).

В § 4.4 обосновывается общая методика оценки безопасности криптографических протоколов, частные случаи применения которой иллюстрируются результатами § 4.2 и § 4.3. При разработке методики рассматривалось несколько зарубежных подходов к моделированию криптографических протоколов: базовая модель Белларе-Рогавея[110] и ее модификации[111,112], модель Конетти-Кравчука[113] и ее модификации[114,115], а также ряд подходов, предназначенных для верификации протоколов. Рассматривался отечественный подход, основанный на классическом криптографическом анализе для получения оценок стойкости используемых базовых преобразований[116], а также возможность применения теории «доказуемой стойкости», позволяющей исследовать безопасность протоколов в

[110]Bellare M., Rogaway P. Entity authentication and key distribution // Advances in Cryptology – Crypto '93. — Vol. 773 Of Lecture Notes Of Computer Science. — Springer, 1993. — P. 232–249.

[111]Blake-Wilson S., Johnson D., Menezes A. Key agreement protocols and their security analysis // Crypography and Coding – 6th IMA Conference. — Springer, 1997. — P. 20–45.

[112]364 Bellare M., Rogaway P., Pointcheval D. Authenticated key exchange secure against dictionary attacks // Advances in Cryptology – EUROCRYPT 2000. — Springer, 2000. — P. 139–155.

[113]Canetti R., Krawczyk H. Analysis of key-exchange protocols and their use for building secure channels // Advances in Cryptology – EUROCRYPT 2001. — P. 453–474.

[114]LaMacchia B.A., Lauter K., Mityagin A. Stronger security of authenticated key exchange // Provable Security, First International Conference, ProvSec 2007 — P. 1–16.

[115]Menezes A., Ustaoglu B. On the importance of public-key validation in the MQV and HMQV key agreement protocols // Progress in Cryptology - INDOCRYPT 2006. — P. 133–147.

[116]Об основных концепциях криптографической стойкости / И.Ф. Качалин, А.С. Кузьмин, Е.А. Суслов и др. // Тезисы XII Всероссийской школы-коллоквиума по стохастическим методам и VI Всероссийского симпозиума по прикладной и промышленной математике. — 2005. — С. 982–983. — Сочи-Дагомыс, 1-7 октября 2005 г.

заданных вероятностных моделях поведения нарушителя с ограниченными вычислительными ресурсами[117,118].

При разработке методики учитывались следующие факторы: необходимость проведения анализа по формальным моделям используемых на практике протоколов; необходимость формализации предъявляемых к протоколу требований (свойств безопасности); вывод численных значений показателей эффективности мер защиты, должен осуществляться с учетом результатов, полученных при оценке стойкости базовых криптографических преобразований, например, алгоритмов блочного шифрования, функций хеширования и выработки имитовставки, алгоритмов выработки электронной подписи и т.п. (в качестве таких показателей, традиционно, выступают вероятность π и трудоемкость Q успешной реализации алгоритмов компрометации криптографических преобразований), необходимость учета атак на криптографические протоколы, успешность применения которых не зависит от свойств используемых криптографических преобразований.

В результате исследований была выработана модель, представляющая криптографический протокол в виде дискретной динамической системы, множество внутренних состояний которой образует информация, приводящая к компрометации предъявляемых к протоколу свойств. В рамках данной модели были описаны основные свойства безопасности, а также метод подсчета численных оценок указанных выше показателей эффективности мер защиты. Разработанная методика была успешно применена при анализе ряда отечественных криптографических протоколов, что привело к утверждению их в качестве национальных рекомендаций по стандартизации.

---

[117]Обзор уязвимостей некоторых протоколов выработки общего ключа с аутентификацией на основе пароля и принципы построения протокола SESPAKE / Е. К. Алексеев, Л. Р. Ахметзянова, И. Б. Ошкин, С. В. Смышляев // Математические вопросы криптографии. — 2016. — Vol. 7, no. 4. — P. 7–28.

[118]Akhmetzyanova L., Alekseev A., Sedov G., Smyshlyaev S. On Security of TLS 1.2 Record Layer with Russian Ciphersuites // Proceedings of 8-th Workshop on Current Trends in Cryptology (CTCrypt 2019). – 2019. – pp. 253-292.

глава 1

Вопросы применения эллиптических кривых в средствах защиты информации

В настоящей главе излагаются результаты, позволившие обосновать целесообразность применения в средствах защиты информации эллиптических кривых, определенных над конечным простым полем $\mathbb{F}_p$.

В первом параграфе ставятся задачи вычисления кратной точки и дискретного логарифмирования в группе точек эллиптической кривой, приводится обзор известных методов решения поставленных задач.

Во втором параграфе рассматривается класс алгоритмов поиска длин циклов в последовательностях и приводится алгоритм решения задачи дискретного логарифмирования, основанный на поиске длины цикла в последовательности точек эллиптической кривой.

В третьем параграфе вводится понятие «слабого» ключа для асимметричной схемы защиты информации, стойкость которой основывается на задаче дискретного логарифмирования, и приводится алгоритм дискретного логарифмирования, алгоритмическая сложность которого зависит от мультипликативного порядка неизвестного ключа.

В четвертом параграфе приводится метод представления эндоморфизмов эллиптических кривых в виде пары рациональных функций, приводятся результаты практической реализации данного метода, а также формы эллиптических кривых, в которых построенные эндоморфизмы имеют невысокую алгоритмическую сложность вычисления. Доказывается теорема о представлении натуральных чисел значениями многочленов с целыми коэффициентами в точках мнимого квадратичного поля, утверждение теоремы используется при реализации алгоритма вычисления кратных точек эллиптических кривых с использованием построенных эндоморфизмов.

В последнем параграфе формулируются требования к эллиптическим кривым, позволяющие минимизировать множество «слабых» ключей, а также приводится алгоритм построения кривых, удовлетворяющих сформулированным требованиям; параметры построенных с помощью данного алгоритма эллиптических кривых приводятся в приложении А.

Изложенные в настоящей главе результаты опубликованы в следующих работах автора [173, 175, 176, 179, 311, 323, 325, 326, 333], из которых шесть работ входят в перечень рецензируемых научных изданий ВАК.

§ 1.1. Мотивация и обзор известных результатов

Основным множеством, на котором реализуются современные механизмы защиты информации, является множество точек эллиптической кривой, определенной над конечным простым полем.

Подробное изложение теории эллиптических кривых может быть найдено в монографиях [109, 233, 314, 384, 393]. Приложения эллиптических кривых к вопросам защиты информации ведут отсчет от работ Н. Коблица [128] и В. Миллера [162], детали применяемых на практике механизмов защиты информации могут быть найдены в книгах [38, 161, 316].

Определение 1.1 (см. [233]). *Пусть $\mathbb{F}$ – произвольное поле, характеристика которого отлична от двух и трех. Рассмотрим элементы $a, b \in \mathbb{F}$ такие, что $4a^3 + 27b^2 \neq 0$, а также однородное уравнение*

$$\mathcal{E}_{a,b}(\mathbb{F}): \quad x_2^2 x_3 = x_1^3 + a x_1 x_3^2 + b x_3^3. \tag{1.1}$$

Множество точек проективного пространства $\mathbb{P}^2(\mathbb{F})$, удовлетворяющих уравнению (1.1), будем называть эллиптической кривой определенной над полем $\mathbb{F}$ в короткой форме Вейерштрасса.

Точку $(0:1:0)$, удовлетворяющую равенству (1.1), будем называть бесконечно удаленной точкой эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F})$ и обозначать символом $\mathcal{O}$.

На множестве точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F})$ определена структура аддитивной абелевой группы. Пусть $P = (x_1 : x_2 : x_3)$ произвольная точка кривой, тогда равенства

$$\begin{aligned} z_1 &= 2x_2x_3(9x_1^4 + 6ax_1^2x_3^2 + a^2x_3^4 - 8x_1x_2^2x_3), \\ z_2 &= 36x_1^3x_2^2x_3 - 27x_1^6 - 27ax_1^4x_3^2 - 8x_2^4x_3^2 + 12ax_1x_2^2x_3^3 - \\ &\qquad -9a^2x_1^2x_3^4 - a^3x_3^6, \\ z_3 &= 8x_2^3x_3^3. \end{aligned} \tag{1.2}$$

задают операцию «удвоения» точки P, см. [132, 233], и определяют точку $[2]P = (z_1 : z_2 : z_3)$, также удовлетворяющую равенству (1.1).

Для двух различных точек $P_1 = (x_1 : x_2 : x_3)$ и $P_2 = (y_1 : y_2 : y_3)$ проективного пространства $\mathbb{P}^2(\mathbb{F})$ равенства

$$\begin{aligned} z_1 &= (y_3x_1 - y_1x_3)\times \\ &\quad \times\big[y_3x_3\left(y_1x_1(x_1y_3 + y_1x_3) + (y_3x_2 - y_2x_3)^2\right) - \\ &\qquad -(x_1^3y_3^3 + y_1^3y_3^3)\big], \\ z_2 &= y_3^4x_2(x_2^2x_3 + x_1^3) + x_3^4y_2(y_2^2y_3 - y_1^3) + \\ &\quad + y_2y_3^2x_3(3x_2^2x_3 - 2x_1^3) + y_1^2y_3x_2x_3^2(2y_1x_3 - 3y_3x_1) + \\ &\qquad + 3y_2y_3^2x_3^2(y_1x_1^2 - y_2x_2x_3), \\ z_3 &= x_3y_3(y_3x_1 - y_1x_3)^3. \end{aligned} \tag{1.3}$$

определяют операцию «сложения», т.е. точку $P_1 + P_2 = (z_1 : z_2 : z_3)$. В частности, выполнены следующие соотношения

$$(x_1 : x_2 : x_3) + (x_1 : -x_2 : x_3) = (x_1 : -x_2 : x_3) + (x_1 : x_2 : x_3) = \mathcal{O},$$
$$(x_1 : x_2 : x_3) + \mathcal{O} = \mathcal{O} + (x_1 : x_2 : x_3) = (x_1 : x_2 : x_3),$$

из которых следует, что $\mathcal{O}$ является нейтральным элементом, а вектор $(x_1 : -x_2 : x_3)$ является обратным к вектору $(x_1 : x_2 : x_3)$.

У каждой точки эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F})$, отличной от $\mathcal{O}$, найдется представитель $(x : y : 1)$. Такой представитель удовлетворяет равенству

$$y^2 = x^3 + ax + b, \tag{1.4}$$

определяющему эллиптическую кривую в аффинной форме. Преобразование из проективной формы в аффинную определено равенствами

$$\varphi : \ (x_1 : x_2 : x_3) \to (x, y)\,,$$
$$x = \frac{x_1}{x_3}, \quad y = \frac{x_2}{x_3},$$

для всех $x_3 \neq 0$. При этом, соотношения (1.2) и (1.3) принимают хорошо известный вид

$$\begin{cases} x_3 &= \mu^2 - x_1 - x_2, \\ y_3 &= \mu(x_1 - x_3) - x_2, \end{cases} \tag{1.5}$$

где $\mu = \dfrac{y_2 - y_1}{x_2 - x_1}$ для (1.2) и $\mu = \dfrac{3x_1^2 + a}{2y_1}$ для (1.3).

Основным преобразованием, используемым в средствах защиты информации, является операция вычисления «кратной» точки эллиптической кривой.

Определение 1.2. *Пусть $p > 3$ простое число и $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ произвольная точка эллиптической кривой, определенной над конечным простым полем $\mathbb{F}_p$. Точка $Q \in E_{a,b}(\mathbb{F}_p)$ называется точкой кратности $k \in \mathbb{N}$, если*

$$Q = [k]P = \underbrace{P + \cdots + P}_{k \text{ раз}}. \tag{1.6}$$

Задача определения неизвестного значения k по известным точкам P, Q называется задачей дискретного логарифмирования в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$.

Алгоритмическая сложность нахождения точки Q по заданным значениям k и P, в большинстве случаев, определяет сложность реализации всего механизма защиты информации. Алгоритмическая сложность решения задачи дискретного логарифмирования определяет стойкость механизма защиты информации. Рассмотрим указанные вопросы более детально.

§ 1.1.1. Вычисление кратной точки

При практической реализации криптографических механизмов в средствах защиты информации принято проводить вычисления в группах точек эллиптических кривых, определяемых над конечными полями $\mathbb{F}_p$ для простых чисел p, принадлежащих интервалу

$$2^{160} < p < 2^{640}. \tag{1.7}$$

Для снижения алгоритмической сложности вычисления кратной точки эллиптической кривой принято использовать комбинации из одного или нескольких описываемых ниже подходов.

1. Оптимизация элементарных операций сложения, умножения, а также взятия обратного элемента в поле $\mathbb{F}_p$.

   Классические алгоритмы для реализации операций сложения и умножения больших целых чисел могут быть найдены в монографии Д.Кнута [303]. При этом асимптотически быстрые алгоритмы такие, как алгоритм А.Шёнхаге и В.Штрассена [221] или алгоритм М.Фюрера [93], не дают выигрыша при вычислениях с целыми числами, лежащими в диапазоне (1.7).

   Используемые на практике оптимизации операции умножения в поле $\mathbb{F}_p$ ведут начало от работ П.Барретта [19] и П.Монтгомери [163]. Различия между указанными работами приводятся в статье [47].

   Несколько модификаций операции умножения в представлении Монтгомери можно найти в работе американских авторов [124]. Алгоритмы, реализующие операцию вычисления обратного элемента в представлении Монтгомери, содержатся в работах Р. Лоренца [148, 149], см. также статью [220]. Обзор разработанных к настоящему времени алгоритмов, использующих представление Монтгомери для реализации операций в поле $\mathbb{F}_p$, можно найти в книге Д.Боса и А.Ленстры [250].

   Другим, менее распространенным подходом, является представление больших целых чисел в системе остаточных классов [265, 267, 386]. Данный подход получил распространение при аппаратной реализации операций в группе точек эллиптической кривой на программируемых логических матрицах (FPGA) и графических ускорителях (GPU). Реализация представления Монтгомери в системе остаточных классов изучалась в работах [16, 117, 206], см. также статью П.А. Лебедева и А.Ю. Нестеренко [311].

Отдельно стоит выделить использование простых чисел специального вида. В рекомендациях [81] используются простые числа вида

$$p = \sum_{n=0}^{\frac{k}{w}} \delta_n 2^{nw}$$

где w – двоичная длина регистра вычислительного средства, как правило, $w = 64$, k натуральное число, удовлетворяющее условиям $k \equiv 0 \pmod{w}$ и $2^{k-1} < p < 2^k$, также выполнено условие $\delta_n \in \{-1, 0, 1\}$ для всех $n = 0, 1, \ldots \frac{n}{w}$.

В ряде работ также предлагается использовать простые числа вида $p = 2^k - \theta$, см. [20, 29]. Такие простые, в частности, используются в отечественных рекомендациях [362]. Обзор алгоритмов, реализующих элементарные операции в поле $\mathbb{F}_p$ для простых чисел указанного вида, может быть найден в работе Д.Бернштейна и Т.Ланге [32].

2. Использование различных представлений эллиптических кривых, позволяющих минимизировать количество элементарных операций в поле $\mathbb{F}_p$, необходимых для реализации операций сложения и удвоения точек эллиптической кривой.

   Помимо короткой формы Вейерштрасса (1.4), используются следующие представления.

   - Форма Монтгомери, см. [164, раздел 10.3.1], а также [58, 186, 250]

     $$by^2 \equiv x^3 + ax^2 + x \pmod{p}, \quad b(a^2 - 4) \not\equiv 0 \pmod{p}.$$

     Порядок эллиптической кривой в форме Монтгомери всегда кратен 2. Предложенный в работе [164] алгоритм не позволяет находить y-координату кратной точки, что, тем не менее, оказывается достаточным для реализации схем выработки общего ключа.

   - Форма Т. Хадано, см. [101], а также § 1.4.5, задаваемая сравнением

     $$y^2 \equiv x^3 + ax^2 + bx \pmod{p}, \quad a^2 + b^2 \not\equiv 0 \pmod{p}.$$

     Порядок эллиптической кривой в форме Хадано всегда кратен 2.

   - Форма Якоби, задаваемая сравнением

     $$y^2 \equiv (1 - x^2)(1 - \lambda x^2) \pmod{p}, \quad \lambda \not\equiv 0, 1 \pmod{p},$$

     см. [35, 58]. Порядок такой кривой всегда кратен 4.

- Форма пересечения Якоби, задаваемая системой сравнений

$$\begin{cases} x^2 + y^2 \equiv 1 \pmod p, \\ \lambda x^2 + z^2 \equiv 1 \pmod p, \end{cases} \qquad \lambda \not\equiv 0, 1 \pmod p,$$

  см. [58, 294]. Данное представление эллиптической кривой исследовалось в кандидатской диссертации автора, см. [323, 326]. Порядок кривой в форме пересечения Якоби всегда кратен 4.

- Форма Гессе

$$x^3 + y^3 + z^3 \equiv dxyz \pmod p, \quad d \not\equiv 0 \pmod p,$$

  см. статьи братьев Чудновских [58] и Н. Смарта [236]. Порядок данной кривой всегда кратен 3.

- Форма Эдвардса

$$x^2 + y^2 \equiv 1 + dx^2y^2 \pmod p, \quad d \not\equiv 0, 1 \pmod p,$$

  см. статьи Г. Эдвардса [75], а также Бернштейна и Ланге [31]. Порядок кривой в форме Эдвардса всегда кратен 4.

- Искривленная форма Эдвардса

$$ax^2 + y^2 \equiv 1 + dx^2y^2 \pmod p, \quad a, d \not\equiv 0, 1 \pmod p,$$

  см. статью [253]. В настоящее время данная форма считается наиболее эффективной для практической реализации, см. рекомендации [82, 362].

Обзоры и сравнительный анализ различных представлений эллиптических кривых см. в работах [3, 31, 58]. Дополнительный перечень публикаций см. в [30].

3. Использование проективных координат для реализации операций в группе точек эллиптической кривой. Начиная с работ [132, 164], см. также статью Г.Коэна [62], вычисление кратных точек с однократным обращением элемента в поле $\mathbb{F}_p$ реализуется повсеместно.

4. Использование алгоритмов вычисления кратной точки, минимизирующих количество операций в группе точек эллиптической кривой.

   Данный класс методов основан на представлениях экспоненты $k \in \mathbb{N}$ в виде

$$k = \sum_{\substack{n=0 \\ n_1 + \cdots + n_m = n}} k_n b_1^{n_1} \cdots b_m^{n_m},$$

где $b_1, \ldots b_m$ взаимно простые натуральные числа, являющиеся основаниями системы счисления, $m \geqslant 1$ – натуральное число, задающее число оснований, $k_0, k_1, \ldots$ – целые коэффициенты разложения экспоненты k, удовлетворяющие неравенствам $B_1 \leqslant k_n \leqslant B_2$ для некоторых натуральных чисел B_1, B_2, определяемых алгоритмом представления.

Обзор методов вычисления кратной точки для представлений с одним основанием системы счисления содержится в книге [38]. Обзор методов с несколькими основаниями см. в работе С.В. Гребнева [99].

5. Использование оптимизаторов программного кода, минимизирующих число используемых переменных и операций пересылки данных между регистрами вычислительного средства. Пример такой оптимизации описывается в работе [227].

6. Использование эндоморфизмов эллиптической кривой.

   Пусть $\phi : \mathcal{E}_{a,b}(\mathbb{F}_p) \to \mathcal{E}_{a,b}(\mathbb{F}_p)$ заданный эндоморфизм эллиптической кривой. В работе [95] Р. Галант, Р. Ламберт и С. Ванстоун предложили использовать для вычисления кратной точки равенство

   $$Q = [k]P = [k_1]P + [k_2]\phi(P),$$

   где k_1, k_2 целые, зависящие от числа k и эндоморфизма ϕ коэффициенты, удовлетворяющие неравенствам $0 \leqslant k_1, k_2 \leqslant c_0\sqrt{\mathtt{ord}(P)}$ для некоторой действительной константы c_0. Оценки для величины c_0 были получены в работе [231].

   Отметим, что независимо от авторов работы [95], данный метод рассматривался в работах А.Г. Ростовцева, см. [374, 375]. В дальнейшем метод развивался в работах [7, 94, 118, 168].

   Краткий перечень эллиптических кривых, для которых использование эндоморфизмов может ускорить операцию вычисления кратной точки, приведен работе [95]. Алгоритм построения большого класса эллиптических кривых с данным свойством предложен в работе А.Ю. Нестеренко [175], см. также § 1.4.

§ 1.1.2. Дискретное логарифмирование

Алгоритмическая сложность решения задачи дискретного логарифмирования определяет стойкость большого числа стандартизированных отечественных механизмов защиты информации, в частности: схемы электронной подписи ГОСТ Р 34.10-2012, см. [280], схем выработки общего

ключа с аутентификацией на основе открытого ключа, см. [352], протоколов безопасности транспортного уровня, см. [359, 367], механизмов обмена защищенными сообщениями, см. [363], механизмов защищенного взаимодействия контрольных и измерительных устройств, см. [365], протокола защищенного обмена для индустриальных систем, см. [366], и т.д.

Первый метод решения задачи дискретного логарифмирования, имеющий сложность, меньшую чем сложность тотального опробования, предложил в 1962 году А.О. Гельфонд, см. [344, гл.6, § 3]. Метод Гельфонда в отечественной литературе принято называть методом «согласования», в зарубежной – методом Д. Шенкса «больших и малых шагов», см. [228]. Данный метод применим к любой абелевой группе, а его сложность оценивается величиной $O(\sqrt{q})$ групповых операций, где q порядок группы. Метод требует хранения $O(\sqrt{q})$ элементов группы, что делает его неприменимым при больших значениях q.

В 1975 году Дж. Поллард, см.[195], основываясь на методе Р. Флойда поиска циклов в последовательностях, предложил вероятностный алгоритм решения задачи дискретного логарифмирования (ρ-метод Полларда). Математическое ожидание сложности его работы оценивается величиной $O(\sqrt{q})$, однако, в отличие от метода согласования, алгоритм Полларда-Флойда использует константный объем памяти. Еще один вариант данного алгоритма (λ-метод) Дж. Поллард предложил работе [196].

В 2001 годе Э. Теске, см. [247], предложила модификацию алгоритма Полларда-Флойда, позволившую незначительно снизить его сложность за счет использования памяти, хранящей $O(\log_2 q)$ элементов группы. Развивая указанные алгоритмы, в 2011 году А.Ю. Нестеренко в работе [325] предложил метод, использующий идеи Б. Госпера, см. [22], для поиска длин циклов в последовательностях. Данный метод рассматривается в § 1.2 настоящей диссертации.

В 1998 году П. ван Ооршотом и М. Винером, см. [188], был предложен универсальный метод параллельного поиска коллизий. Математическое ожидание сложности его работы также оценивается величиной $O(\sqrt{q})$, однако при совместной работе n независимых вычислителей, алгоритмическая сложность работы одного вычислительного средства составляет $O\left(\frac{\sqrt{q}}{n}\right)$ групповых операций, что позволяет сократить в n раз общее время работы алгоритма.

Вопросы применения метода Ооршота-Винера к группе точек эллиптических кривых рассматривались в работах [46, 258]. Результаты практических вычислений для простых значений p порядка 2^{112} и 2^{114} могут быть найдены в работах [238, 239]. Случай эллиптических кривых, определенных над полем характеристики два, рассматривался в работе [85].

Указанные выше методы являются универсальными и применимы к

любой абелевой группе. Для решения задачи дискретного логарифмирования в мультипликативной группе $\mathbb{F}_p^*$ известны алгоритмы, имеющие алгоритмическую сложность, меньшую чем методы Полларда-Флойда и ван Ооршота-Винера, см. [98, 123]. Вместе с тем, для группы точек эллиптической кривой, определенной над конечным простым полем $\mathbb{F}_p$, вопрос о построении алгоритма, имеющего алгоритмическую сложность меньшую, чем $O(\sqrt{q})$, остается открытым.

Вопросы решения задачи дискретного логарифмирования для эллиптических кривых частного вида рассматривались в работах Т. Сато и К. Араки [219], И.А. Семаева [225], Н. Смарта [235], Р. Шипси и К. Сварт [229].

Методы сведения задачи дискретного логарифмирования в группе точек эллиптической кривой к другим трудноразрешимым математическим задачам рассматривались в работах А. Менезеса, С. Ванстоуна, Т. Окамото [160, 161] (сведение к задаче дискретного логарифмирования в конечном поле), а также К. Пети, М. Костерса и А. Мессенга [194] (сведение к решению нелинейных систем уравнений над конечными полями). Следует также отметить носящие теоретический характер работы Дж. Сильвермена [232] и И.А. Семаева [226].

При оценке показателей эффективности мер защиты, реализуемых криптографическими механизмами, важным является вопрос о существовании так называемых «слабых» ключей, то есть секретных значений, использование которых приводит к снижению алгоритмической сложности решения задачи, обосновывающей стойкость средства защиты информации. Примеры «слабых» ключей, применительно к задачам анализа алгоритмов блочного шифрования, можно найти в работах Н. Фергюссона [89], Дж. Ким [125], а также рекомендациях [18, раздел 3.4.2].

Для задачи дискретного логарифмирования вопрос о «слабых» ключах был решен А.Ю. Нестеренко в работе [176]. Автором предложен алгоритм дискретного логарифмирования, применимый к любой конечной абелевой группе, алгоритмическая сложность которого зависит от мультипликативного порядка неизвестного значения, см. далее § 1.3. В том же параграфе приводится оценка мощности множества «слабых» ключей.

Для защиты от указанного алгоритма, а также защиты от методов решения задачи дискретного логарифмирования в частных случаях, необходимо выбирать параметры эллиптических кривых специальным образом. Перечни требований к параметрам эллиптических кривых могут быть найдены в стандартах [246, 280], а также в работах Д.Бернштейна и Т.Ланге [31], П.Баррето, Г.Перейры и Дж.Рикардини [20], А.Ю. Нестеренко [179]. Алгоритмы построения эллиптических кривых, удовлетворяющих указанным требованиям, рассматриваются в § 1.5.

§ 1.2. Алгоритмы поиска длин циклов в последовательностях и задача дискретного логарифмирования

Рассмотрим конечное множество $\mathcal{M}$, состоящее из q элементов. Зафиксируем некоторое отображение $f : \mathcal{M} \to \mathcal{M}$ множества $\mathcal{M}$ в себя, произвольный элемент $a_0 \in \mathcal{M}$ и рассмотрим последовательность элементов $a_0,\ a_1,\ a_2,\ \ldots$, определяемую равенством

$$a_{n+1} = f(a_n), \quad n = 0,\ 1,\ \ldots \tag{1.8}$$

Поскольку множество $\mathcal{M}$ конечно, то начиная с некоторого момента последовательность (1.8) зациклится. Цикл может начинаться с произвольного элемента a_λ так, что для некоторых целых значений λ и τ выполнено равенство

$$a_n = a_{n+\tau}, \quad \text{для всех} \quad n \geqslant \lambda \geqslant 0 \quad \text{и} \quad \tau \geqslant 1 \tag{1.9}$$

Определение 1.3. *Будем называть величину $\lambda \in \mathbb{N}_0$ длиной подхода к циклу, а величину $\tau \in \mathbb{N}$ длиной цикла.*

Задачу определения величин λ и τ, при известном отображении f и заданном начальном элементе a_0, будем называть задачей поиска цикла в последовательности $\{a_n\}_{n=0}^{\infty}$.

Известно несколько алгоритмов решения данной задачи. Первый и самый известный алгоритм был предложен в 1968 году Р. Флойдом, см. [303, п.3.1, задача 6b]. Годом позже Д. Кнутом был опубликован, см. [303, п.3.1, задача 7b], алгоритм Р. Брента.

Стоит также отметить еще два алгоритма. Первый, был предложен Р. Седжвиком и Т. Сжимански в 1981 году, оценки времени его работы опубликованы в [224]. Алгоритм Седжвика-Сжимански обладает наименьшей трудоемкостью среди всех известных алгоритмов поиска длин циклов в последовательностях. Вместе с тем, алгоритм использует столь большой объем памяти для хранения промежуточных элементов последовательности, что делает его неприменимым на практике.

Другой алгоритм предложен Г. Нивашем в 2004 году, см. [183], и основан на идее поиска минимального элемента, лежащего внутри цикла. Алгоритм Ниваша может быть использован только для тех множеств $\mathcal{M}$, на которых можно ввести эффективно вычислимое отношение упорядоченности элементов.

Однако наиболее эффективный способ решения рассматриваемой задачи заключается в использовании алгоритма, предложенного в 1972 году

Б. Госпером, см. [22, п.132]. Госпер дал описание алгоритма без обоснования корректности работы и оценки сложности.

В параграфе § 1.2 приводится полученное автором диссертации оригинальное обоснование алгоритма Госпера и предлагается метод решения задачи дискретного логарифмирования, в основе которого лежит алгоритм Госпера. Изложение следует работе [325].

§ 1.2.1. Алгоритм Флойда

Алгоритм Флойда основан на выполнении следующего условия: если выполнено равенство

$$a_n = a_{2n}, \tag{1.10}$$

то величина τ делит n. Для вычисления τ можно использовать следующий алгоритм.

Алгоритм 1.1: Алгоритм Флойда

Вход : Отображение $f : \mathcal{M} \to \mathcal{M}$ и начальный элемент a_0.
Выход : Значение длины цикла τ.

1 Определить начальные значения: $a \leftarrow a_0$ и $b \leftarrow f(a)$.
2 **Пока** $a \neq b$ **выполнять**
3 $\quad a \leftarrow f(a)$, $c \leftarrow f(b)$, $b \leftarrow f(c)$.
4 **конец**
5 Определить: $b \leftarrow f(a)$ и $\tau \leftarrow 1$.
6 **Пока** $a \neq b$ **выполнять**
7 $\quad b \leftarrow f(b)$ и $\tau \leftarrow \tau + 1$.
8 **конец**

Алгоритм выполняется до тех пор, пока не будет найдено равенство (1.10), см. проверку в строке 2, при этом точное значение индекса n не вычисляется. Поскольку задача поиска всех делителей числа n, в общем случае, является весьма трудоемкой, см. [276, 316], то последний цикл, см. строки 6 – 8, используется для вычисления точного значения τ непосредственным перебором.

Легко видеть, что минимально возможное значение индекса n, при котором выполняется равенство $a_n = a_{2n}$, равно $\tau \left\lceil \frac{\lambda}{\tau} \right\rceil$. Таким образом трудоемкость алгоритма Флойда равна $\tau \left(3 \left\lceil \frac{\lambda}{\tau} \right\rceil + 1\right)$ операций вычисления функции f.

§ 1.2.2. Алгоритм Брента

Прежде чем описывать алгоритм вычисления длины цикла последовательности (1.8), докажем теорему, утверждения которой служат его обоснованием. При доказательстве теоремы мы будем следовать идеям,

изложенным в монографии [61, п.8.2.2]. Определим функцию целочисленного аргумента

$$l(n) = 2^{\lfloor \log_2 n \rfloor}, \quad n = 1,\ 2,\ \ldots,$$

которая возвращает максимальное целое число, являющееся степенью двойки и не превосходящее числа n. Из определения функции $l(n)$ следует, что $l(n) \leqslant n < 2l(n)$. Определим параметр k равенством

$$k = \lceil \log_2 \max\{\lambda + 1, \tau\} \rceil,$$

где τ означает длину цикла, а λ длину подхода к циклу в последовательности, порожденной элементом a_0. Из определения параметра k следуют неравенства $\tau \leqslant 2^k$ и $\lambda \leqslant 2^k - 1$. Определим индекс n_0 равенством

$$n_0 = 2^k + \tau \left\lceil \frac{l(\lambda) + 1}{\tau} \right\rceil - 1 \tag{1.11}$$

Теорема 1.1 (см. [325]). *Выполнены следующие утверждения*

1. *Верны неравенства* $2^k \leqslant n_0 < 2^{k+1}$,
2. *Для индекса* n_0 *выполнено равенство* $a_{n_0} = a_{l(n_0)-1}$,
3. *Выполнено* $\frac{3}{2} l(n_0) \leqslant n_0 < 2l(n_0)$.

Доказательство. Начнем с доказательства первого утверждения теоремы. Поскольку длина цикла τ является целым числом, то $\tau \geqslant 1$. Поскольку неравенство $\left\lceil \frac{l(\lambda)+1}{\tau} \right\rceil \geqslant 1$ выполнено по определению, то неравенство $2^k \leqslant n_0$ очевидно. Для оценки сверху заметим, что нам достаточно показать, что

$$\tau \left\lceil \frac{l(\lambda) + 1}{\tau} \right\rceil \leqslant 2^k,$$

тогда $n_0 \leqslant 2 \cdot 2^k - 1 < 2^{k+1}$.

1. В начале рассмотрим случай, когда $\tau > l(\lambda)$. Тогда выполнено $\tau \geqslant l(\lambda) + 1$ и $\frac{l(\lambda)+1}{\tau} \leqslant 1$. Следовательно

$$\left\lceil \frac{l(\lambda) + 1}{\tau} \right\rceil = 1 \quad \text{и} \quad \tau \left\lceil \frac{l(\lambda) + 1}{\tau} \right\rceil = \tau \leqslant 2^k.$$

   Последнее неравенство выполнено в силу выбора параметра k.

2. Рассмотрим второй случай $\tau \leqslant l(\lambda)$. Тогда выполнено

$$\left\lceil \frac{l(\lambda) + 1}{\tau} \right\rceil = \frac{l(\lambda) + 1 + \delta}{\tau} \leqslant \frac{l(\lambda) + \tau}{\tau} \leqslant \frac{2l(\lambda)}{\tau},$$

при некотором натуральном $\delta < \tau$.

Используя равенство $\lfloor \log_2 \lambda \rfloor + 1 = \lceil \log_2(\lambda + 1) \rceil$, выполненное при натуральных значениях λ, мы можем записать

$$\tau \left\lceil \frac{l(\lambda)+1}{\tau} \right\rceil \leqslant 2l(\lambda) = 2^{\lfloor \log_2 \lambda \rfloor + 1} = 2^{\lceil \log_2(\lambda+1) \rceil} \leqslant 2^k.$$

Поскольку для обоих рассмотренных случаев выполнено необходимое неравенство, то первое утверждение теоремы доказано.

Для доказательства второго утверждения теоремы заметим, что из первого утверждения следует

$$k \leqslant \log_2 n_0 < k + 1 \quad \text{и} \quad l(n_0) = 2^k.$$

Тогда разность $n_0 - (l(n_0) - 1) = \tau \left\lceil \frac{l(\lambda)+1}{\tau} \right\rceil$ кратна длине цикла τ, то есть равенство $a_{n_0} = a_{l(n_0)-1}$ действительно имеет место.

Нам осталось доказать последнее утверждение теоремы. Оценка сверху тривиально вытекает из определения функции $l(n)$. Для получения нижней оценки заметим справедливость неравенств

$$\tau \left\lceil \frac{l(\lambda)+1}{\tau} \right\rceil - 1 \geqslant \tau - 1 = 2^{\log_2 \tau} - 1 \geqslant 2^{\lceil \log_2 \tau \rceil - 1} \tag{1.12}$$

$$\tau \left\lceil \frac{l(\lambda)+1}{\tau} \right\rceil - 1 \geqslant l(\lambda) = 2^{\lceil \log_2 \lambda + 1 \rceil - 1}. \tag{1.13}$$

Следовательно, выполнено неравенство $\tau \left\lceil \frac{l(\lambda)+1}{\tau} \right\rceil - 1 \geqslant 2^{k-1}$, откуда следует

$$n_0 = 2^k + \tau \left\lceil \frac{l(\lambda)+1}{\tau} \right\rceil - 1 \geqslant 2^k + 2^{k-1} = \frac{3l(n_0)}{2}$$

и доказательство последнего утверждения теоремы. □

Доказанная теорема 1.1 в явном виде задает значение индекса n_0 которое необходимо вычислить для определения длины цикла. К сожалению, на практике, нам неизвестно значение n_0.

Р. Брент предложил следующую идею: для поиска значения, кратного величине τ, нам необходимо воспользоваться тем фактом, что $a_{n_0} = a_{2^k-1}$ при некотором натуральном значении значении k таком, что

$$3 \cdot 2^{k-1} \leqslant n_0 < 2^{k+1}.$$

Тогда алгоритм поиска τ может быть записан следующим образом.

Алгоритм 1.2: Алгоритм Брента

Вход : Отображение $f : \mathcal{M} \to \mathcal{M}$ и начальный элемент a_0.
Выход : Значение длины цикла τ.

1 Определить начальные значения: $c \leftarrow a_0$, $a \leftarrow f(a_0)$, $n \leftarrow 1$ и $t \leftarrow 1$.
2 **Если** $a = c$, **то**
3 | вернуть значение $\tau \leftarrow 1$ и завершить алгоритм.
4 **конец**
5 **Если** $n = t$, **то**
6 | положить $c = a$ и вычислить $t = 2t$.
7 **конец**
8 Определить $a \leftarrow f(a)$ и вычислить $n \leftarrow n + 1$.
9 **Если** $n \geqslant 3t/4$, **то**
10 | проверить выполняется ли $a = c$. Если нет, то вернуться к строке 5.
11 **конец**
12 Определить $\tau \leftarrow 1$ и $a \leftarrow f(c)$.
13 **Пока** $a \neq c$ **выполнять**
14 | $a \leftarrow f(a)$ и $\tau \leftarrow \tau + 1$.
15 **конец**

Алгоритм Брента, как и алгоритм Флойда, не позволяет в явном виде определить значение длины цикла τ. В девятой строке приведенного алгоритма находится совпадение $a_{n_0} = a_{2^k-1}$ для некоторого натурального числа k. Последние два шага приведенного алгоритма позволяют определить значение τ простым перебором.

Для оценки трудоемкости алгоритма Брента заметим, что из третьего утверждения теоремы 1.1 и определения величины k следует оценка снизу

$$n_0 \geqslant \frac{3l(n_0)}{2} = \frac{3}{2} \cdot 2^k \geqslant \frac{3}{2} \max\{\lambda + 1, \tau\},$$

для числа шагов, необходимых для поиска совпадения в девятой строке алгоритма алгоритма. Таким образом, общая трудоемкость алгоритма Брента составит не менее $\tau + \frac{3}{2} \max\{\lambda + 1, \tau\}$ операций вычисления функции f.

§ 1.2.3. Алгоритм Госпера

В алгоритме Госпера для поиска двух совпадающих элементов последовательности (1.8)

$$a_{n+1} = f(a_n), \quad n = 0,\ 1,\ \ldots$$

производится сравнение a_n с элементами некоторого множества $M(n)$. Фиксируем значение $n > 0$ и поместим в множество $M(n)$ элементы $a_{n_0},\ a_{n_1},\ \ldots$ последовательности (1.8), с условием

$$n_i = \max_{r<n}\{r | \nu_2(r+1) = i\}, \tag{1.14}$$

для всех возможных значений $i = 0,\ 1,\ \ldots$, где функция $\nu_2(r+1)$ возвращает наибольшую степень двойки, делящую величину $r+1$.

Из определения следует, что для фиксированного значения n множество $M(n)$ конечно, содержит не более $\lfloor \log_2 n \rfloor + 1$ чисел и отличается от множества $M(n+1)$ лишь одним элементом.

Мы можем построить множество $M(n)$ следующим образом. Разобьем последовательность (1.8) на несколько подпоследовательностей так, что первая подпоследовательность содержит все элементы с индексами i такими, что $i+1$ нечетно, т.е.

$$a_0, a_2, a_4, a_6, a_8, \ldots \tag{1.15}$$

вторая — элементы индексами i такими, что $i+1$ делится в точности на двойку, т.е.

$$a_1, a_5, a_9, a_{13}, a_{17}, \ldots \tag{1.16}$$

третья — элементы с индексами i такими, что $i+1$ делится в точности на четверку, т.е.

$$a_3, a_{11}, a_{19}, a_{27}, \ldots$$

четвертая — элементы с индексами i такими, что $i+1$ делится в точности на восемь, т.е.

$$a_7, a_{23}, a_{39}, a_{55}, \ldots$$

и так далее. Тогда в множество $M(n)$ входит по одному элементу из каждой подпоследовательности с максимальным индексом, не превосходящим n. В таблице 1.1 указан явный вид множеств $M(n)$ для всех индексов $n = 1, \ldots, 16$.

n	$M(n)$	n	$M(n)$
1	$\{a_0\}$	9	$\{a_8,\ a_5,\ a_3,\ a_7\}$
2	$\{a_0,\ a_1\}$	10	$\{a_8,\ a_9,\ a_3,\ a_7\}$
3	$\{a_2,\ a_1\}$	11	$\{a_{10},\ a_9,\ a_3,\ a_7\}$
4	$\{a_2,\ a_1,\ a_3\}$	12	$\{a_{10},\ a_9,\ a_{11},\ a_7\}$
5	$\{a_4,\ a_1,\ a_3\}$	13	$\{a_{12},\ a_9,\ a_{11},\ a_7\}$
6	$\{a_4,\ a_5,\ a_3\}$	14	$\{a_{12},\ a_{13},\ a_{11},\ a_7\}$
7	$\{a_6,\ a_5,\ a_3\}$	15	$\{a_{14},\ a_{13},\ a_{11},\ a_7\}$
8	$\{a_6,\ a_5,\ a_3,\ a_7\}$	16	$\{a_{14},\ a_{13},\ a_{11},\ a_7,\ a_{15}\}$

Таблица 1.1: Множества $M(n)$ для $n = 1, \ldots, 16$.

Отметим, что распределение элементов последовательности (1.8) по подмножествам $M(n)$ не зависит от значений $a_0, a_1, a_2, \ldots$, а определяется только равенством (1.14).

Теорема 1.2 (см. [325]). *Пусть заданы параметры λ и τ, определяющие длину подхода к циклу и длину цикла последовательности (1.8). Тогда найдутся натуральные индексы r и $n = r + \tau$ такие, что*

1. *элемент a_r принадлежит множеству $M(n)$ и выполнено равенство $a_n = a_r$,*
2. *$\lambda + \tau \leqslant n < \lambda + 2\tau$.*

Доказательство. Определим в качестве параметра i целое число, удовлетворяющее неравенствам $2^i \leqslant \tau < 2^{i+1}$ и рассмотрим целые числа

$$r = 2^i \left\lceil \frac{\lambda + 1}{2^i} \right\rceil - 1 \quad \text{и} \quad n = r + \tau.$$

Очевидно, что $r < n$ для любого целого $\tau \geqslant 1$. Представим $\left\lceil \frac{\lambda+1}{2^i} \right\rceil = 2^l s$, где $l \geqslant 0$ целое число и $s = 2h + 1$ нечетное целое число. Тогда r имеет вид

$$r = 2^{i+l}s - 1 = 2^{i+l}(2h + 1) - 1 = 2^{i+l} - 1 + h2^{i+l+1}, \tag{1.17}$$

то есть $r \equiv 2^{i+l} - 1 \pmod{2^{i+l+1}}$ и мы получаем, что $\nu_2(r + 1) = i + l$. Поскольку выполнено неравенство

$$r + 2^{i+l+1} \geqslant r + 2^{i+1} > r + \tau = n$$

мы получаем, что индекс r принадлежит множеству $M(n)$. Учитывая, что τ есть период последовательности (1.8), то $a_r = a_{r+\tau} = a_n$. Первое утверждение теоремы доказано.

Для доказательства второго утверждения теоремы получим оценки снизу и сверху на величину r. Воспользовавшись неравенством

$$x \leqslant \lceil x \rceil < x + 1,$$

выполненным для любого действительного $x > 0$, получим

$$\lambda = 2^i \left(\frac{\lambda + 1}{2^i} \right) - 1 \leqslant 2^i \left\lceil \frac{\lambda + 1}{2^i} \right\rceil - 1 = r$$
$$< 2^i \left(\frac{\lambda + 1}{2^i} + 1 \right) - 1 = \lambda + 2^i \leqslant \lambda + \tau,$$

то есть неравенство $\lambda \leqslant r < \lambda + \tau$, из которого следует утверждение теоремы. □

Второе условие теоремы 1.2 дает оценку на число элементов последовательности $a_0, a_1, a_2, \ldots$, которые необходимо вычислить для нахождения равенства $a_n = a_r$.

При практической реализации алгоритма поиска цикла будем хранить множество $M(n)$ в массиве T – по адресу 0 будет располагаться элемент подпоследовательности (1.15), по адресу 1 – элемент подпоследовательности (1.16) и так далее. Тогда алгоритм Госпера может быть записан следующим образом.

Алгоритм 1.3: Алгоритм Госпера

Вход : Отображение $f : \mathcal{M} \to \mathcal{M}$ и начальный элемент a_0.
Выход : Значение длины цикла τ.

1 Определить начальные значения: $a \leftarrow a_0$, $n \leftarrow 1$, $t \leftarrow 1$ и $T[0] \leftarrow a_0$.
2 Вычислить $a \leftarrow f(a)$.
3 **Для всех** $i = 0,\ \ldots,\ t-1$ **выполнять**
4 Если выполнено равенство $T[i] = a$, то завершить алгоритм с результатом
$$\tau = n - 2^i \left(1 + 2\left\lfloor \frac{n - 2^i + 1}{2^{i+1}} \right\rfloor\right) + 1.$$
5 **конец**
6 Вычислить $n \leftarrow n + 1$ и $i \leftarrow \nu_2(n)$.
7 **Если** $i = t$, **то**
8 вычислить $t = t + 1$
9 **конец**
10 Определить $T[i] = a$ и вернуться на шаг 2.

Значение τ определяется в четвертой строке алгоритма. Из (1.14) и (1.17) следует, что при завершении работы алгоритма величина h принимает значение $h = \left\lfloor \frac{n-(2^i-1)}{2^{i+1}} \right\rfloor$, а величина r значение $r = 2^i - 1 + h2^{i+1}$. Учитывая равенство $\tau = n - r$, мы получаем явное выражение для длины периода τ.

Алгоритм	*Трудомкость*	*Объем памяти*
Флойд	$\tau\left(3\left\lceil \frac{\lambda}{\tau} \right\rceil + 1\right)$	3
Брент	не менее $\tau + \frac{3}{2}\max\{\lambda + 1, \tau\}$	4
Госпер	не более $\lambda + 2\tau$	$\lceil \log_2(\lambda + 2\tau) \rceil + 4$

Таблица 1.2: Оценки трудоемкости алгоритмов поиска длин циклов в последовательностях

Для сравнения, приведем в таблице 1.2 различные характеристики описанных нами выше алгоритмов. Во второй колонке мы приводим оценку трудоемкости алгоритма, измеряемую в количестве вычислений функции f. В третьей колонке содержится количество ячеек памяти, необходимых для вычисления длины цикла τ. Из приведенных значений следует,

что алгоритм Госпера имеет наименьшую трудоемкость среди рассматриваемых алгоритмов используя при этом несколько больший объем памяти.

Рассмотрение вопроса о трудоемкости приведенных в таблице 1.2 алгоритмов в зависимости от мощности q множества $\mathcal{M}$ сводится к определению значений математического ожидания $E_q(\lambda, f)$ и $E_q(\tau, f)$ для величин λ и τ, соответственно, при случайном выборе отображения f. Хорошо известно, см., например, [91], [308, п.5.3] или обзор в [321], что

$$\lim_{q\to\infty} \frac{E_q(\lambda, f)}{\sqrt{q}} = \lim_{q\to\infty} \frac{E_q(\tau, f)}{\sqrt{q}} = \sqrt{\frac{\pi}{8}}.$$

Тогда можно считать, что асимптотическая оценка числа шагов в алгоритме Госпера (в предположении о независимости случайных величин λ и τ) следует из равенства

$$\lim_{q\to\infty} \frac{E_q(\lambda, f) + 2E_q(\tau, f)}{\sqrt{q}} = 3\sqrt{\frac{\pi}{8}}. \tag{1.18}$$

§ 1.2.4. Алгоритм дискретного логарифмирования

Покажем, как описанный ранее алгоритм Госпера, см. алгоритм 1.3, может быть использован для решения задачи дискретного логарифмирования в группе точек эллиптической кривой.

Зафиксируем простое число $p > 3$ и рассмотрим эллиптическую кривую, заданную в аффинной форме сравнением (1.4)

$$\mathcal{E}_{a,b}(\mathbb{F}_p): \quad y^2 \equiv x^3 + ax + b \pmod p,$$

а также равенство (1.6)

$$Q = [k]P, \quad P, Q \in \mathcal{E}_{a,b}(\mathbb{F}_p), \quad k \in \mathbb{Z}_q^*, \quad q = \mathtt{ord}(P). \tag{1.19}$$

Будем считать, что q – простое число, делящее порядок всей группы точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$.

Задача, которую необходимо решить, заключается в определении величины k по известным параметрам эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и точкам $P, Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$.

Определим в качестве множества $\mathcal{M}$ — подгруппу порядка q, порожденную точкой P, т.е.

$$\mathcal{M} = \{P,\ [2]P,\ [3]P,\ \ldots,\ [q]P = \mathcal{O}\}, \quad \mathcal{M} \subseteq \mathcal{E}_{a,b}(\mathbb{F}_p)$$

где $\mathcal{O}$ – бесконечно удаленная точка кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$. Из равенства (1.19) следует, что $Q \in \mathcal{M}$ (если $Q \notin \mathcal{M}$, то уравнение (1.19), очевидно, неразрешимо относительно неизвестной k).

Зафиксируем[1] $s = \lceil \log_2 q \rceil \in \mathbb{N}$ и разобьем множество $\mathcal{M}$ на s не пересекающихся подмножеств

$$\mathcal{M} = \bigcup_{l=0}^{s-1} J_l$$

следующим образом – будем относить к подмножеству J_l те точки $P = (x, y)$ множества $\mathcal{M}$, для которых выполнено условие

$$x \equiv l \pmod{s},$$

т.е. точки, у которых x-координата сравнима с l по модулю фиксированного ранее значения s.

Теперь построим случайное отображение f множества $\mathcal{M}$ в себя. Для этого выберем случайным образом вычеты

$$\gamma_i, \omega_i \in_R \mathbb{Z}_q^*, \quad i = 1, \ldots, s,$$

и для произвольной точки $R \in \mathcal{M}$ такой, что $R = (x, y)$ и $x \equiv l \pmod{s}$, определим

$$f(R) \;=\; R + [\gamma_l]P + [\omega_l]Q, \quad \text{если} \quad R \in J_l. \tag{1.20}$$

Поскольку $R \in \mathcal{M}$, то найдется индекс $j \in \mathbb{Z}_q$ такой, что $R = [j]P$ и, учитывая условие (1.19), получаем равенство

$$f(R) = [j]P + [\gamma_n]P + [\omega_n]Q = [j + \gamma_n + \omega_n k \pmod{q}]P \in \mathcal{M}.$$

Следовательно, отображение $f : \mathcal{M} \to \mathcal{M}$ определено корректно.

Используем построенное отображение для выработки последовательности точек $\{R_n\}_{n=0}^{\infty}$ множества $\mathcal{M}$. Выберем в качестве начальной точки

$$R_0 = [\alpha_0]P + [\beta_0]Q, \quad \alpha_0, \beta_0 \in_R \mathbb{Z}_q^*, \tag{1.21}$$

где α_0, β_0 – случайные вычеты из $\mathbb{Z}_q^*$, и определим

$$R_{n+1} = f(R_n) = R_n + [\gamma_l]P + [\omega_l]Q, \quad \text{если} \quad R_n \in J_l. \tag{1.22}$$

Отметим, что каждая из точек последовательности $\{R_n\}_{n=0}^{\infty}$ может быть представлена в виде (1.21). Действительно, для каждой точки R_n найдется номер $l_n \in \mathbb{Z}_s$ такой, что $R_n \in J_{l_n}$. Тогда, используя (1.21), для любого индекса $n = 0, 1, \ldots$получаем

[1]Заметим, что в оригинальной статье Дж. Полларда [196] значение s предлагалось брать равным трем. Позднее, для решения задачи дискретного логарифмирования в мультипликативной группе поля $\mathbb{F}_p$ Э.Теске, см. [247], предложила выбирать значение $s = \lceil \log_2 q \rceil$. Для группы точек эллиптических кривых автором диссертации используется то же самое значение.

$$R_{n+1} = R_n + [\gamma_{l_n}]P + [\omega_{l_n}]Q = \\ = R_{n-1} + [\gamma_{l_{n-1}}]P + [\omega_{l_{n-1}}]Q + [\gamma_{l_n}]P + [\omega_{l_n}]Q = \\ \cdots = R_0 + \left[\sum_{j=0}^{n} \gamma_{l_j} \pmod q\right] P + \left[\sum_{j=0}^{n} \omega_{l_j} \pmod q\right] Q = \\ = \left[\alpha_0 + \sum_{j=0}^{n} \gamma_{l_j} \pmod q\right] P + \left[\beta_0 + \sum_{j=0}^{n} \omega_{l_j} \pmod q\right] Q = \\ = [\alpha_{n+1}]P + [\beta_{n+1}]Q, \quad (1.23)$$

где $\alpha_{n+1}, \beta_{n+1} \in \mathbb{Z}_q$.

Используя алгоритм Госпера можно найти два элемента R_t и R_l последовательности точек $\{R_n\}_{n=0}^{\infty}$ такие, что $R_t = R_l$, тогда, используя равенства (1.19) и (1.23), можно записать

$$[\alpha_t + \beta_t k \pmod q]P = \\ = [\alpha_t]P + [\beta_t k \pmod q]P = [\alpha_t]P + [\beta_t]Q = R_t = \\ = R_l = [\alpha_l]P + [\beta_l]Q = [\alpha_l]P + [\beta_l k \pmod q]P = \\ = [\alpha_l + \beta_l k \pmod q]P.$$

Последнее равенство позволяет выразить неизвестное k через значения величин $\alpha_t, \alpha_l, \beta_t, \beta_l \in \mathbb{Z}_q$

$$k \equiv \frac{\alpha_t - \alpha_l}{\beta_l - \beta_t} \pmod q. \quad (1.24)$$

Поскольку q простое число, то сравнение (1.24) позволяет найти неизвестное значение k только в случае, когда $\alpha_t \not\equiv \alpha_l \pmod q$. В противном случае, алгоритм Госпера должен продолжить свое выполнение для поиска следующей пары совпадающих точек R_t, R_l.

Для практической реализации описанного алгоритма необходимо использование двух массивов. Первый массив – S, будет хранить точки, используемые для реализации отображения f согласно равенству (1.20), т.е.

$$S[l] = [\gamma_l]P + [\omega_l]Q, \quad l = 0, 1, \ldots, s-1,$$

где $s = \lceil \log_2 q \rceil$, а также вычеты $\gamma_l, \omega_l \in \mathbb{Z}_q^*$, которые будут выбираться случайным образом.

Во втором массиве – T будут храниться элементы используемого в алгоритме Госпера множества $M(n)$, определяемого при вычислении n-го элемента последовательности $\{R_n\}_{n=0}^{\infty}$. Каждый элемент массива должен хранить в себе точку R, находящуюся во множестве $M(n)$, а также коэффициенты α, β, выражающие точку R через исходные точки P и Q согласно (1.23).

Размер массива T определим равенством $h = 2 + \lfloor \log_2 \sqrt{q} \rfloor$. Учитывая второе утверждение теоремы 1.2 и равенство (1.18) можно ожидать[2], что максимальное число шагов алгоритма не превысит величины

$$n < \lambda + 2\tau \sim 3\sqrt{\frac{\pi q}{8}}. \tag{1.25}$$

Поскольку множество $M(n)$ содержит не более $\lfloor \log_2 n \rfloor + 1$ элементов последовательности $\{R_n\}_{n=0}^{\infty}$, получаем неравенство

$$\lfloor \log_2 n \rfloor + 1 < \left\lfloor \log_2 3\sqrt{\frac{\pi q}{8}} \right\rfloor + 1 = \\ = \left\lfloor \frac{1}{2}\Big(\log_2 9\pi + \log_2 q - 3 \Big) \right\rfloor + 1 < \left\lfloor \frac{\log_2 q}{2} \right\rfloor + 2 = h,$$

из которого следует, что ожидаемый размер множества $M(n)$ не превысит величины h.

Описываемый алгоритм дискретного логарифмирования был реализован автором на практике, см. далее алгоритм 1.4. Для простого числа $p = 20946419752860000901$ такого, что $2^{64} < p < 2^{65}$, была рассмотрена кривая

$$y^2 \equiv x^3 + 20000878653677493608x + \\ + 18397449348597994624 \pmod{20946419752860000901},$$

и точки

$$\begin{aligned} P &= (8965431374621232772, 6559412994221936909), \\ Q &= -P. \end{aligned}$$

При этом, было выполнено равенство $\mathtt{ord}(P) = q = 56774958479$. Порядок всей группы точек кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ равнялся 20946419760089834350.

В ходе выполнения алгоритма, для вычисления отображения f, использовалась таблица из $s = \lceil \log_2 q \rceil = 36$ точек вида $[\gamma_i]P + [\omega_i]Q$. После 435287 итераций вычисления функции f, было найдено совпадение двух точек

$$2450565259Q + 35503301414P = 27039682333Q + 3317460009P,$$

что позволило определить неизвестное значение $k = 56774958478$. Время работы программы на ПЭВМ с процессором AMD Athlon X64 составило не более 20 сек.

[2]Равенство (1.18) дает лишь предельное значение математического ожидания числа шагов алгоритма при $q \to \infty$. Реальное число шагов алгоритма зависит от выбора отображения f, а также начальной точки R_0, и может отличаться от приведенного в (1.25) значения.

Ожидаемое количество шагов алгоритма, согласно (1.25), не должно было превосходить величины

$$3\sqrt{\frac{\pi q}{8}} \sim 447950.0714629683.$$

Полученное на практике значение 435287 хорошо согласуется с ожидаемым теоретическим значением.

Алгоритм 1.4: Алгоритм дискретного логарифмирования методом Госпера

Вход : Точки P, Q кривой (1.4), связанные соотношением (1.6) и $\mathtt{ord}\, P = q$.
Выход : Значение величины k, удовлетворяющей (1.19).

1 Определить $s = \lceil \log_2 q \rceil$.
2 **Для всех** $l = 0,\ 1,\ \ldots,\ s-1$ **выполнять**
3 | Выработать вычеты $\gamma_l, \omega_l \in_R \mathbb{Z}_q^*$ и определить $S[l] \leftarrow [\gamma_l]P + [\omega_l]Q$.
4 **конец**
5 Выработать вычеты $\alpha_0, \beta_0 \in_R \mathbb{Z}_q^*$ и определить $R_0 \leftarrow [\alpha_0]P + [\beta_0]Q$.
6 Определить значения: $R \leftarrow R_0$, $n \leftarrow 1$, $t \leftarrow 1$, $\alpha \leftarrow 0$, $\beta \leftarrow 0$ и $T[0] \leftarrow \{R_0, \alpha_0, \beta_0\}$.
7 Определить индекс l такой, что $R \in J_l$, и определить $R \leftarrow R + S[l]$, см. (1.22), а также $\alpha \leftarrow \alpha + \gamma_l \pmod q$, $\beta \leftarrow \beta + \omega_l \pmod q$.
8 **Для всех** $i = t-1,\ \ldots,\ 1,\ 0$ **выполнять**
9 | **Если** *выполнено равенство* $T[i] = R$, **то**
10 | | Определить $\alpha_t \leftarrow T[i]$, $\beta_t \leftarrow T[i]$.
11 | | **Если** $\beta_t \not\equiv \beta\ (mod\, q)$ **то**
12 | | | Определить $k \equiv \frac{\alpha_t - \alpha}{\beta - \beta_t} \pmod q$ и завершить алгоритм.
13 | | **конец**
14 | **конец**
15 **конец**
16 Вычислить $n \leftarrow n+1$ и $i \leftarrow \nu_2(n)$, см. (1.14).
17 **Если** $i = t$, **то**
18 | вычислить $t = t+1$. Если $t \geqslant 2 + \lfloor \log_2 \sqrt{q} \rfloor$, то вернуться к строке 2.
19 **конец**
20 Определить $T[i] = R$ и вернуться к строке 7.

Заключение к § 1.2

В § 1.2 рассмотрен класс алгоритмов поиска длин циклов в последовательностях. Получено обоснование корректности работы алгоритма Госпера, оценка его алгоритмической сложности, см. терему 1.2, и показано, что данная оценка является наименьшей среди всех рассматриваемых алгоритмов.

Алгоритм Госпера применен для решения задачи дискретного логарифмирования в группе точек эллиптической кривой. Предложена и подтверждена результатами практических экспериментов оценка асимпто-

тической сложности решения задачи дискретного логарифмирования с помощью алгоритма Госпера.

§ 1.3. Алгоритмы дискретного логарифмирования, использующие информацию о мультипликативном порядке неизвестного

Алгоритмическая сложность рассмотренных ранее алгоритмов решения задачи дискретного логарифмирования (1.19) зависела только от параметров эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$. В данном параграфе рассматривается класс алгоритмов, алгоритмическая сложность которых зависит не только от параметров эллиптической кривой, но и от неизвестного значения $k \in \mathbb{Z}_q$, а также некоторой дополнительной информации о неизвестном значении. Рассмотренные алгоритмы позволят нам позднее определить понятие «слабого» ключа, см. обзор в § 1.1.2. Дальнейшее изложение следует работе [176].

Пусть $q \in \mathbb{N}$ и $k \in \mathbb{Z}_q$, $\mathbf{НОД}\,(k,\ q) = 1$. Напомним, что символом $\mathtt{ord}_q\, k$ мы обозначаем показатель вычета k по модулю q, т.е.

$$\mathtt{ord}_q\, k = \min_{n\in\mathbb{N}}\{n :\ k^n \equiv 1 \pmod q\}.$$

Верна следующая теорема, см. [176].

Теорема 1.3. *Пусть $p > 3$ – простое число, $\mathcal{E}_{a,b}(\mathbb{F}_p)$ эллиптическая кривая и $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ – точка кривой, порождающая циклическую подгруппу $\langle P\rangle \subseteq E_{a,b}(\mathbb{F}_p)$ простого порядка q.*

Пусть $Q = [k]P$, $k \in \mathbb{Z}_q$ и $\mathtt{ord}_q\, k = r$. Если $q > 6$ и $r \geqslant 6$, то алгоритмическая сложность нахождения величины k не превосходит $8\sqrt{r}\log_2 q$ групповых операций.

Доказательство. Приведем алгоритм поиска неизвестного $k \in \mathbb{Z}_q$ и оценим его алгоритмическую сложность.

Пусть $r = \mathtt{ord}_q\, k$ и g – произвольный первообразный корень по модулю q, тогда $r|q-1$. Определим

$$\alpha \equiv g^{\frac{q-1}{r}} \pmod q\,, \tag{1.26}$$

тогда $\mathtt{ord}_q\, \alpha = r$ и найдется вычет $x \in \mathbb{Z}_r$ такой, что

$$k \equiv \alpha^x \pmod q\,.$$

Следовательно, выполнено равенство

$$Q = [k]P = [\alpha^x \pmod q]P \tag{1.27}$$

и задача нахождения неизвестного k сводится к нахождению неизвестной величины $x \in \mathbb{Z}_r$.

Определим $h = \lfloor \sqrt{r} \rfloor + 1$. Тогда найдутся, см. [291, лемма 6.1], такие целые числа x_0, x_1, что

$$1 \leqslant x_0, x_1 \leqslant h, \quad x = x_1 h - x_0.$$

Тогда равенство (1.27) принимает вид

$$[\alpha^{x_0} \pmod q]Q = [\left(\alpha^h\right)^{x_1} \pmod q]P. \tag{1.28}$$

Для поиска величин x_0, x_1 достаточно применить алгоритм согласования, см. [344, гл.6, § 3]. Вычислим две последовательности

$$[\alpha]Q,\ [\alpha^2 \pmod q]Q,\ [\alpha^3 \pmod q]Q,\ \ldots,\ [\alpha^h \pmod q]Q$$

и

$$[\alpha^h \pmod q]P,\ [(\alpha^h)^2 \pmod q]P,\ [(\alpha^h)^3 \pmod q]P,\ \ldots,\ [(\alpha^h)^h \pmod q]P.$$

В силу (1.27) найдется элемент, который будет принадлежать одновременно двум последовательностям. Индекс этого элемента в первой последовательности даст нам значение x_0, а индекс во второй последовательности – значение x_1.

Поскольку суммарно обе последовательности содержат не более $2h$ элементов группы $\mathcal{E}_{a,b}(\mathbb{F}_p)$, а каждый элемент вычисляется не более чем за $2\lceil \log_2 q \rceil$ групповых операций, то общая алгоритмическая сложность алгоритма согласования не превосходит величины $4(\lfloor \sqrt{r} \rfloor + 1)\lceil \log_2 q \rceil$. Из условий теоремы следует, что $\sqrt{r} + \log_2 q + 1 < \sqrt{r} \log_2 q$, тогда выполнено неравенство

$$\begin{aligned}(\lfloor \sqrt{r} \rfloor + 1)\lceil \log_2 q \rceil < (\sqrt{r} + 1)(\log_2 q + 1) = \\ = \sqrt{r} \log_2 q + (\sqrt{r} + \log_2 q + 1) < 2\sqrt{r} \log_2 q,\end{aligned}$$

которое завершает доказательство теоремы. □

Доказательство теоремы конструктивно, т.е. оно содержит алгоритм поиска неизвестного значения $k \in \mathbb{Z}_q$. Для данного алгоритма существенным является значение величины $r = \mathtt{ord}_q\, k$, однако, если это значение неизвестно, то алгоритм всё равно может быть реализован для $\alpha = g$ и

$r = q-1$, см. равенство (1.26). Таким образом, алгоритмическая сложность поиска любого неизвестного значения $k \in \mathbb{Z}_q$ не превосходит $8\sqrt{q-1}\log_2 q$.

Поскольку алгоритм согласования требует хранения $2\bigl(\lfloor\sqrt{r}\rfloor+1\bigr)$ точек эллиптической кривой в памяти вычислительного средства, он не может быть практически реализован на ЭВМ при больших значениях r.

Для практического подтверждения утверждений теоремы 1.3 в работе [176] автором был предложен алгоритм, являющийся модификацией λ-метода Полларда [196]. Приведем описание этого алгоритма, а также параллельный вариант алгоритма, основанный на идеях, предложенных в статье Ооршота-Винера, см. [188].

§ 1.3.1. Вариант алгоритма, основанный на идеях λ-метода Полларда

Зафиксируем произвольный первообразный корень g по модулю q и определим, согласно (1.26), вычет

$$\alpha \equiv g^{\frac{q-1}{r}} \pmod q .$$

Для поиска неизвестного значения $k \in \mathbb{Z}_r$ рассмотрим подмножество точек $\mathcal{M} \subset \mathcal{E}_{a,b}(\mathbb{F}_p)$, определяемое равенством

$$\mathcal{M} = \{[\alpha]P,\ [\alpha^2 \pmod q]P,\ \ldots,\ [\alpha^r \pmod p]P = P\}.$$

Легко видеть, что точка Q также принадлежит рассматриваемому подмножеству $\mathcal{M}$, поскольку $Q = [k]P = [\alpha^x]P \in \mathcal{M}$ для некоторого значения $x \in \mathbb{Z}_r$.

Определим случайное отображение $f : \mathcal{M} \to \mathcal{M}$. Для этого зафиксируем $s = \lceil \log_2 r \rceil$ и выберем случайным образом вычеты

$$\xi_0,\ \ldots,\ \xi_{s-1} \in_R \mathbb{Z}_r^*.$$

Теперь для любой точки $R \in \mathcal{M}$, заданной в аффинной форме координатами (x_R, y_R), положим

$$f(R) = [\alpha^{\xi_l} \pmod q]R, \quad \text{где} \quad l \equiv x_R \pmod s \tag{1.29}$$

для некоторого $l \in \mathbb{Z}_s$.

С помощью отображения f построим две[3] последовательности точек $\{R_n\}_{n=0}^{\infty}$ и $\{U_i\}_{i=0}^{\infty}$, принадлежащих подмножеству $\mathcal{M}$.

[3] В этом заключается отличие данного метода от рассматривавшихся ранее в § 1.2 методов, где использовалась только одна последовательность $\{R_n\}_{n=0}^{\infty}$.

Выберем случайные вычеты $\gamma_0, \omega_0 \in_R \mathbb{Z}_r^*$ и определим начальные точки последовательностей равенствами

$$R_0 = [\alpha^{\gamma_0} \pmod q]P, \quad U_0 = [\alpha^{\omega_0} \pmod q]Q.$$

Остальные элементы последовательностей определим равенствами

$$R_{n+1} = f(R_n), \quad U_{n+1} = f(U_n), \quad n = 0, 1, \ldots$$

Легко видеть, что все точки последовательности $\{R_n\}_{n=0}^{\infty}$ принадлежат подмножеству $\mathcal{M}$. Действительно, учитывая (1.29), получим

$$\begin{aligned} R_{n+1} = [\alpha^{\xi_{l_n}} \pmod q]R_n = [\alpha^{\xi_{l_n}}\alpha^{\xi_{l_{n-1}}} \pmod q]R_{n-1} = \cdots \\ = [\alpha^{\gamma_0+\sum_{i=0}^{n}\xi_{l_i}} \pmod q]P = [\alpha^{\gamma_n} \pmod q]P \in \mathcal{M}, \end{aligned}$$

для $\gamma_n \equiv \gamma_0 + \sum_{i=0}^{n} \xi_{l_i} \pmod r$.

Аналогично, для точек последовательности $\{U_i\}_{i=0}^{\infty}$ выполнены равенства

$$\begin{aligned} U_{i+1} = [\alpha^{\xi_{l_i}} \pmod q]U_n = [\alpha^{\xi_{l_i}}\alpha^{\xi_{l_{i-1}}} \pmod q]U_{i-1} = \cdots \\ = [\alpha^{\omega_0+\sum_{j=0}^{n}\xi_{l_j}} \pmod q]Q = [\alpha^{\omega_i} \pmod q]Q = [\alpha^{x+\omega_i} \pmod q]P \in \mathcal{M}, \end{aligned}$$

для $\omega_i \equiv \omega_0 + \sum_{j=0}^{n} \xi_{l_j} \pmod r$.

Выберем индекс n_0, удовлетворяющий неравенству

$$n_0 \geqslant \left\lfloor \sqrt{\frac{\pi r}{2}} \right\rfloor$$

и зафиксируем точку R_{n_0}. Поскольку n_0 достаточно велико, можно ожидать, что точка R_0 лежит на цикле последовательности $\{R_n\}_{n=0}^{\infty}$.

Если найдется индекс i такой, что $R_{n_0} = U_i$, то выполнено равенство

$$[\alpha^{\gamma_{n_0}} \pmod q]P = [\alpha^{\omega_i} \pmod q]Q = [k\alpha^{\omega_i} \pmod q]P.$$

и можно определить неизвестное k сравнением

$$k \equiv \alpha^{\gamma_{n_0}-\omega_i} \pmod q. \tag{1.30}$$

Заметим также, что если выполнено равенство $R_{n_0} = U_i$, то, в силу определения отображения f, выполнены и равенства

$$R_{n_0+j} = U_{i+j}, \quad j = 0, 1, \ldots.$$

Следовательно, общие точки двух последовательностей будут принадлежать одному и тому циклу.

При случайном выборе точек R_0 и U_0 может случиться так, что последовательности $\{R_n\}_{n=0}^{\infty}$ и $\{U_i\}_{i=0}^{\infty}$ не будут иметь общих точек. Согласно [91] отображение f разбивает множество $\mathcal{M}$ на m непересекающихся областей. При этом, для математического ожидания $E_r(m, f)$ величины m выполнено равенство

$$\lim_{r\to\infty} \frac{E_r(m,f)}{\log_2 r} = 1.$$

Следовательно, можно ожидать, что для нахождения совпадающих точек $R_{n_0} = U_i$ и, как следствие, определения неизвестного k, потребуется выбрать $\log_2 r$ случайных пар R_0 и U_0. Суммируем сказанное в виде алгоритма.

Алгоритм 1.5: Алгоритм, использующий информацию о мультипликативном порядке

Вход : Мультипликативный порядок r, отображение f, заданное набором значений $\xi_0, \ldots, \xi_{s-1} \in \mathbb{Z}_r^*$, α – образующий мультипликативной подгруппы порядка r в $\mathbb{Z}_q^*$, см. (1.26), а также точки $P, Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$, связанные соотношением (1.19).

Выход : Значение k такое, что $Q = [k]P$.

1 Определить $c = 0$ и $n_0 = \lfloor \sqrt{\frac{\pi r}{2}} \rfloor$.
2 Выбрать случайное $\gamma_0 \in_R \mathbb{Z}_r^*$ и определить $R_0 = [\alpha^{\gamma_0} \pmod q]P$.
3 Используя (1.29) вычислить $\gamma_{n_0} \in \mathbb{Z}_r$ такое, что $R_{n_0} = [\alpha^{\gamma_{n_0}} \pmod q]P$.
4 Выбрать случайное $\omega_0 \in_R \mathbb{Z}_r^*$ и определить $U_0 = [\alpha^{\omega_0} \pmod q]Q$.
5 **Для всех** $i = 1, 2, \ldots, n_0$ **выполнять**
6 Вычислить точку $U_i = f(U_{i-1})$ и $\omega_i \in \mathbb{Z}_r$ такое, что $U_i = [\alpha^{\omega_i} \pmod q]Q$.
7 **Если** $U_i = R_{n_0}$ **то**
8 Определить неизвестное k сравнением $k \equiv \alpha^{\gamma_{n_0} - \omega_i}$ и завершить алгоритм.
9 **конец**
10 **конец**
11 Определить $c = c + 1$.
12 **Если** $c < \lfloor \log_2 r \rfloor$ **то**
13 вернуться к строке 2.
14 **конец**
15 Завершить алгоритм с уведомлением о неудаче.

Поскольку для каждой из последовательностей $\{R_n\}_{n=0}^{\infty}$ и $\{U_i\}_{i=0}^{\infty}$ вычисляется не более n_0 элементов с трудоемкостью не более, чем $2\lceil \log_2 q \rceil$ групповых операций, а общее количество вырабатываемых пар R_0 и U_0 не превышает $\lfloor \log_2 r \rfloor$, то алгоритмическая сложность алгоритма 1.5 не превышает

$$4n_0 \lfloor \log_2 r \rfloor \lceil \log_2 q \rceil < 2\sqrt{2\pi r} \log_2 r (\log_2 q + 1).$$

операций в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$. Данная оценка несколько хуже, чем утверждение теоремы 1.3. Вместе с тем в алгорит-

ме 1.5 требуется хранение лишь трех точек эллиптической кривой, что делает возможной его реализацию на ЭВМ при любых значениях r.

§ 1.3.2. Параллельный вариант алгоритма, основанный на идеях работы Ооршота-Винера

Алгоритм 1.5 может быть легко модифицирован для проведения параллельных вычислений. Опишем одну из возможных модификаций, использующую идеи Ооршота-Винера, см. [188], а также, так называемые, множества «точек-ловушек».

Пусть w натуральное число и у нас имеется $2w$ потоков, обладающих общей памятью. Мы предполагаем, что каждый поток может проводить вычисления независимо от остальных. Разделим все множество потоков на две равные группы – первая группа будет формировать множество «точек-ловушек» $\mathcal{S}(P)$, лежащих на циклах последовательностей, образованных точкой P, а вторая группа – множество «точек-ловушек» $\mathcal{S}(Q)$, лежащих на циклах последовательностей, образованных точкой Q.

Каждый поток из первой группы вырабатывает последовательности следующим образом. Выбирается случайный, отличный от нуля вычет $\gamma_0 \in \mathbb{Z}_r^*$ и формируется точка $R_0 = [\alpha^{\gamma_0}]P$. Остальные элементы последовательности формируются также, как и ранее

$$R_n = f(R_{n-1}), \quad n = 1, 2, \ldots,$$

где f определено (1.29) и

$$R_n = [\alpha^{\gamma_n} \pmod q]P, \quad \gamma_n \equiv \gamma_0 + \sum_{i=0}^{n} \xi_{l_i} \pmod r.$$

Аналогичным образом каждый поток из второй группы вычисляет последовательность

$$U_n = [\alpha^{\omega_n} \pmod q]Q, \quad \omega_k\omega_n \equiv \omega_0 + \sum_{j=0}^{n} \xi_{l_j} \pmod r,$$

которая стартует со случайной точки $U_0 = [a^{\omega_0}]Q$ и $\omega_0 \in \mathbb{Z}_r^*$.

После того, как каждый поток вычислит не менее $\dfrac{\sqrt{\pi r}}{2}$ элементов последовательности, начинается формирование множеств $\mathcal{S}(P)$ и $\mathcal{S}(Q)$. Для этого фиксируется некоторое натуральное число u (параметр алгоритма),

являющееся степенью двойки. Данное число является общим для всех вычислительных потоков. Каждый поток из первой группы выполняет следующую процедуру. Если точка $R_n = (x_{R_n}, y_{R_n})$ удовлетворяет условию

$$x_R \equiv 0 \pmod{u},$$

то

1) точка R_n помещается во множество $\mathcal{S}(P)$; также сохраняется соответствующее ей значение γ_n;

2) ищется совпадение точки R с элементами множества $\mathcal{S}(Q)$; если такое совпадение найдено, т.е. найдется точка $U_i = [\alpha^{\omega_i}]Q \in \mathcal{S}(Q)$ такая, что
$$[\alpha^{\gamma_n}]P = R_n = U_i = [k\alpha^{\omega_i}]P$$
то, используя аналогичное (1.30) равенство, поток определяет значение неизвестного k.

Такие же действия выполняет каждый поток из второй группы, формируя множество $\mathcal{S}(Q)$ и сравнивая вычисляемые точки U_n с элементами множества $\mathcal{S}(P)$.

Проиллюстрируем данный параллельный вариант алгоритма результатом проведенного эксперимента. При практических вычислениях мы рассматривали эллиптическую кривую, определяемую сравнением

$$y^2 \equiv x^3 - 2x + 831611549129771623857790233716768722 67 \pmod{p},$$

где $p = 170144519623114011343322490539658030031$ простое число, для которого выполнено равенство $\lceil \log_2 p \rceil = 128$. Порядок данной кривой равен $2q$, где простое число q удовлетворяет равенствам

$$\begin{aligned} q &= 85072259811557005664489355982895124223, \\ q-1 &= 2 \cdot 3 \cdot 139 \cdot 643 \cdot 12281 \cdot 51593 \cdot 53887 \cdot 4646248420547414411. \end{aligned}$$

Задача дискретного логарифмирования заключалась в поиске $k \in \mathbb{Z}_q^*$ такого, что $Q = [k]P$, где

$$P = (127079991335379663215392766670928701845,$$
$$1464786513378857537600045478132450557 80),$$

$$Q = (135768511924514909185266977775889591902,$$
$$1048338267121924343081112061656152496 92),$$

$|\langle P\rangle| = q$ и $Q \in \langle P\rangle$. Дополнительно было известно, что мультипликативный порядок неизвестного k равен $r = 34143537841471 = 12281{\cdot}51593{\cdot}53887$ и $\lceil \log_2 r \rceil = 46$.

Сперва мы выбрали образующий элемент $\alpha \in \mathbb{F}_q^*$, порядок которого равен r

$$a = 690386279342874007585445795480296580 20,$$

а также 46 случайных значений $\xi_0, \ldots, \xi_{45} \in \mathbb{Z}_r^*$, используемых дл определения случайного отображения f.

Случайные значения $\xi_0, \ldots, \xi_{45}$ выбирались таким образом, что бы количество ненулевых элементов в двоичном представлении вычетов

$$\zeta_i \equiv \alpha^{\xi_i} \pmod q, \quad i = 0, \ldots, 45$$

не превосходило 40 (константа 40 была подобрана экспериментально таким образом, чтобы с одной стороны, снизить время работы алгоритма, а с другой – не ухудшить свойств отображения f).

После этого компьютер (AMD процессор с 8 ядрами, 2.5Hz) вычислил 8 последовательностей (число вырабатываемых последовательностей $2w$ совпадало с числом ядер используемого процессора) для фиксированного значения $u = 1024$. Нашлись две точки R_n, U_i удовлетворяющие равенству

$$R_n = [178574163682345398152859188282742571 5]P = [392246999408260179955993732826343084 7]Q = U_i,$$

для некоторых индексов n,i и мы сразу же нашли неизвестное значение

$$k = 111993268900250420930399627452392772 10.$$

Время работы программы составило 224 секунд, множество $\mathcal{S}(P)$ содержало 2270, а множество $\mathcal{S}(Q)$ – 2356 точек эллиптической кривой.

Размер построенных в ходе выполнения алгоритма множеств $\mathcal{S}(P)$ и $\mathcal{S}(Q)$, очевидно, зависит от выбранного ранее параметра u. Предполагая, что x-координаты точек эллиптической кривой распределены равномерно в кольце $\mathbb{Z}_p$ можно ожидать, что размер множеств $\mathcal{S}(P)$ и $\mathcal{S}(Q)$ не будет превосходить величины

$$\frac{1}{u}\sqrt{\frac{\pi r}{8}} = 3576.$$

Данная величина определяется как средняя доля точек кривой, принадлежащих одному циклу последовательности $\{R_n\}_{n=0}^{\infty}$ или $\{U_n\}_{n=0}^{\infty}$. Полученные на практике значения оказались меньше ожидаемого максимального значения, поскольку совпадение $R_n = U_i$ было найдено до того, как был пройден весь цикл последовательности.

§ 1.3.3. Множество «слабых» ключей

Из теоремы 1.3 следует, что алгоритмическая сложность решения задачи дискретного логарифмирования $Q = [k]P$ зависит от мультипликативного порядка неизвестного $k \in \mathbb{Z}_q^*$. Теперь можно дать следующее определение.

Определение 1.4. *Пусть $p > 3$ – простое число, $\mathcal{E}_{a,b}(\mathbb{F}_p)$ эллиптическая кривая и $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ – точка кривой, порождающая циклическую подгруппу простого порядка q.*

Предположим, что нарушитель может за приемлемое время выполнить не более B операций в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, и будем называть «слабыми» ключами значения $k \in \mathbb{Z}_q$, принадлежащие множеству

$$K(B) = \left\{k : \ \mathtt{ord}_q\, k = r, \ \ 8\sqrt{r}\log_2 q < B\right\}. \tag{1.31}$$

Как следует из данного определения, множеству «слабых» ключей $K(B)$ принадлежат те значения k, для которых оценка теоремы 1.3 не превосходит величины B. При этом, само множество $K(B)$ существенно зависит как от параметров кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и порядка точки P, так и от возможностей нарушителя, т.е. величины B.

Поскольку применение «слабых» ключей в средствах защиты информации нецелесообразно, перед применением эллиптической кривой необходимо дать ответы на два вопроса:

- какое количество ключей является «слабыми»,
- чему будет равно среднее значение алгоритмической сложности решения задачи дискретного логарифмирования, если мы будем выбирать ключи случайным образом из всего множества возможных значений?

Хорошо известно, см. [275], что для каждого r количество k таких, что $\mathtt{ord}_q\, k = r$ и $r|q-1$ в точности равно $\varphi(r)$, где $\varphi()$ – функция Эйлера. Тогда для ответа на первый вопрос необходимо вычислить множество

$$\mathcal{R}_q = \left\{r : \quad r|q-1, \quad r < B_0 = \left(\frac{B}{8\log_2 q}\right)^2\right\} \tag{1.32}$$

и определить количество «слабых» ключей равенством

$$|K(B)| = \sum_{r\in\mathcal{R}_q} \varphi(r).$$

Запишем равенство

$$q - 1 = t \times q_1,$$

где t максимальный делитель числа $q-1$ такой, что $t < B_0$. Тогда, множество $\mathcal{R}_q$ составляют все делители числа t и, в силу равенства $\sum_{r|t} \varphi(r) = t$, выполнено $|K(B)| = t$.

Вероятность выбрать «слабый» ключ при случайном выборе секретного значения k равна

$$p(B) = \frac{|K(B)|}{q-1} = \frac{1}{q_1}.$$

Для ответа на второй вопрос обозначим символом $T(k,r)$ алгоритмическую сложность задачи определения неизвестного значения величины k при известном зачении параметра r. Определим равенством

$$T = \frac{1}{q-1} \sum_{0<k<q} T(k,r)$$

величину, которую будем называть *средним значением алгоритмической сложности* алгоритма дискретного логарифмирования. Легко заметить, что если величина $T(k,r)$ не зависит от k, то величина T совпадает со значением $T(k,r)$.

Применительно к алгоритму теоремы 1.3 имеем неравенство

$$T(k,r) < 8\sqrt{r}\log_2 q$$

и оценку

$$T = \frac{1}{q-1} \sum_{0<k<q} T(k,r) < \frac{8\log_2 q}{q-1} \sum_{r|q-1} \sqrt{r},$$

которая существенным образом зависит от разложения величины $q-1$ на простые сомножители.

Для иллюстрации сказанного, в работе [179] автором были получены оценки числа «слабых» ключей для эллиптических кривых, рекомендованных к применению в отечественных средствах защиты информации, см. Р 1323565.1.024-2019 [362].

В настоящее время рекомендуется применять эллиптические кривые вида

$$\mathcal{E}_{a,b}(\mathbb{F}_p): \quad y^2 \equiv x^3 - 3x + b \pmod p,$$

у которых простое число q – порядок подгруппы, порожденной точкой P, удовлетворяет равенству $q - 1 = t \times q_1$ при некотором большом простом q_1 и натуральном составном t. В таблице 1.3 приведены параметры эллиптических кривых из [362], для которых выполнено неравенство

$$2^{254} < q < 2^{256}.$$

набор	b	p	t
«A»	166	$2^{256}-617$	$2\cdot 3\cdot 7\cdot 17\cdot 37\cdot 127\cdot 121493\cdot 5592900119$
«B»	см. [362]	$2^{255}+3225$	$2\cdot 47336631894758162101$
«C»	32858	см. [362]	$2^3\cdot 3^2\cdot 5^2\cdot 47\cdot 207130852417\cdot 15398703602419036183$

Таблица 1.3: Параметры кривых из Р 1323565.1.024-2019, см. [362].

Для оценки возможностей нарушителя воспользуемся результатами работ [238, 239] и будем считать, что максимальное значение порядка подгруппы, в которой возможно решение задачи дискретного логарифмирования, не превосходит величины 2^{114}. Тогда, полагая величину B равной

$$B = 2^{57} = \sqrt{2^{114}},$$

для эллиптических кривых, параметры которых указаны в таблице 1.3, имеем оценку на размер мультипликативного порядка r

$$r < B_0 = \left(\frac{B}{8\log_2 q}\right)^2 < 2^{92}.$$

Воспользовавшись (1.32) легко проверить, что для указанных в таблице 1.3 эллиптических кривых выполнены неравенства

$$\begin{array}{rcl} 4.9\times 10^{28} & < & |K(B)|_A = 4951760157141521099596496896 < 2^{92}, \\ 9.4\times 10^{20} & < & |K(B)|_B = 94673263789516324202 < 2^{92}, \\ 1.3\times 10^{25} & < & |K(B)|_C = 1302730342287920575560000 < 2^{92}, \end{array}$$

где $|K(B)|_A$, $|K(B)|_B$, $|K(B)|_C$ определяют количество «слабых» ключей для кривых из наборов «A», «B» и «C».

Полученные значения для мощности множеств «слабых» ключей позволяют говорить о том, что секретные ключи в схемах защиты информации, использующих рекомендованные в [362] эллиптические кривые, не должны выбираться случайным образом.

Для выбора секретного ключа k может быть использовано следующее соображение. Пусть g – произвольный первообразный корень по модулю q. Тогда, секретный ключ $k \in \mathbb{Z}_q^*$ должен удовлетворять сравнению

$$k \equiv g^{\xi} \pmod p,$$

где $\xi \in_R \mathbb{Z}_q^*$ случайно выработанное значение, удовлетворяющее равенству $\mathbf{НОД}(\xi,\ q-1)=1$. В этом случае, согласно [275], величина k будет являться первообразным корнем по модулю q и $\mathtt{ord}_q\, k = q-1$.

§ 1.3.4. Обобщения предложенных алгоритмов

В качестве заключения рассмотрим, без вынесения на защиту, различные подходы к обобщению алгоритма, предложенного в ходе доказательства теоремы 1.3.

§ 1.3.4.1. Алгоритм, основанный на случайных сдвигах

Рассмотрим случай, когда $\mathtt{ord}_q\, k > B_0$ и величина B_0 определена ранее в (1.32). Предположим, что нам известен вычет $y \in \mathbb{Z}_q^*$ такой, что

$$\mathtt{ord}_q(k+y) < B_0. \tag{1.33}$$

Тогда, определив точку R равенством $R = [y]P$, получим равенство

$$Q_1 = Q + R = [k]P + [y]P = [\alpha^n]P,$$

для α такого, что $\mathtt{ord}_q\, \alpha = \mathtt{ord}_p(k+y)$. Полученное равенство позволяет применить алгоритм теоремы 1.3.

Поскольку при случайном выборе $y \in \mathbb{Z}_q^*$ вероятность выполнения условия (1.33) мала, то описанный метод не интересен с практической точки зрения. Вместе с тем, он позволяет свести задачу дискретного логарифмирования в группе точек эллиптической кривой к решению другой сложной задачи, рассматриваемой в кольце вычетов $\mathbb{Z}_q$.

Пусть $s \in \mathbb{N}$ и $k_1, \ldots, k_{s-1} \in \mathbb{Z}_q$ – известные фиксированные вычеты кольца $\mathbb{Z}_q$. Если существует алгоритм такой, что для любого неизвестного значения $k = k_0 \in \mathbb{Z}_q$ могут быть найдены значения $a_0, \ldots, a_{s-1} \in \mathbb{Z}_q^*$ такие, что найдется некоторый элемент $k_s \in \mathbb{Z}_q^*$,

$$k_s = \sum_{i=0}^{s-1} a_i k_i \quad \text{и} \quad \mathtt{ord}_q\, k_s < B_0,$$

то такой алгоритм позволит найти решение задачи дискретного логарифмирования в группе точек эллиптической кривой. Действительно, в силу равенства

$$Q_s = [k_s]P = [a_0]Q + \sum_{i=1}^{s-1} [a_i k_i]P$$

величина k_s может быть найдена с помощью утверждения теоремы 1.3, а потом значение k_0 определено сравнением

$$k_0 \equiv a_0^{-1} \left(k_s - \sum_{i=1}^{s-1} a_i k_i \right) \pmod q .$$

§ 1.3.4.2. Алгоритм, основанный на применении рекуррентных последовательностей

Формулировка теоремы 1.3 допускает следующее обобщение.

Теорема 1.4. *Пусть $p > 3$ – простое число, $\mathcal{E}_{a,b}(\mathbb{F}_p)$ эллиптическая кривая и $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ – точка кривой, порождающая циклическую подгруппу $\langle P\rangle \subseteq E_{a,b}(\mathbb{F}_p)$ простого порядка q.*

Пусть $Q = [k]P$ для некоторого неизвестного $k \in \mathbb{Z}_q^*$ и $\mathcal{M}$ произвольное конечное подмножество в $\langle P\rangle$ такое, что ему принадлежат точки P и Q. Обозначим символом $r \in \mathbb{N}$ мощность множества $|\mathcal{M}|$. Если найдется способ перечисления элементов множества $\mathcal{M}$, т.е. эффективно вычислимые отображения

$$f:\ \mathbb{Z}_r \to \mathcal{M}, \quad g:\ \mathbb{Z}_r \to \mathbb{Z}_q$$

такие, что

1. $f(i) \neq f(j)$ *при* $i \neq j \in \mathbb{Z}_r$,
2. $f(i) = [g(i)]P$,

то найдется алгоритм, вычисляющий неизвестное k с алгоритмической сложностью, не превосходящей величины $2T_f\sqrt{r}$ групповых операций, где T_f алгоритмическая сложность вычисления отображения f.

Доказательство сформулированной теоремы аналогично доказательству теоремы 1.3.

Первый вариант такого множества $\mathcal{M}$ рассматривал Дж.Поллард, см. работу [196, стр. 922], предложивший искать с помощью λ-метода неизвестное k, удовлетворяющее неравенствам $A \leqslant k \leqslant B$ для некоторых заданных значений $A, B \in \mathbb{Z}_q^*$. Для перечисления элементов множества $\mathcal{M}$ Поллардом использовались отображения

$$\begin{aligned} &g(i) \equiv A + i \pmod q, \\ &f(i) = [g(i)]P, \quad i = 0, 1, \ldots, r-1, \quad r = B - A + 1. \end{aligned}$$

Аналогично, в доказательстве теоремы 1.3 для перечисления множества $\mathcal{M}$ автором использовались отображения

$$\begin{aligned} &g(i) \equiv \alpha^i \pmod q, \\ &f(i) = [\alpha^i \pmod q]P, \quad i = 0, 1, \ldots, r-1, \end{aligned}$$

для $\alpha \in \mathbb{Z}_q^*$ такого, что $\mathtt{ord}_q\,\alpha = r$.

Исходя из сказанного, можно описать большой класс множеств $\mathcal{M}$, для которых существуют эффективные перечисления, введенные в утверждении теоремы 1.4. Пусть

$$a(x) = x^n - a_{n-1}x^{n-1} - \cdots - a_0 \in \mathbb{F}_q[x]$$

неприводимый многочлен степени $n \geqslant 2$. Пусть α корень многочлена $a(x)$ в поле $\mathbb{F}_{q^n}$ такой, что $\mathtt{ord}_{q^n}\alpha = r$ и $r > n$.

Выберем произвольные начальные значения $g_0, \ldots, g_{n-1} \in \mathbb{Z}_q$, тогда рекуррентная последовательность

$$g_{i+n} \equiv a_{n-1}g_{i+(n-1)} + \cdots + a_0 g_i \pmod{q}, \tag{1.34}$$

где $g_i \in \mathbb{F}_q$, $i = 0, 1, \ldots$, является чисто периодической последовательностью с периодом r, см. [315], при этом известно, что

$$r | q^n(q-1)(q^2-1)\cdots(q^n-1).$$

Тогда, в качестве множества $\mathcal{M}$, зависящего от многочлена $a(x)$ и начальных значений $g_0, \ldots, g_{n-1} \in \mathbb{Z}_q^*$, можно определить

$$\mathcal{M} = \{[g_i]P, \quad i = 0, \ldots, r-1\},$$

при этом отображение $g(i) = g_i$ задает перечисление множества $\mathcal{M}$.

Если известно, что точка $Q = [k]P$ принадлежит множеству $\mathcal{M}$, то существует индекс $i \in \mathbb{Z}_r$ такой, что $k = g_i$, и значение k может быть определено с помощью описанных ранее алгоритмов.

Вместе с тем, в случае, когда множество $\mathcal{M}$ неизвестно, применение описываемого подхода не целесообразно. Действительно, при фиксированном множестве $\mathcal{M}$ и случайном выборе точки $Q \neq \mathcal{O}$, вероятность того, что $Q \in \mathcal{M}$, будет равна $\frac{r}{q-1}$. При значениях r, близких к единице, полученная вероятность ничтожна, а при r, близких к $q-1$, алгоритмическая сложность поиска k, согласно утверждению теоремы 1.4, будет превышать алгоритмическую сложность методов, изложенных в § 1.2

Заключение к § 1.3

В § 1.3 доказана теорема о том, что алгоритмическая сложность решения задачи дискретного логарифмирования в группе точек эллиптической кривой зависит от значения мультипликативного порядка неизвестного.

Предложен и практически реализован алгоритм, решающий задачу дискретного логарифмирования в группе точек эллиптической кривой с алгоритмической сложностью $O(\sqrt{r}\log q)$, где q порядок подгруппы, в которой решается задача дискретного логарифмирования, а r мультипликативный порядок неизвестного.

Предложен способ оценки множества «слабых» ключей, для которых алгоритмическая сложность решения задачи дискретного логарифмирования в группе точек эллиптической кривой меньше, чем $O(\sqrt{q})$. Для эллиптических кривых, рекомендованных к использованию в отечественных средствах защиты информации, получены точные значения мощностей множеств «слабых» ключей.

§ 1.4. Эндоморфизмы эллиптических кривых

В этом параграфе рассматриваются вопросы применения эндоморфизмов эллиптических кривых для реализации операции вычисления кратной точки эллиптической кривой.

В начале параграфа приводятся дополнительные сведения из теории комплексного умножения. Данная теория разрабатывалась в работах Г. Вебера, Г. Хассе, М. Дойринга, Г. Шимуры и др. Полное изложение соответствующих результатов может быть найдено в [109, 233, 255, 314, 382], исторический обзор см. в [266, гл. 11]. Далее в параграфе:

- излагается и обосновывается алгоритм поиска эндоморфизмов эллиптических кривых, см. далее алгоритм 1.6,
- приводятся результаты практической реализации изложенного алгоритма,
- предлагается способ выбора формы эллиптической кривой, минимизирующей трудоемкость вычисления найденных эндоморфизмов,
- в заключение, рассматривается способ вычисления кратной точки эллиптической кривой, использующий найденные эдоморфизмы.

§ 1.4.1. Сведения из теории комплексного умножения

Пусть $\mathbb{C}$ поле комплексных чисел. Зафиксируем пару чисел $\omega_1, \omega_2 \in \mathbb{C}$ и рассмотрим решетку

$$\Lambda = \Lambda(\omega_1, \omega_2) = \{n\omega_1 + m\omega_2,\ n, m \in \mathbb{Z}\}\,.$$

Следуя [294, ч. 2, гл. 1], см. также [382, лекция 2, § 5], определим зависящие от решетки Λ величины

$$g_2(\Lambda) = 60G_2(\Lambda), \quad g_3(\Lambda) = 140G_3(\Lambda), \tag{1.35}$$

где $G_k(\Lambda)$ определено равенством

$$G_k(\Lambda) = \sum_{\omega\in\Lambda\setminus\{0\}} \frac{1}{\omega^{2k}}, \quad k = 2, 3, \ldots,$$

и рассмотрим дифференциальное уравнение

$$\left(\wp'(z)\right)^2 = 4\wp(z)^3 - g_2(\Lambda)\wp(z) - g_3(\Lambda).$$

Решением данного уравнения является функция Вейерштрасса, определяемая равенством

$$\wp(z,\Lambda) = \wp(z) = \frac{1}{z^2} + \sum_{n=2}^{\infty} c_n z^{2n-2}, \quad z, c_n \in \mathbb{C}, \tag{1.36}$$

в котором коэффициенты разложения функции $\wp(z)$ в ряд, см. [294, ч.2, гл.1, стр. 222], удовлетворяют рекуррентному соотношению

$$\begin{aligned} 20c_2 &= g_2(\Lambda), \\ 28c_3 &= g_3(\Lambda), \\ (n-3)(2n+1)c_n &= 3\sum_{k=2}^{n-2} c_k c_{n-k}, \quad n = 4, 5, \ldots \end{aligned} \tag{1.37}$$

Функция Вейерштрасса $\wp(z)$ является четной эллиптической функцией, т.е. функцией удовлетворяющей равенствам

$$\wp(-z) = \wp(z) \quad \text{и} \quad \wp(z) = \wp(z+\omega),$$

для любого $\omega \in \Lambda$.

Определим в качестве эллиптической кривой $\mathcal{E}_\Lambda(\mathbb{C})$ множество точек $(x_1 : x_2 : x_3)$ проективного пространства $\mathbb{P}^2(\mathbb{C})$, удовлетворяющих однородному уравнению[4]

$$\mathcal{E}_\Lambda(\mathbb{C}): \quad x_2^2 x_3 = 4x_1^3 - g_2(\Lambda)x_1 x_3^2 - g_3(\Lambda)x_3^3, \tag{1.38}$$

а также бесконечно удаленную точку $\mathcal{O} = (0 : 1 : 0)$. Функция Вейерштрасса $\wp(z)$ задает параметризацию множества $\mathcal{E}_\Lambda(\mathbb{C})$ точек эллиптической кривой:

$$\begin{aligned} \mathbb{C}/\Lambda &\to \mathcal{E}_\Lambda(\mathbb{C}), \\ z &\to (\wp(z) : \wp'(z) : 1), \\ 0 &\to \mathcal{O}. \end{aligned}$$

[4]Заметим, что замена переменных $(x_1 : x_2 : x_3) \to (x_1 : \frac{x_2}{2} : x_3)$ приводит эллиптическую кривую (1.38) к виду (1.1) со следующими параметрами: $a = -\frac{g_2}{4}$ и $b = -\frac{g_3}{4}$.

§ 1.4.1.1. Отображения эллиптических кривых

Пусть Λ_1 и Λ_2 две решетки в $\mathbb{C}$ и $\alpha \in \mathbb{C}^*$ такой элемент, что $\alpha\Lambda_1 \subseteq \Lambda_2$. Тогда, умножение на α индуцирует гомоморфизм

$$\phi_\alpha(z) = \alpha z \pmod{\Lambda_2}$$

и отображение между эллиптическими кривыми

$$\begin{array}{rcl} \mathcal{E}_{\Lambda_1}(\mathbb{C}) & \to & \mathcal{E}_{\Lambda_2}(\mathbb{C}) \\ \mathcal{O} & \to & \mathcal{O}, \\ \big(\wp(z,\Lambda_1) : \wp'(z,\Lambda_1) : 1\big) & \to & \big(\wp(\alpha z,\Lambda_2) : \wp'(\alpha z,\Lambda_2) : 1\big). \end{array} \tag{1.39}$$

Теорема 1.I (см. [233, гл. VI, § 4])**.** *Две эллиптические кривые, определяемые, соответственно, решетками Λ_1 и Λ_2 изоморфны тогда и только тогда, когда $\alpha\Lambda_1 = \Lambda_2$ для некоторого $\alpha \in \mathbb{C}^*$.*

Рассмотрим модулярную функцию

$$j(\Lambda) = 1728 \cdot \frac{g_2^3(\Lambda)}{g_2^3(\Lambda) - 27g_3^2(\Lambda)}, \tag{1.40}$$

которую принято называть j-инвариантом эллиптической кривой $\mathcal{E}_\Lambda(\mathbb{C})$. Из равенств $\alpha\Lambda_1 = \Lambda_2$ и (1.35) следует, что

$$g_2(\Lambda_2) = \alpha^4 g_2(\Lambda_1), \quad g_3(\Lambda_2) = \alpha^6 g_3(\Lambda_1). \tag{1.41}$$

Тогда выполнено равенство

$$j(\Lambda_2) = 1728 \cdot \frac{g_2^3(\Lambda_2)}{g_2^3(\Lambda_2) - 27g_3^2(\Lambda_2)} = 1728 \cdot \frac{\alpha^{12} g_2^3(\Lambda_1)}{\alpha^{12} g_2^3(\Lambda_1) - 27\alpha^{12} g_3^2(\Lambda_1)} = j(\Lambda_1),$$

из которого следует, что инварианты изоморфных кривых совпадают.

Для произвольной решетки Λ, без ограничения общности, будем считать, что $\operatorname{Im} \frac{\omega_2}{\omega_1} \geqslant 0$ и определим[5] элемент $\tau = \frac{\omega_2}{\omega_1} \in \mathbb{C}_+$.

Выбирая $\alpha = \frac{1}{\omega_1}$ из теоремы 1.I получаем, что эллиптическая кривая $\mathcal{E}_\Lambda(\mathbb{C})$ изоморфна эллиптической кривой $\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$, где $\Lambda_\tau = \{n + m\tau, \tau \in \mathbb{C}_+\}$. Далее будем рассматривать $j(\Lambda_\tau)$ как функцию переменного $\tau \in \mathbb{C}_+$ и использовать обозначение $j(\tau) = j(\Lambda_\tau)$.

Для вычисления значений функции $j(\tau)$ можно использовать следующие соотношения. Определим функцию Дедекинда $\eta(\tau)$ и функцию Вебера $\mathfrak{f}_1(\tau)$, см. [255, § 34],

$$\eta(\tau) = q^{24} \prod_{n=1}^{\infty} (1 - q^n), \quad \mathfrak{f}_1(z\tau) = \frac{\eta(2\tau)}{\eta(\tau)}, \qquad \text{где} \quad q = e^{2\pi i \tau},$$

[5] Если выполнено обратное неравенство $\operatorname{Im} \frac{\omega_2}{\omega_1} < 0$, то можно определить $\tau = \frac{\omega_1}{\omega_2} \in \mathbb{C}_+$.

тогда

$$j(\tau) = \frac{(\mathfrak{f}_1(\tau)^{24} + 16)^3}{\mathfrak{f}_1(\tau)^{24}}.$$

В [382, лекция 6, § 6] приводится равенство

$$j(\tau) = \frac{(1 + 240 \sum_{n=1}^{\infty} \sigma_3(n) q^n)^3}{q \prod_{n=1}^{\infty} (1 - q^n)^{24}}, \tag{1.42}$$

где $q = e^{2\pi i \tau}$, а $\sigma_k(n)$ – есть сумма k-х степеней всех положительных делителей числа n, т.е. $\sum_{d|n} d^k$, которое также можно использовать для эффективного вычисления значений функции $j(\tau)$.

Значение $j(\tau)$, отличное от 0 и 1728, может быть использовано для определения[6] множества изоморфных кривых, j-инвариант которых совпадает со значением $j(\tau)$. Положим

$$k = \frac{j(\tau)}{j(\tau) - 1728},$$

и определим коэффициенты эллиптической кривой $\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$ равенствами

$$g_2(\Lambda_\tau) = 3k, \quad g_3(\Lambda_\tau) = k.$$

Выберем произвольное число $\alpha \in \mathbb{C}^*$ и определим решетку $\Lambda_\alpha = \alpha\Lambda_\tau$. Тогда, с учетом (1.38), (1.41), получаем

$$\begin{array}{rcl} g_2(\Lambda_\alpha) & = & 3k\alpha^4, \\ g_3(\Lambda_\alpha) & = & k\alpha^6, \\ \mathcal{E}_{\Lambda_\alpha}(\mathbb{C}) : y^2 z & = & 4x^3 - 3k\alpha^4 x z^2 - k\alpha^6 z^3, \end{array} \tag{1.43}$$

и

$$j(\Lambda_\alpha) = 1728 \frac{g_2^3(\Lambda_\alpha)}{g_2^3(\Lambda_\alpha) - 27 g_3^2(\Lambda_\alpha)} = 1728 \frac{(3kc^2)^3}{(3kc^2)^3 - 27(kc^3)^2} =$$

$$= 1728 \frac{k}{k-1} = 1728 \frac{\frac{j(\tau)}{j(\tau)-1728}}{\frac{j(\tau)}{j(\tau)-1728} - 1} = j(\tau).$$

Теперь рассмотрим случай, когда найдется элемент $\alpha \in \mathbb{C}^*$ такой, что $\alpha\Lambda \subset \Lambda$. Множество таких значений α индуцирует множество определяемых (1.39) эндоморфизмов ϕ_α эллиптической кривой $\mathcal{E}_\Lambda(\mathbb{C})$

$$\mathtt{End}(\mathcal{E}) = \{\phi_\alpha : \mathcal{E}_\Lambda(\mathbb{C}) \to \mathcal{E}_\Lambda(\mathbb{C})\} \sim \{\alpha \in \mathbb{C}^* : \ \alpha\Lambda \subset \Lambda\}.$$

[6] Эллиптические кривые, со значением j-инварианта равным 0 или 1728 определяются, например, в [233, гл. III, § 1].

Если $\phi_\alpha, \phi_\beta \in \mathtt{End}(\mathcal{E})$ два различных эндоморфизма, то равенства

$$(\phi_\beta \cdot \phi_\alpha)(P) = \phi_\beta(\phi_\alpha(P)), \quad (\phi_\alpha + \phi_\beta)(P) = \phi_\alpha(P) + \phi_\beta(P),$$

выполненные для любой точки $P \in \mathcal{E}_\Lambda(\mathbb{C})$, определяют на $\mathtt{End}(\mathcal{E})$ структуру кольца, нулем которого является отображение бесконечно удаленной точки $\mathcal{O}$ в себя, а единицей – тождественное отображение.

Теорема 1.II (см. [233, гл. III, § 9]). *Кольцо эндоморфизмов* $\mathtt{End}(\mathcal{E})$ *эллиптической кривой* $\mathcal{E}_\Lambda(\mathbb{C})$ *изоморфно кольцу целых чисел* $\mathbb{Z}$*, либо порядку некоторого мнимого квадратичного поля, либо порядку некоторой алгебры кватернионов.*

Согласно [294, ч.2, гл.1, § 9], см. также [64], эндоморфизм ϕ_α может быть определен в явном виде отображением

$$\phi_\alpha : (x, y) \to \left(R(x), \frac{R'(x)y}{\alpha} \right), \tag{1.44}$$

где $R(x)$ некоторая рациональная функция, однозначно определяемая величиной $\alpha \in \mathbb{C}^*$. Алгоритм вычисления рациональной функции $R(x)$ для заданной эллиптической кривой $\mathcal{E}_\Lambda(\mathbb{C})$ и заданного значения α рассматривается далее в § 1.4.2.

§ 1.4.1.2. Редукция в конечное простое поле

Рассмотрим $SL_2(\mathbb{Z})$ – группу квадратных матриц

$$SL_2(\mathbb{Z}) = \left\{ \mathcal{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \det \mathcal{A} = ad - bc = 1 \right\}$$

и определим ее действие на $\mathbb{C}$ равенством $\mathcal{A}(z) = \dfrac{az + b}{cz + d}$, $z \in \mathbb{C}$.

Определим фундаментальную область $\mathbb{D} \subset \mathbb{C}_+$ – множество комплексных чисел z, удовлетворяющих условиям

$$\mathbb{D} = \left\{ z \in \mathbb{C}_+, \ -\frac{1}{2} < z \leqslant \frac{1}{2}, \text{ если } \operatorname{Re} z > 0, \text{то } |z| \geqslant 1, \text{иначе } |z| > 1 \right\}.$$

Теорема 1.III (см. [290, 382]). *Для любого* $\tau \in \mathbb{C}_+$ *выполнены следующие утверждения:*

1. *найдется матрица* $\mathcal{A} \in SL_2(\mathbb{Z})$ *такая, что* $\mathcal{A}(\tau) \in \mathbb{D}$.
2. *для любой матрицы* $\mathcal{A} \in SL_2(\mathbb{Z})$ *выполнено* $j(\tau) = j(\mathcal{A}(\tau))$.

Из утверждения теорем 1.I и 1.III следует, что для любой эллиптической кривой, заданной решеткой $\Lambda = \{n\omega_1 + n\omega_2\}$, найдется изоморфная ей эллиптическая кривая, заданная решеткой Λ_τ и $\tau \in \mathbb{D}$. Алгоритм вычисления матрицы $\mathcal{A} \in \mathbb{SL}_2(\mathbb{Z})$ такой, что $\mathcal{A}\left(\frac{\omega_2}{\omega_1}\right) \in \mathbb{D}$ может быть найден в монографии [277, гл. 2].

Рассмотрим случай, когда $\tau \in \mathbb{D}$ является мнимой квадратичной иррациональностью. Пусть d – натуральное свободное от квадратов число. Определим равенством

$$\Delta = \begin{cases} d, & \text{если} \quad d \equiv 3 \pmod 4, \\ 4d, & \text{если} \quad d \equiv 1, 2 \pmod 4, \end{cases} \tag{1.45}$$

величину, называемую фундаментальным дискриминантом, см. [61], и зафиксируем мнимое квадратичное поле $\mathbb{K} = \mathbb{Q}(\sqrt{-d})$, образованное присоединением к полю рациональных чисел величины $\sqrt{-d}$. Следуя [273, гл. 2, § 7] рассмотрим кольцо целых поля $\mathbb{K}$

$$\mathbb{Z}_{\mathbb{K}} = \{n + m\omega\},$$

где

$$\omega = \begin{cases} \sqrt{-d}, & \text{если} \quad d \equiv 1, 2 \pmod 4, \\ \frac{1+\sqrt{-d}}{2}, & \text{если} \quad d \equiv 3 \pmod 4. \end{cases} \tag{1.46}$$

и обозначим $\tau = f\omega$ для некоторого $f \in \mathbb{N}$, тогда $\Lambda_\tau \subseteq \mathbb{Z}_{\mathbb{K}}$.

Пусть $QF(-\Delta) =$

$$\left\{ Q(x, y) = Ax^2 + Bxy + Cy^2, \quad A, B, C \in \mathbb{Z}, \quad B^2 - 4AC = -\Delta \right\}$$

множество всех квадратичных форм, дискриминант которых равен $-\Delta$. Согласно гауссовой теории квадратичных форм, см. [290], будем говорить, что две формы $Q, Q_1 \in QF(-\Delta)$ эквивалентны, если

$$Q_1(x, y) = \mathcal{A} \cdot Q(x, y) = Q(ax + by, cx + dy), \tag{1.47}$$

для некоторой $\mathcal{A} \in SL_2(\mathbb{Z})$. Условие (1.47) разбивает все множество квадратичных форм $SL_2(\mathbb{Z})$ на классы эквивалентности.

Напомним, см., например, [277, гл.2, § 3], что квадратичная форма $Q(x, y) \in QF(-\Delta)$ называется приведенной, если выполнены следующие условия:

1. $\mathbf{НОД}(A,\ B,\ C) = 1$,
2. $|B| \leqslant A \leqslant C$,
3. если $|B| = A$ или $A = C$, то $B \geqslant 0$.

С каждой квадратичной формой $Q(x,y) \in QF(-\Delta)$ можно однозначно связать решетку Λ_τ, где $\tau \in \mathbb{K}$ – мнимая квадратичная иррациональность такая, что $\tau \in \mathbb{C}_+$ и $Q(\tau,1) = 0$. Условие приведенности квадратичной формы $Q(x,y)$ эквивалентно тому, что связанная с ней квадратичная иррациональность τ принадлежит фундаментальной области $\mathbb{D}$.

Теорема 1.IV (Гаусс, см. [277, гл.3, § 3], [290, раздел 5]). *Выполнены следующие утверждения.*

1. *Для любой квадратичной формы $Q(x,y) \in QF(-\Delta)$ существует, и при том единственная, эквивалентная ей приведенная квадратичная форма.*

2. *Число неэквивалентных приведенных форм в $QF(-\Delta)$ конечно.*

Теорема 1.V (см. [382, лекция 3]). *Пусть h – число классов эквивалентных квадратичных форм из $QF(-\Delta)$ и $\tau_1, \ldots, \tau_h \in \mathbb{K} = \mathbb{Q}(\sqrt{-d})$ мнимые квадратичные иррациональности, связанные с приведенными формами из различных классов эквивалентности. Выполнены следующие утверждения:*

1. *значения модулярной функции $j(\tau_i)$, $i \in 1, \ldots, h$, на классе эквивалентных квадратичных форм из $QF(-\Delta)$ совпадают и являются целыми алгебраическими числами;*

2. *найдется унитарный[7] многочлен $H(x) \in \mathbb{Z}[x]$ такой, что*

$$H(j(\tau_i)) = 0, \quad \deg H(x) = h,$$

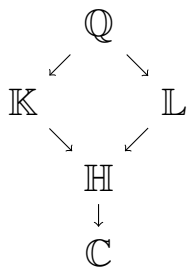
   для всех $i \in 1, \ldots, h$;

3. *поле $\mathbb{H} = \mathbb{Q}\left(\sqrt{-d}, j(\tau_i)\right)$, образованное присоединением к полю $\mathbb{K}$ значения модулярной функции $j(\tau_i)$, не зависит от выбора индекса $i \in [1, \ldots, h]$ и является максимальным неразветвленным расширением поля $\mathbb{K}$.*

Поле $\mathbb{H}$, определенное в утверждении теоремы 1.V, принято называть полем классов Гильберта. Алгоритм построения многочлена $H(x)$ основывается на переборе всех приведенных квадратичных форм фиксированного дискриминанта и может быть найден в работах [8, 61].

Для $\tau \in \mathbb{K} = \mathbb{Q}(\sqrt{-d})$ обозначим символом $\mathbb{L} = \mathbb{Q}\left(j(\tau)\right)$ поле, образованное присоединением к полю рациональных чисел значения модулярной функции $j(\tau)$. Далее будем рассматривать следующую башню полей

[7] Под унитарным подразумевается многочлен со старшим коэффициентом равным единице.



Основываясь на равенствах (1.43) и выбирая $\alpha \in \Lambda_\tau \subseteq \mathbb{Z}_{\mathbb{K}} \subset \mathbb{Q}(\sqrt{-d})$ будем считать, что рассматриваемые нами эллиптические кривые и отображения между ними определены над полем $\mathbb{H}$. Поскольку в криптографических приложениях используются эллиптические кривые, определенные над конечным простым полем $\mathbb{F}_p$, $p > 3$, необходима процедура редукции эллиптических кривых в конечное поле $\mathbb{F}_p$. Реализация этой процедуры основана на следующей теореме.

Теорема 1.VI (см. [109, глава 13] и [314]). *Пусть $p > 3$ нечетное простое число и d натуральное, свободное от квадратов такое, что уравнение $4p = t^2 + ds^2$ разрешимо относительно целых значений t, s.*

Пусть $\Lambda_\tau \subseteq \mathbb{Z}_{\mathbb{K}}$ – некоторый порядок мнимого квадратичного поля $\mathbb{Q}(\sqrt{-d})$, где $\tau = f\omega$, ω определяется равенством (1.46) и $f \in \mathbb{N}$.

Рассмотрим минимальный многочлен $H(x) \in \mathbb{Z}[x]$ величины $j(\tau)$, определяемой равенством (1.42), и выберем произвольный корень j_p многочлена $H(x) \pmod p$.

Тогда найдется эллиптическая кривая $\mathcal{E}_{a,b}(\mathbb{F}_p)$, определенная над конечным полем $\mathbb{F}_p$ такая, что

1. *кольцо эндоморфизмов $\mathtt{End}(\mathcal{E}_{a,b}(\mathbb{F}_p))$ изоморфно Λ_τ;*
2. *величина $j(\mathcal{E}_{a,b})$, определенная сравнением*

$$j(\mathcal{E}_{a,b}) \equiv 1728 \cdot \frac{4a^3}{4a^3 - 27b^2} \pmod p, \tag{1.48}$$

   удовлетворяет сравнению $j(\mathcal{E}_{a,b}) \equiv j_p \pmod p$;
3. *выполнено равенство $|\mathcal{E}_{a,b}(\mathbb{F}_p)| = p + 1 + t$;*
4. *найдется другая эллиптическая кривая[8] $\mathcal{E}_{a',b'}(\mathbb{F}_p)$, определенная над тем же полем $\mathbb{F}_p$ такая, что*
   - *4.1. $j(\mathcal{E}_{a',b'}) \equiv j(\mathcal{E}_{a,b}) \pmod p$,*
   - *4.2. выполнено равенство $|E_{a',b'}| = p + 1 - t$.*

[8] Эллиптическую кривую $\mathcal{E}_{a',b'}(\mathbb{F}_p)$, удовлетворяющую утверждению 4 теоремы 1.VI, принято называть твистом эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, см. [127].

Величину $j(\mathcal{E}_{a,b})$, определенную сравнением (1.48) принято, по аналогии с комплексным случаем, называть j-инвариантом кривой. Коэффициенты эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ могут быть выражены через инвариант кривой $j(\mathcal{E}_{a,b})$ следующим образом

$$\begin{cases} a \equiv 3kc^2 \pmod p, \\ b \equiv 2kc^3 \pmod p, \end{cases} \quad \text{где} \quad k \equiv \frac{j(\mathcal{E}_{a,b})}{1728 - j(\mathcal{E}_{a,b})} \pmod p, \tag{1.49}$$

для любого вычета $c \in \mathbb{F}_p^*$. Отметим, что в § 1.5 приводится разработанный автором алгоритм построения эллиптических кривых специального вида, основанный на утверждениях теоремы 1.VI.

Если мы рассмотрим произвольную эллиптическую кривую

$$\mathcal{E}_{a,b}(\mathbb{F}_p): \quad y^2 \equiv x^3 + ax + b \pmod p,$$

для $a, b \in \mathbb{F}_p$ и $4a^3 + 27b^2 \not\equiv 0 \pmod p$, то ее кольцо эндоморфизмов, см. [66] или [109, гл. 12], изоморфно порядку некоторой алгебры кватернионов только в том случае, когда порядок группы точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ равен $p+1$. Такие кривые принято называть суперсингулярными.

В остальных случаях эллиптическая кривая называется ординарной, а ее кольцо эндоморифзмов $\mathtt{End}(\mathcal{E}_{a,b}(\mathbb{F}_p))$ изоморфно некоторому порядку $\Lambda_\tau \subseteq \mathbb{Z}_\mathbb{K}$ мнимого квадратичного поля $\mathbb{K}$. Заметим, что алгоритмы вычисления кольца эндоморфизмов для ординарной эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, заданной своими коэффициентами $a, b \in \mathbb{F}_p$, могут быть найдены в работах [36, 56, 129].

В заключение отметим, что редукция определяемых в (1.39) отображений между эллиптическими кривыми сводится к редукции коэффициентов рациональных функций $R(x), R'(x)$, см. (1.44), где

$$\wp(\alpha z) = R(\wp(z)) = \frac{P(\wp(z))}{Q(\wp(z)}, \quad P(x), Q(x) \in \mathbb{H}[x].$$

Если $j(\tau) \notin \mathbb{K}$, то определим $\xi = \tau + j(\tau)$ такой, что

$$\mathbb{H} = \mathbb{Q}\left(\sqrt{-d}, j(\tau)\right) = \mathbb{Q}(\xi).$$

Рассмотрим минимальный многочлен $m(x) \in \mathbb{Q}(x)$, корнем которого является ξ. Тогда, $\deg m(x) = 2h$, где h – степень участвующего в утверждениях теорем 1.V и 1.VI многочлена $H(x)$, корнем которого является величина $j(\tau)$.

Любой элемент $\gamma \in \mathbb{H}$ может быть записан в виде

$$\gamma = c_{2h-1}\xi^{2h-1} + c_{2h-2}\xi^{2h-2} + \cdots + c_1\xi + c_0,$$

где $c_{2h-1}, \ldots, c_0 \in \mathbb{Q}$.

Пусть $e \in \mathbb{F}_p$ корень многочлена $m(x)$ в поле $\mathbb{F}_p$, где p удовлетворяет условиям теоремы 1.VI. Тогда редукция элемента γ из поля $\mathbb{H}$ в поле $\mathbb{F}_p$ определяется следующим образом

$$\gamma \ \to \ c_{2h-1}e^{2h-1} + c_{2h-2}e^{2h-2} + \cdots + c_1e + c_0 \pmod p .$$

§ 1.4.2. Алгоритм вычисления эндоморфизмов эллиптической кривой

Пусть заданы $\tau \in \mathbb{D} \subset \mathbb{C}_+$ – мнимая квадратичная иррациональность, решетка

$$\Lambda_\tau = \{n + m\tau,\ n, m \in \mathbb{Z}\} \subseteq \mathbb{Z}_{\mathbb{K}} \subset \mathbb{K} = \mathbb{Q}(\sqrt{-d}),$$

определяемая для некоторого свободного от квадратов числа $d \in \mathbb{N}$, а также отличный от нуля элемент $\alpha \in \Lambda_\tau$. Также мы будем считать, что величина $j(\tau)$ принимает значения, отличные от 0 и 1728.

В настоящем разделе приводится алгоритм явного определения эндоморфизма $\phi_\alpha \in \mathtt{End}(\mathcal{E}) : \ \mathcal{E}_{\Lambda_\tau}(\mathbb{C}) \to \mathcal{E}_{\Lambda_\tau}(\mathbb{C})$, соответствующего числу α. Изложение следует работе [175].

Согласно (1.39), соответствующий α эндоморфизм индуцируется отображением

$$\phi_\alpha : \ (\wp(z) : \wp'(z) : 1) \ \to \ (\wp(\alpha z) : \wp'(\alpha z) : 1),$$

в котором функция Вейерштрасса $\wp(z)$ определена равенством (1.36)

$$\wp(z) = \frac{1}{z^2} + \sum_{n=2}^{\infty} c_n z^{2n-2}, \quad z, c_n \in \mathbb{C},$$

а коэффициенты c_n, $n = 2, 3, \ldots$, удовлетворяют рекуррентному соотношению (1.37) и принадлежат полю $\mathbb{L} = \mathbb{Q}(j(\tau))$.

Определим функцию $\wp(\alpha z)$, тогда выполнено равенство

$$\wp(\alpha z) = \frac{1}{\alpha^2 z^2} + \sum_{n=2}^{\infty} c_n \alpha^{2n-2} z^{2n-2} = \sum_{n=0}^{\infty} d_{0,n} z^{2n-2} \tag{1.50}$$

в котором коэффициенты разложения $d_{0,n} \in \mathbb{H} = \mathbb{Q}(\sqrt{-d}, j(\tau))$ и определяются следующим образом:

$$\begin{aligned} d_{0,0} &= \alpha^{-2}, \\ d_{0,1} &= 0, \\ d_{0,n} &= c_n \alpha^{2n-2},\ n = 2, 3, \ldots . \end{aligned}$$

Поскольку эллиптические функции образуют поле, а функция $\wp(z)$ – четна, то $\wp(\alpha z)$ и $\wp'(\alpha z)$ как функции переменной z могут быть рационально выражены через $\wp(z)$ и $\wp'(z)$, см. [294, ч.2, гл.1, § 9], а также [64]. Следовательно, найдется такая рациональная функция $R(x) = \dfrac{P(x)}{Q(x)}$, где $P(x), Q(x) \in \mathbb{H}[x]$, что

$$\wp(\alpha z) = R(\wp(z)), \quad \wp'(\alpha z) = \frac{R'(\wp(z))\wp'(z)}{\alpha}. \tag{1.51}$$

Степень многочлена $P(x)$ определяется числом полюсов (без учета их кратности) функции $\wp(\alpha z)$ в основном параллелограмме периодов функции $\wp(z)$, см. [294, ч.2, гл.1]. Согласно [243], степени многочленов $P(x), Q(x)$, как функции от величины α, определяются равенствами[9]

$$\deg P(x) = N(\alpha), \quad \deg Q(x) = N(\alpha) - 1.$$

Таким образом, задача явного определения эндоморфизма ϕ_α сводится к построению рациональной функции $R(x) = \dfrac{P(x)}{Q(x)}$, удовлетворяющей равенствам (1.51).

Предположим, что многочлены $P(x), Q(x)$ известны. Тогда, используя алгоритм Эвклида в кольце $\mathbb{H}[x]$, можно определить последовательности многочленов $l_0(x), \ldots, l_m(x), r_0(x), \ldots, r_m(x) \in \mathbb{H}[x]$ такие, что

$$\begin{array}{rcll} P(x) & = & l_0(x)Q(x) + r_1(x), & \deg l_0(x) = 1, \\ Q(x) & = & l_1(x)r_1(x) + r_2(x), & \deg l_1(x) \geqslant 1, \\ & \cdots & & \\ r_{m-1}(x) & = & l_m(x)r_m(x), & \end{array}$$

а величину m – равенством

$$N(\alpha) = \deg l_0(x) + \cdots + \deg l_m(x).$$

При этом, если $\deg l_0(x) = \cdots = \deg l_m(x) = 1$, то $m = N(\alpha) - 1$.

Многочлены $l_0(x), \ldots, l_m(x)$ позволяют представить функцию $R(x)$ в виде непрерывной дроби

$$R(x) = \frac{P(x)}{Q(x)} = l_0(x) + \cfrac{1}{l_1(x) + \cfrac{1}{\cdots + \cfrac{1}{l_m(x)}}},$$

следовательно, для построения функции $R(x)$ достаточно определить последовательность многочленов $l_0(x), \ldots, l_m(x) \in \mathbb{H}[x]$. В основе предлагаемого далее способа определения указанной последовательности многочленов лежит метод обращения функции $\wp(z)$, см. [385, гл. VI, § 76].

[9] Напомним, что символом $N(\alpha)$ обозначается норма алгебраического числа $\alpha \in \Lambda_\tau \in \mathbb{K}$.

Для произвольного $k \in \mathbb{N}_0$ рассмотрим четную эллиптическую функцию $f_k(z)$, имеющую в нуле полюс второго порядка и определяемую рядом

$$f_k(z) = \sum_{n=0}^{\infty} d_{k,n} z^{2n-2} = \frac{d_{k,0}}{z^2} + d_{k,1} + d_{k,2} z^2 + \cdots, \quad d_{k,n} \in \mathbb{H}. \tag{1.52}$$

Примером такой функции могут служить $\wp(z)$ или $\wp(\alpha z)$.
Воспользовавшись (1.36) запишем равенство

$$\begin{aligned} f_k(z) = & \\ & = d_{k,0} \underbrace{\left(\frac{1}{z^2} + \sum_{n=2}^{\infty} c_n z^{2n-2} \right)}_{\wp(z)} + d_{k,1} + \sum_{n=2}^{\infty} \left(d_{k,n} - d_{k,0} c_n \right) z^{2n-2} = \\ & = l_k(\wp(z)) + f'_{k+1}(z), \end{aligned}$$

где

$$l_k(x) = d_{k,0} x + d_{k,1} \in \mathbb{H}[x] \quad \text{и} \quad \deg l_k(x) = 1. \tag{1.53}$$

Функция $f'_{k+1}(z) = f_k(z) - l_k(\wp(z))$ не имеет полюса в нуле. Если она отлична от нуля, то для нее может быть в явном виде определена функция $f_{k+1}(z)$ такая, что $f_{k+1}(z) f'_{k+1}(z) = 1$ и

$$f_k(z) = l_k(\wp(z)) + \frac{1}{f_{k+1}(z)}.$$

Для того, чтобы в явном виде вычислить функцию $f_{k+1}(z)$, запишем

$$\begin{aligned} f'_{k+1}(z) = & \sum_{n=2}^{\infty} \left(d_{k,n} - d_{k,0} c_n \right) z^{2n-2} = \\ & = \left(d_{k,2} - d_{k,0} c_2 \right) z^2 \left(1 + \frac{d_{k,3} - d_{k,0} c_3}{d_{k,2} - d_{k,0} c_2} z^2 + \frac{d_{k,4} - d_{k,0} c_4}{d_{k,2} - d_{k,0} c_2} z^4 + \cdots \right) = \\ & = e_{-1} z^2 \sum_{n=0}^{\infty} e_n z^{2n}, \end{aligned}$$

где $e_{-1}, e_0, e_1, \ldots \in \mathbb{H}$ и

$$\begin{aligned} e_{-1} &= d_{k,2} - d_{k,0} c_2, \\ e_n &= \frac{d_{k,n+2} - d_{k,0} c_{n+2}}{e_{-1}}, \quad n = 0, 1, \ldots \end{aligned} \tag{1.54}$$

Теперь, следуя [294, ч.1, гл.2], определим ряд

$$\sum_{n=0}^{\infty} u_n z^{2n},$$

в котором коэффициенты $u_n \in \mathbb{H}$ определяются рекуррентными соотношениями

$$\begin{array}{rcl} u_0 & = & 1, \\ u_1 & = & -e_1, \\ u_2 & = & -(e_1u_1 + e_2), \\ & \cdots & \\ u_n & = & -(e_1u_{n-1} + \cdots + e_{n-1}u_1 + e_n), \quad n = 3, 4, \ldots \end{array} \tag{1.55}$$

Теперь, учитывая равенство

$$\left(\sum_{n=0}^{\infty} e_n z^{2n}\right)\left(\sum_{n=0}^{\infty} u_n z^{2n}\right) = \\ = (1 + e_1z^2 + e_2z^4 + \cdots)(1 + u_1z^2 + u_2z^4 + \cdots) = \\ = 1 + (e_1 + u_1)z^2 + (e_2 + e_1u_1 + u_2)z^4 + \cdots = 1,$$

определим функцию $f_{k+1}(z)$ равенством

$$f_{k+1}(z) = \frac{1}{f'_{k+1}(z)} = \frac{1}{e_{-1}z^2\sum\limits_{n=0}^{\infty} e_n z^{2n}} = \frac{1}{e_{-1}z^2}\sum_{n=0}^{\infty} u_n z^{2n}.$$

Записывая $f_{k+1}(z)$ в виде

$$f_{k+1}(z) = \frac{1}{e_{-1}z^2} + \frac{u_1}{e_{-1}} + \frac{u_2}{e_{-1}}z^2 + \cdots = \sum_{n=0}^{\infty} d_{k+1,n} z^{2n-2},$$

где

$$d_{k+1,n} = \frac{u_n}{e_{-1}} \in \mathbb{H}, \tag{1.56}$$

мы получаем, что функция $f_{k+1}(z)$ также имеет вид (1.52).

Теперь, полагая $f_0(z) = \wp(\alpha z)$, мы можем записать равенства

$$\wp(\alpha z) = f_0(z) = l_0(\wp(z)) + \frac{1}{f_1(z)} = l_0(\wp(z)) + \frac{1}{l_1(\wp(z)) + \dfrac{1}{f_2(z)}} = \cdots \\ \cdots = l_0(\wp(z)) + \cfrac{1}{l_1(\wp(z)) + \cfrac{1}{\cdots + \cfrac{1}{l_{m-1}(\wp(z)) + \cfrac{1}{f_m(z)}}}}.$$

Поскольку представление $\wp(\alpha z)$ в виде рациональной функции от $\wp(z)$ единственно и $\deg l_k(x) = 1$ для всех $k \in \mathbb{N}_0$, то при $m = N(\alpha) - 1$ будет выполнено равенство $f_m(z) = l_m(\wp(z))$ и мы получим искомое разложение функции $\wp(\alpha z)$ в непрерывную дробь

$$\wp(\alpha z) = l_0(\wp(z)) + \cfrac{1}{l_1(\wp(z)) + \cfrac{1}{\cdots + \cfrac{1}{l_{m-1}(\wp(z)) + \cfrac{1}{l_m(\wp(z))}}}}.$$

Воспользовавшись рекуррентными соотношениями

$$\begin{aligned} &P_k(x) = l_k(x)P_{k-1}(x) + P_{k-2}(x), \\ &Q_k(x) = l_k(x)Q_{k-1}(x) + Q_{k-2}(x), \quad k = 1, 2, \ldots, m, \end{aligned} \tag{1.57}$$

при $P_{-1}(x) = 1, Q_{-1}(x) = 0$ и $P_0(x) = l_0(x), Q_0(x) = 1$, можно определить рациональную функцию

$$R(x) = \frac{P_m(x)}{Q_m(x)}, \quad P_m(x), Q_m(x) \in \mathbb{H}[x].$$

Для того, чтобы окончательно определить эндоморфизм ϕ_α, согласно (1.39) и (1.51), осталось вычислить рациональную функцию

$$\wp'(\alpha z) = \frac{R'(\wp(z))\wp'(z)}{\alpha}, \quad \text{где} \quad R'(x) = \frac{P'_m(x)Q_m(x) - P_m(x)Q'_m(x)}{Q^2_m(x)}.$$

Прежде, чем суммировать сказанное в виде единого алгоритма, сделаем несколько замечаний. В начале необходимо определить минимальные многочлены для целых алгебраических чисел τ и $j(\tau)$ и, с их помощью, построить поле $\mathbb{H} = \mathbb{Q}(\sqrt{-d}, j(\tau))$. Это позволит проводить вычисления с алгебраическими числами, а не с комплексными и, тем самым, реализовывать арифметические операции без потери точности.

Поскольку мы не можем проводить вычисления на ЭВМ с бесконечными рядами, то при разложении в цепную дробь используются частичные суммы некоторой длины $r \in \mathbb{N}$, т.е. мы считаем, что

$$f_k(z) = \sum_{n=0}^{r-1} d_{k,n} z^{2n-2}, \quad k = 0, 1, \ldots, m. \tag{1.58}$$

Выбор величины r основан на следующих рассуждениях. При вычислении многочлена $l_k(x)$ нам требуется знать только два старших коэффициента $d_{k,0}, d_{k,1}$ частичной суммы $f_k(z)$, задаваемой равенством (1.58).

Оставшиеся $r-2$ коэффициента используются при обращении и вычислении коэффициентов частичной суммы $f_{k+1}(z)$.

С другой стороны, из равенств (1.55) и (1.56) следует, что для частичной суммы $f_{k+1}(z)$ можно определить лишь на 2 коэффициента меньше, чем у частичной суммы $f_k(x)$, и с каждым новым значением k количество значащих знаков в последовательности $d_{k,0},\dots,d_{k,r-1}$ уменьшается на два, т.е. $f_0(z)$ задается r коэффициентами, $f_1(z)$ – $r-2$ коэффициентами, $f_2(z)$ – $r-4$ коэффициентами и т.д.

Алгоритм 1.6: Алгоритм построения эндоморфизма ϕ_α

Вход : Свободное от квадратов число $d\in\mathbb{N}$, мнимая квадратичная иррациональность $\tau\in\mathbb{D}\subset\mathbb{C}_+$, решетка $\Lambda_\tau=\{n+m\tau,\ n,m\in\mathbb{Z}\}\subseteq\mathbb{Z}_{\mathbb{K}}\subset\mathbb{K}=\mathbb{Q}(\sqrt{-d})$, коэффициенты $g_2(\Lambda_\tau)$ и $g_2(\Lambda_\tau)$ эллиптической кривой $\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$ такой, что $\mathtt{End}(\mathcal{E})\sim\Lambda_\tau$, а также отличный от нуля элемент $\alpha\in\Lambda_\tau$.

Выход : Эндоморфизм $\phi_\alpha:\ \mathcal{E}_{\Lambda_\tau}(\mathbb{C})\to\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$.

1 Определить минимальные многочлены для алгебраических чисел τ и $j(\tau)$.

2 Определить поле $\mathbb{H}=\mathbb{Q}(\sqrt{-d},j(\tau))$.

3 Определить $m=N(\alpha)-1$ и $r=2N(\alpha)+4$.

4 Используя равенства (1.37) определить $c_2,c_3,\dots c_r\in\mathbb{H}$ – коэффициенты разложения функции $\wp(z)$ в ряд Лорана.

5 Используя равенства (1.50) определить $d_{0,0},d_{0,1},\dots,d_{0,r}\in\mathbb{H}$ – коэффициенты разложения функции $\wp(\alpha z)$ в ряд Лорана.

6 Согласно (1.53) и (1.57) определить многочлены $P_{-1}(x)=1$, $Q_{-1}(x)=0$ и $P_0(x)=d_{0,0}x+d_{0,1}$, $Q_0(x)=1$.

7 **Для всех** $k=1,2,\dots,m$ **выполнять**

8 Используя (1.54) определить коэффициенты $e_{-1},e_0,e_1,\dots,e_r\in\mathbb{H}$.

9 Используя (1.55) определить коэффициенты $u_0,u_1,\dots,u_r\in\mathbb{H}$.

10 Определить коэффициенты $d_{k,n}=\dfrac{u_n}{e_{-1}}\in\mathbb{H}$ для всех $n=0,1,\dots,r$.

11 Согласно (1.53) определить многочлен $l_k(x)=d_{k,0}x+d_{k,1}\in\mathbb{H}[x]$.

12 Согласно (1.57) определить

$$P_k(x)=l_k(x)P_{k-1}(x)+P_{k-2}(x),\quad Q_k(x)=l_k(x)Q_{k-1}(x)+Q_{k-2}(x).$$

13 **конец**

14 Определить эндоморфизм ϕ_α следующим образом

$$\phi_\alpha:(x:y:1)\to\left(\frac{P_m(x)}{Q_m(x)}:\frac{\big(P'_m(x)Q_m(x)-P_m(x)Q'_m(x)\big)y}{\alpha Q_m^2(x)}:1\right).$$

Поскольку необходимо определить $N(\alpha)$ многочленов $l_k(x)$, то требуется не менее $2N(\alpha)$ коэффициентов разложения $\wp(\alpha z)$ в ряд Лорана. Согласно третьей строке, в алгоритме 1.6 используется значение

$$r=2N(\alpha)+4,$$

что не влияет существенно на трудоемкость алгоритма, однако позволяет контролировать корректность проводимых вычислений.

Проверка того, что рациональная функция $R(x) = \dfrac{P_m(x)}{Q_m(x)}$ вычислена правильно выполняется с помощью равенства

$$4\left(\frac{P_m(x)}{Q_m(x)}\right)^3 - g_2(\Lambda_\tau)\frac{P_m(x)}{Q_m(x)} - g_3(\Lambda_\tau) = \\ = \frac{\left(4x^3 - g_2(\Lambda_\tau)x - g3(\Lambda_\tau)\right)\left(P'_m(x)Q_m(x) - P_m(x)Q'_m(x)\right)^2}{\alpha^2 Q_m^4(x)},$$

где $g_2(\Lambda_\tau)$, $g_3(\Lambda_\tau)$ коэффициенты эллиптической кривой $\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$.

Легко видеть, что трудоемкость предложенного алгоритма оценивается величиной $O\left(N(\alpha)^2\right)$ элементарных операций сложения, умножения и обращения элементов поля $\mathbb{H}$.

Несмотря на малую оценку числа шагов алгоритма 1.6, время его практической реализации может быть весьма велико. Это связано с тем, что каждый элемент поля $\mathbb{H}$ представляется в виде вектора длины $2h$, где h степень минимального многочлена для $j(\tau)$. Кроме того, числители и знаменатели рациональных чисел, образующих коэффициенты указанного вектора, растут весьма быстро с ростом величины d, см. тексты программ из приложения А.

Для иллюстрации работы алгоритма 1.6 рассмотрим поле $\mathbb{K} = \mathbb{Q}(\sqrt{-1})$ и выберем в качестве $\tau = 2\sqrt{-1}$. Тогда

$$j(\tau) = 287496, \quad j(\tau) \in \mathbb{Z},$$

и поле $\mathbb{H}$ совпадает с полем $\mathbb{K}$, степень расширения $[\mathbb{H} : \mathbb{Q}] = 2$. Рассмотрим кривую

$$\mathcal{E}_{\Lambda_\tau}(\mathbb{K}) : \quad y^2 = 4x^3 - 11x - 7,$$

и построим эндоморфизм ϕ_α этой кривой для $\alpha = \tau = 2\sqrt{-1}$ и $N(\alpha) = 4$.

Определим $r = 12$ и, воспользовавшись равенствами (1.37), вычислим

$$\wp(z) = \frac{1}{z^2} + \frac{11}{20}z^2 + \frac{1}{4}z^4 + \frac{121}{1200}z^6 + \frac{3}{80}z^8 + \frac{2081}{156000}z^{10} + \\ + \frac{11}{2400}z^{12} + \frac{32641}{21216000}z^{14} + \frac{211}{416000}z^{16} + \frac{524801}{3182400000}z^{18} + \frac{11249}{212160000}z^{20}.$$

Тогда, согласно равенствам (1.50), выполнено

$$f_0(z) = \wp(2\sqrt{-1}z) = \\ = -\frac{1}{4z^2} - \frac{11}{5}z^2 + 4z^4 - \frac{484}{75}z^6 + \frac{48}{5}z^8 - \frac{66592}{4875}z^{10} + \frac{1408}{75}z^{12} - \\ - \frac{2089024}{82875}z^{14} + \frac{54016}{1625}z^{16} - \frac{268698112}{6215625}z^{18} + \frac{23037952}{414375}z^{20}$$

и

$$l_0(x) = -\frac{1}{4}x.$$

Имеем

$$f_1'(z) = f_0(z) - l_0(\wp(z)) = \\ -\frac{33}{16}z^2\left(1 - \frac{65}{33}z^2 + \frac{187}{60}z^4 - \frac{205}{44}z^6 + \frac{14567}{2200}z^8 - \right. \\ \left. -\frac{3277}{360}z^{10} + \frac{8388737}{686400}z^{12} - \frac{850963}{52800}z^{14} + \frac{667022071}{31824000}z^{16} - \frac{9436347389}{350064000}z^{18}\right),$$

тогда, используя (1.55) и (1.56) для обращения стоящей в скобках суммы, получим

$$f_0(z) = l_0(\wp(z)) + \frac{1}{f_1(z)},$$

где

$$f_1(z) = \\ -\frac{16}{33z^2} - \frac{1040}{1089} - \frac{66476}{179685}z^2 - \frac{13316}{1185921}z^4 - \frac{105835987}{978384825}z^6 + \frac{53072957}{6457339845}z^8 - \\ -\frac{2360507941003}{138509939675250}z^{10} + \frac{56219150567}{70320430912050}z^{12} - \\ -\frac{17020809926916907}{20513876105663226000}z^{14} - \frac{36087106902840211}{39821053616875674000}z^{16}$$

и

$$l_1(x) = -\frac{16}{33}x - \frac{1040}{1089} = -\frac{16}{33}\left(x + \frac{65}{33}\right).$$

Аналогично,

$$f_2(z) = -\frac{35937}{3712z^2} - \frac{1109691}{107648} - \frac{42297849}{7804480}z^2 - \frac{843261705}{362127872}z^4 - \\ -\frac{265824060381}{262542707200}z^6 - \frac{216041708223}{609099080704}z^8 - \\ -\frac{7497368010716067}{57407588356352000}z^{10} - \frac{905182450965837}{20490093074882560}z^{12}$$

и

$$l_2(x) = -\frac{35937}{3712}x - \frac{1109691}{107648} = -\frac{1089}{3712}\left(33x + \frac{1019}{3712}\right).$$

И последнее,

$$f_3(z) = -\frac{12487168}{1185921z^2} - \frac{12056576}{1185921} - \frac{3121792}{539055}z^2 - \\ -\frac{3121792}{1185921}z^4 - \frac{780448}{735075}z^6 - \frac{780448}{1976535}z^8$$

и

$$l_3(x) = -\frac{12487168}{1185921}x - \frac{12056576}{1185921} = -\frac{430592}{1185921}(29x + 28).$$

Используя (1.57) и построенные многочлены $l_0(x), \ldots, l_3(x)$, определим последовательность подходящих дробей

$$\begin{aligned}
\tfrac{P_0(x)}{Q_0(x)} &= -\tfrac{1}{4}x, \\
\tfrac{P_1(x)}{Q_1(x)} &= \frac{-\frac{1}{4}x^2 - \frac{65}{132}x - \frac{33}{16}}{x + \frac{65}{33}}, \\
\tfrac{P_2(x)}{Q_2(x)} &= \frac{-\frac{1}{4}x^3 - \frac{22}{29}x^2 - \frac{1225}{464}x - \frac{1019}{464}}{x^2 + \frac{88}{29}x + \frac{67}{29}}, \\
R(x) = \tfrac{P_3(x)}{Q_3(x)} &= \frac{-\frac{1}{4}x^4 - x^3 - \frac{27}{8}x^2 - \frac{19}{4}x - \frac{137}{64}}{x^3 + 4x^2 + \frac{21}{4}x + \frac{9}{4}}.
\end{aligned}$$

Для завершения алгоритма осталось вычислить производную

$$R'(x) = \frac{-\frac{1}{4}x^5 - \frac{13}{8}x^4 - \frac{17}{8}x^3 - \frac{1}{16}x^2 + \frac{67}{64}x + \frac{47}{128}}{x^5 + \frac{13}{2}x^4 + \frac{67}{4}x^3 + \frac{171}{8}x^2 + \frac{27}{2}x + \frac{27}{8}}.$$

§ 1.4.3. Результаты практических вычислений

Согласно [266, гл. 13, § 2, стр. 438] существует только 11 решеток $\Lambda_\tau \subseteq Z_{\mathbb{K}} \subset \mathbb{Q}(\sqrt{-d})$ с числом классов эквивалентных квадратичных форм, равным единице, и значением $j(\tau)$ отличным от 0 и 1728. Каждой такой решетке соответствует одна, с точностью до изоморфизма, эллиптическая кривая $\mathcal{E}(\Lambda_\tau)$ с целым значением инварианта. Перечень таких кривых приведен в таблице 1.4, см. также [374].

№	τ	$N(\tau)$	$j(\tau)$	$E_{\Lambda_\tau}(\mathbb{C})$
1	$2\sqrt{-1}$	4	287496	$y^2 = 4x^3 - 11x - 7$
2	$\sqrt{-2}$	2	8000	$y^2 = 4x^3 - 30x - 28$
3	$\sqrt{-3}$	3	54000	$y^2 = 4x^3 - 15x - 11$
4	$\frac{3}{2}(1+\sqrt{-3})$	9	−12288000	$y^2 = 4x^3 - 120x - 253$
5	$\frac{1}{2}(1+\sqrt{-7})$	2	−3375	$y^2 = 4x^3 - 35x - 49$
6	$\sqrt{-7}$	7	16581375	$y^2 = 4x^3 - 595x - 2793$
7	$\frac{1}{2}(1+\sqrt{-11})$	3	−32768	$y^2 = 4x^3 - 264x - 847$
8	$\frac{1}{2}(1+\sqrt{-19})$	5	−884736	$y^2 = 4x^3 - 152x - 361$
9	$\frac{1}{2}(1+\sqrt{-43})$	11	−884736000	$y^2 = 4x^3 - 3440x - 38829$
10	$\frac{1}{2}(1+\sqrt{-67})$	17	−147197952000	$y^2 = 4x^3 - 29480x - 974113$
11	$\frac{1}{2}(1+\sqrt{-163})$	41	−262537412640768000	$y^2 = 4x^3 - 8697680x - 4936546769$

Таблица 1.4: Эллиптические кривые с целым j-инвариантом.

Предложенный ранее алгоритм 1.6 был реализован автором на ЭВМ, что позволило определить явный вид эндоморфизмов ϕ_α для всех эллип-

тических кривых, указанных в таблице 1.4. Результаты вычислений приводятся[10] ниже.

Отметим, что для эллиптических кривых с номерами 2 и 5, явный вид эндоморфизмов ϕ_α, соответствующих $\alpha = \tau$, приводился ранее в работе [95].

№ 1. Кривая $y^2 = 4x^3 - 11x - 7$, $\alpha = 2\sqrt{-1}$ и $N(\alpha) = 4$:

$$\phi_\alpha : (x, y) \to \left(-\frac{16x^4 + 64x^3 + 216x^2 + 304x + 137}{16(x+1)(2x+3)^2}, \right.$$
$$\left. \frac{-32x^5 - 208x^4 - 272x^3 - 8x^2 + 134x + 47}{16\alpha(x+1)^2(2x+3)^3} y \right).$$

№ 2. Кривая $y^2 = 4x^3 - 30x - 28$, $\alpha = \sqrt{-2}$ и $N(\alpha) = 2$:

$$\phi_\alpha : (x, y) \to \left(-\frac{2x^2 + 4x + 9}{4(x+2)}, \ \frac{-2x^2 - 8x + 1}{4\alpha(x+2)^2} y \right).$$

№ 3. Кривая $y^2 = 4x^3 - 15x - 11$, $\alpha = \sqrt{-3}$ и $N(\alpha) = 3$:

$$\phi_\alpha : (x, y) \to \left(-\frac{4x^3 + 12x^2 + 33x + 28}{3(2x+3)^2}, \ \frac{-8x^3 - 36x^2 - 6x + 13}{3\alpha(2x+3)^3} y \right).$$

№ 4. Кривая $y^2 = 4x^3 - 120x - 253$, $\alpha = \frac{3}{2}(1 + \sqrt{-3})$ и $N(\alpha) = 9$:

$$\phi_\alpha : (x, y) \to$$
$$\left(\frac{1}{27f^2(x)} \times \Big(-\alpha x^9 - 18(\alpha+3)x^8 + 9(50\alpha - 297)x^7 + 6(2021\alpha - 6210)x^6 + \right.$$
$$+ 9(12538\alpha - 26529)x^5 + 9(61340\alpha - 85743)x^4 + 3(519281\alpha - 349893)x^3 +$$
$$\left. + 36(70243\alpha + 13647)x^2 + 9(239966\alpha + 335061)x + 728569\alpha + 2442393 \Big), \right.$$
$$\frac{-y}{27f^3(x)} \times (x^3 + 9x^2 + 33x + 47) \times \Big(\alpha x^9 + 9(9 + 2\alpha)x^8 +$$
$$+ 9(81 + 40\alpha)x^7 + (3912\alpha - 3159)x^6 + 9(1988\alpha - 7227)x^5 +$$
$$+ 18(470\alpha - 17523)x^4 + 3(200151 + 77831\alpha)x^3 - 18(1575 + 51791\alpha)x^2 -$$
$$\left. - 9(166580\alpha - 156789)x + 1385100 - 912277\alpha \Big) \right),$$

где $f(x) = (x+3)\left(x^3 + (15 - 3\alpha)x^2 + (57 - 18\alpha)x + (62 - 27\alpha)\right)$.

[10] Во всех случаях коэффициенты построенных рациональных функций принадлежат $\mathbb{Z}[\alpha]$.

№ 5. Кривая $y^2 = 4x^3 - 35x - 49$, $\alpha = \frac{1}{2}(1 + \sqrt{-7})$ и $N(\alpha) = 2$:

$$\phi_\alpha : (x, y) \to \Big(\frac{-4(\alpha+1)x^2 - 4(\alpha+3)x + 7(5\alpha-7)}{8(2x-\alpha+4)}, \\ -\frac{4(\alpha+1)x^2 + 8(\alpha+3)x + 7(5\alpha-3)}{4\alpha(2x-\alpha+4)^2} y \Big).$$

№ 6. Кривая $y^2 = 4x^3 - 595x - 2793$, $\alpha = \sqrt{-7}$ и $N(\alpha) = 7$:

$$\phi_\alpha : (x, y) \to \\ \Big(-\frac{1}{7f^2(x)} \times \Big(64x^7 + 4032x^6 + 193648x^5 + 4900000x^4 + 65275644x^3 + \\ + 472046204x^2 + 1765121561x + 2683223144 \Big), \\ \frac{y}{7\alpha f^3(x)} \times \Big(-512x^9 - 48384x^8 - 1257984x^7 - 12757248x^6 - 18411456x^5 + \\ 695999136x^4 + 6221339488x^3 + 22527872304x^2 + 34125504238x + 11533585259 \Big),$$

где $f(x) = 8x^3 + 252x^2 + 2422x + 7357$.

№ 7. Кривая: $y^2 = 4x^3 - 264x - 847$, $\alpha = \frac{1}{2}(1 + \sqrt{-11})$ и $N(\alpha) = 3$:

$$\phi_\alpha : (x, y) \to \Big(-\frac{(\alpha+2)x^3 + 6(\alpha+5)x^2 - 33(4\alpha-13)x - 11(59\alpha-134)}{9(x-\alpha+6)^2}, \\ -\frac{(\alpha+2)x^3 + 9(\alpha+5)x^2 + 33(4\alpha-1)x + 11(19\alpha-70)}{9\alpha(x-\alpha+6)^3} y \Big).$$

№ 8. Кривая: $y^2 = 4x^3 - 152x - 361$, $\alpha = \frac{1}{2}(1 + \sqrt{-19})$ и $N(\alpha) = 5$:

$$\phi_\alpha : (x, y) \to \Big(\frac{1}{f^2(x)} \times \big(-(\alpha+4)x^5 - 10(\alpha+9)x^4 + 95(2\alpha-17)x^3 + \\ + 380(7\alpha-32)x^2 + 5415(2\alpha-7)x + 1805(8\alpha-23) \big), \\ \frac{-5y}{\alpha f^3(x)} \times \big((\alpha+4)x^6 + 15(\alpha+9)x^5 + 209(\alpha+4)x^4 + \\ + 19(73\alpha-8)x^3 + 1083(3\alpha-13)x^2 - 361(\alpha+104)x - 6859(\alpha+4) \big) \Big),$$

где $f(x) = 5x^2 + 5(10-\alpha)x + (114 - 19\alpha)$.

№ 9. Кривая: $y^2 = 4x^3 - 3440x - 38829$, $\alpha = \frac{1}{2}(1 + \sqrt{-43})$ и $N(\alpha) = 11$:

$$\phi_\alpha : (x, y) \to$$
$$\Big(\frac{1}{f^2(x)} \times \big(-(\alpha+10)x^{11} - 132(\alpha+21)x^{10} + 946(20\alpha - 581)x^9 +$$
$$+ 473(10717\alpha - 126382)x^8 + 101695(4505\alpha - 37219)x^7 + 6996616(3287\alpha - 21635)x^6 +$$
$$+ 874577(828781\alpha - 4587030)x^5 + 874577(17123327\alpha - 82078613)x^4 +$$
$$+ 37606811(5410144\alpha - 22883347)x^3 + 75213622(23349061\alpha - 88276902)x^2 +$$
$$+ 1617092873(5412111\alpha - 18461213)x + 1617092873(11878584\alpha - 36810755)\big),$$
$$\frac{-y}{\alpha f^3(x)} \times \big((\alpha+10)x^{15} + 198(\alpha+21)x^{14} + 43(560\alpha + 12497)x^{13} +$$
$$+ 129(16759\alpha + 266326)x^{12} + 22188(5991\alpha + 53981)x^{11} + 5547(957664\alpha + 3343567)x^{10} +$$
$$+ 636056(204783\alpha - 361643)x^9 + 238521(6346253\alpha - 79067847)x^8 -$$
$$- 10256403(1319376\alpha + 45425861)x^7 - 27350408(32732861\alpha + 234409806)x^6 -$$
$$- 441025329(40240641\alpha + 107283901)x^5 - 294016886(695797144\alpha + 172857357)x^4 -$$
$$- 151712713176(9706277\alpha - 16452654)x^3 - 37928178294(167920463\alpha - 641046237)x^2 -$$
$$- 271818611107(52619384\alpha - 372703907)x - 271818611107(39005407\alpha - 616133530)\big)\Big),$$

где

$$f(x) = 11x^5 - 66(\alpha - 22)x^4 - 473(11\alpha - 145)x^3 -$$
$$- 473(320\alpha - 3213)x^2 - 20339(95\alpha - 794)x - 1849(4951\alpha - 36037).$$

Точные значения эндоморфизмов для остальных эллиптических кривых из таблицы 1.4 приведены в приложении А. Там же приведен текст программы, проверяющей корректность полученных результатов.

Для иллюстрации работы алгоритма 1.6 в случае, когда $[\mathbb{H} : \mathbb{Q}] > 2$, рассмотрим пример из работы [175].

Пусть $\tau = \sqrt{-5}$, тогда $j(\tau) = 320(1975 + 884\sqrt{5})$ является корнем неприводимого над полем рациональных чисел многочлена

$$x^2 - 1264000x - 681472000.$$

Поле $\mathbb{H}$, над которым определяются коэффициенты эллиптической кривой, задается равенством

$$\mathbb{H} = \mathbb{Q}\left(\sqrt{-5}, j(\tau)\right) = \mathbb{Q}(\xi), \quad [\mathbb{H} : \mathbb{Q}] = 4,$$

где $\xi = \tau + j(\tau) = \sqrt{-5} + 320\left(884\sqrt{5} + 1975\right)$ есть корень неприводимого над полем рациональных чисел многочлена 4-й степени

$$x^4 - 2528000x^3 + 1596333056010x^2 +$$
$$+ 1722761203360000x + 4644120820787200025 \in \mathbb{Z}[x].$$

Эллиптическая кривая $\mathcal{E}_{\Lambda_\tau}(\mathbb{H})$ такая, что $\mathtt{End}(\mathcal{E}) \sim \Lambda_\tau$, определяется равенством

$$y^2 = 4x^3 - (3\lambda + 31944000)x - 6688(10648000 + \lambda),$$

где $\lambda = 27j(\tau)$. Представляя коэффициенты данной кривой как элементы поля $\mathbb{H}$ в виде $a_3\xi^3 + a_2\xi^2 + a_1\xi + a_0$, где $a_0, a_1, a_2, a_3 \in \mathbb{Q}$, можно записать равенства

$$\begin{aligned} g_2 &= \frac{-81\xi^3 + 153576000\xi^2 + 165597695595\xi + 25527052481599320000}{800210944010} \\ g_3 &= \frac{-90288\xi^3 + 171186048000\xi^2 + 1845862313565 60\xi + 28454154499489375360000}{400105472005}. \end{aligned}$$

Тогда, эндоморфизм ϕ_α, соответствующий $\alpha = \sqrt{-5}$, задается отображением (1.44)

$$\phi_\alpha(x, y) \to \left(R(x), \frac{R'(x)y}{\alpha} \right),$$

где

$$\begin{aligned} R(x) = \Big(&-\frac{1}{5}x^5 + \frac{9\xi^3 - 17064000\xi^2 - 18399743955\xi - 1737434015015496640}{2829545898019360}x^4 + \\ &+ \frac{18477\xi^3 - 35032392000\xi^2 - 37774674339615\xi - 2213885820761140804800}{353693237252420}x^3 + \\ &+ \frac{2433582\xi^3 - 4614071472000\xi^2 - 4975253965944090\xi + 225567145721735518640000}{5201371136065}x^2 + \\ &+ \frac{1}{88423309313105}(129371544864\xi^3 - 2452884490621440 00\xi^2 - \\ &- 26448925561782169968 0\xi + 1127978108160531510541184 0000)x + \\ &+ \frac{1}{17684661862621}(29818774923264\xi^3 - 5653639725450854 4000\xi^2 - \\ &- 60961980404425819207680\xi + 3109730909814757318049906688000)\Big) \times \\ &\times \Big(x^4 + \frac{-9\xi^3 + 17064000\xi^2 + 18399743955\xi + 1737434015015496640}{565909179603872}x^3 + \\ &+ \frac{-247113\xi^3 + 468526248000\xi^2 + 505201769772435\xi - 12350597662172876520000}{1768466186262100}x^2 + \\ &+ \frac{1}{6801793024085}(-2839878\xi^3 + 5384408688000\xi^2 + \\ &+ 5805892007048610\xi - 9752345047403332952 0000)x + \\ &+ \frac{1}{17684661862621}(-7545046176\xi^3 + 1430540754969600 0\xi^2 + \\ &+ 1542521308522798512 0\xi + 3857145182244940838346035 20)\Big)^{-1}. \end{aligned}$$

§ 1.4.4. Выбор формы, минимизирующей трудоемкость вычислений

Для некоторых эллиптических кривых, указанных в таблице 1.4, можно предъявить форму, в которой вычисление построенных эндоморфизмов ϕ_α потребует наименьшего количества элементарных операций сложения, умножения и т.д. элементов поля $\mathbb{H}$.

Докажем следующую теорему.

Теорема 1.5. *Пусть $\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$ эллиптическая кривая, заданная равенством*

$$y^2 = f(x), \quad \text{где} \quad f(x) = 4x^3 - g_2 x - g_3.$$

Рассмотрим произвольные комплексные числа $\theta \in \mathbb{C}$ и $\gamma \in \mathbb{C}^*$. Тогда, отображение

$$\begin{array}{rccl} & \mathcal{E}_{\Lambda_\tau}(\mathbb{C}) & \to & \mathcal{H}(\mathbb{C}) \\ \psi_\theta : & (0:1:0) = \mathcal{O} & \to & \mathcal{O} = (0:0:1), \\ & (x:y:1) & \to & \left(u = (x-\theta)\gamma : v = \frac{1}{2}\gamma^{\frac{3}{2}} y : 1\right), \end{array} \tag{1.59}$$

является изоморфизмом кривых, и $\mathcal{H}(\mathbb{C})$ – эллиптическая кривая, определяемая равенством

$$v^2 = u^3 + a_2\gamma u^2 + a_4\gamma^2 u + a_6\gamma^3, \tag{1.60}$$

в котором

$$a_2 = 3\theta, \quad a_4 = 3\theta^2 - \frac{g_2}{4}, \quad a_6 = \theta^3 - \frac{g_2\theta}{4} - \frac{g_3}{4}. \tag{1.61}$$

Утверждение данной теоремы вытекает из соотношений, полученных Дж. Тейтом, см. [314, Прил. 1]. Тем не менее, мы приведем доказательство теоремы 1.5 и получим равенства, которые будут использованы в дальнейшем для определения эндоморфизмов на эллиптической кривой $\mathcal{H}(\mathbb{C})$.

Доказательство теоремы 1.5. Пусть $(x : y : 1)$ произвольная, отличная от $\mathcal{O}$ точка эллиптической кривой $\mathcal{E}_{\Lambda_\tau}(\mathcal{C})$. Воспользуемся (1.59) и запишем равенства

$$x = \frac{u}{\gamma} + \theta, \quad y = \frac{2v}{\gamma^{\frac{3}{2}}}.$$

Подставляя полученные равенства в уравнение кривой, получим

$$\frac{4v^2}{\gamma^3} = y^2 = 4x^3 - g_2x - g_3 =$$
$$= 4\left(\frac{u}{\gamma} + \theta\right)^3 - g_2\left(\frac{u}{\gamma} + \theta\right) - g_3 =$$
$$\frac{4}{\gamma^3}\left(u^3 + 3\theta\gamma u^2 - \frac{g_2 - 12\theta^2}{4}\gamma^2 u + \gamma^3\left(\theta^3 - \frac{g_2\theta}{4} - \frac{g_3}{4}\right)\right).$$

Замечая, что последнее слагаемое обращается в нуль, сокращая на общий, отличный от нуля множитель $\frac{4}{\gamma^3}$ и обозначая

$$a_2 = 3\theta, \quad a_4 = 3\theta^2 - \frac{g_2}{4}, \quad a_6 = \frac{f(\theta)}{4}$$

получим равенство (1.60). Теперь, используя полученные выше равенства, определим обратное отображение

$$\psi_\theta^{-1}: \begin{array}{rcl} \mathcal{H}_{A,B}(\mathbb{C}) & \to & \mathcal{E}_{\Lambda_\tau}(\mathbb{C}) \\ (0:0:1) = \mathcal{O} & \to & \mathcal{O} = (0:1:0), \\ (u:v:1) & \to & \left(x = \dfrac{u}{\gamma} + \theta : y = 2\gamma^{-\frac{3}{2}}v : 1\right) \end{array} \tag{1.62}$$

и запишем

$$u = (x - \theta)\gamma, \quad v = \frac{1}{2}\gamma^{\frac{3}{2}}y.$$

Подставляя полученные равенства в уравнение кривой (1.60), получим

$$\frac{y^2\gamma^3}{4} = (x-\theta)^3\gamma^3 + 3\theta\gamma(x-\theta)^2\gamma^2 +$$
$$+ \left(3\theta^2 - \frac{g_2}{4}\right)\gamma(x-\theta)\gamma^2 + \left(\theta^3 - \frac{g_2\theta}{4} - \frac{g_3}{4}\right)\gamma^3 =$$
$$= \gamma^3\left(x^3 - \frac{g_2}{4}x - \theta\left(3\theta^2 - \frac{g_2}{4}\right) - \frac{g_2\theta}{4} - \frac{g_3}{4} + 3\theta^3\right) =$$
$$= \frac{1}{4}\gamma^3\big(4x^3 - g_2x - g_3\big).$$

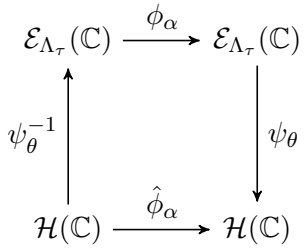
Сокращая на отличный от нуля множитель $\frac{1}{4}\gamma^3$ получим исходное уравнение эллиптической кривой $\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$.

Тот факт, что отображение $\psi_\theta^{-1} \cdot \psi_\theta(x, y)$ определено для каждой точки кривой $\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$ и оставляет эту точку на месте, завершает доказательство теоремы. □

Утверждение теоремы 1.5 позволяет определить отображение эллиптической кривой $\mathcal{H}(\mathbb{C})$ в себя

$$\hat{\phi}_\alpha(u, v) = \psi_\theta \cdot \phi_\alpha \cdot \psi_\theta^{-1}(u, v).$$

Схематично, отображение $\hat{\phi}_\alpha$ может быть изображено следующим образом



и записано в явном виде следующим образом

$$\hat{\phi}_\alpha : (u, v) \to \left(\gamma R\left(\frac{u}{\gamma} + \theta\right) - \gamma\theta,\ \frac{v}{\alpha} R'\left(\frac{u}{\gamma} + \theta\right)\right). \tag{1.63}$$

Выпишем в таблице 1.5 некоторые эллиптические кривые в форме $\mathcal{H}(\mathbb{C})$, соответствующие приведенным в таблице 1.4 эллиптическим кривым в короткой форме Вейерштрасса. При этом, параметр $\theta \in \mathbb{H}$ будет подобран таким образом, чтобы упростить знаменатели определяющих $\hat{\phi}_\alpha$ рациональных функций, а параметр $\gamma \in \mathbb{H}$ — таким образом, чтобы минимизировать их коэффициенты.

Заметим, что условие $\theta, \gamma \in \mathbb{H}$ позволяет говорить о том, что эндоморфизм $\hat{\phi}_\alpha$ не выводит за пределы поля $\mathbb{H}$ и может быть успешно редуцирован в конечное поле $\mathbb{F}_p$.

№	τ	θ	γ	$\mathcal{H}(\mathbb{C})$
1	$2\sqrt{-1}$	-1	2	$v^2 = u^3 - 6u^2 + u$
2	$\sqrt{-2}$	-2	$\frac{2}{3}$	$v^2 = u^3 - 4u^2 + 2u$
3	$\sqrt{-3}$	-1	2	$v^2 = u^3 - 6u^2 - 3u$
4	$\frac{3}{2}\left(1 + \sqrt{-3}\right)$	-3	1	$v^2 = u^3 - 9u^2 - 3u - \frac{1}{4}$
5	$\frac{1}{2}\left(1 + \sqrt{-7}\right)$	$\frac{\alpha-4}{2}$	$-\frac{(\alpha+10)}{112}$	$v^2 = u^3 - \frac{3}{32}(\alpha - 6)u^2 - \frac{1}{64}(3\alpha - 2)u$
5	$\frac{1}{2}\left(1 + \sqrt{-7}\right)$	$-\frac{1}{2}$	$\frac{1}{2}$	$v^2 = u^3 - \frac{3}{4}u^2 - 2u - 1$

Таблица 1.5: Эллиптические кривые с целым j-инвариантом.

Некоторые их эллиптических кривых в такой форме были впервые опубликованы, по-видимому, в работе Т. Хадано [101]. Используя (1.63) и, при необходимости, равенство

$$u^2 + a_2 u + a_4 = \frac{v^2 - a_6}{u}$$

можно записать следующее.

№ 1. Кривая $v^2 = u^3 - 6u^2 + u$, $\alpha = 2\sqrt{-1}$ и $N(\alpha) = 4$,

$$\hat{\phi}_\alpha : (u, v) \to \left(-\frac{(u-1)^2 v^2}{4u^2(u+1)^2}, -\frac{(u^5 + 3u^4 - 30u^3 + 30u^2 - 3u - 1)}{4\alpha u^2 (u+1)^3} v \right).$$

№ 2. Кривая $v^2 = u^3 - 4u^2 + 2u$, $\alpha = \sqrt{-2}$ и $N(\alpha) = 2$,

$$\hat{\phi}_\alpha : (u, v) \to \left(\frac{-v^2}{2u^2}, \ \frac{2 - u^2}{2\alpha u^2} v \right).$$

Отметим, что в работах [8] и [374, 375] эндоморфизм $\hat{\phi}_\alpha$ использовался именно в таком виде.

№ 3. Кривая $v^2 = u^3 - 6u^2 - 3u$, $\alpha = \sqrt{-3}$ и $N(\alpha) = 3$:

$$\hat{\phi}_\alpha : (x, y) \to \left(-\frac{u(u-3)^2}{3(u+1)^2}, \ -\frac{(u^3 + 3u^2 - 21u + 9)\, v}{3\alpha(u+1)^3} \right).$$

№ 4. Кривая $v^2 = u^3 - 9u^2 - 3u - \frac{1}{4}$, $\alpha = \frac{3}{2}(1 + \sqrt{-3})$ и $N(\alpha) = 9$:

$$\begin{aligned}
&\hat{\phi}_\alpha : (x, y) \to \\
&\left(\frac{\left(-\alpha u^3 + 3\,(9 - 2\alpha)\, u^2 + 2\alpha u + \alpha\right)\left(u^6 - 15u^5 + 24u^4 + 88u^3 + 51u^2 + 12u + 1\right)}{27u^2\left(u^3 + 3\,(2 - \alpha)\, u^2 - 6u - 1\right)^2}, \right. \\
&\qquad - \frac{\left(u^3 + 6u + 2\right) v}{27\alpha u^3\left(u^3 + 3\,(2 - \alpha)\, u^2 - 6u - 1\right)^3} \times \\
&\times \left(\alpha u^9 + 9(9 - \alpha)u^8 + 9(28\alpha - 135)u^7 + 12(162 - 115\alpha)u^6 + 18(396 - 83\alpha)u^5 + \right. \\
&\left. \left. + 9(459 - 62\alpha)u^4 + 3(324 - 107\alpha)u^3 + 9(9 - 13\alpha)u^2 - 18\alpha u - \alpha \right) \right).
\end{aligned}$$

№ 5. Кривая $v^2 = u^3 - \frac{3}{32}(\alpha - 6)u^2 - \frac{1}{64}(3\alpha - 2)u$, $\alpha = \frac{1}{2}(1 + \sqrt{-7})$ и $N(\alpha) = 2$:
Подставляя $\theta = \frac{\alpha - 4}{2}$ и $\gamma = -\frac{\alpha + 10}{112}$ в равенства (1.61) и учитывая, что

$$\begin{aligned}
\alpha^2 &= \alpha - 2, \\
\alpha^3 &= -(\alpha + 2), \\
(\alpha + 10)^2 &= 7(3\alpha + 14), \\
(\alpha + 10)^3 &= 7(47\alpha + 134),
\end{aligned}$$

получим коэффициенты кривой

$$\begin{aligned}
a_2\gamma &= \frac{-3(\alpha-4)(\alpha+10)}{224} &&= -\frac{3}{32}(\alpha - 6), \\
a_4\gamma^2 &= \frac{(\alpha+10)^2\left(3\alpha^2 - 24\alpha + 13\right)}{50176} &&= -\frac{1}{64}(3\alpha - 2), \\
a_6\gamma^3 &= -\frac{(\alpha+10)^3\left(\alpha^3 - 12\alpha^2 + 13\alpha - 22\right)}{11239424} &&= 0.
\end{aligned}$$

и эндоморфизм

$$\hat{\phi}_\alpha : (u, v) \to \left(-\frac{(\alpha+1)u^2 + u - \mu}{4u},\ -\frac{(\alpha+1)u^2 + \mu}{4\alpha u^2} v, \right),$$

где $\mu = \frac{\alpha-2}{16} = \left(\frac{\alpha}{4}\right)^2 = \frac{1}{4(\alpha+1)}$.

Отметим, что в [61, § 7.2.3] приводится эндоморфизм $\hat{\phi}_\alpha$, построенный для той же кривой при $\theta = -\frac{1}{2}$ и $\gamma = \frac{1}{2}$. В этом случае уравнение кривой принимает вид

$$v^2 = u^3 - \frac{3}{4}u^2 - 2u - 1,$$

а эндоморфизм $\hat{\phi}_\alpha$

$$\hat{\phi}_\alpha : (u, v) \to \left(-\frac{(\alpha+1)u^2 + 2(1-\alpha)}{4u + 3 - \alpha},\ -\frac{4(\alpha+1)u^2 + 2(\alpha+5)u + 8(\alpha-1)}{\alpha(4u+3-\alpha)^2} v \right).$$

Теперь, воспользовавшись равенством $-\frac{1}{4}\alpha^2(\alpha+1) = 1$ и обозначая $\omega = \frac{\alpha-3}{4}$, можно записать равенства

$$-(\alpha+1)u^2 - 2(1-\alpha) = \frac{4}{\alpha^2}\left(u^2 - \frac{(1-\alpha)\alpha^2}{2}\right) = \frac{4}{\alpha^2}(u^2 - \alpha),$$

$$\begin{aligned} &-4(\alpha+1)u^2 - 2(\alpha+5)u - 8(\alpha-1) = \\ &\qquad = \frac{16}{\alpha^2}\left(u^2 - \frac{(\alpha+5)\alpha^2}{8}u - \frac{(\alpha-1)\alpha^2}{2}\right) = \frac{16}{\alpha^2}\left(u^2 - 2\omega u + \alpha\right) \end{aligned}$$

и эндоморфизм $\hat{\phi}_\alpha$ в виде

$$\hat{\phi}_\alpha : (u, v) \to \left(\frac{u^2 - \alpha}{\alpha^2(u-\omega)},\ \frac{u^2 - 2\omega u + \alpha}{\alpha^3(u-\omega)^2} \right).$$

Для остальных кривых, указанных в таблице 1.4, также могут быть предъявлены значения θ и γ, снижающие алгоритмическую сложность вычисления эндоморфизма $\hat{\phi}_\alpha$.

§ 1.4.5. Алгоритм вычисления кратной точки

Пусть $p > 3$ – простое число, $\mathcal{H}(\mathbb{F}_p)$ – эллиптическая кривая, заданная в аффинной форме (1.60) сравнением

$$v^2 \equiv u^3 + a_2 u^2 + a_4 u + a_6 \pmod{p},$$

где $a_2, a_4, a_6 \in \mathbb{F}_p$. Будем считать, что кольцо эндоморфизмов $\mathtt{End}(\mathcal{H})$ этой кривой изоморфно решетке $\Lambda_\tau = \{n + m\tau, n, m \in \mathbb{Z}\}$ и $\Lambda_\tau \subseteq \mathbb{Z}_{\mathbb{K}} \subset \mathbb{Q}(\sqrt{-d})$, где d – свободное от квадратов, натуральное число, а τ определено равенством (1.46).

Будем считать, что задан элемент $\alpha \in \Lambda_\tau$, его минимальный многочлен

$$m_\alpha(x) = x^2 - \mathrm{tr}(\alpha)x + N(\alpha) \in \mathbb{Z}[x], \tag{1.64}$$

где

$$\mathrm{tr}(\alpha) = \alpha + \overline{\alpha}, \quad N(\alpha) = \alpha \cdot \overline{\alpha},$$

$\overline{\alpha}$ – сопряженный с α второй корень многочлена $m_\alpha(x)$, а также соответствующий α эндоморфизм $\hat{\phi}_\alpha \in \mathtt{End}(\mathcal{H})$.

Зафиксируем точку $P \in \mathcal{H}(\mathbb{F}_p)$ и будем считать, что $\mathtt{ord}\, P = q$ – простое число. Наиболее известный алгоритм вычисления кратной точки

$$Q = [k] = \underbrace{P + \cdots + P}_{k \text{ раз}}, \quad k \in \mathbb{F}_q,$$

ведет начало от работы [95] и основывается на представлении

$$k \equiv k_1 + k_2\alpha_q \pmod q, \tag{1.65}$$

где $\alpha_q \in \mathbb{F}_q$ корень минимального многочлена $m_\alpha(x) \pmod q$. Тогда

$$Q = [k]P = [k_1 + k_2\alpha_q]P = [k_1]P + [k_2]\left([\alpha_q]P\right) = [k_1]P + [k_2]\hat{\phi}_\alpha(P).$$

Очевидно, что представление (1.65) не единственно. Полагая

$$k^* \equiv k\alpha_q^{-1} \pmod q$$

можно видеть, что для любого $n = 0, 1, \ldots, q-1$ выполнено сравнение

$$k \equiv \underbrace{n\alpha_q}_{k_1} + \underbrace{(k^* - n)}_{k_2}\alpha_q \pmod q.$$

Используя подход, в основе которого лежит метод гауссового приведения, см. теорему 1.IV, в работе [7], см. также [375], была получена следующая оценка для коэффициентов k_1, k_2, участвующих в сравнении (1.65).

Теорема 1.VII. *Пусть $\Lambda_\alpha = \{s_0 + s_1\alpha,\ s_0, s_1 \in \mathbb{Z}\} \subseteq \Lambda_\tau \subset \mathbb{Q}(\sqrt{-d})$ – решетка, порождаемая элементом α. Если найдется элемент $\lambda \in \Lambda_\alpha$ такой, что $q|N(\lambda)$, то найдется и элемент $\rho = k_1 + k_2\alpha \in \Lambda_\tau$ такой, что*

$$k \equiv \rho \ (mod\, \lambda), \quad \text{и} \quad N(\rho) \leqslant c_0 N(\alpha)$$

где

$$0 < c_0 \leqslant \begin{cases} \frac{9+4N(\alpha)}{16}, & \text{если} \quad \mathrm{tr}(\alpha) - \text{нечетно}, \\ \frac{1+N(\alpha)}{4}, & \text{иначе}. \end{cases}$$

Из утверждения теоремы 1.VII сразу вытекают равенства $k = \beta\lambda + \rho$ и

$$Q = [k]P = \beta\left(\lambda(P)\right) + \rho(P) = \rho(P) = [k_1]P + [k_2]\hat{\phi}_\alpha(P),$$

выполненные для некоторого $\beta \in \Lambda_\alpha$, а также неравенство, которому удовлетворяют коэффициенты k_1, k_2

$$N(\rho) = k_1^2 + k_1 k_2 \operatorname{tr}(\alpha) + k_2^2 N(\alpha) \leqslant c_0 N(\alpha).$$

Далее в разделе описывается другой подход к вычислению кратной точки. В начале приведем небольшой пример и рассмотрим эллиптическую кривую $\mathcal{H}() : v^2 = u^3 - \frac{3}{4}u^2 - 2u - 1$ и эндоморфизм $\alpha = \frac{1}{2}(1 + \sqrt{-7})$, являющийся корнем неприводимого в $\mathbb{Z}$ многочлена $m_\alpha(x) = x^2 - x + 2$. Тогда выполнены точные равенства

$$\begin{array}{rcl}
1 & = & 1, \\
2 & = & -\alpha^2 + \alpha, \\
3 & = & -\alpha^2 + \alpha + 1, \\
4 & = & \alpha^5 + \alpha^2, \\
5 & = & \alpha^5 + \alpha^2 + 1, \\
6 & = & \alpha^5 + \alpha, \\
7 & = & \alpha^5 + \alpha + 1, \\
8 & = & \alpha^5 - \alpha^3, \\
9 & = & \alpha^5 - \alpha^3 + 1, \\
10 & = & \alpha^5 - \alpha^3 - \alpha^2 + \alpha, \\
11 & = & \alpha^5 - \alpha^3 - \alpha^2 + \alpha + 1, \\
12 & = & -\alpha^7 + \alpha^6 + \alpha^5 - \alpha^4 + \alpha^3 + \alpha^2, \\
13 & = & -\alpha^7 + \alpha^6 + \alpha^5 - \alpha^4 + \alpha^3 + \alpha^2 + 1, \\
14 & = & -\alpha^7 + \alpha^6 + \alpha^5 - \alpha^4 + \alpha^3 + \alpha, \\
15 & = & -\alpha^7 + \alpha^6 + \alpha^5 - \alpha^4 + \alpha^3 + 1, \\
16 & = & -\alpha^7 + \alpha^6 + \alpha^5 - \alpha^4 \text{ и т.д.}
\end{array}$$

Из приведенного примера видно, что любое натуральное число k может быть в точности представлено значением некоторого многочлена с целыми коэффициентами в точке α. Более точно, можно сформулировать следующую теорему.

Теорема 1.6. *Пусть $d > 1$ – свободное от квадратов, целое число и задан элемент $\alpha \in \Lambda_\tau \subseteq \mathbb{Z}_\mathbb{K} \subset \mathbb{Q}(\sqrt{-d})$ такой, что $N(\alpha) \geqslant 2$. Определим натуральное число*

$$n_\alpha = \frac{N(\alpha) - \delta_\alpha}{2}, \quad \textit{где} \quad \delta_\alpha \equiv N(\alpha) \ (mod\, 2)\,, \tag{1.66}$$

и множество

$$\mathcal{N} = [-n_\alpha, -n_\alpha + 1,\ \ldots,\ n_\alpha - 1, n_\alpha]\,.$$

Тогда, если α *удовлетворяет неравенству* $|\operatorname{tr}(\alpha)-1| \leqslant n_\alpha$, *то для любого натурального* k *найдется многочлен* $g(z) \in \mathcal{N}[z]$ *такой, что*

$$k = g(\alpha) = \sum_{i=0}^{w+c_1} x_i \alpha^i, \quad x_i \in \mathcal{N},$$

где $\deg g(z) \leqslant w + c_1$, *где* $w = \left\lceil 2 \log_{N(\alpha)} k \right\rceil$ *и*

$$c_1 = \begin{cases} 4, & \text{если} \quad \alpha = 1 \pm \sqrt{-2}, \\ 3, & \text{иначе.} \end{cases} \tag{1.67}$$

Заметим, что вопросы представления целых чисел в системах счисления с произвольным действительным основанием α ведут начало от работ А. Реньи [211] и В. Перри [190]. Используемый в данной работе метод представления натурального числа k в системе счисления с комплексным основанием α базируется на результатах работ В. Мюллера [167] и Н. Смарта [234]. Отметим, что в более слабой форме, теорема 1.6 формулировалась в работах [118, 374].

Рассмотрим функцию $||\xi||$, возвращающую длину вектора, определяемого комплексным числом ξ, и докажем вспомогательные леммы.

Лемма 1.1. *Пусть* $\alpha \in \Lambda_\tau$ *удовлетворяет условиям теоремы 1.6 и* $\Lambda_\alpha \subseteq \Lambda_\tau$ *– решетка, порожденная элементом* α. *Тогда для любого* $\xi \in \Lambda_\alpha$ *найдутся значения* $x \in \mathcal{N}$ *и* $\xi_1 \in \Lambda_\alpha$ *такие, что*

$$\xi = x + \alpha\xi_1, \quad \text{и} \quad ||\xi_1|| \leqslant \frac{||\xi||}{\sqrt{N(\alpha)}} + \frac{\sqrt{N(\alpha)}}{2}.$$

Доказательство. Пусть $\xi = s_0 + s_1\alpha$, тогда, используя алгоритм деления с остатком, определим $s_0 = qN(\alpha) + r$, где $0 \leqslant r < N(\alpha)$. Если $r \leqslant n_\alpha$, то положим $x = r$. В противном случае, положим $x = r - N(\alpha)$ и $q = q + 1$, тогда

$$0 > x > -(N - n_\alpha) = -(2n_\alpha - \delta_\alpha - n_\alpha) = -n_\alpha + \delta_\alpha > n_\alpha.$$

В обоих случаях, $s_0 = qN(\alpha) + x$ и $x \in \mathcal{N}$.

Поскольку α является корнем своего минимального многочлена $m_\alpha(x)$, то из (1.64) следует, что выполнены равенства

$$N(\alpha) = -\alpha^2 + \operatorname{tr}(\alpha)\alpha = \alpha(\operatorname{tr}(\alpha) - \alpha)$$

и

$$\xi = x + qN(\alpha) + s_1\alpha = x + \alpha(q\operatorname{tr}(\alpha) - q\alpha + s_1) = x + \alpha\xi_1,$$

где $\xi_1 = \big(q\,\mathrm{tr}(\alpha) + s_1\big) - q\alpha$. Записывая равенство

$$\xi_1 = \frac{\xi - x}{\alpha}$$

и используя неравенство треугольника, получим необходимую оценку

$$||\xi_1|| = \frac{||\xi - x||}{||\alpha||} \leqslant \frac{||\xi|| + ||x||}{\sqrt{N(\alpha)}} \leqslant \frac{||\xi|| + n_\alpha}{\sqrt{N(\alpha)}} \leqslant \frac{||\xi||}{\sqrt{N(\alpha)}} + \frac{\sqrt{N(\alpha)}}{2}.$$

□

Лемма 1.2. *Пусть $\alpha \in \Lambda_\tau$ удовлетворяет условиям теоремы 1.6, элемент $\xi \in \Lambda_\alpha$ и*

$$||\xi|| < c_2 + \sqrt{N(\alpha)},$$

где $c_2 = 2$ при $N(\alpha) \in \{2, 3\}$ и $c_2 = 1$ иначе. Тогда найдутся такие элементы $x_0, \dots x_{c_1} \in \mathcal{N}$, что

$$\xi = \sum_{i=0}^{c_1} x_i \alpha^i,$$

где константа c_1 определена равенством (1.67).

Доказательство. Запишем $\xi = s_0 + s_1\alpha$. Используя неравенство треугольника и равенство $||\alpha|| = \sqrt{N(\alpha)}$, получим неравенство

$$||\xi|| = ||s_0 + s_1\alpha|| \leqslant |s_0| + |s_1|\sqrt{N(\alpha)} < c_2 + \sqrt{N(\alpha)}$$

из которого следуют оценки на величины s_0 и s_1:

$$|s_0| < c_2 + \sqrt{N(\alpha)}, \quad |s_1| < 1 + \frac{c_2}{\sqrt{N(\alpha)}}. \tag{1.68}$$

Следовательно, при $N(\alpha) \in \{2, 3\}$ выполнено

$$|s_0| < 2 + \sqrt{N(\alpha)} < 2N(\alpha),$$

а для остальных значений $N(\alpha)$ выполнено $|s_0| < N(\alpha)$ или, обобщая,

$$|s_0| < c_2 N(\alpha).$$

Полученные оценки не позволяют говорить, что $s_0, s_1 \in \mathcal{N}$. Следовательно, необходимо выполнить преобразования, аналогичные тем, что были выполнены в ходе доказательства леммы 1.1.

Начнем с общего случая: $N(\alpha) \geqslant 4$, тогда $n_\alpha \geqslant 2$ и $|s_1| < 1 + \frac{1}{2}$. Поскольку s_1 целое, то $|s_1| \leqslant 1 \leqslant n_\alpha - 1$ и $s_1 \in \mathcal{N}$. Следовательно, если $|s_0| \leqslant n_\alpha$, то искомое представление получено.

Предположим, что $s_0 > 0$ (случай $s_0 < 0$ рассматривается аналогично), тогда определим $x_0 = N - s_0$ и

$$\xi = -x_0 + N + s_1\alpha = -x_0 + \alpha(\operatorname{tr}(\alpha) - \alpha + s_1) = -x_0 + \alpha\xi_1,$$

где

$$\xi_1 = \operatorname{tr}(\alpha) + s_1 - \alpha.$$

Если $|\operatorname{tr}(\alpha) + s_1| \leqslant n_\alpha$, то искомое представление получено. В противном случае, учтем ограничение $|\operatorname{tr}(\alpha) - 1| \leqslant n_\alpha$, введенное в условии теоремы 1.6, и получим, что

$$|\operatorname{tr}(\alpha) + s_1| \leqslant |\operatorname{tr}(\alpha)| + |s_1| \leqslant n_\alpha + 1 + n_\alpha - 1 \leqslant N(\alpha).$$

Теперь, обозначая $x_1 = N(\alpha) - (\operatorname{tr}(\alpha) + s_1)$ получим равенство

$$\xi_1 = -x_1 + N - \alpha = -x_1 + \alpha(\operatorname{tr}(\alpha) + \alpha - 1) = -x_1 + \alpha\xi_2,$$

где $\xi_2 = x_2 + \alpha$ и $x_2 = \operatorname{tr}(\alpha) - 1$, следовательно, $|x_2| = |\operatorname{tr}(\alpha) - 1| \leqslant n_\alpha$ и искомое представление получено:

$$\begin{aligned}\xi = -x_0 + \alpha\xi_1 = -x_0 + \alpha(-x_1 + \alpha\xi_2) = \\ = -x_0 + \alpha(-x_1 + \alpha(x_2 + \alpha)) = \\ = -x_0 - x_1\alpha + x_2\alpha^2 + \alpha^3, \quad x_0, x_1, x_2 \in \mathcal{N}.\end{aligned}$$

Теперь осталось рассмотреть частные случаи. Начнем со случая, когда $N(\alpha) = 2$, тогда $n_\alpha = 1$ и

$$0 \leqslant \operatorname{tr}(\alpha) \leqslant n_\alpha + 1 = N(\alpha).$$

Равенство $\operatorname{tr}(\alpha) = N(\alpha)$ для мнимой квадратичной иррациональности α возможно только для значений

$$\alpha = \frac{1 + \sqrt{-3}}{2}, \quad \alpha = 1 + \sqrt{-1}, \quad \alpha = \frac{3 + \sqrt{-3}}{2}.$$

Из перечисленных значений, только $\alpha = 1 + \sqrt{-1}$ удовлетворяет частному случаю $N(\alpha) = 2$. Вместе с тем, это значение α не удовлетворяет условию теоремы 1.6.

Тогда, для остальных значений α таких, что $N(\alpha) = 2$, выполнено $0 \leqslant \operatorname{tr}(\alpha) \leqslant 1$ и $\operatorname{tr}(\alpha) \in \mathcal{N}$. Далее, учитывая, что s_0, s_1 целые, из (1.68) следует

$$|s_0| \in \{0, 1, 2, 3\}, \quad |s_1| \in \{0, 1, 2\}.$$

Если $|s_0| \in \{0, 1\}$, то $s_0 \in \mathcal{N}$. Если $|s_1| \in \{0, 1\}$, то $s_1 \in \mathcal{N}$ и искомое представление получено. Если же $|s_1| = 2$, то можно записать

$$\xi = s_0 + 2\varepsilon\alpha = s_0 + \varepsilon N(\alpha)\alpha = s_0 + \varepsilon\alpha^2(\operatorname{tr}(\alpha) - \alpha) = s_0 + \varepsilon\operatorname{tr}(\alpha)\alpha^2 - \varepsilon\alpha^3,$$

где $\varepsilon = \pm 1$ и все коэффициенты полученного разложения принадлежат множеству $\mathcal{N}$. Осталось рассмотреть случай, когда $s_0 \in \{2, 3\}$, тогда

$$\xi = N(\alpha) + x_0 + s_1\alpha = x_0 + \alpha(\mathrm{tr}(\alpha) - \alpha + s_1) = x_0 + \alpha\xi_1,$$

где $x_0 \in \{0, 1\} \in \mathcal{N}$ и $\xi_1 = \mathrm{tr}(\alpha) + s_1 - \alpha$. Если $\mathrm{tr}(\alpha) + s_1 \in \mathcal{N}$, то искомое представление получено, в противном случае

$$\mathrm{tr}(\alpha) + s_1 \in \{-2, 2, 3\}.$$

Запишем равенство $\mathrm{tr}(\alpha) + s_1 = \varepsilon N(\alpha) + x_1$, где $x_1 \in \{0, 1\} \in \mathcal{N}$ и $\varepsilon = \pm 1$. Тогда

$$\xi_1 = x_1 + \varepsilon N - \alpha = x_1 + \alpha(\varepsilon\,\mathrm{tr}(\alpha) - 1 - \alpha) = x_1 + \alpha\xi_2,$$

где

$$\xi_2 = \varepsilon\,\mathrm{tr}(\alpha) - 1 - \alpha.$$

Рассматривая все возможные варианты

$\mathrm{tr}(\alpha) + s_1$	s_1	$\mathrm{tr}(\alpha)$	ε	$\varepsilon\,\mathrm{tr}(\alpha) - 1$
-2	-2	0	-1	-1
2	2	0	1	-1
2	1	1	1	0
3	3	0	1	-1

можно убедиться, что во всех случаях коэффициенты ξ_2 удовлетворяют необходимым ограничениям и для случая $N(\alpha)$ также выполнено равенство

$$\xi = x_0 + x_1\alpha + x_2\alpha^2 - \alpha^3.$$

Для завершения доказательства леммы, нам осталось рассмотреть случай $N(\alpha) = 3$, тогда $n_\alpha = 1$ и

$$0 \leqslant \mathrm{tr}(\alpha) \leqslant n_\alpha + 1 = 2 < N(\alpha).$$

Учитывая, что s_0, s_1 целые, из (1.68) следует

$$|s_0| \in \{0, 1, 2, 3\}, \quad |s_1| \in \{0, 1, 2\}.$$

Если $|s_0| \in \{0, 1\}$, то $s_0 \in \mathcal{N}$. Если $|s_1| \in \{0, 1\}$, то $s_1 \in \mathcal{N}$ и искомое представление получено. Если же $|s_1| = 2$, то можно записать

$$\xi = s_0 + 2\varepsilon\alpha = s_0 + \varepsilon\,(N(\alpha) - 1)\,\alpha = s_0 - \varepsilon\alpha + \varepsilon\alpha^2(\mathrm{tr}(\alpha) - \alpha) = \\ = s_0 - \varepsilon\alpha + \varepsilon\,\mathrm{tr}(\alpha)\alpha^2 - \varepsilon\alpha^3,$$

где $\varepsilon = \pm 1$.

Если $0 \leqslant \mathrm{tr}(\alpha) \leqslant 1$, то коэффициенты полученного разложения принадлежат множеству $\mathcal{N}$. Исключением является случай, когда $\mathrm{tr}(\alpha) = 2$. Тогда, можно записать равенство

$$\begin{aligned}\xi = s_0 - \varepsilon\alpha + \varepsilon\,\mathrm{tr}(\alpha)\alpha^2 - \varepsilon\alpha^3 = \\ = s_0 - \varepsilon\alpha + \varepsilon(N(\alpha) - 1)\alpha^2 - \varepsilon\alpha^3 = \\ = s_0 - \varepsilon\alpha - \varepsilon\alpha^2 + \varepsilon\alpha^3(2 - \alpha) - \varepsilon\alpha^3 = \\ = s_0 - \varepsilon\alpha - \varepsilon\alpha^2 + \varepsilon\alpha^3 - \varepsilon\alpha^4,\end{aligned}$$

и все коэффициенты полученного разложения принадлежат множеству $\mathcal{N}$. Отметим, что подобная ситуация возможна для α, являющегося корнем многочлена $x^2 - 2x + 3$, т.е. для $\alpha = 1 \pm \sqrt{-2}$ и $\xi = \pm 2\alpha$.

Осталось рассмотреть случай, когда $s_0 \in \{2, 3\}$, тогда

$$\xi = N(\alpha) - x_0 + s_1\alpha = -x_0 + \alpha(\mathrm{tr}(\alpha) - \alpha + s_1) = -x_0 + \alpha\xi_1,$$

где $x_0 \in \{0, 1\} \in \mathcal{N}$ и $\xi_1 = \mathrm{tr}(\alpha) + s_1 - \alpha$. Если $\mathrm{tr}(\alpha) + s_1 \in \mathcal{N}$, то искомое представление получено, в противном случае

$$\mathrm{tr}(\alpha) + s_1 \in \{-2, 2, 3, 4\}.$$

Записывая $\mathrm{tr}(\alpha) + s_1 = \varepsilon N(\alpha) + x_1$, где $x_1 \in \{0, \pm 1\} \in \mathcal{N}$ и $\varepsilon = \pm 1$. Тогда

$$\xi_1 = x_1 + \varepsilon N - \alpha = x_1 + \varepsilon\alpha(\mathrm{tr}(\alpha) - \alpha) - \alpha = x_1 + \alpha\xi_2,$$

где

$$\xi_2 = \varepsilon\,\mathrm{tr}(\alpha) - 1 - \varepsilon\alpha.$$

Если $\varepsilon = 1$, то коэффициенты ξ_2 принадлежат множеству $\mathcal{N}$. Случай $\varepsilon = -1$ возможен только при $\mathrm{tr}(\alpha) + s_1 = -2$, поскольку $\mathrm{tr}(\alpha) \geqslant 0$, то это равносильно одновременному выполнению равенств $\mathrm{tr}(\alpha) = 0$ и $s_1 = -2$. Но в этом случае, $\xi_2 = -1 + \alpha$. Лемма доказана. □

Доказательство теоремы 1.6. Поскольку для любого $k \in \mathbb{K}$ выполнено равенство $k = k + 0 \cdot \alpha$, мы будем считать, что $k \in \Lambda_\alpha$. Тогда, воспользовавшись утверждением леммы 1.1, запишем последовательность равенств

$$k = x_0 + \alpha\xi_1 = x_0 + \alpha(x_1 + \alpha\xi_2) = \cdots = \sum_{i=0}^{w-1} x_i\alpha^i + \alpha^w\xi_w,$$

для любого натурального $w = 1, 2, \ldots$ и $x_0, \ldots, x_{w-1} \in \mathcal{N}$. Также, из равенства $||k|| = k$ и утверждения леммы 1.1 следует цепочка неравенств

$$||\xi_w|| \leqslant \frac{||\xi_{w-1}||}{\sqrt{N(\alpha)}} + \frac{\sqrt{N(\alpha)}}{2} \leqslant$$

$$\leqslant \frac{1}{\sqrt{N(\alpha)}} \left(\frac{||\xi_{w-2}||}{\sqrt{N(\alpha)}} + \frac{\sqrt{N(\alpha)}}{2} \right) + \frac{\sqrt{N(\alpha)}}{2} =$$

$$= \frac{||\xi_{w-2}||}{\left(\sqrt{N(\alpha)}\right)^2} + \frac{\sqrt{N(\alpha)}}{2} \left(1 + \frac{1}{\sqrt{N(\alpha)}} \right) \leqslant$$

$$\leqslant \cdots \leqslant \frac{k}{\left(\sqrt{N(\alpha)}\right)^w} + \frac{\sqrt{N(\alpha)}}{2} \left(1 + \frac{1}{\sqrt{N(\alpha)}} + \cdots + \frac{1}{\left(\sqrt{N(\alpha)}\right)^{w-1}} \right) <$$

$$< \frac{k}{\left(\sqrt{N(\alpha)}\right)^w} + \frac{\sqrt{N(\alpha)}}{2} \sum_{i=0}^{\infty} \frac{1}{\left(\sqrt{N(\alpha)}\right)^i} = \frac{k}{\left(\sqrt{N(\alpha)}\right)^w} + \frac{N(\alpha)}{2\left(\sqrt{N(\alpha)} - 1\right)}.$$

Следовательно, выбирая значение w равным $\left\lceil 2 \log_{N(\alpha)} k \right\rceil$ получим неравенства $\left(\sqrt{N(\alpha)}\right)^w > k$ и

$$||\xi_w|| < 1 + \frac{N(\alpha)}{2\left(\sqrt{N(\alpha)} - 1\right)}.$$

Далее, из неравенства

$$\frac{N(\alpha)}{2\left(\sqrt{N(\alpha)} - 1\right)} \cdot \frac{1}{\sqrt{N(\alpha)}} = \frac{\sqrt{N(\alpha)} - 1 + 1}{2\left(\sqrt{N(\alpha)} - 1\right)} = \frac{1}{2} + \frac{1}{2\left(\sqrt{N(\alpha)} - 1\right)} \leqslant 1,$$

выполненного для всех $N(\alpha) \geqslant 4$, следует, что

$$||\xi_w|| < c_2 + \sqrt{N(\alpha)}$$

где $c_2 = 2$ при $N(\alpha) \in \{2, 3\}$ и $c_2 = 1$ иначе. Теперь, утверждение теоремы следует из леммы 1.2. □

Следует отметить, что все значения $\alpha = \tau$, перечисленные в таблицах 1.4 и 1.5, удовлетворяют условиям только что доказанной теоремы. Доказательство теоремы 1.6 конструктивно, что позволяет предъявить алгоритм вычисления кратной точки для каждой из кривых, содержащихся в указанных таблицах, см. алгоритм 1.7.

Алгоритм 1.7 состоит из двух частей. В первой части используется представление $\xi = s_0 + s_1\alpha$ и для заданного натурального k вычисляется последовательность целых коэффициентов $x_0, \ldots, x_w$ такая, что

$$k = x_0 + x_1\alpha + x_2\alpha^2 + \cdots x_w\alpha^w, \quad w \in \mathbb{N}$$

и $x_o, \ldots, x_w \in \mathcal{N} = [-n_\alpha, \ldots, n_\alpha]$. На втором шаге для вычисления искомой точки Q используется схема Горнера и равенство

$$Q = [k]P = [x_0]P + [x_1]\phi_\alpha(P) + [x_2]\phi_\alpha^2(P) + \cdots + [x_w]\phi_\alpha^w(P) = \\ = \phi_\alpha(\cdots(\phi_\alpha(\phi_\alpha([x_w]P) + [x_{w-1}]P)\cdots) + [x_1]P)) + [x_0]P.$$

При этом точки $R_1 = P, R_2 = [2]P, \ldots, R_{n_\alpha} = [n_\alpha]P$ могут быть вычислены заранее.

Алгоритм 1.7: Алгоритм вычисления кратной точки эллиптической кривой с использованием эндоморфизма ϕ_α

Вход : Натуральное число k, эллиптическая кривая $\mathcal{E}$ и точка $P \in \mathcal{E}$, а также эндоморфизм $\phi_\alpha : \mathtt{End}(\mathcal{E}) \to \mathtt{End}(\mathcal{E})$ и значение $N(\alpha)$.
Выход : Точка $Q \in \mathcal{E}$, удовлетворяющая равенству $Q = [k]P$.

Определить $n_\alpha = \frac{N(\alpha)-\delta_\alpha}{2}$, где $\delta_\alpha \equiv N(\alpha) \pmod 2$, $s_0 = k$, $s_1 = 0$ и $w = 0$.
Для всех $i = 1, \ldots, n_\alpha$ **выполнять**
  Определить $R_i = [i]P$.
конец
Пока ($s_0 \not\equiv 1$ *и* $s_1 \neq 0$) **выполнять**
  Используя алгоритм деления с остатком, определить $s_0 = qN + x_w$.
  Если $x_w > n_\alpha$ **то**
    Определить $x_w = N - x_w$ и $q = q + 1$.
  конец
  Определить $s_0 = q\,\mathrm{tr}(\alpha) + s_1$, $s_1 = -q$ (см. доказательство леммы 1.1).
  Определить $w = w + 1$.
конец
Определить[a] $Q = [\mathrm{sign}(x_w)]R_{x_w}$.
Для всех $i = w - 1, \ldots, 0$ **выполнять**
  Вычислить $Q = \phi_\alpha(Q)$. **Если** $x_i > 0$ **то**
    Определить $Q + R_{x_i}$.
  конец
  Определить $Q = Q - R_{x_i}$.
конец

[a] Напомним, что функция $\mathrm{sign}(x)$ возвращает знак числа x.

Заключение к § 1.4

В § 1.4 предложен и обоснован алгоритм вычисления явного представления эндоморфизмов эллиптических кривых.

Приведены результаты практической реализации данного алгоритма и предъявлены эндоморфизмы для всех эллиптических кривых $\mathcal{E}$, чье кольцо $\mathtt{End}(\mathcal{E})$ изоморфно порядку мнимого квадратичного поля с числом классов равным единице, см. таблицу 1.4.

Предъявлены формы представления эллиптических кривых с указанным свойством, допускающие минимальную сложность вычисления построенных эндоморфизмов.

Доказана теорема о представлении натуральных чисел значениями многочленов в точках мнимого квадратичного поля. Утверждение доказанной теоремы применено для реализации алгоритма вычисления кратной точки на эллиптической кривой.

§ 1.5. Алгоритмы построения эллиптических кривых

В последнем параграфе данной главы формулируются требования к параметрам эллиптических кривых и описывается разработанный автором алгоритм построения данных параметров, позволяющий сделать нецелесообразным применение известных методов дискретного логарифмирования в группе точек эллиптической кривой, см. обзор в § 1.1, а также результаты § 1.3. Завершают параграф результаты практических вычислений.

Дальнейшее изложение следует статье [179].

§ 1.5.1. Определение требований

Дадим следующие определения.

Определение 1.5. *Пусть p нечетное простое число. Мы будем называть число p безопасным простым, если число $\frac{p-1}{2}$ также является простым.*

Отметим, что в этом случае простое число $\frac{p-1}{2}$ принято называть простым числом Софи Жермен, см. [230, § 5.5.5].

Определение 1.6. *Пусть $0 < \alpha < \beta$ натуральные числа, $p > 3$ простое число. Мы будем называть эллиптическую кривую $\mathcal{E}_{a,b}(\mathbb{F}_p)$, определенную сравнением (1.4)*

$$y^2 \equiv x^3 + ax + b \pmod p,$$

безопасной, если найдется точка $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ такая, что $\operatorname{ord} P = q$ и выполняются следующие условия:

1. *$m = |\mathcal{E}_{a,b}(\mathbb{F}_p)|$ и $m \neq p$;*
2. *p безопасное простое, т.е. $\frac{p-1}{2}$ также простое число;*

3. $j(\mathcal{E}_{a,b}) \not\equiv 0$ *или* $1728 \pmod p$, *где величина* $j(\mathcal{E}_{a,b})$ *определена сравнением (1.48);*

4. $2^\alpha < q < 2^\beta$;

5. q *безопасное простое, т.е.* $\frac{q-1}{2}$ *также простое число;*

6. *для фиксированного значения* B *условие* $p^t \not\equiv 1 \pmod q$ *выполняется для всех* $t = 1, 2, \ldots, B$.

Заметим, что безопасная эллиптическая кривая удовлетворяет требованиям из ГОСТ Р 34.10-2012, см. [280], дополненным требованиями простоты чисел $\frac{p-1}{2}$ и $\frac{q-1}{2}$.

Первое условие из определения 1.6 делает невозможным применение методов Т. Сато и К. Араки [219], И.А. Семаева [225] и Н. Смарта [235]. Условие безопасности простого числа p делает неприменимым алгоритм К. Пети, М. Костерса и А. Мессенга [194]. Условие безопасности простого числа q минимизирует мощность множества «слабых» ключей. Последнее, шестое условие делает нецелесообразным применение метода А. Менезеса, С. Ванстоуна, Т. Окамото [160].

Константы α, β и B могут выбираться различными способами в зависимости от национальных требований по безопасности. Так в национальном стандарте на электронную подпись ГОСТ Р 34.10-2012 предусмотрено два набора значений:

- $\alpha = 254$, $\beta = 256$ и $B = 31$, а также
- $\alpha = 508$, $\beta = 512$ и $B = 131$,

зависящие от используемой в стандарте функции хэширования; в [246] регламентируется другой набор параметров $\alpha = 224$, $\beta > \alpha$ и $B = 10^4$.

Заметим, что шестое условие из определения 1.6 может быть заменено условием

$$p^2 \not\equiv 1 \pmod q,$$

если $B < q_1 = \frac{q-1}{2}$. Действительно, поскольку $q = 2q_1 + 1$ и q_1 простое число, то возможны только три варианта

$$p^2 \equiv 1, \quad p^{q_1} \equiv 1, \quad p^{q-1} \equiv 1 \pmod q.$$

Также заметим, что для безопасного простого p условие $j(E_{a,b}) \not\equiv 0$ или 1728 влечет за собой неравенство $m \neq p + 1$, см. [109, гл. 13].

Используя некоторые недоказанные предположения, мы сформулируем более строгие требования к параметрам эллиптических кривых, рекомендуемых к применению в средствах защиты информации.

Введенная нами ранее в формулировке теоремы 1.V, см. стр. 93, величина h не только определяет число неэквивалентных приведенных квадратичных форм фиксированного дискриминанта, но и существенным образом влияет на эффективность вычислений, проводимых в поле рациональных функций $\mathbb{H}(x)$.

На настоящий момент времени автору не известен алгоритм решения задачи дискретного логарифмирования в группе точек эллиптической кривой, использующий вычисления в $\mathbb{H}(x)$. Однако еще в 2001 году в первой версии рекомендаций [246] было выдвинуто требование существования нижней оценки для величины h, а именно, $h \geqslant 200$. За прошедшие годы эта оценка не изменилась.

Более эмоциональная оценка для h предлагалась в работе [31]. Бернштейн и Ланге предложили использовать эллиптические кривые, для которых фундаментальный дискриминант Δ кольца эндоморизмов, определяемый равенством (1.45), удовлетворяет неравенству $\Delta > 2^{100}$. Такое ограничение делает полностью невозможными какие-либо практические вычисления в поле рациональных функций $\mathbb{H}(x)$. На взгляд автора оценка $h > 500$ является достаточной.

Еще одно требование основано на том факте, что кривая $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и ее твист $\mathcal{E}_{a',b'}(\mathbb{F}_p)$, рассматриваемые как кривые с коэффициентами из поля $\mathbb{H}$, имеют одинаковые инварианты и, следовательно, изоморфны, см. [109, 233]. Это позволяет предположить, что может быть найден алгоритм дискретного логарифмирования, имеющий одинаковую сложность для кривых $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и $\mathcal{E}_{a',b'}(\mathbb{F}_p)$. Следовательно, мы вправе требовать, чтобы твист $|\mathcal{E}_{a',b'}(\mathbb{F}_p)|$ также имел большой простой делитель.

Пусть эллиптическая кривая $\mathcal{E}_{a,b}(\mathbb{F}_p)$ является безопасной. Обозначим ее порядок символом $m_\delta = |\mathcal{E}_{a,b}(\mathbb{F}_p)| = uq$, где q также безопасное простое число. Учитывая приведенные выше значения α,β и ограничения на величину q, мы также будем считать, что $1 \leqslant u \leqslant 4$.

Обозначим порядок твиста $\mathcal{E}_{a',b'}(\mathbb{F}_p)$ символом $m_{-\delta} = |\mathcal{E}_{u,v}(\mathbb{F}_p)| = vr$, где r простое число. Верна следующая лемма, см. [179].

Лемма 1.3. *Пусть $p > 7$ безопасное простое число, тогда*

1. $p \equiv 11 \pmod{12}$,
2. *выполнено сравнение* $m_{-\delta} \equiv u \pmod{12}$ *и* $u|v$,
3. *если r безопасное простое число, то* $v \equiv -u \pmod{12}$.
4. *если* $u = v$, *то* $r \equiv 1 \pmod{\frac{12}{u}}$.

Доказательство. Поскольку $p > 7$ безопасное, нечетное простое число, то $p_1 = \frac{p-1}{2} > 3$ также является нечетным простым числом и может быть

записано в виде $p_1 = 2p_2 + 1$. Следовательно, мы сразу заключаем, что $p = 2(2p_2 + 1) + 1$ и $p \equiv 3 \pmod 4$.

Предположим, что $p \equiv 1 \pmod 3$, тогда из равенства

$$p = 3s + 1 = 2p_1 + 1,$$

выполненного для некоторого натурального s следует, что $s|2p_1$, и, в силу простоты p_1, либо $s = 2$, либо $s = p_1$. В первом случае мы заключаем, что $p = 2 \cdot 2 + 1 < 7$, а во втором, что $2 = 3$. Оба варианта невозможны, следовательно, наше предположение неверно и $p \equiv 2 \pmod 3$. Теперь, воспользовавшись китайской теоремой об остатках, мы сразу заключаем, что $p \equiv 11 \pmod{12}$.

Далее, учтем третье и четвертое утверждения теоремы 1.VI и рассмотрим сумму

$$uq + vr = m_\delta + m_{-\delta} = 2(p + 1). \tag{1.69}$$

Тогда, учитывая, что p, q безопасные простые, получаем второе утверждение леммы

$$m_{-\delta} = vr = 2(p + 1) - uq \equiv u \pmod{12}.$$

Поскольку $1 \leqslant u \leqslant 4$, то $u|12$. Учитывая, что r простое число, мы сразу заключаем, что $u|v$.

Равенство (1.69) дает нам также и третье утверждение леммы. Действительно, если r безопасное простое, то $r \equiv q \equiv p \equiv 11 \pmod{12}$ и мы получаем

$$(u + v)11 \equiv 0 \pmod{12}, \quad \text{или} \quad v \equiv -u \pmod{12}.$$

Если $u = v$, то равенство (1.69) дает нам

$$u(11 + r) \equiv 0 \pmod{12},$$

что равносильно $r - 1 \equiv 0 \left(\mathrm{mod}\ \frac{12}{u}\right)$. Лемма доказана. □

Из двух последних утверждений леммы следует, что простой делитель r не может быть одновременно безопасным простым и принадлежать интервалу $2^\alpha < r < 2^\beta$.

Определение 1.7. *Мы будем называть безопасную эллиптическую кривую $\mathcal{E}_{a,b}(\mathbb{F}_p)$ строго безопасной, если выполнены следующие условия:*

1. *число h классов неэквивалентных квадратичных форм дискриминанта $-\Delta$ не менее 500,*

2. *порядок твиста содержит простой делитель r, удовлетворяющий неравенствам $2^\alpha < r < 2^\beta$.*

§ 1.5.2. Алгоритмы построения эллиптической кривой, удовлетворяющей сформулированным требованиям

Построение параметров эллиптической кривой, удовлетворяющей определениям 1.6 или 1.7, может быть реализовано при помощи двух принципиально различных подходов:

- в первом случае мы случайно выбираем коэффициенты эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, а потом, используя алгоритм Шуфа-Элкиса-Аткина, см. [67, 222] или [38, гл. 7], находим порядок группы точек эллиптической кривой и проверяем выполнимость определений 1.6 или 1.7;

- во втором случае мы сначала определяем порядок эллиптической кривой, удовлетворяющий определению 1.6 или 1.7, а потом используя результаты, описанные в § 1.4.1, определяем коэффициенты эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$; впервые такой подход к построению эллиптических кривых был реализован на практике в работах Ф. Морейна, см. [8, 165].

Далее мы остановимся на втором подходе. Для его практической реализации может быть использовано несколько стратегий поиска простых чисел p, q и r, определяющих $\mathbb{F}_p$ – поле определения эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, а также простые делители ее порядка и порядка ее твиста.

Опишем указанные стратегии предполагая, что параметры $\alpha, \beta \in \mathbb{N}$, и $\alpha < \beta$, фиксированы.

1. Безопасные простые числа p ищутся в убывающей последовательности целых чисел $p_n = p_0 - 12n$, где $n = 1, 2, \ldots$, а p_0 – наименьшее натуральное число такое, что

$$p_0 > 2^\beta, \quad p_0 \equiv 11 \pmod{12}.$$

   Автором было экспериментально подтверждено, что такая стратегия позволяет строить эллиптические кривые достаточно эффективно, см. далее алгоритм 1.8.

2. Аналогично, простые числа могут искаться в возрастающей последовательности целых чисел $p_n = p_0 + 12n$, где $n = 1, 2, \ldots$, а p_0 – наибольшее натуральное число такое, что

$$p_0 < 2^\alpha, \quad p_0 \equiv 11 \pmod{12}.$$

Отметим, что для простых чисел вида $p = 2^\beta - \theta$ или $p = 2^\alpha + \theta$ элементарные алгоритмы, реализующие операции в поле $\mathbb{F}_p$, могут быть выполнены несколько быстрее. Кривые, определенные над такими полями, используются в средствах защиты информации достаточно часто, см. например, кривые «E-382» [20], «Curve25519» [29] или кривые «paramsetA» и «paramsetB», определяемые [362].

3. Безопасное простое число p может выбираться псевдослучайным образом. Для генерации числа p может использоваться псевдослучайная функция $h() : \mathbb{V}_n \to [2^\alpha, \ldots, 2^\beta] \subset \mathbb{N}$, отображающая произвольные битовые строки в натуральные числа в заданном интервале. Примером такой функции для $\beta = 256$ может служить функция хеширования, регламентированная ГОСТ Р 34.11-2012, см. [281].

   Использование псевдослучайных функций позволяет реализовать поиск простого числа p в виде детерминированной последовательности действий, которая при каждом повторении будет выдавать один и тот же результат. Это позволяет провести независимую проверку того, что алгоритм построения параметров эллиптической кривой корректен и не содержит в себе преднамеренных или непреднамеренных ошибок.

Основываясь на утверждениях теоремы 1.VI для каждого найденного безопасного простого числа p может быть реализована процедура поиска порядка группы точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, удовлетворяющей определениям 1.6 и 1.7, Эта процедура также может быть реализована несколькими способами.

1. Наиболее простой способ поиска порядка m заключается в последовательном переборе чисел

$$d = 1, 2, 5, 7, 10, 11, 13, 14, \ldots,$$

и проверки разрешимости равенства

$$4p = x^2 + dy^2 \tag{1.70}$$

для целых значений x, y. Если значения x, y существуют, то могут быть определены величины

$$m_- = p + 1 - |x|, \quad m_+ = p + 1 + |x|,$$

значения которых должны проверяться на соответствие определениям 1.6 и 1.7. Следующая лемма, см. [179], позволяет несколько сократить множество перебираемых значений.

Лемма 1.4. *Пусть $p \equiv 11\ (mod\, 12)$, d свободно от квадратов и уравнение (1.70) разрешимо в целых числах x, y. Тогда*

$$d \equiv \{2, 7, 10, 11\}\ (mod\, 12)\,. \tag{1.71}$$

Доказательство. Поскольку d свободно от квадратов, то

$$d \not\equiv 0, 4, 8\ (\mathrm{mod}\, 12)\,.$$

Далее, из условия $p \equiv 2\ (\mathrm{mod}\, 3)$ получаем $4p \equiv 2 \equiv x^2 + dy^2\ (\mathrm{mod}\, 3)$. Поскольку $\left(\frac{2}{3}\right) = -1$, то сравнение $2 \equiv x^2\ (\mathrm{mod}\, 3)$ не разрешимо, следовательно, $d \not\equiv 0\ (\mathrm{mod}\, 3)$.

В заключение рассмотрим случай $d \equiv 1\ (\mathrm{mod}\, 4)$. Из (1.70) следует $x^2 + y^2 \equiv 0\ (\mathrm{mod}\, 4)$, то есть величины x, y одновременно четные, тогда (1.70) принимает вид $p = x_1^2 + dy_1^2$, где $x = 2x_1$ и $y = 2y_1$.

Поскольку для любых x_1, y_1 сравнения

$$11 \equiv x_1^2 + y_1^2\ (\mathrm{mod}\, 12)\,, \quad 11 \equiv x_1^2 + 5y_1^2\ (\mathrm{mod}\, 12)$$

неразрешимы, то мы делаем вывод, что $d \not\equiv 1\ (\mathrm{mod}\, 4)$. Лемма доказана. □

2. Для оптимизации перебора может быть зафиксировано некоторое конечное множество

$$\mathcal{D} = \{d_0, d_1, \ldots\},$$

содержащее свободные от квадратов натуральные числа d такие, что h – число классов неэквивалентных приведенных квадратичных форм из $CL(-\Delta)$, где Δ определено в (1.45), удовлетворяет неравенству $h > 500$.

При проведении практических экспериментов автором было построено множество $\mathcal{D}$, состоящее из $2 \cdot 10^6$ отсортированных по возрастанию значений d, удовлетворяющих (1.71). Приведем, для информации, первые и последние элементы множества $\mathcal{D}$.

n	d	h	$d\ (\mathrm{mod}\, 12)$	n	d	h	$d\ (\mathrm{mod}\, 12)$
0	72446	504	5		...	...	
1	75206	504	5	1999995	8368991	2101	11
2	76634	514	5	1999996	8368994	4360	2
3	78014	516	5	1999997	8368999	1214	7
4	78146	512	2	1999998	8369002	1276	10
	...	...		1999999	8369003	741	11

Таблица 1.7: Множество $\mathcal{D}$ отобранных значений d.

Алгоритм 1.8: Алгоритм поиска порядка группы точек эллиптической кривой.

Вход : Натуральные числа $\alpha < \beta$.
Выход : Простое число p и параметры эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, удовлетворяющей определениям 1.6 или 1.7.

Определить минимальное нечетное число p_0, удовлетворяющее условиям

$$p_0 \equiv 11 \pmod{12}, \quad p_0 > 2^\beta.$$

Для всех $n = 1, 2, \ldots$ **выполнять**
  Определить $p = p_0 - 12n$.
  Используя тест Миллера-Рабина, см. [208], проверить, что числа p и $\frac{p-1}{2}$ являются простыми, иначе перейти к следующему индексу n.
  Для всех $d \in \mathcal{D}$ **выполнять**
    Используя алгоритм Корначчи, см. [61, гл. 2], проверить разрешимость равенства $4p = x^2 + dy^2$.
    Если $4p = x^2 + dy^2$, **то**
      Определить $m_\delta = p + 1 + \delta|x|$ для всех $\delta \in \{-1, 1\}$.
      Для всех $\delta \in \{-1, 1\}$ **выполнять**
        Используя алгоритм факторизации, отсеивающий маленькие простые делители, например [143], найти представление $m_\delta = uq$, где q – максимальный простой делитель m_δ.
        Если $2^\alpha < q < 2^\beta$ и $p^2 \not\equiv 1 \ (mod\, q)$, **то**
          Если $\frac{q-1}{2}$ – простое число, **то**
            /* найден порядок кривой, удовлетворяющей (1.6) и первому условию из (1.7). Осталось проверить порядок твиста. */
            Найти представление $m\_\delta = vr$, где r – максимальный простой делитель m_δ.
            Если $2^\alpha < r < 2^\beta$, **то**
              /* выполнено второе условие из (1.7). */
            конец
            Используя алгоритм 1.9, определить коэффициенты найденной эллиптической кривой.
          конец
          иначе
            Перейти к следующему значению δ.
          конец
        конец
        иначе
          Перейти к следующему значению δ.
        конец
      конец
    конец
  конец
конец

Алгоритм 1.9: Алгоритм построения коэффициентов эллиптической кривой

Вход : Простое число p и натуральное, свободное от квадратов число d такое уравнение $4p = x^2 + dy^2$ разрешимо в целых числах, а также натуральные числа $\alpha < \beta$.

Выход : Коэффициенты a, b эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, удовлетворяющей определениям 1.6 и 1.7.

Используя алгоритм Корначчи, см. [61, гл. 2], найти целые значения x, y такие, что $4p = x^2 + dy^2$.

Определить $m_\delta = p + 1 + \delta|x|$, где $\delta \in \{-1, 1\}$.

Выбрать $\delta \in \{-1, 1\}$ такое, что $m_\delta = uq$, q – безопасное простое и $2^\alpha < q < 2^\beta$.

Определить фундаментальный дискриминант Δ равенством (1.45).

Определить величину ω равенством (1.46).

Используя алгоритм перебора приведенных квадратичных форм из $CL(-\Delta)$, см. [61], построить многочлен $H(x) \in \mathbb{Z}[x]$ такой, что $H(j(\omega)) = 0$.

Построить список

$$\mathcal{L} = \{j_p \in \mathbb{F}_p : \ H(j_p) \equiv 0 \pmod p\},$$

содержащий все корни многочлена $H(x)$ по модулю p, отсортированные по возрастанию (количество корней в списке должно совпадать с h – числом неэквивалентных приведенных форм из $CL(-\Delta)$).

Определить $i = 0$ – порядковый номер корня j_p в списке $\mathcal{L}$.

Для всех $j_p \in \mathcal{L}$ **выполнять**

  Вычислить $i = i + 1$.

  Вычислить $k \equiv \frac{j_p}{1728 - j_p} \pmod p$.

  Если $\left(\frac{-k^{-1}}{p}\right) = -1$, **то**

    необходимо выбрать следующий корень из списка $\mathcal{L}$.

  конец

  Используя алгоритм Тонелли-Шенкса, см. [324], вычислить $c \in \mathbb{F}_p$ такой, что

$$c^2 \equiv -k^{-1} \pmod p, \quad \text{и} \quad 0 < c < p - c.$$

  Определить коэффициенты $a \equiv -3 \pmod p$ и $b \equiv -2c \pmod p$.

  Используя алгоритм Шуфа-Элкиса-Аткина, см. [222], определить порядок m построенной эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$.

  Если $m = m_\delta$, **то**

    завершить выполнение алгоритма.

  конец

  Определить $b = p - b$ и завершить выполнение алгоритма.

конец

Выше приведен алгоритм 1.8, реализующий поиск параметров с заранее выбранным множеством дискриминантов $\mathcal{D}$, а также вспомогательный алгоритм 1.9, предназначенный для нахождения коэффициентов a, b эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, а также точки P, порождающей подгруппу простого порядка q.

Дадим некоторые пояснения к приведенным алгоритмам. Несмотря на то, что для построения эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ нам достаточно

найти всего один корень многочлена $H(x)$, на седьмом шаге мы вычисляем все корни многочлена и сортируем их по возрастанию. Это делается для того, чтобы алгоритм стал детерминированным и при нескольких независимых запусках результат его работы не зависел от датчика случайных чисел, используемого в вероятностном алгоритме нахождения корней многочлена $H(x)$.

Как говорилось ранее в § 1.4.1, для любого вычета $c \in \mathbb{F}_p^*$ коэффициенты a, b эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ могут быть выражены через инвариант кривой $j(\mathcal{E}_{a,b})$ следующим образом, см. (1.49)

$$\begin{cases} a \equiv 3kc^2 \pmod p, \\ b \equiv 2kc^3 \pmod p, \end{cases} \quad \text{где} \quad k \equiv \frac{j(\mathcal{E}_{a,b})}{1728 - j(\mathcal{E}_{a,b})} \pmod p.$$

В зависимости от того, является ли величина c квадратичным вычетом или невычетом по модулю p, сравнения (1.49), согласно теореме 1.VI, дают нам либо искомую кривую, либо ее твист.

Для снижения трудоемкости реализации операции сложения двух точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ в средствах защиты информации принято использовать кривые, для которых выполнено равенство $a = -3$. Из (1.49) для таких кривых сразу следует условие

$$3kc^2 \equiv -3 \pmod p \quad \text{или} \quad \left(\frac{-k^{-1}}{p}\right) = 1. \tag{1.72}$$

Поскольку многочлен $H(x)$ имеет много корней, то мы надеемся, что найдется корень j_p, для которого условие (1.72) будет выполнено, см. двенадцатый шаг алгоритма 1.9. Как показывают эксперименты, такой корень находится всегда. С учетом (1.72) значение второго коэффициента эллиптической кривой принимает вид $b \equiv -2c \pmod p$.

Поскольку $p \equiv 11 \pmod{12}$, то для некоторого натурального n символ Лежандра удовлетворяет равенству

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = (-1)^{5+6n} = -1.$$

Следовательно, в паре вычетов $c, -c \in \mathbb{F}_p^*$ один является квадратичным вычетом, а второй невычетом по модулю p. Тогда мы можем рассмотреть две эллиптические кривые

$$y^2 \equiv x^3 - 3x \pm 2c \pmod b,$$

отличающиеся друг от друга знаком при коэффициенте b – одна из этих кривых будет той, что мы разыскиваем. Проверка этого производится прямым подсчетом порядка группы точек с применением алгоритма Шуфа-Элкиса-Аткина.

§ 1.5.3. Результаты экспериментов

При проведении практических экспериментов мы зафиксировали

$$\alpha = 254, \quad \beta = 256,$$

и искали безопасные простые числа p вида $p_n = 2^\beta - \theta_n$, $n = 1, 2, \ldots$

Для первых четырех безопасных простых чисел мы использовали стратегию неограниченного перебора величин d и нашли следующие значения.

n	θ_n	d	h
1	36113	6333787	293
1	36113	185015326	10056
2	188069	354691387	2464
3	241457	88727446	5632
4	243017	1468413670	21754

Таблица 1.8: Параметры безопасных простых чисел вида $2^{256} - \theta_n$.

В приведенной таблице величина h, как и раньше, обозначает число классов неэквивалентных приведенных квадратичных форм дискриминанта $-\Delta$, см. (1.45).

Для следующего безопасного простого числа $p_5 = 2^{256} - 315053$ было экспериментально проверено, что для всех $d \leqslant 2147387651$ параметров эллиптической кривой, удовлетворяющей требованиям определения 1.6 не существует.

Приведем расширенный, по сравнению со статьей [179], перечень из шестидесяти пяти эллиптических кривых, найденных с использованием стратегии перебора ограниченных значений d. В приводимой далее таблице предполагается, что:

- эллиптическая кривая $\mathcal{E}_{a,b}(\mathbb{F}_p)$ задана сравнением
$$y^2 \equiv x^3 - 3x - 2\varepsilon c \pmod p,$$
где $\varepsilon \in \{-1, 1\}$, а вычет $c \in \mathbb{F}_p^*$ удовлетворяет условиям
$$c^2 \equiv 1 - 1728 \cdot j_p^{-1} \pmod p, \quad 0 < c < p - c;$$
- индекс i определяет порядковый номер корня j_p в списке $\mathcal{L}$;
- величина δ определена равенствами
$$4p = x^2 + dy^2, \quad m_\delta = p + 1 + \delta|x|, \quad m_\delta = uq,$$
где q – безопасное простое, удовлетворяющее $2^\alpha < q < 2^\beta$;
- r максимальный простой делитель величины $m_{-\delta}$.

N	θ	d	δ	u	h	i	ε	$\log_2 r$
0	188069	354691387	-1	1	2464	3	-1	230
1	596057	2942326	-1	2	1072	1	-1	104
2	3308933	4711954	1	2	1024	1	-1	116
3	4909637	2767810	-1	2	748	5	-1	121
4	5210777	2166070	-1	2	676	2	1	109
5	5460857	640030	-1	2	544	1	1	139
6	7982057	4683971	-1	3	1282	1	1	133
7	12516053	2965114	-1	2	1248	1	-1	250
8	17775197	2763419	1	3	792	4	-1	183
9	20094329	5936782	-1	2	1060	7	1	188
10	21924557	2447434	1	2	904	2	-1	202
11	22686929	6494062	1	2	1044	1	-1	212
12	24386669	1904482	1	2	572	2	1	188
13	27679193	2105014	1	2	692	5	-1	242
14	30043733	1449922	1	2	648	2	1	103
15	30508457	6847462	1	2	1380	1	-1	97
16	37291877	4450978	-1	2	1056	1	1	123
17	56522129	1235854	-1	2	512	1	-1	68
18	61750517	5309194	-1	2	1020	1	1	136
19	66973697	1917790	-1	2	656	1	-1	107
20	72275273	6899806	-1	2	1768	1	1	150
21	72397793	2674102	-1	2	616	1	-1	212
22	75003869	2181898	1	2	650	4	1	107
23	76294889	5302246	1	2	1648	1	-1	131
24	86037437	2307178	1	2	630	2	1	127
25	90054089	935518	-1	2	576	2	1	147
26	94763309	6502618	-1	2	1020	2	-1	179
27	95543717	2511370	-1	2	576	1	-1	125
28	95604857	6317710	1	2	1248	1	1	197
29	99835277	6531466	-1	2	1278	2	1	102
30	101083457	1645030	-1	2	724	2	-1	95
31	102644933	1215034	-1	2	570	1	-1	132
32	103961777	6801790	-1	2	1064	2	1	170
33	104855009	2754502	-1	2	672	1	1	90
34	105256073	4675246	1	2	1156	6	1	205
35	106396757	1325146	-1	2	616	4	1	116
36	106579673	2245198	-1	2	524	4	1	115
37	107902613	3847762	1	2	1024	1	-1	216
38	111803417	2773595	-1	3	576	3	1	135
39	115564757	1329946	-1	2	790	4	-1	112
40	120449333	6456754	1	2	2440	1	-1	158
41	127419473	977566	-1	2	576	1	1	249
42	130599857	1591774	1	2	784	1	1	170
43	133942793	6810214	-1	2	1220	1	-1	175
44	134458637	1884010	1	2	672	1	-1	153
45	141432953	1770334	1	2	608	1	-1	182

46	141925217	974830	1	2	704	1	-1	90
47	143245037	2646898	1	2	520	1	1	141
48	143371757	5793539	1	3	1050	2	1	159
49	148379873	835054	-1	2	628	2	1	97
50	155701517	2711290	1	2	720	1	1	118
51	157512569	2111422	-1	2	596	1	-1	132
52	166031789	2589274	1	2	1056	4	-1	110
53	167246093	2797339	-1	1	523	1	-1	129
54	174798749	4264714	1	2	1340	1	1	152
55	175398437	1328986	-1	2	688	2	-1	150
56	181229417	6456982	1	2	1072	3	-1	142
57	182813909	2662211	-1	3	671	1	-1	91
58	185038409	1648331	-1	3	540	1	1	110
59	188247449	2594902	-1	2	502	1	1	106
60	189374657	2677739	1	3	720	1	1	172
61	191005073	6125782	-1	2	1116	3	-1	203
62	193543013	6657706	-1	2	1856	2	1	152
63	196757453	2217739	-1	1	536	1	1	184
64	199081553	3709294	1	2	1468	1	1	103

Таблица 1.10: Параметры строго безопасных эллиптических кривых.

Первая из указанных в таблице 1.10 кривых, записывается в короткой форме Вейерштрасса следующим образом

$$y^2 \equiv x^3 - 3x + \\ + 3024818943147551221418867269063791031023404613954261875826530956434811262719 9 \pmod{p},$$

где

$$p = 2^{256} - 188069 = \\ 115792089237316195423570985008687907853269984665640564039457584007913129451867.$$

В приложении А приводится полный текст программы, реализующей алгоритм 1.8 в системе компьютерной алгебры `Magma`. Там же приведены параметры построенных эллиптических кривых в виде, пригодном для применения в средствах защиты информации.

Заключение к § 1.5

В § 1.5 определены требования к параметрам эллиптических кривых, рекомендуемых к применению в средствах защиты информации.

Предложен алгоритм, позволяющий строить эллиптические кривые, удовлетворяющие предъявленным требованиям, а также приведены конкретные параметры построенных эллиптических кривых.

глава 2

Представление действительных иррациональных чисел в заданной системе счисления и генерация псевдослучайных последовательностей

В настоящей главе излагаются результаты исследований, позволяющие обосновать целесообразность применения в средствах защиты информации подхода к генерации псевдослучайных последовательностей, основанного на представлении действительных иррациональных чисел в виде систематических дробей по заданному основанию.

В первом параграфе приводится мотивация проведенных исследований, а также обзор известных результатов.

Во втором параграфе проводится выбор классов действительных иррациональных чисел, используемых в дальнейшем для генерации псевдослучайных последовательностей, а также исследуется ряд свойств рассматриваемых чисел. В третьем параграфе приводятся разработанные автором алгоритмы эффективного представления чисел из рассматриваемых классов в виде систематических дробей. Данные алгоритмы позволяют эффективно порождать последовательности псевдослучайный чисел, образованные коэффициентами систематических дробей.

В четвертом параграфе рассматриваются вопросы восстановления параметров чисел из рассматриваемых классов по известным рациональным приближениям.

В пятом параграфе доказан критерий иррациональности чисел из рассматривающих классов, позволяющий свести гипотезу о равномерном распределении элементов вырабатываемых последовательностей к проверке статистической гипотезы о равномерном распределении на интервале $[0, 1)$ величин, определяемых в ходе выработки псевдослучайных чисел. В заключение, в шестом параграфе приведен пример практического применения разработанного подхода – метод локальной аутентификации пользователей средств защиты информации.

Изложенные в настоящей главе результаты опубликованы в следующих работах автора [57, 329, 330, 332, 335, 338, 392], четыре из которых входят в перечень рецензируемых научных изданий ВАК.

§ 2.1. Мотивация и обзор известных результатов

Область применения генераторов псевдослучайных последовательностей в средствах защиты информации достаточно широка. Можно привести следующие примеры:

- псевдослучайные последовательности, вырабатываемые поточными шифрами, используются для обеспечения конфиденциальности (шифрования) информации, см. [316, 351];
- в асимметричных криптографических схемах и протоколах генераторы псевдослучайных чисел используются для выработки случайных значений, используемых при формировании электронной подписи или при шифровании с открытым ключом, см. также [161, 347];
- алгоритмы развертывания исходной ключевой информации представляют собой генераторы псевдослучайных последовательностей, используемые для увеличения мощности ключевого множества или выработки производных ключей, см. [354, 360]; данный класс алгоритмов, в частности, может быть использован для аутентификации пользователей и субъектов информационных процессов;
- методы инженерно-криптографической защиты программных и аппаратных реализаций средств защиты информации используют генераторы псевдослучайных чисел для маскирования информации, хранящейся в памяти средства защиты, или при вычислении случайного адреса хранения информации, см. [355].

Дадим формальное определение генератора псевдослучайных последовательностей.

Определение 2.1. *Пусть n, m и $b > 1$ натуральные числа, $x_1, \dots, x_m \in \mathbb{Q}$ - произвольные рациональные числа. Тогда генератором псевдослучайных последовательностей мы будем называть отображение*

$$G(n, x_1, \dots, x_m) \longrightarrow (a_1, \dots, a_n) = (a_k)_{k=1}^{n}, \quad a_1, \dots, a_n \in \mathbb{Z}_b.$$

Широкая область применения вынуждает предъявлять к генераторам псевдослучайных последовательностей большой набор требований, позволяющих обеспечить необходимый уровень безопасности защищаемой информации. Среди таких требований мы будем выделять следующие.

1. Необходимо, чтобы для любых значений входных параметров n, $x_1, \ldots, x_m$ вырабатываемая генератором последовательность $(a_k)_{k=1}^n$ представляла собой реализацию случайной величины, равномерно распределенной на интервале $[0, b-1]$, см., например, [270, 347], а также [389, глава IX, стр. 243]

   Для некоторых из используемых в средствах защиты информации генераторов псевдослучайных последовательностей сформулированное требование доказать не удается. Поэтому, на практике, оно заменяется на более слабое – требование статистической неотличимости выработанной последовательности от равномерно распределенной.

   Мы сформулируем требование статистической неотличимости следующим образом. Пусть θ фиксированное значение уровня значимости, принимающее действительные значения на интервале $(0, \frac{1}{2})$. Зафиксируем некоторое множество статистических критериев. Тогда для любой выборки $(a_k)_{k=1}^n$ не найдется статистических критерия из фиксированного ранее множества, отвергающего гипотезу о равномерном распределении элементов выборки $(a_k)_{k=1}^n$ с уровнем значимости, большим θ, см. [307, гл. 4, § 21].

   Отметим, что обязательность проверки требования статистической неотличимости, для фиксированного набора статистических критериев, накладывается на вырабатываемые генератором случайные значения при проведении динамического контроля средств криптографической защиты информации, см. [355, раздел 5.2].

2. Необходимо, чтобы задача определения любого подмножества элементов последовательности $(a_k)_{k=1}^n$ по известному другому подмножеству элементов той же последовательности имела высокую алгоритмическую сложность, см. [270]. В случае выполнимости данного требования говорят, что генератор псевдослучайных чисел обладает свойством невозможности «чтения вперед» или «чтения назад», в зависимости от того, где располагается подпоследовательность, которую необходимо определить. Частными случаями данного свойства являются:

   - высокая трудоемкость определения начальных значений генератора $x_1, \ldots, x_m$ по элементам последовательности $(a_k)_{k=1}^n$;
   - отсутствие у последовательности $(a_k)_{k=1}^n$ периода длиной τ при $\tau < n$.

В настоящее время известно достаточно большое число способов построения генераторов псевдослучайных последовательностей, используемых в средствах криптографической защиты информации. Среди таких способов стоит отметить:

- использование линейных форм над элементами конечных множеств (сюда можно отнести линейные аддитивные и линейные рекуррентные последовательности);
- использование преобразований, в основе которых лежат трудно разрешимые задачи теории чисел, такие, как задача факторизации целых чисел и задача дискретного логарифмирования;
- использование односторонних криптографических преобразований, например, функций хэширования или алгоритмов блочного шифрования;
- использование поточных алгоритмов шифрования.

Русскоязычные обзоры известных генераторов псевдослучайных последовательностей могут быть найдены в книгах [297, 303, 316, 351], также см. публикации [268, 354].

Большинство из перечисленных способов позволяют строить генераторы, удовлетворяющие только одному из двух предъявленных на предыдущей странице требований. Поэтому далее мы рассматриваем подход, основанный на представлении иррациональных чисел в виде систематических дробей по заданному основанию, и показываем, что таким способом можно строить генераторы псевдослучайных последовательностей, удовлетворяющие обоим предъявленным требованиям.

Напомним следующие известные результаты.

Определение 2.2. *Пусть $\alpha > 0$ действительное число и $b > 1$ некоторое фиксированное натуральное число. Мы будем говорить, что число α представлено в виде систематической дроби в системе счисления по основанию b, если выполнено равенство*

$$\alpha = \sum_{k=0}^{\infty} a_k b^{-k}, \quad a_k \in \mathbb{Z}, \quad \text{и} \quad 0 \leqslant a_k < b \quad \text{при} \quad k > 0. \tag{2.1}$$

Величина a_0 называется целой частью числа α, а неотрицательные числа $a_1,\ a_2, \ldots$ — коэффициентами разложения числа α.

Последовательность коэффициентов $(a_k)_{k=1}^{\infty}$, определяемая рядом (2.1), называется:

- *периодической, если найдутся такие натуральные τ и λ, что для всех $k \geqslant \lambda$ выполнено равенство $a_k = a_{k+\tau}$.*

- *конечной, если найдется такое натуральное n_0 такое, что для всех $k \geqslant n_0$ выполнено равенство $a_k = 0$.*

Конечная систематическая дробь может рассматриваться как пример периодической систематической дроби с равным единице периодом и нулевым коэффициентом a_k, лежащим на периоде. Верна следующая теорема, см., например, [210, гл. 4, § 1] или [324, п. 4.6].

Теорема 2.I. *Последовательность коэффициентов $(a_k)_{k=1}^{\infty}$, определяемая рядом (2.1), конечна или периодична тогда и только тогда, когда число α рационально.*

Из утверждения теоремы следует, что последовательность коэффициентов $(a_k)_{k=1}^{\infty}$ разложения любого действительного иррационального числа является бесконечной и не имеет периода. Рассмотрим вопрос о распределении элементов этой последовательности.

Для некоторого натурального t зафиксируем произвольные целые числа $\delta_1, \ldots, \delta_t$ такие, что $0 \leqslant \delta_i < b$ для всех $i = 1, \ \ldots, \ t$ и определим величину $N_n(\alpha, b, \delta_1, \ldots, \delta_t)$, задающую число равенств

$$(a_{k+1}, \ \ldots, \ a_{k+t}) = (\delta_1, \ \ldots, \ \delta_t), \quad \text{для} \quad k = 0, \ 1, \ \ldots, \ n-1,$$

Следуя монографиям Э. Бореля [274] и А.Г. Постникова [350] дадим следующее определение.

Определение 2.3. *Мы будем называть число α нормальным по основанию b, если при любом натуральном t и любом наборе $(\delta_1, \ \ldots, \ \delta_t)$ выполнено равенство*

$$\lim_{n\to\infty} \frac{N_n(\alpha, b, \delta_1, \ldots, \delta_t)}{n} = \frac{1}{b^t}.$$

Мы будем называть число абсолютно нормальным, если оно является нормальным по любому основанию b.

Впервые вопрос о нормальности произвольных действительных иррациональных чисел, по-видимому, был поставлен Э. Борелем, см. [44] или [274, гл. 5].

Отдельно стоит выделить случай числа π, для которого исследования начались существенно ранее. Здесь можно отметить ряд результатов – Мадхавы из Сангамаграма (1350-1425), Авраама Шарпа (1699), Джона Мечина (1706), Захария Дазе (1844), Уильяма Шенкса (1874), направленных исключительно на вычисление знаков десятичного разложения

числа π. Исторический обзор указанных вычислений можно найти в книге [295, гл. 1], см. также [205], последние результаты о вычислении мантиссы числа π можно найти в [260]. Гипотеза о нормальности числа π сформулирована в работе [15]. Результаты экспериментальных исследований с помощью вычислительных машин, направленных на подтверждение гипотезы о нормальности числа π могут быть найдены в работах [15, 138, 252, 330].

В настоящее время доказана нормальность только ряда иррациональных чисел специального вида, примеры см. в работах [55, 63, 309, 350]. Однако для произвольного иррационального числа α, определяемого рядом (2.1), не известен критерий, позволяющий определить, является ли число α нормальным или нет.

В начале 20-го века Г. Вейль, Г.Харди и Дж.Литллвуд, доказали, см. [302, обзор к § 8, гл.1], что для почти всех иррациональных чисел α последовательность[1] действительных чисел $\{\alpha b^n\}$, где $b > 1$ натуральное число, равномерно распределена на отрезке $[0, 1)$. Позднее Н.М. Коробов показал, см. [309], что из равномерного распределения элементов указанной последовательности следует равномерное распределение коэффициентов разложения числа α в системе счисления по основанию b. Верна следующая теорема.

Теорема 2.II (Коробов Н.М.)**.** *Пусть $\alpha > 0$ иррациональное число. Тогда α является нормальным числом по основанию b, если последовательность величин $\{\alpha b^n\}$, $n = 0,\ 1,\ \ldots$, является реализацией равномерно распределенной на интервале $[0, 1)$ случайной величины.*

Подробное изложение дальнейших результатов, полученных в данном направлении, может быть найдено в монографии А.Г. Постникова, см. [350].

§ 2.2. Выбор множеств действительных чисел

Для практической реализации алгоритмов выработки псевдослучайных последовательностей, нам потребуется определить множества действительных чисел, допускающих эффективное представление в виде систематической дроби для произвольного целого числа $b > 1$.

§ 2.2.1. Первое множество чисел

Дадим определение первого множества чисел.

[1] Напомним, что символом $\{\alpha\}$ мы обозначаем дробную часть числа α, то есть $\{\alpha\} = \alpha - \lfloor\alpha\rfloor$.

Пусть $b > 1$, $d > 1$ – целые числа, m – натуральное и $x_1, \ldots, x_m \in \mathbb{N}$ также натуральные, попарно различные числа, удовлетворяющие неравенствам $0 < x_k \leqslant d$ для всех $k = 1, \ldots, m$. Пусть $u_1, \ldots, u_m \in \mathbb{Q}$ — не все одновременно равные нулю рациональные числа. К первому множеству мы будем относить рассматривавшиеся в работах [14, 249] действительные числа α вида

$$\alpha = \sum_{n=0}^{\infty} \left(\frac{u_1}{dn + x_1} + \cdots + \frac{u_m}{dn + x_m} \right) b^{-n} = \sum_{n=0}^{\infty} \sum_{k=1}^{m} \frac{u_k}{dn + x_k} b^{-n}. \tag{2.2}$$

Введенные ограничения на натуральные числа $x_1, \ldots, x_m$ позволяют заключить, что $m \leqslant d$. Тогда, определяя нулями отсутствующие значения u_k, мы можем определить число (2.2) равенством

$$\alpha = \alpha\left(b, d, (u_1, \ldots, u_d)\right) = \sum_{n=0}^{\infty} \sum_{k=1}^{d} \frac{u_k}{dn + k} b^{-n}. \tag{2.3}$$

Введенное обозначение будет использовано в дальнейшем.

Иррациональность чисел вида (2.2) является следствием теоремы, доказанной Р. Тайдеманом и его соавторами в работе [251, теорема 4 и следствие 4.1], см. также [249].

Теорема 2.III. *Пусть b, m натуральные числа, а $P(n), Q(n) \in \mathbb{Q}[n]$ многочлены такие, что $\deg Q(n) = m$, $\deg P(n) < \deg Q(n)$ и $Q(n)$ имеет в точности m различных рациональных корней, принадлежащих интервалу $[-1, 0)$. Тогда действительное число*

$$\alpha = \sum_{n=0}^{\infty} \frac{P(n)}{Q(n)} b^{-n} \tag{2.4}$$

либо равно нулю, либо трансцендентно.

Заметим, что при доказательстве теоремы 2.III использовался метод, основанный на оценках линейных форм от логарифмов алгебраических чисел, см. результат А. Бейкера [17], а также результат Д. Лемера о значениях двойных сумм вида $\sum_{n=0}^{\infty} \sum_{k=1}^{m} \frac{u_k}{dn+x_k}$, см. [141]. Схожая техника используется нами далее при доказательстве теоремы 2.2.

Очевидно, что числа вида (2.2) могут быть записаны в виде (2.4). Действительно, поскольку $x_1, \ldots, x_m$ различные натуральные числа и для всех $k = 1, \ldots, m$ выполнены неравенства $-1 \leqslant -\frac{x_k}{d} < 0$, то определяя

$$Q(n) = \prod_{k=1}^{m} (dn + x_k), \quad P(n) = \sum_{k=1}^{m} u_k \prod_{i \neq k}^{m} (dn + x_i), \quad P(n), Q(n) \in \mathbb{Q}[n],$$

получим искомое представление.

Верно и обратное — каждое число вида (2.4) представимо в виде (2.2). Пусть

$$Q(t) = q_m t^m + \cdots + q_0 = q_m(t - e_1)\cdots(t - e_m),\ Q(t) \in \mathbb{Q}[t],$$

где $e_k \in \mathbb{Q}$ и $-1 \leqslant e_k < 0$ для всех $k = 1, \ldots, m$. Представим e_k в виде несократимой дроби $e_k = \frac{v'_k}{w_k}$, где $w_k > 0$ при $k = 1, \ldots, m$, и обозначим $d = \mathbf{НОК}\,(w_1,\ \ldots,\ \ w_m)$. Тогда

$$Q(t) = \frac{q_m}{d^m}\prod_{k=1}^{m}(dt + v_k),$$

где

$$v_k = -v'_k\frac{d}{w_k} = -e_k d \in \mathbb{Z}, \quad \text{и} \quad 0 < v_k \leqslant d, \quad k = 1, \ldots, m.$$

Теперь мы можем определить рациональные коэффициенты $u_1, \ldots, u_m$. Для этого запишем равенство

$$\sum_{k=1}^{m} u'_k \prod_{i\neq k}^{m}(dt + x_i) = P(t) = p_{m-1}t^{m-1} + \cdots + p_0$$

и приравняем коэффициенты при одинаковых степенях переменной t. Это даст нам систему из m линейных уравнений, любое решение которой позволит определить значения неизвестных $u'_1, \ldots, u'_m \in \mathbb{Q}$. Теперь равенство

$$\alpha = \sum_{n=0}^{\infty}\frac{P(n)}{Q(n)}b^{-n} = \sum_{n=0}^{\infty}\frac{\sum_{k=1}^{m} u'_k \prod_{i\neq k}^{m}(dn + x_i)}{q_m d^{-m}\prod_{k=1}^{m}(dn + v_k)}b^{-n} = \sum_{n=0}^{\infty}\sum_{k=1}^{m}\frac{u_k}{dn + x_k}b^{-n},$$

где $u_k = \frac{u'_k d^m}{q_m}$, очевидно.

Рассматриваемый нами класс чисел (2.2) содержит в себе большое число известных математических констант, например,

$$\pi = \alpha\,(18, 8, (4, 0, 0, -2, -1, -1, 0, 0)) = \\ = \sum_{n=0}^{\infty}\left(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6}\right)16^{-n}, \quad (2.5)$$

см. [14], или

$$\begin{aligned} 2\ln 2 &= \alpha\,(2, 1, (1))\,, \text{см. [122] и [292, ф-ла 1.513.4]}, \\ \ln 3 &= \alpha\,(4, 2, (1, 0))\,, \text{см. [13]}, \\ \arctan 2 &= \alpha\,(16, 8, (8, 0, 4, 0, -2, 0, -1, 0))\,, \text{см. [2]}, \\ \sqrt{3}\arctan\left(\tfrac{\sqrt{3}}{7}\right) &= \alpha(27, 3, (3, 1, 0)), \text{см. [13]}, \\ \tfrac{1}{\tan\phi} &= \alpha(16, 8, (8, 16, 4, 0, -2, -4, -1, 0)), \text{см. [13]}, \end{aligned}$$

где $\phi = \frac{1+\sqrt{5}}{2}$ — «золотое сечение». Более подробный перечень констант, представимых в виде (2.4), может быть найден в работе [13].

§ 2.2.2. Второе множество чисел

Дадим определение второго множества чисел.

Пусть m – натуральное число и $(x_n)_{n=0}^{\infty}$ — чисто периодическая последовательность рациональных чисел с периодом длины m такая, что существует индекс $k \geqslant 0$ для которого $x_k \neq 0$. Определим действительное число α равенством

$$\alpha = \sum_{n=0}^{\infty} \frac{x_n}{n!}. \tag{2.6}$$

Иррациональность чисел вида (2.6) в случае непериодической последовательности коэффициентов $(x_n)_{n=0}^{\infty}$ изучалась в работе [103]. Случай периодической последовательности $(x_n)_{n=0}^{\infty}$ рассматривался автором, совместно с В.Г. Чирским.

Дальнейшее изложение следует работе [392], см. также [57]. Верна следующая теорема.

Теорема 2.1. *Пусть m – натуральное число и $(x_n)_{n=0}^{\infty}$ — чисто периодическая последовательность рациональных чисел с периодом длины m такая, что существует индекс $k \geqslant 0$ для которого $x_k \neq 0$. Пусть α определено равенством (2.6) и $\alpha \neq 0$, тогда α — иррационально.*

В работе [392] для доказательства теоремы 2.1 использовался классический метод Зигеля-Шидловского в теории трансцендентных чисел. С помощью этого метода удалось получить некоторые количественные оценки, связанные с рассматриваемой задачей.

В начале потребуется провести некоторые вычисления. Поскольку последовательность $(x_k)_{k=0}^{\infty}$ чисто периодическая, то для любого индекса $k = 0, 1, 2, ...$ имеет место равенство $x_{k+m} = x_k$, тогда

$$\alpha = \sum_{k=0}^{\infty} \frac{x_k}{k!} = \sum_{k=0}^{m-1} x_k \sum_{s=0}^{\infty} \frac{1}{(k+sm)!}. \tag{2.7}$$

Используя обозначение $(x)_s = x(x+1)\cdots(x+s-1)$, определим функции комплексного переменного z

$$f_0(z) = \sum_{s=0}^{\infty} \frac{(\frac{z}{m})^{ms}}{(1)_s(\frac{1}{m})_s \ldots (\frac{m-1}{m})_s}, \tag{2.8}$$

$$f_k(z) = \sum_{s=0}^{\infty} \frac{(\frac{z}{m})^{ms}}{(1)_s(\frac{1}{m}+1)_s(\frac{k}{m}+1)_s(\frac{k+1}{m})_s \ldots (\frac{m-1}{m})_s}, \quad k = 1, ..., m-1. \tag{2.9}$$

Отметим, что введенные в (2.8) и (2.9) функции являются, так называемыми, E-функциями, см. [394, гл. 3].

При $k = 1, \ldots, m-1$ имеют место равенства

$$(k+sm)! = 1 \cdot 2 \cdot 3 \cdots (k+sm) =$$
$$1(1+m)\cdots(1+sm) \times 2(2+m)\cdots(2+sm) \times k(k+m)\cdots(k+sm) \times$$
$$\times (k+1)(k+1+m)\cdots(k+1+(s-1)m)\cdots m(2m)...(sm) =$$
$$= k! \cdot s! \cdot m^{sm} \left(\frac{1}{m}+1\right)\cdots\left(\frac{1}{m}+s\right)\cdots\left(\frac{k}{m}+1\right)\cdots\left(\frac{k}{m}+s\right) \times$$
$$\times \left(\frac{k+1}{m}\right)\cdots\left(\frac{k+1}{m}+s-1\right)\cdots\left(\frac{m-1}{m}\right)\cdots\left(\frac{m-1}{m}+s-1\right) =$$
$$= k! \cdot m^{sm} \cdot (1)_s \left(\frac{1}{m}+1\right)_s \cdots \left(\frac{k}{m}+1\right)_s \left(\frac{k+1}{m}\right)_s \cdots \left(\frac{m-1}{m}\right)_s . \quad (2.10)$$

Кроме того, при $k = 0$

$$m^{ms} \cdot (1)_s \left(\frac{1}{m}\right)_s \cdots \left(\frac{m-1}{m}\right)_s =$$
$$m^s \cdot s! \cdot m^s \left(\frac{1}{m}\right) \cdot \left(\frac{1}{m}+1\right)\cdots\left(\frac{1}{m}+s-1\right)\cdots m^s \times$$
$$\times \left(\frac{m-1}{m}\right) \cdot \left(\frac{m-1}{m}+1\right)\cdots\left(\frac{m-1}{m}+s-1\right) =$$
$$= m \cdot 2m \cdots sm \cdot 1 \cdot (m+1)\cdots(m+(s-1)m) \times$$
$$\times (m-1) \cdot (2m-1)\cdots(sm-1) = (ms)!. \quad (2.11)$$

Следовательно, согласно (2.8), (2.9), (2.10), (2.11), получаем

$$\sum_{s=0}^{\infty} \frac{1}{(k+sm)!} =$$
$$= \sum_{s=0}^{\infty} \frac{1}{k! \cdot m^{sm} \cdot (1)_s \cdot (\frac{1}{m}+1)_s \cdots (\frac{k}{m}+1)_s (\frac{k+1}{m})_s \cdots (\frac{k+1}{m})_s} =$$
$$= \frac{1}{k!} \sum_{s=0}^{\infty} \frac{(\frac{1}{m})^{ms}}{(1)_s \cdot (\frac{1}{m}+1)_s \cdots (\frac{k}{m}+1)_s (\frac{k+1}{m})_s \cdots (\frac{k+1}{m})_s} =$$
$$= \frac{1}{k!} f_k(1). \quad (2.12)$$

Ввиду (2.7), (2.12), получаем равенство

$$\alpha = \sum_{k=0}^{\infty} \frac{x_k}{k!} = \sum_{k=0}^{m-1} x_k \sum_{s=0}^{\infty} \frac{1}{(k+sm)!} = \sum_{k=0}^{m-1} \frac{x_k}{k!} f_k(1). \quad (2.13)$$

Таким образом, для доказательства теоремы 2.1 достаточно доказать линейную независимость чисел

$$1, f_0(1), f_1(1), \ldots, f_{m-1}(1) \tag{2.14}$$

над полем рациональных чисел. Действительно, если α представимо в виде несократимой дроби $\alpha = \frac{p}{q}$, то из равенства

$$-\frac{p}{q} \cdot 1 + \sum_{k=0}^{m-1} \frac{x_k}{k!} f_k(1) = 0$$

сразу следует линейная зависимость чисел (2.14).

Линейная независимость чисел (2.14) над полем рациональных чисел является следствием более общей теоремы. Для её формулировки потребуется следующее определение, см. [394].

Определение 2.4. *Пусть l, H — натуральные числа. Мерой линейной независимости чисел $\alpha_1, ..., \alpha_l$ называется функция*

$$L(\alpha_1, ..., \alpha_l, H) = \min |h_1\alpha_1 + \ldots + h_l\alpha_l|,$$

где минимум берётся по всевозможным наборам целых чисел $h_1, ..., h_l$ таким, что

1. *$|h_k| < H$ для всех $k = 1, \ldots, l$,*
2. *$|h_1| + \cdots + |h_l| > 0$.*

Теорема 2.2. *Пусть последовательность рациональных чисел $(x_k)_{k=0}^{\infty}$ является чисто периодической последовательностью с длиной периода m. Если найдется хотя бы один индекс k такой, что $x_k \neq 0$, то для любого ε такого, что $0 < \varepsilon < \frac{1}{2}$, существует эффективная постоянная $c > 0$, зависящая от чисел $x_0, \ldots, x_{m-1}$, и m, ε, такая, что*

$$L(1, f_0(1), f_1(1), \ldots, f_{m-1}(1), H) > cH^{-m-\varepsilon}. \tag{2.15}$$

Для того, чтобы применить метод Зигеля-Шидловского для доказательства теоремы 2.2 докажем сначала, что функции

$$1, f_0(z), f_1(z), \ldots, f_{m-1}(z)$$

линейно независимы над полем рациональных функций $\mathbb{C}(z)$. Это утверждение можно доказать различными способами. Использованный ниже способ, хотя и не самый короткий, даёт наиболее полное представление о структуре рассматриваемого множества функций.

Пусть $S_1 = \{f_0(z), \ldots, f_{m-1}(z)\}$, $S_2 = \{h_0(z), \ldots, h_{m-1}(z)\}$ два набора функций комплексного переменного. Мы будем говорить, что они линейно эквивалентны над полем рациональных функций $\mathbb{C}(z)$ и обозначать $S_1 \sim S_2$, если найдутся такие $a_{ij}(z) \in \mathbb{C}(z)$, что

$$h_j(z) = \sum_{i=0}^{m-1} a_{ij}(z) f_i(z), \quad j = 0, 1, \ldots, m-1,$$

и $\det |a_{ij}(z)| \neq 0$.

Лемма 2.1. *Выполнено условие эквивалентности*

$$\{f_0(z), ..., f_{m-1}(z)\} \sim \{f_0(z), ..., f_0^{(m-1)}(z)\}.$$

Доказательство. Введем оператор $\delta : \mathbb{C}(z) \to \mathbb{C}(z)$ равенством $\delta = z\frac{d}{dz}$ и докажем равенства

$$\begin{array}{rcl} \delta f_0(z) & = & \frac{z^m}{(m-1)!} f_{m-1}(z), \\ (\delta + k) f_k(z) & = & k f_{k-1}(z), \quad k = 1, \ldots, m-1. \end{array} \tag{2.16}$$

Действительно,

$$\begin{aligned} (\delta + k) f_k(z) = \\ = \left(z\frac{d}{dz} + k\right) \left(\sum_{s=0}^{\infty} \frac{(\frac{z}{m})^{ms}}{(1)_s (\frac{1}{m}+1)_s (\frac{k}{m}+1)_s (\frac{k+1}{m})_s \cdots (\frac{m-1}{m})_s}\right) = \\ = \sum_{s=0}^{\infty} \frac{(ms+k)(\frac{z}{m})^{ms}}{(1)_s (\frac{1}{m}+1)_s (\frac{k}{m}+1)_s (\frac{k+1}{m})_s \cdots (\frac{m-1}{m})_s}. \end{aligned}$$

Далее, поскольку выполнено равенство $(\frac{k}{m}+1)_s = (\frac{k}{m}+1)\cdots(\frac{k}{m}+s)$, то

$$\begin{aligned} \frac{(ms+k)}{(\frac{k}{m}+1)_s} = \frac{m(\frac{k}{m}+s)}{(\frac{k}{m}+1)\cdots(\frac{k}{m}+s)} = \\ = \frac{m}{(\frac{k}{m}+1)\cdots(\frac{k}{m}+s-1)} = \frac{k}{\frac{k}{m}\cdots(\frac{k}{m}+s-1)} = \frac{k}{(\frac{k}{m})_s}. \end{aligned}$$

Таким образом,

$$\begin{aligned} (\delta + k) f_k(z) = \\ = k \left(\sum_{s=0}^{\infty} \frac{(\frac{z}{m})^{ms}}{(1)_s (\frac{1}{m}+1)_s (\frac{k-1}{m}+1)_s (\frac{k}{m})_s \cdots (\frac{m-1}{m})_s}\right) = \\ = k f_{k-1}(z), \quad k = 1, \ldots, m-1. \end{aligned}$$

Так как

$$f_0(z)=\sum_{s=0}^{\infty}\frac{(\frac{z}{m})^{ms}}{(1)_s(\frac{1}{m})_s\cdots(\frac{m-1}{m})_s}=1+\sum_{s=1}^{\infty}\frac{(\frac{z}{m})^{ms}}{(1)_s(\frac{1}{m})_s\ldots(\frac{m-1}{m})_s},$$

выполнено равенство

$$\delta f_0(z)=z\frac{d}{dz}\left(1+\sum_{s=1}^{\infty}\frac{(\frac{z}{m})^{ms}}{(1)_s(\frac{1}{m})_s\cdots(\frac{m-1}{m})_s}\right)=\sum_{s=1}^{\infty}\frac{ms(\frac{z}{m})^{ms}}{(1)_s(\frac{1}{m})_s\cdots(\frac{m-1}{m})_s}.$$

Для любого $s\geq 1$ имеем $(1)_s=s!$, поэтому $\frac{s}{(1)_s}=\frac{1}{(1)_{s-1}}$ и

$$\sum_{s=1}^{\infty}\frac{ms(\frac{z}{m})^{ms}}{(1)_s(\frac{1}{m})_s\cdots(\frac{m-1}{m})_s}=m\sum_{s=1}^{\infty}\frac{(\frac{z}{m})^{ms}}{(1)_{s-1}(\frac{1}{m})_s\cdots(\frac{m-1}{m})_s}=$$
$$=m\sum_{r=0}^{\infty}\frac{(\frac{z}{m})^{m(r+1)}}{(1)_r(\frac{1}{m})_{r+1}\cdots(\frac{m-1}{m})_{r+1}}.$$

По определению, $(a)_{r+1}=a(a+1)\cdots(a+r+1)=a(a+1)_r$, следовательно,

$$\left(\frac{1}{m}\right)_{r+1}\cdots\left(\frac{m-1}{m}\right)_{r+1}=$$
$$=\frac{1}{m}\left(\left(\frac{1}{m}\right)+1\right)_r\cdots\frac{m-1}{m}\left(\left(\frac{m-1}{m}\right)+1\right)_r=$$
$$=\frac{(m-1)!}{m^{m-1}}\left(\left(\frac{1}{m}\right)+1\right)_r\cdots\left(\left(\frac{m-1}{m}\right)+1\right)_r.$$

Используем это равенство и получим

$$\delta f_0(z)=m\sum_{r=0}^{\infty}\frac{(\frac{z}{m})^{m(r+1)}}{(1)_r(\frac{1}{m})_{r+1}\cdots(\frac{m-1}{m})_{r+1}}=$$
$$=m\left(\frac{z}{m}\right)^m\cdot\frac{m^{m-1}}{(m-1)!}f_{m-1}(z)=\frac{z^m}{(m-1)!}f_{m-1}(z).$$

Таким образом, равенства (2.16) доказаны, а из них следует утверждение леммы:

$$\{f_0(z),\ldots,f_{m-1}(z)\}\sim\{f_0(z),\delta f_0(z),\ldots,\delta^{m-1}f_0(z)\}\sim\{f_0(z),\ldots,f_0^{(m-1)}(z)\}.$$

□

Следствие 2.2.А. *Функции* $1, f_0(z), \ldots, f_{m-1}(z)$ *составляют решение системы дифференциальных уравнений вида*

$$y_i'(z) = Q_i(z) + \sum_{j=0}^{m-1} Q_{i,j}(z) y_j(z), \quad Q_i(z), Q_{i,j}(z) \in \mathbb{C}(z), \quad i,j = 0, \ldots, m-1.$$

Докажем еще несколько вспомогательных утверждений.

Лемма 2.2. *Выполнено равенство*

$$f_0(z) = \frac{1}{m} \sum_{j=0}^{m-1} \exp(\zeta^j z), \quad \textit{где} \quad \zeta = \exp\left(\frac{2\pi i}{m}\right).$$

Доказательство. Из равенств (2.8) и (2.11) следует, что

$$f_0(z) = \sum_{s=0}^{\infty} \frac{(\frac{z}{m})^{ms}}{(1)_s (\frac{1}{m})_s \cdots (\frac{m-1}{m})_s} = \sum_{s=0}^{\infty} \frac{z^{ms}}{m^{ms} (1)_s (\frac{1}{m})_s \cdots (\frac{m-1}{m})_s} = \sum_{s=0}^{\infty} \frac{z^{ms}}{(ms)!}.$$

Далее,

$$\frac{1}{m} \sum_{j=0}^{m-1} \exp(\zeta^j z) = \frac{1}{m} \sum_{j=0}^{m-1} \sum_{r=0}^{\infty} \frac{\zeta^{jr} z^r}{r!} = \frac{1}{m} \sum_{r=0}^{\infty} \left(\sum_{j=0}^{m-1} \zeta^{jr} \right) \frac{z^r}{r!}.$$

Так как $\zeta = \exp(\frac{2\pi i}{m})$, то при m, делящемся на r, сумма $\sum_{j=0}^{m-1} \zeta^{jr}$ равна m. Если же m не делится на r, то $\sum_{j=0}^{m-1} \zeta^{jr} = \frac{1-\zeta^{jm}}{1-\zeta^j} = 0$. Таким образом,

$$\frac{1}{m} \sum_{j=0}^{m-1} \exp(\zeta^j z) = \frac{1}{m} \sum_{s=0}^{\infty} \frac{m z^{ms}}{(ms)!} = f_0(z).$$

Лемма доказана. □

Лемма 2.3. *Имеет место линейная эквивалентность*

$$\{f_0(z), ..., f_{m-1}(z)\} \sim \{\exp(\zeta^j z), \quad j = 0, 1, \ldots, \ m-1\}.$$

Доказательство. По лемме 2.2,

$$\begin{aligned} f_0'(z) &= \frac{1}{m} \textstyle\sum_{j=0}^{m-1} \zeta^j \exp(\zeta^j z), \\ f_0''(z) &= \frac{1}{m} \textstyle\sum_{j=0}^{m-1} \zeta^{2j} \exp(\zeta^j z), \\ &\cdots \\ f_0^{(m-1)}(z) &= \frac{1}{m} \textstyle\sum_{j=0}^{m-1} \zeta^{(m-1)j} \exp(\zeta^j z). \end{aligned}$$

Следовательно, матрица перехода от системы $\{f_0(z), ..., f_{m-1}(z)\}$ к системе $\{\exp(\zeta^j z), \quad j = 0, 1, \ldots, m-1\}$ — невырождена, поскольку ее определитель равен

$$\det \begin{vmatrix} 1 & 1 & ... & 1 \\ 1 & \zeta & ... & \zeta^{m-1} \\ 1 & \zeta^2 & ... & \zeta^{2(m-1)} \\ ... & ... & ... & ... \\ 1 & \zeta^{m-1} & ... & \zeta^{(m-1)^2} \end{vmatrix},$$

т.е. представляет собой отличный от нуля определитель Вандермонда. □

Последнее необходимое нам утверждение является хорошо известным, см., например, [349, док-во теоремы 4, § 7].

Лемма 2.4. *Если $\alpha_1, \ldots, \alpha_l$ — различные комплексные числа, то функции $\exp(\alpha_1 z), \ldots, \exp(\alpha_l z)$ линейно независимы над полем рациональных функций $\mathbb{C}(z)$.*

Доказанные нами леммы позволяют получить доказательство теоремы 2.2 как простое следствие следующей теоремы, см. [394, гл. 11, теорема 1].

Теорема 2.IV. *Пусть $l \geqslant 2$ натуральное число. Пусть совокупность E-функций $f_1(z), \ldots, f_l(z) \in \mathbb{C}(z)$ линейно независима над полем $\mathbb{C}(z)$ и удовлетворяет системе линейных дифференциальных уравнений*

$$y_i'(z) = Q_{i,0}(z) + \sum_{j=1}^{l} Q_{i,j}(z) y_j(z), \quad i = 1, \ldots, l, \quad Q_{i,j} \in \mathbb{C}(z).$$

Обозначим $T(z)$ — многочлен, являющийся наименьшим общим кратным знаменателей рациональных функций $Q_{i,j}$ для всех возможных значений индексов i, j.

Пусть ξ — алгебраическое число такое, что $\xi T(\xi) \neq 0$, тогда для любого действительного числа $0 < \varepsilon < \frac{1}{2}$ существует постоянная c, зависящая от функций $f_1(z), \ldots, f_l(z)$, а также чисел l, ξ и ε, такая, что выполнено неравенство

$$L(f_1(\xi), \ldots, f_l(\xi), H) > cH^{1-l-\varepsilon}.$$

Лемма 2.4, вместе с доказанными леммами 2.1, 2.3, дает нам утверждение о линейной независимости над полем $\mathbb{C}(z)$ функций $1, f_0(z), \ldots, f_{m-1}(z)$. По следствию из леммы 2.1 эти функции удовлетворяют системе дифференциальных уравнений с коэффициентами из $\mathbb{C}(z)$. Следовательно, функции $1, f_0(z), \ldots, f_{m-1}(z)$ удовлетворяют условию теоремы 2.IV при $\xi = 1$ и теорема 2.2 верна.

Отметим, что эффективность постоянной c следует из доказанной в работе [33] эффективности постоянной n_0, см. также [394, стр. 106].

Заключение к § 2.2

В § 2.2 введены в рассмотрение два класса действительных иррациональных чисел, определяемых соотношениями (2.2) и (2.6). Кроме того:

- доказана теорема 2.1, из которой следует иррациональность чисел вида (2.6);
- получено представление чисел вида (2.6) в виде конечной суммы (2.13) значений функций, задаваемых равенствами (2.8) и (2.9); данное представление позволило получить оценку меры линейной независимости значений функций, задаваемых равенствами (2.8) и (2.9),

Определенные в настоящем параграфе классы чисел будут использованы для генерации псевдослучайных последовательностей. Более того, представление (2.13) будет использовано, см. § 2.4, для оценки возможности восстановления неизвестных коэффициентов $(x_k)_{k=0}^{m-1}$ иррационального числа вида (2.6) по его представлению в виде систематической дроби.

§ 2.3. Эффективные алгоритмы разложения

В данном параграфе описываются разработанные автором алгоритмы, позволяющие эффективно представлять рассмотренные выше иррациональные числа вида (2.2) и (2.6) в виде систематической дроби.

Рассмотрим иррациональное число α, задаваемое быстро сходящимся рядом

$$\alpha = \sum_{n=0}^{\infty} \omega_n \tag{2.17}$$

таким, что $\omega_n \in \mathbb{Q}$ и существует индекс $n_0 \in \mathbb{N}$ такой, что для любого индекса $n \geqslant n_0$ будет выполнено неравенство

$$0 < |\omega_n| < f(n)b^{-n}, \quad 0 < f(n) < 1, \quad \lim_{n\to\infty} f(n) = 0,$$

для некоторой функции $f(n)$ натурального аргумента n.

Пусть $b > 1$ натуральное число, определяющее основание системы счисления. Мы хотим представить в этой системе счисления действительное число α вида (2.17), т.е. записать его в виде равенства (2.1)

$$\alpha = \sum_{n=0}^{\infty} a_n b^{-n} = [a_0; a_1, a_2, \ldots]_b, \tag{2.18}$$

где $a_0 \in \mathbb{Z}$ и $0 \leqslant a_n < b$ для всех $n = 1, 2, \ldots$.

Следует отметить, что схожая задача решалась в работе [301]. Однако результаты практической реализации на ЭВМ изложенного в [301] метода автору, в настоящее время, не известны.

Предложенные автором алгоритмы используют идеи, схожие с идеями, высказанными в работах [14], [15]. Дальнейшее изложение этого параграфа следует работе [338].

Для начала покажем, что числа из рассматриваемых нами множеств могут быть записаны в виде (2.17). Рассмотрим числа вида (2.2), для которых выполнено равенство

$$\omega_n = \sum_{k=1}^{m} \frac{u_k}{dn + x_k} b^{-n} = \frac{P(n)}{Q(n)} b^{-n}, \quad \text{и} \quad f(n) = \left| \frac{P(n)}{Q(n)} \right|.$$

Поскольку для всех $n = 0, 1, \ldots$ выполнено неравенство

$$\deg P(n) < \deg Q(n)$$

мы заключаем, что $\lim_{n\to\infty} f(n) = 0$. Определим константу n_0 равенством

$$n_0 = \max \left\{ \lceil |u_k| \rceil \in \mathbb{N}, \quad k = 1, \ldots, m \right\}. \tag{2.19}$$

Тогда, вспоминая, что $0 < x_k \leqslant d$ и $m \leqslant d$, для всех $n \geqslant n_0$, получаем строгое неравенство

$$0 < f(n) \leqslant \sum_{k=1}^{m} \frac{|u_k|}{dn + x_k} < \sum_{k=1}^{m} \frac{|u_k|}{dn} \leqslant \sum_{k=1}^{m} \frac{1}{d} = \frac{m}{d} \leqslant 1.$$

Таким образом мы показали, что числа вида (2.2) также являются числами вида (2.17).

Теперь рассмотрим числа вида (2.6). Для них выполнено равенство

$$\omega_n = \frac{x_n}{n!}, \quad n = 0, 1, \ldots.$$

Положим $x = \max\{\frac{1}{\lfloor be \rfloor}, |x_0|, \ldots, |x_{m-1}|\}$, $x \in \mathbb{Q}$, $x > 0$ и определим

$$c_0 = \max\{c \in \mathbb{N} : \ bex^{\frac{1}{c}} > c\} \tag{2.20}$$

где e — основание натурального логарифма и

$$n_0 = \max \left\{ \left\lceil bex^{\frac{1}{c_0}} \right\rceil, \ \left\lceil \frac{b}{\sqrt{2\pi c_0}} - 1 \right\rceil \right\}. \tag{2.21}$$

Теперь воспользуемся формулой Стирлинга для величины $n!$, см. [390, Глава 11, § 7], и получим, для некоторого действительного $0 < \theta < 1$ и любого $n \geqslant n_0$, неравенство

$$n! b^{-(n+1)} = \frac{\sqrt{2\pi n} n^n e^{\frac{\theta}{12n}}}{b(be)^n} > \frac{\sqrt{2\pi n}}{b} \left(\frac{n}{be} \right)^n. \tag{2.22}$$

Поскольку $n_0 \geqslant bex^{\frac{1}{c_0}}$, то

$$\frac{n}{be} \geqslant \frac{n_0}{be} \geqslant x^{\frac{1}{c_0}}$$

и

$$\frac{\sqrt{2\pi n}}{b}\left(\frac{n}{be}\right)^n \geqslant \frac{\sqrt{2\pi bex^{\frac{1}{c_0}}}}{b}\left(x^{\frac{1}{c_0}}\right)^{bex^{\frac{1}{c_0}}} \geqslant \frac{\sqrt{2\pi c_0}}{b}x. \tag{2.23}$$

Теперь, для всех индексов $n > n_0$, с учетом неравенств (2.22), (2.23), можно записать

$$\omega_{n+1} = \frac{x_{n+1}}{(n+1)} \leqslant \frac{x}{(n+1)n!} = \frac{\frac{\sqrt{2\pi c_0}}{b}x}{\frac{\sqrt{2\pi c_0}}{b}(n+1)n!} \leqslant$$
$$\leqslant \frac{n!b^{-(n+1)}}{\frac{\sqrt{2\pi c_0}}{b}(n+1)n!} = \frac{b}{\sqrt{2\pi c_0}(n+1)}b^{-(n+1)} = f(n+1)b^{-(n+1)},$$

где $f(n) = \frac{b}{n\sqrt{2\pi c_0}}$.

Из условия (2.21) получаем неравенство $0 < f(n+1) < 1$, из которого следует, что числа вида (2.6) также представимы в виде быстро сходящегося ряда (2.17).

§ 2.3.1. Элементарный алгоритм представления чисел в виде систематической дроби

Мы начнем с того, что опишем элементарный алгоритм представления действительного числа $\alpha = \sum_{n=0}^{\infty}\omega_n$ вида (2.17) в виде систематической дроби по основанию $b > 1$. Данный алгоритм является обобщением алгоритма, предложенного автором в работе [330], см. также [15].

Определим начальное значение $\delta_{-1} = 0$, последовательность рациональных величин

$$\alpha_n = b\delta_{n-1} + \omega_n b^n, \quad a_n = \lfloor \alpha_n \rfloor, \quad \delta_n = \alpha_n - a_n, \quad n = 0,\ 1,\ \ldots, \tag{2.24}$$

где $\delta_n, \omega_n, \alpha_n \in \mathbb{Q}$, $\delta_{-1}, a_n \in \mathbb{Z}$, а также частичную сумму

$$s_k(\alpha) = \sum_{n=0}^{k} a_n b^{-n}, \quad s_k \in \mathbb{Q}, \quad k = 0,\ 1,\ \ldots. \tag{2.25}$$

Заметим, что для чисел вида (2.2) равенства (2.24) принимают совсем простой вид, а именно,

$$\alpha_n = b\delta_{n-1} + \frac{P(n)}{Q(n)}, \quad a_n = \lfloor \alpha_n \rfloor, \quad \delta_n = \alpha_n - a_n, \quad n = 0,\ 1,\ \ldots.$$

Верна следующая лемма, см. [338].

Лемма 2.5. *Для введенных в (2.24) коэффициентов a_n, $n = 0,\ 1,\ \ldots$, выполнено равенство $\alpha = \sum_{n=0}^{\infty} a_n b^{-n}$. Более того,*

$$\alpha - s_k(\alpha) = b^{-k}\left(\delta_k + b^k \sum_{n=1}^{\infty} \omega_{k+n}\right). \tag{2.26}$$

Доказательство. Зафиксируем произвольный индекс $n > 0$, тогда из (2.24) следует равенство

$$\frac{\alpha_n}{b^n} = \frac{\alpha_{n-1} - a_{n-1}}{b^{n-1}} + \omega_n \quad \text{или} \quad \frac{\alpha_{n-1}}{b^{n-1}} - \frac{\alpha_n}{b^n} = \frac{a_{n-1}}{b^{n-1}} - \omega_n$$

Просуммируем полученные равенства для всех $n = 1,\ 2,\ \ldots$ и получим

$$\alpha_0 = \sum_{n=0}^{\infty} a_n b^{-n} - \sum_{n=1}^{\infty} \omega_n.$$

Поскольку $\alpha_0 = \omega_0$ и ряд $\sum_{n=1}^{\infty} \omega_n$ сходится, получим искомое равенство.

Для доказательства второго утверждения леммы заметим, что из равенства $\delta_n = \alpha_n - a_n$ следует равенство $\delta_n - b\delta_{n-1} = \omega_n b^n - a_n$. Пользуясь этим, находим

$$\begin{aligned}
\alpha - s_k(\alpha) &= \sum_{n=0}^{k}(\omega_n - a_n b^{-n}) + \sum_{n=k+1}^{\infty} \omega_n = \\
&= \sum_{n=0}^{k}(\omega_n b^n - a_n)b^{-n} + \sum_{n=k+1}^{\infty} \omega_n = \sum_{n=0}^{k}(\delta_n b^{-n} - \delta_{n-1} b^{-n+1}) + \sum_{n=k+1}^{\infty} \omega_n = \\
&= b^{-k}\left(\delta_k + b^k \sum_{n=1}^{\infty} \omega_{k+n}\right).
\end{aligned}$$

Лемма доказана. □

Отметим, что равенства (2.24) не дают нам окончательное представление числа α в виде (2.1). Действительно, пусть n_0 индекс такой, что для всех $n \geqslant n_0$ выполнено неравенство $0 < |\omega_n| < b^{-n}$. Тогда, учитывая, что величины δ_n принимают значения в интервале $0 \leqslant \delta_n < 1$ и неравенства

$$-1 < \omega_n b^n \leqslant \alpha_n = \delta_{n-1} b + \omega_n b^n < b + 1, \tag{2.27}$$

получаем, что величины $a_n = \lfloor \alpha_n \rfloor$, вычисленные в соответствии с (2.24), удовлетворяют неравенствам $-1 \leqslant a_n \leqslant b$ при $n \geqslant n_0$.

Рассмотрим два граничных случая, в которых не выполнено необходимое неравенство $0 \leqslant a_n < b$ при $n \geqslant n_0$.

1. Значение $a_n = b$ означает, что в представлении (2.1) возникает равенство

$$\cdots + a_{n-1}b^{-n+1} + b \cdot b^{-n} + \cdots = \cdots + (a_{n-1}+1)b^{-n+1} + 0 \cdot b^{-n} + \cdots ,$$

то есть возникает необходимость изменить значение коэффициентов a_{n-1}, a_n и сделать замену $a_{n-1} = a_{n-1} + 1$, $a_n = 0$. Очевидно, что если новое значение коэффициента $a_{n-1} \geqslant b$, то процедуру переноса единицы нужно повторить для индекса $n-1$.

2. Значение $a_n = -1$ означает, что в представлении (2.1) возникает равенство

$$\cdots + a_{n-1}b^{-n+1} + (-1)\cdot b^{-n} + \cdots = \cdots + (a_{n-1}-1)b^{-n+1} + (b-1)\cdot b^{-n} + \cdots ,$$

то есть возникает необходимость изменить значение коэффициентов a_{n-1}, a_n и сделать замену $a_{n-1} = a_{n-1} - 1$, $a_n = b - 1$.

Поскольку

$$-1 = -1 \cdot b + (b-1), \quad 0 \leqslant b-1 < b,$$

то мы можем объединить два случая в один на основе следующего условия. Если

$$a_n = qb + r, \quad q \in \{-1, 1\}, \quad r \in \{b-1, 0\},$$

то $a_{n-1} = a_{n-1} + q$, $a_n = r$.

Отметим, что для значений индекса n, меньших n_0, значения коэффициентов $a_1, \ldots, a_{n_0}$ могут принимать произвольные значения. Приведем пример — первые десять знаков разложения числа e в системе счисления по основанию $b = 16$, вычисленные с помощью соотношений (2.24), имеют вид[2]

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = [\mathtt{1};\, \mathtt{10},\, \mathtt{80},\, \mathtt{2aa},\, \mathtt{ab5},\, \mathtt{2227},\, \mathtt{5b0d},\, \mathtt{d00f},\, \mathtt{1a021},\, \mathtt{2e3ca},\, \mathtt{49f9c}, \ldots]_{16}. \qquad (2.28)$$

Теперь, используя равенства

$$\begin{aligned} \mathtt{49f9c} &= \mathtt{10} \cdot \mathtt{49f9} + \mathtt{c}, \\ \mathtt{2e3ca} + \mathtt{49f9} &= \mathtt{10} \cdot \mathtt{32dc} + \mathtt{3}, \\ \mathtt{1a021} + \mathtt{32dc} &= \mathtt{10} \cdot \mathtt{1d2f} + \mathtt{d}, \\ \mathtt{d00f} + \mathtt{1d2f} &= \mathtt{10} \cdot \mathtt{ed3} + \mathtt{e}, \end{aligned}$$

[2]Здесь и далее, запись производится в шестнадцатеричной системе счисления, в которой числа от 0 до 15 записываются символами $\mathtt{0}, \mathtt{1}, \ldots, \mathtt{9}, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}$.

$$\begin{aligned}
\mathtt{5b0d} + \mathtt{ed3} &= \mathtt{10} \cdot \mathtt{69e} + \mathtt{0},\\
\mathtt{2227} + \mathtt{69e} &= \mathtt{10} \cdot \mathtt{28c} + \mathtt{5},\\
\mathtt{ab5} + \mathtt{28c} &= \mathtt{10} \cdot \mathtt{d4} + \mathtt{1},\\
\mathtt{2aa} + \mathtt{d4} &= \mathtt{10} \cdot \mathtt{37} + \mathtt{e},\\
\mathtt{80} + \mathtt{37} &= \mathtt{10} \cdot \mathtt{b} + \mathtt{7},\\
\mathtt{10} + \mathtt{b} &= \mathtt{10} \cdot \mathtt{1} + \mathtt{b},\\
\mathtt{1} + \mathtt{1} &= \mathtt{10} \cdot \mathtt{0} + \mathtt{2}
\end{aligned}$$

и представление (2.28), мы можем записать приближение для числа e в виде

$$e \sim [\mathtt{2}; \mathtt{b}, \mathtt{7}, \mathtt{e}, \mathtt{1}, \mathtt{5}, \mathtt{0}, \mathtt{e}, \mathtt{d}, \mathtt{3}, \mathtt{c}, \ldots]_{16}.$$

В тоже время, более точное представление числа e в шестнадцатеричной системе счисления имеет вид

$$e = [\mathtt{2}; \mathtt{b}, \mathtt{7}, \mathtt{e}, \mathtt{1}, \mathtt{5}, \mathtt{1}, \mathtt{6}, \mathtt{2}, \mathtt{8}, \mathtt{a}, \mathtt{e}, \mathtt{d}, \mathtt{2}, \mathtt{a}, \mathtt{6}, \mathtt{a}, \mathtt{b}, \mathtt{f}, \mathtt{7}, \mathtt{1}, \mathtt{5}, \mathtt{8}, \mathtt{8}, \ldots]_{16}.$$

Из сказанного следует, что для получения точного равенства (2.1) может потребоваться корректировка значений, полученных с помощью соотношений (2.24). При этом, для индексов $n \geqslant n_0$ корректировка значений потребует переноса единицы, а для индексов $0 < n < n_0$, как в приведенном примере, может потребоваться перенос больших значений.

В общем случае, если величина $a_n = qb + r \geqslant 0$, где $q \in \mathbb{Z}$ и $0 \leqslant r < b$, то необходимо перенести величину q в разряд с индексом, меньшим на единицу, то есть сделать замену $a_n = r$, $a_{n-1} = a_{n-1} + q$. После чего повторить эту же процедуру для индекса $n - 1$.

Таким образом, если для некоторого заданного значения $k \geqslant 1$ требуется вычислить частичную сумму $s_k(\alpha)$ или, другими словами, точные значения коэффициентов $a_0, \ldots, a_k$, то с помощью соотношений (2.24) необходимо вычислить значения коэффициентов $a_0, a_1, \ldots, a_h$, где индекс h удовлетворяет условиям

$$a_h < b - 1, \quad h \geqslant \max\{k + 1, n_0\}. \tag{2.29}$$

Условие $a_h < b - 1$ означает, что даже в случае переноса из $h + 1$ разряда в h-й разряд, переноса в $(h - 1)$-й разряд не произойдет[3].

Далее надо провести процедуру коррекции в соответствии с описанными выше правилами для всех коэффициентов $a_{h-1}, a_{h-2}, \ldots, a_0$. Сформулируем сказанное выше в виде алгоритма.

[3]Отметим, что в приведенном выше представлении (2.28) для числа e количество вычисленных коэффициентов не удовлетворяло условию (2.29).

Алгоритм 2.1: Алгоритм представления иррационального числа в заданной системе счисления.

Вход : $\alpha = \sum_{n=0}^{\infty} \omega_n$, где $\omega_n \in \mathbb{Q}$, а также $b, k \in \mathbb{N}$.
Выход : $\alpha = [a_0; a_1, \ldots, a_k]_b$, где $a_0 \in \mathbb{Z}$ и $0 \leqslant a_n < b$, $n = 1, \ldots, k$.

Используя (2.19) или (2.20) определить величину индекса n_0 такого, что для любого $n \geqslant n_0$ выполнено $0 < |\omega_n| < b^{-n}$.
Определить индекс $h = \max\{k+1, n_0\}$ и $\delta_{-1} = 0$.
/ Вычисляем последовательность коэффициентов достаточной длины */*
Для всех $n = 0, \ldots, h$ **выполнять**
    Вычислить $\alpha_n = b\delta_{n-1} + \omega_n b^n$.
    Определить $a_n = \lfloor \alpha_n \rfloor$ и $\delta_n = \alpha_n - a_n$.
конец
Пока $a_h \geqslant b - 1$ **выполнять**
    Вычислить $h = h + 1$ и $\alpha_h = b\delta_{h-1} + \omega_h b^h$.
    Определить $a_h = \lfloor \alpha_h \rfloor$ и $\delta_h = \alpha_h - a_h$.
конец
/ Начинаем процедуру коррекции вычисленных значений */*
Для всех $n = h-1, h-2, \ldots, 1$ **выполнять**
    Если $a_n < 0$ или $a_n \geqslant b$ **то**
        Вычислить q, r такие, что $q_n = qb + r$, $q \in \mathbb{Z}$, $0 \leqslant r < b$.
        Определить $a_n = r$, $a_{n-1} = a_{n-1} + q$.
    конец
конец

Все операции в алгоритме 2.1 производятся только с целыми числами, представляющими числители и знаменатели чисел α_n, δ_n и ω_n. Разработанная автором программа позволила вычислить ряд представлений чисел вида (2.2) в шестнадцатеричной системе счисления, в частности, представление числа π, заданного равенством (2.5).

Приведем некоторые из вычисленных коэффициентов числа π.

n	$a_{n+1},\ \ldots,\ a_{n+70}$
1000	49f1c09b075372c980991b7b25d479d8f6e8def7e3fe501ab6794c3b976ce0bd04c006
5000	cad181156b2395e0333e92e13b240b62eebeb92285b2a20ee6ba0d99de720c8c2da2f7
10000	8ac8fcfb8016cbdb8bbc1f476982c71185c7da7a58811477cd67fad1d764d9b4c81029
20000	e937c2354a69ff6665b0fdc5e27f9505fdec5b5707a41cb556226e58f0342f9de47b29
30000	06835924037620630f26652b5091303be43ffb88858122103205ada9f0d3640a228071
40000	1bb9156d760228b4201a0d18c3496472259447a2b72cde08c5e7acb0def7f9b4f36e62
50000	940c2140010f05cd2cd44341f02ea221fcbbcd2ca6af2b780d8b66778177a89e785ca8
60000	eb9f9a6a0991f3e036a4cb413ed76c5b8a82571a3bc95bfd75628cf031a7be002e87fb
70000	ee89e2df509021bda17981995ef49937c0bf2956fe895d22ca9d45a34182a1da118729
80000	8c4fd06beb8b28c7927d5b4a2759b9775abad7ba0c71ace4e702bf3b674d8db1d76205
90000	a49069a85006380285b1643ca6317e51cf4e753f0462502bb1ec82eeea0913fd99ad34
100000	35ea16c406363a30bf0b2e693992b58f7205a7232c4168840b6a48ecb67eaa2a5b9d3c
200000	4c96d097b2f16e01a0ae8f3a25a49c2d6154cf6b745fd8b07a4cf7080251518e49279d
300000	27678c8fdb2eacf6fb2e117a4a19fc7bb94ac6d7a175a9b0e094375fed07b1ba63fa10
400000	3c3eb81ef29e15d79beea3a2adf67cdbdd056c07c1feba909c36a83c7ebca1cd5da7e6

500000	dd637c0a02dd73a323b5fa6d02042029a758fe035ef96ba180f08be656aa1017dcaacc
600000	a3eee097149a8db6b5dcf68749ac4bf3b0a51dc9d94dd7f14866e5dc9988f4e09b9a3c
700000	3d044dab65b83375c7a33cab96edcf5ff6716a1e7760f104c60b485cbfa330131dc146
800000	a0d95ad0687805d5894fc7440a652a6ad9727c85e0356c669ffe0b4804556dc59d966c
900000	0b64622852fda1b8523cf8bac7e117446cc693b725abe7d59da6c9526016ca2f3448a7
1000000	6c65e52cb459350050e4bb178f4c67a0fcf7bf27206290fbe70f93b828cd939c475c72

Для проверки корректности разработанной программы приведенные выше значения сравнивались со значениями, вычисленными с помощью других формул, см. [23, 260].

Несмотря на простоту реализации на ЭВМ, алгоритм 2.1 обладает рядом особенностей:

- алгоритм не допускает параллельной реализации,
- если мы хотим получить k знаков в представлении числа α то для проведения коррекции нам потребуется хранение в памяти всех элементов последовательности $a_0, \ldots, a_h$, где h удовлетворяет (2.29),
- алгоритм использует очень большой объем памяти.

Для оценки объема используемой памяти еще раз рассмотрим равенства (2.24)

$$\alpha_n = b\delta_{n-1} + \omega_n b^n, \quad a_n = \lfloor \alpha_n \rfloor, \quad \delta_n = b\delta_{n-1} + \omega_n b^n - a_n, \quad n = 0, 1, \ldots.$$

и получим асимптотические оценки роста знаменателей величин α_n при $n \geqslant n_0$. Обозначим $\omega_n = \frac{p_n}{q_n}$, где $q_n > 0$ и

$$\begin{cases} p_n = P(n), & q_n = Q(n)b^n, \quad \text{для чисел вида (2.2)}, \\ p_n = x_n, & q_n = n!, \quad \text{для чисел вида (2.6)}. \end{cases} \tag{2.30}$$

Тогда величина δ_n принимает следующие значения

$$\begin{aligned} \delta_{-1} &= 0, \\ \delta_0 &= \frac{p_0}{q_0} - a_0 = \frac{p_0 - a_0 q_0}{q_0} = \frac{l_0(b)}{q_0}, \\ \delta_1 &= \frac{bl_0(b)}{q_0} + \frac{p_1 b}{q_1} - a_1 = \frac{bq_1 l_0(b) + p_1 q_0 b - a_1 q_0 q_1}{q_0 q_1} = \frac{l_1(b)}{q_0 q_1}, \\ &\ldots \\ \delta_n &= \frac{bl_{n-1}(b)}{q_0 \cdots q_{n-1}} + \frac{p_n b^n}{q_n} - a_n = \frac{bq_n l_{n-1}(b) + p_n q_0 \cdots q_{n-1} b - a_1 q_0 \cdots q_n}{q_0 \cdots q_n} = \frac{l_n(b)}{q_0 \cdots q_n}, \end{aligned}$$

для некоторых многочленов $l_n(b) \in \mathbb{Z}[b]$, $\deg l_n(b) = n$, при $n = 0, 1, \ldots$. Поскольку величина δ_n удовлетворяет неравенствам $0 \leqslant \delta_n < 1$, а величины q_n определены равенствами (2.30), то выполнено неравенство

$$0 \leqslant l_n(b) < q_0 \cdots q_n, \quad n = 1, 2, \ldots.$$

Теперь, с учетом неравенства (2.27), мы можем записать

$$0 \leqslant a_n = \lfloor \alpha_n \rfloor = \left\lfloor \frac{bl_{n-1}(b)}{q_0 \cdots q_{n-1}} + \frac{p_n b^n}{q_n} \right\rfloor = \left\lfloor \frac{f_n(b)}{q_0 \cdots q_n} \right\rfloor \leqslant b,$$

для некоторого многочлена $f_n(b) \in \mathbb{Z}[b]$ и $\deg f_n(b) = n$. Таким образом оценка величины $q_0 \cdots q_n$ дает нам оценку объема используемой при вычислениях памяти.

Для чисел вида (2.2) обозначим

$$Q(n) = \theta_m n^m + \theta_{n-1} n^{m-1} + \cdots + \theta_0 \in \mathbb{Q}[n]$$

и

$$\theta = \max\{|\theta_m|, |\theta_{m-1}|, \cdots, |\theta_0|\}.$$

Тогда

$$\begin{aligned} Q(0) &= \theta_0 \leqslant \theta, \\ Q(1) &= \textstyle\sum_{k=0}^{m} \theta_k \leqslant \sum_{k=0}^{m} |\theta_k| \leqslant (m+1)\theta, \end{aligned}$$

и, для всех $n = 2, 3, \ldots$, выполнено

$$Q(n) \leqslant \sum_{k=0}^{m} |\theta_k| n^k \leqslant \theta \left(1 + n + n^2 + \cdots + n^m\right) =$$

$$= n^m \theta \left(1 + \frac{1}{n} + \cdots + \frac{1}{n^m}\right) < n^m \theta \sum_{k=0}^{\infty} \frac{1}{n^k} = \frac{n^{m+1}\theta}{n-1}.$$

Теперь мы можем записать неравенство

$$q_0 \cdots q_n = Q(0) \cdot Q(1) \cdots Q(n) \cdot b \cdot b^2 \cdots b^{n-1} \cdot b^n \leqslant$$

$$\leqslant \theta \cdot (m+1)\theta \cdot \left(\prod_{k=2}^{n} \frac{k^{m+1}\theta}{(k-1)} \right) \cdot b^{\frac{n(n+1)}{2}} =$$

$$= (m+1)\theta^{m+1} n (n!)^m \cdot b^{\frac{n(n+1)}{2}}.$$

Следовательно, для фиксированного многочлена $Q(n)$, выполнено асимптотическое равенство

$$q_0 \cdots q_n = O\left(n(n!)^m b^{\frac{n(n+1)}{2}}\right), \quad \text{при} \quad n \to \infty. \tag{2.31}$$

Для чисел вида (2.6), из условий (2.30), следует точное равенство

$$q_0 \cdots q_n = \prod_{k=0}^{n} k!. \tag{2.32}$$

Отметим, что полученные оценки (2.31) и (2.32) являются оценками сверху и не учитывают тот факт, что величины p_n, q_n могут иметь нетривиальный общий делитель. В этом случае, после сокращения, рост величин q_n происходит существенно медленнее, а его характер зависит от конкретных значений многочленов $P(n), Q(n)$ или последовательности значений $x_0, \ldots, x_{m-1}$.

Для снижения объема памяти, используемой при вычислениях на ЭВМ, можно использовать вариации алгоритма 2.1, описываемые далее.

§ 2.3.2. Модификации элементарного алгоритма

При проведении вычислений мы можем использовать тот факт, что рассматриваемые нами числа α вида (2.17) могут быть представлены в виде линейной формы с рациональными коэффициентами от действительных чисел, также представимых в виде (2.17). Это позволяет свести вычисление частичной суммы для исходного числа α к операциям с частичными суммами входящих в α слагаемых. Нам потребуется следующая лемма, см. [338].

Лемма 2.6. *Пусть $k, b > 1, m \geqslant 1$ — натуральные числа, $u_1, \ldots, u_m$ — произвольные рациональные числа и $\xi_1, \ldots, \xi_m$ — положительные действительные числа, определенные рядом (2.17). Пусть символ* $\operatorname{sign}(u)$ *обозначает знак действительного числа u.*

Обозначим

$$\begin{aligned} \mu &= \lceil \log_b m \rceil, \\ l &= \max\{0, \lceil \log_b |u_1| \rceil, \ldots, \lceil \log_b |u_m| \rceil\}, \\ z &= \max\{0, \lceil \log_b |\xi_1| \rceil, \ldots, \lceil \log_b |\xi_m| \rceil\}. \end{aligned}$$

Тогда, для величины $\alpha = \sum_{i=1}^{m} u_i \xi_i$ выполнено неравенство

$$\left| \alpha - \sum_{i=1}^{m} \operatorname{sign}(u_i) s_{k+3+z+\mu}(|u_i|) s_{k+3+l+\mu}(\xi_i) \right| < b^{-k},$$

где частичные суммы s_k определены равенством (2.25).

Доказательство. Для любого положительного действительного числа ξ запишем

$$\xi = \sum_{n=0}^{\infty} a_n b^{-n}, \quad s_{k+1}(\xi) = \sum_{n=0}^{k+2} a_n b^{-n}, \quad \varepsilon_\xi = \xi - s_{k+1}(\xi),$$

где $a_0 \in \mathbb{N}$, $0 \leqslant a_n < b$ для всех $n = 1, 2, \ldots$. Тогда верно

$$0 \leqslant \xi - s_{k+1}(\xi) = \sum_{n=k+2}^{\infty} a_n b^{-n} = \frac{1}{b^{k+2}} \left(a_{k+2} + \frac{a_{k+3}}{b} + \frac{a_{k+4}}{b^2} \cdots \right) <$$

$$< \frac{b}{b^{k+2}} \sum_{n=0}^{\infty} b^{-n} = \frac{1}{b^k (b-1)} \leqslant b^{-k}$$

и из условия $b > 1$, получаем неравенство

$$0 \leqslant \xi - s_{k+1}(\xi) = \varepsilon_\xi < b^{-k}. \tag{2.33}$$

Теперь, учитывая, что величины $\xi_i > 0$ для всех $i = 1, \ldots, m$, и неравенства

$$s_{k+1}(\xi_i) < \xi_i \leqslant b^z, \quad s_{k+1}(|u_i|) < |u_i| \leqslant b^l,$$

из (2.33) имеем

$$\left| \alpha - \sum_{i=1}^{m} \operatorname{sign}(u_i) s_{k+3+z+\mu}(|u_i|) s_{k+3+l+\mu}(\xi_i) \right| =$$

$$= \left| \sum_{i=1}^{m} \operatorname{sign}(u_i) \left(|u_i| \xi_i - s_{k+1+z}(|u_i|) s_{k+1+l}(\xi_i) \right) \right| =$$

$$= \left| \sum_{i=1}^{m} \operatorname{sign}(u_i) \left(s_{k+3+l+\mu}(\xi_i) \varepsilon_{u_i} + s_{k+3+z+\mu}(|u_i|) \varepsilon_{\xi_i} + \varepsilon_{u_i} \varepsilon_{\xi_i} \right) \right| \leqslant$$

$$\leqslant \sum_{i=1}^{m} \left(s_{(k+2+l+\mu)+1}(\xi_i) \varepsilon_{u_i} + s_{(k+2+z+\mu)+1}(|u_i|) \varepsilon_{\xi_i} + \varepsilon_{u_i} \varepsilon_{\xi_i} \right) <$$

$$< m \left(\frac{b^z}{b^{k+2+z+\mu}} + \frac{b^l}{b^{k+2+l+\mu}} + \frac{1}{b^{2(k+1)+l+z+2\mu}} \right) \leqslant$$

$$\leqslant b^\mu \cdot b^{-(k+\mu)} \cdot \frac{1}{b^2} \left(2 + \frac{1}{b^{k+\mu+l+z}} \right) \leqslant \frac{3}{4} b^{-k} < b^{-k}.$$

□

Доказанная лемма позволяет свести вычисление $s_k(\alpha)$ к элементарным операциям сложения и умножения для частичных сумм действительных чисел ξ_i, через которые выражается число α. Лемма 2.6 также позволяет оценить количество слагаемых в частичных суммах, необходимых для вычисления величины α с заданной точностью.

В утверждении леммы 2.6 возникают приближения $s_k(u)$ для рациональных чисел u. Рассмотрим способ вычисления указанных приближений более подробно.

§ 2.3.2.1. Представление рациональных чисел

Мы начнем с детального описания алгоритма представления рациональных чисел в заданной системе счисления, то есть алгоритма вычисления равенства

$$u = \sum_{n=0}^{\infty} u_n b^{-n} = [u_0; u_1, \ldots]_b,$$

где $u_0 \in \mathbb{Z}$ и $0 \leqslant u_n < b$ для всех $n = 1, 2, \ldots$.

В случае, когда u является целым числом, то очевидно выполнено равенство $u = [u_0]_b$. Далее, если $u < 0$, то выполнено равенство

$$-u = -[v_0; v_1, \ldots,]_b, \quad \text{где} \quad |u| = \sum_{n=0}^{\infty} v_n b^{-n}.$$

Тогда , из равенств

$$0 = u + (-u) = \sum_{n=0}^{\infty} (u_n - v_n) b^n$$

получаем, что $u_n = b - u_n$, $v_{n-1} = v_{n-1} - 1$.

Таким образом, нам достаточно рассмотреть случай рационального, не являющегося целым числа $u \in \mathbb{Q}$ такого, что $u > 0$. В этом случае выполнено равенство

$$u = \lfloor u \rfloor + \frac{P}{Q}, \quad 0 < P < Q, \quad \textbf{НОД}\,(P,\ Q) = 1. \tag{2.34}$$

Пусть l минимальное целое неотрицательное число такое, что $u < b^l$, тогда мы легко можем определить коэффициенты $u_{1-l}, \ldots, u_{-1}, u_0$ такие, что

$$\lfloor u \rfloor = \sum_{n=1-l}^{0} u_l b^{-n}, \quad 0 \leqslant u_n < b, \quad n = 1 - l, \ldots, -1, 0.$$

Таким образом, для представления числа u в системе счисления по основанию b нам надо определить коэффициенты $u_1, u_2, \ldots$, удовлетворяющие условию

$$\frac{P}{Q} = \sum_{n=1}^{\infty} u_n b^{-n}, \quad 0 \leqslant u_n < b, \quad n = 1, 2, \ldots.$$

Хорошо известно, см. теорему 2.I, что последовательность коэффициентов $u_1, u_2, \ldots$ периодична. Для определения периода, который мы будем обозначать символом τ, можно воспользоваться следующими рассуждениями.

Пусть основание системы счисления b представимо в виде

$$b = p_1^{\beta_1} \cdots p_s^{\beta_s}.$$

Отметим, что в практических приложениях, как правило, выполнено равенство $s = 1$ и величина b принимает вид $b = 2^\beta$. Для всех $i = 1, \ldots, s$ определим величины

$$\alpha_i = \nu_{p_i}(Q), \quad \lambda_i = \left\lceil \frac{\alpha_i}{\beta_i} \right\rceil, \quad \lambda = \mathbf{НОК}\left(\lambda_1, \ \ldots, \ \lambda_s\right), \tag{2.35}$$

где $\nu_p(Q)$ — максимальная степень, в которой p в точности делит Q. Тогда выполнено $\lambda_i\beta_i \geqslant \alpha_i$ и

$$\frac{P}{Q} = \frac{P}{q\prod_{i=1}^s p_i^{\alpha_i}} = \frac{P\prod_{i=1}^s p_i^{\frac{\lambda(\lambda_i\beta_i-\alpha_i)}{\lambda_i}}}{b^\lambda q} = \frac{P'}{b^\lambda q}, \quad \text{где} \quad \mathbf{НОД}\left(b, \ q\right) = 1.$$

Если выполнено равенство $q = 1$, то искомое представление найдено. Действительно, поскольку $1 < \frac{P}{Q}$, то $P' < b^\lambda$ и мы можем записать равенство $P' = u_\lambda + u_{\lambda-1}b + \cdots + u_1 b^{\lambda-1}$. Тогда

$$\frac{P}{Q} = \frac{P'}{b^\lambda} = \sum_{n=1}^{\lambda} u_n b^{-n}.$$

При этом, представление числа $\frac{P}{Q}$ в системе счисления по основанию b конечно и содержит в точности λ коэффициентов.

Далее будем считать, что $q > 1$. Определим величины t, r равенствами

$$t = \left\lfloor \frac{P'}{q} \right\rfloor, \quad r = P' - tq,$$

из которых следуют неравенства $0 \leqslant r < q$. Используя введенные обозначения, запишем равенство

$$\frac{P}{Q} = \frac{P'}{b^\lambda q} = b^{-\lambda}\left(t + \frac{r}{q}\right).$$

При этом, из неравенств

$$1 > \frac{P}{Q} = \frac{1}{b^\lambda} \cdot \frac{P'}{q} \geqslant b^{-\lambda}t$$

следует, что $t < b^\lambda$.

Обозначим $\tau = \mathtt{ord}_q\, b$ минимальное натуральное число такое, что $b^\tau \equiv 1 \pmod q$. При этом будет выполнено равенство $b^\tau = 1 + zq$ для некоторого натурального числа z. Тогда

$$\frac{r}{q} = \frac{zr}{b^\tau - 1} = zr \sum_{n=1}^{\infty} b^{-n\tau},$$

где $zr < zq = b^\tau - 1 < b^\tau$. Последнее равенство позволяет нам окончательно записать (2.34) в виде

$$u = \lfloor u \rfloor + tb^{-\lambda} + \frac{zr}{b^\lambda} \sum_{n=1}^{\infty} b^{-n\tau} = \sum_{n=1-l}^{\infty} u_n b^{-n}, \tag{2.36}$$

где

$$\begin{aligned} \lfloor u \rfloor &= u_{1-l} b^{l-1} + \cdots + u_{-1} b + u_0, \\ t &= u_1 b^{\lambda-1} + \cdots + u_{\lambda-1} b + u_\lambda, \\ zr &= u_{\lambda+1} b^{\tau-1} + \cdots + u_{\lambda+\tau-1} b + u_{\lambda+\tau}, \end{aligned}$$

и $0 \leqslant u_n < b$ для всех $n = 1 - l, \ldots, 0, 1, \ldots$.

Легко видеть, что коэффициенты $u_{\lambda+1}, \ldots, u_{\lambda+\tau}$ образуют период разложения числа α, коэффициенты $u_{1-l}, \ldots, w_\lambda$ — подход к периоду, а число u может быть записано в виде

$$u = [u_{1-l}, \ldots, u_0; u_1, \ldots, u_\lambda, \overline{u_{\lambda+1}, \ldots, u_{\lambda+\tau}}]_b.$$

В случае произвольного знаменателя q нахождение периода τ может оказаться сложной задачей. Действительно, из определения τ следует, что $\tau | \varphi(q)$, где φ — функция Эйлера, и поиск τ сводится к перебору делителей числа $\varphi(q)$, то есть к разложению числа $\varphi(q)$ на простые сомножители.

Использование равенства (2.36) целесообразно только в случае, когда величина $\varphi(q)$ невелика и может быть легко разложена на простые сомножители, а величина τ не превосходит количества коэффициентов в частичной сумме $s_k(u)$, т.е. $\tau \leqslant k$.

В качестве примера рассмотрим дробь

$$u_n = \frac{P_n}{Q_n} = \frac{1}{8n+5},$$

возникающую в формуле для числа π, см. равенство (2.5).

Легко видеть, что для любого $n \geqslant 0$ знаменатель Q_n есть нечетное число, поэтому для $b = 16$, период τ_n представления числа u_n удовлетворяет сравнению

$$16^{\tau_n} \equiv 1 \pmod{8n+5}.$$

Предположим, что мы хотим найти приближение $s_{1025}(u_n)$, которое позволит определить значение u_n с точностью до 16^{-1024}. Легко рассчитать, что для всех $n = 0, \ldots, 515$ выполнено неравенство $\tau_n < 1025$.

n	Q_n	τ_n
0	5	1
1	13	3
2	21	3
	$\ldots$	
511	4093	1023
512	4101	683
513	4109	879
514	4117	979
515	4125	25
516	4133	1033

Следовательно, для всех $0 \leqslant n \leqslant 515$ использование равенства (2.36) позволит вычислить менее 1025 необходимых коэффициентов частичной суммы $s_{1025}(u_n)$.

В случае, когда длина периода представления рационального числа u превышает количество членов частичной суммы $s_k(u)$, для вычисления $s_k(u)$ можно воспользоваться следующими простыми соотношениями.

Пусть $u = \dfrac{P}{Q}$ и $0 < u < 1$. Определим начальное значение $\delta_1 = u$ и последовательность коэффициентов $u_1, u_2, \ldots$ равенствами

$$u_n = \lfloor \delta_n b \rfloor, \quad \delta_{n+1} = b\delta_n - u_n, \quad n = 1, 2, \ldots. \tag{2.37}$$

Поскольку для всех $n = 1, 2, \ldots$, в силу определения, выполнены неравенства $0 \leqslant \delta_n < 1$, то

$$0 \leqslant u_n = \lfloor b\delta_n \rfloor < b.$$

Теперь, из (2.37), для любого $k \geqslant 1$ имеем

$$u = \frac{P}{Q} = \frac{1}{b}\left(b\delta_1\right) = \frac{1}{b}\left(u_1 + \delta_2\right) = \frac{1}{b}\left(u_1 + \frac{1}{b}(b\delta_2)\right) =$$

$$= \frac{1}{b}\left(u_1 + \frac{1}{b}\left(u_2 + \delta_3\right)\right) = \sum_{n=1}^{k} u_n b^{-n} + \frac{\delta_{k+1}}{b^k} = s_k(u) + \frac{\delta_{k+1}}{b^k}.$$

Поскольку $0 \leqslant \delta_{k+1} < 1$, то

$$u - s_k(u) = \frac{\delta_{k+1}}{b^k} < b^{-k}. \tag{2.38}$$

Сравнивая полученную оценку с утверждением леммы 2.6, легко видеть, для рациональных чисел частичная сумма $s_k(u)$ дает приближение с точностью b^{-k} на меньшем числе слагаемых.

§ 2.3.2.2. Алгоритм для первого множества чисел

Теперь мы можем применить полученные ранее результаты к представлению в системе счисления по основанию $b > 1$ чисел вида (2.2). Верно следующее равенство

$$\alpha = \sum_{i=1}^{m} u_i \xi_i, \quad \text{где } u_i \in \mathbb{Q}, \quad \xi_i = \sum_{n=0}^{\infty} \frac{b^{-n}}{dn + x_i} \in \mathbb{R}.$$

При этом числа ξ_i также имеют вид (2.2) для всех $i = 1, \ldots, m$.

Запишем число ξ в виде

$$\xi = \sum_{n=0}^{\infty} \frac{b^{-n}}{(dn + x)} = \sum_{n=0}^{\infty} R_n b^{-n}, \quad x \in \mathbb{N}, \quad R_n \in \mathbb{Q},$$

то есть

$$R_0 = \frac{1}{x}, \quad R_1 = \frac{1}{d + x}, \quad R_2 = \frac{1}{2d + x}, \quad \ldots$$

Поскольку выполнено неравенство $0 < x \leqslant d$, то $R_n < 1$ для всех индексов $n = 0, 1, \ldots$.

Теперь воспользуемся одним из описанных нами ранее, см. § 2.3.2.1, методов и запишем равенство

$$R_n = \sum_{i=1}^{\infty} r_i^{(n)} b^{-i} = [0; r_1^{(n)}, \ldots, r_{\lambda_n}^{(n)}, \overline{r_{\lambda_n+1}^{(n)}, \ldots, r_{\lambda_n+\tau_n}^{(n)}}],$$

где $0 \leqslant r_j^{(n)} < b$, $\lambda_n, \tau_n \in \mathbb{N}$, для всех возможных значений индексов n. Тогда

$$\xi = \sum_{n=0}^{\infty} R_n b^{-n} = \sum_{n=0}^{\infty} \left(\sum_{i=1}^{\infty} r_i^{(n)} b^{-i} \right) b^{-n} = \sum_{n=1}^{\infty} \left(\sum_{i=0}^{n-1} r_{n-i}^{(i)} \right) b^{-n}. \tag{2.39}$$

С другой стороны, обозначим $\delta = \lceil \log_b(k+1) \rceil$. Теперь, пользуясь (2.38) и вспоминая, что $d(k+1) > b > 1$, получаем

$$\xi - \sum_{n=0}^{k} s_{k+2+\delta-n}(R_n) b^{-n} =$$

$$= \sum_{n=0}^{k} \left(R_n - s_{k+2+\delta-n}(R_n) \right) b^{-n} + \sum_{n=k+1}^{\infty} R_n b^{-n} < \tag{2.40}$$

$$< \sum_{n=0}^{k} b^{-(k+2+\delta)} + R_{k+1} b^{-(k+1)} \sum_{n=0}^{\infty} b^{-n} <$$

$$< \frac{k+1}{b^{k+2+\delta}} + \frac{b}{d(k+1)(b-1)b^{k+1}} \leqslant$$

$$\leqslant b^{-(k+1)} \left(\frac{1}{b} + \frac{1}{(b-1)} \right) < \frac{3}{2} b^{-(k+1)} < b^{-k}.$$

Последнее неравенство дает нам выражение, позволяющее вычислить приближение к числу ξ с заданной точностью b^{-k}. Заметим, что при этом с ростом индекса n количество членов в частичной сумме $s_{k+2+\delta-n}(R_n)$ убывает. Теперь мы можем сформулировать алгоритм, основывающийся на неравенстве (2.40) и позволяющий вычислять значение ξ с заданной точностью.

Алгоритм 2.2: Алгоритм представления иррационального числа вида $\sum_{n=0}^{\infty} R_n b^{-n}$ в заданной системе счисления

Вход : $\xi = \sum_{n=0}^{\infty} R_n b^{-n}$, где $R_n = (dn+x)^{-1}$, а также $b, d, x, k \in \mathbb{N}$ такие, что $d(k+1) > b$.

Выход : $s_k(\xi) = [0; c_1, \ldots, c_k]_b$, где $0 \leqslant c_i < b$, $i = 1, \ldots, k$.

1 Определить $\delta = \lceil \log_b(k+1) \rceil$ и $c_i = 0$ для всех $i = 0, \ldots, k+2+\delta$.
2 **Для всех** $n = 0, \ldots, k$ **выполнять**
3 Вычисляем $s_{k+2+\delta-n}(R_n) = \sum_{i=1}^{k+\delta-n} r_i^{(n)} b^{-i}$. Для этого используем либо представление (2.36), либо соотношения (2.37).
4 **Для всех** $i = k+\delta-n, \ldots, 1$ **выполнять**
5 Определить $c_{i+n} = c_{i+n} + r_i^{(n)}$.
6 **конец**
7 /* *Редукция значений* c_i */
8 Определить $\sigma = 0$.
9 **Для всех** $i = k+2+\delta, \ldots, 1$ **выполнять**
10 Вычислить значения q, r такие, что $0 \leqslant r < b$ и $c_i + \sigma = qb + r$.
11 Определить $\sigma = q$ и $c_i = r$.
12 **конец**
13 Определить $c_0 = \sigma$.
14 **конец**
15 Вернуть в качестве результата вектор $s_k(\xi) = [0; c_1, \ldots, c_k]_b$.

Алгоритм 2.2 находит приближение к числу $\xi = \sum_{n=0}^{\infty} R_n b^{-n}$ с заданной точностью k путем последовательного вычисления приближений к рациональным числам R_n с точностью $k+2+\delta-n$, где δ определено в первой строке алгоритма.

Такое сведение позволяет существенно уменьшить объем используемой памяти. Действительно, в случае, когда период разложения числа R_n в системе счисления по основанию b не превосходит величины k для получения значений лежащих на периоде коэффициентов $r_i^{(n)}$ мы исполь-

зуем представление (2.36) (как мы показали ранее, см. § 2.3.2.1, эта ситуация возникает при небольших значениях индекса n). В случае, когда $\mathtt{ord}_{(dn+x)}\, b > k$, нам остается последовательно применять равенства (2.37). Однако для больших значений n нам необходимо вычислить только $k+2+\delta-n$ коэффициентов разложения $r_i^{(n)}$, $i = 1, \ldots, k+2+\delta-n$ (в частности, при $n = k$, только $2+\delta$ коэффициентов), что также приводит к медленному росту знаменателей величин δ_n, участвующих в равенстве (2.37). Объем памяти, используемой для хранения промежуточных значений $c_1, \ldots, c_{k+2+\delta}$, не слишком сильно отличается от величины k, поскольку $\delta = \lceil \log_b(k+1) \rceil$.

В качестве последнего замечания отметим, что в алгоритме 2.2 третий шаг — шаг вычисления представления рационального числа R_n в системе счисления по основанию b, может быть вычислен параллельно для различных значений индекса n.

§ 2.3.2.3. Алгоритм для второго множества чисел

Соображения, аналогичные высказанным выше, могут быть использованы для представления в системе счисления по основанию b иррациональных чисел вида (2.6). Действительно, согласно (2.13), выполнено равенство

$$\alpha = \sum_{k=0}^{\infty} \frac{x_k}{k!} = \sum_{i=0}^{m-1} x_i \xi_i(m), \tag{2.41}$$

где m период последовательности рациональных чисел $(x_n)_{n=0}^{\infty}$, $x_0, \ldots, x_{m-1}$ числа, лежащие на периоде и

$$\xi_i(m) = \sum_{n=0}^{\infty} \frac{1}{(nm+i)!}, \quad i = 0, \ldots, m-1.$$

Легко видеть, что числа $\xi_i(m)$ также имеют вид (2.17) для всех возможных значений индекса i. Следовательно, равенство (2.41) также позволяет применить для вычисления числа α утверждение леммы 2.6.

Рассмотрим числа $\xi_i(m)$ более детально. Для них выполнено следующее свойство. При $m = 1$ выполнены равенства

$$\xi_0(1) = \sum_{n=0}^{\infty} \frac{1}{n!} = e, \quad \alpha = x_0 e,$$

где e основание натурального логарифма.

При $m > 1$ выполнено

$$e=\sum_{n=0}^{\infty}\frac{1}{n!}=$$
$$=\frac{1}{0!}+\frac{1}{1!}+\frac{1}{2!}+\cdots+\frac{1}{m!}+\frac{1}{(m+1)!}+\cdots+\frac{1}{(2m)!}+\frac{1}{(2m+1)!}+\cdots=$$
$$+\frac{1}{0!}+\frac{1}{m!}+\frac{1}{(2m)!}+\cdots+$$
$$+\frac{1}{1!}+\frac{1}{(m+1)!}+\frac{1}{(2m+1)!}+\cdots+$$
$$+\frac{1}{2!}+\frac{1}{(m+2)!}+\frac{1}{(2m+2)!}+\cdots=$$
$$=\sum_{n=0}^{\infty}\frac{1}{(nm)!}+\sum_{n=0}^{\infty}\frac{1}{(nm+1)!}+\sum_{n=0}^{\infty}\frac{1}{(nm+2)!}+\cdots=$$
$$=\xi_0(m)+\xi_1(m)+\xi_2(m)+\cdots+\xi_{m-1}(m),$$

или

$$\xi_0(m)+\cdots+\xi_{m-1}(m)=e.$$

Вернемся к вычислению $\xi_i(m)$ с заданной точностью b^{-k}, предполагая, что k достаточно велико и выполнено неравенство $\frac{1}{k!}<b^{-k}$. Определим минимальное натуральное число k_0 такое, что $k_0m+i\geqslant k$, тогда k_0 определяет число слагаемых, необходимых для вычисления $\xi_i(m)$. Действительно, воспользовавшись неравенством

$$((k_0+s)m+i)!=(k_0m+i)!\times$$
$$\times(k_0m+i+1)\cdots(k_0m+i+m)\times\cdots$$
$$\cdots\times(k_0m+i+(s-1)m+1)\cdots(k_0m+i+sm)>$$
$$>(k_0m+i)!(k_0m+i)^{sm}=k!k^{sm},$$

получаем

$$\xi_i(m)-\sum_{n=0}^{k_0}\frac{1}{(nm+i)!}=\sum_{n=k_0+1}^{\infty}\frac{1}{(nm+i)!}=$$
$$<\frac{1}{k!}\left(\frac{1}{k^m}+\frac{1}{k^{2m}}+\cdots\right)=\frac{1}{k!(k^m-1)}<b^{-k}.$$

Обозначим $Q_n=(nm+i)!$, тогда

$$Q_{n+1}=Q_n\times(nm+i+1)\times\cdots\times(mn+i+m)$$

и $Q_n|Q_{n+1}$ для всех $n=0,\ldots,k_0$. Условие делимости позволяет нам использовать только равенство (2.36) для вычисления частичной суммы к

рациональному числу $\frac{1}{Q_n}$. Если обозначить $\tau_n = \mathtt{ord}_{Q_n} b$, то из свойств показателей следует, что $\tau_n | \tau_{n+1} = \mathtt{ord}_{Q_{n+1}} b$ и

$$\tau_{n+1} = s\tau_n, \quad \text{где} \quad s \Big| \prod_{j=1}^{m} \varphi(nm+i+j), \quad n = 0, \ldots, k_0. \tag{2.42}$$

Следовательно, определив начальное значение $\tau_0 = \mathtt{ord}_{i!} b$, остальные значения мы можем находить последовательно применяя условие (2.42).

И последнее. Если необходимо представлять в виде систематической дроби по основанию b сразу несколько чисел α вида (2.6), то величины $\xi_0(m), \ldots, \xi_{m-1}(m)$ могут быть определены заранее, поскольку их значения не зависят от последовательностей коэффициентов $\{x_n\}_{n=0}^{\infty}$, определяющих α.

Заключение к § 2.3

В § 2.3 показано, что действительные иррациональные числа, определяемые равенствами (2.2) и (2.6), могут быть представлены в виде быстро сходящихся рядов (2.17). Рассмотрен ряд вопросов, относящихся к эффективному вычислению на ЭВМ систематических дробей рассматриваемых классов чисел:

- в первом разделе параграфа доказана лемма 2.5, на основе которой предложен элементарный алгоритм представления чисел вида (2.17) в виде систематической дроби по произвольному основанию $b > 1$; получены оценки сверху на объем памяти, необходимой для практической реализации предложенного алгоритма на ЭВМ;

- во втором разделе предложены две модификации элементарного алгоритма, позволяющие снизить объем используемой при вычислениях памяти, а также допускающие параллельную реализацию на ЭВМ (в основу предложенных модификаций положено представление действительных иррациональных чисел, определяемых равенствами (2.2) и (2.6), в виде линейных форм с рациональными коэффициентами).

§ 2.4. Методы определения элементов последовательности

В § 2.1 на стр. 138 было сформулировано требование высокой трудоемкости определения любого подмножества последовательности $(a_k)_{k=1}^n$ по заданному другому подмножеству элементов последовательности. В данном параграфе мы исследуем частные случаи сформулированного требования, в частности, исследуем сложность восстановления коэффициентов $(x_k)_{k=1}^m$, определяющих число α, по известной последовательности коэффициентов разложении в систематическую дробь.

§ 2.4.1. Восстановление неизвестных коэффициентов иррационального числа

Задача о восстановлении начального состояния генератора псевдослучайных чисел является хорошо известной, см., например [297, 316, 351]. Вместе с тем, применительно к систематическим дробям действительных иррациональных чисел данная задача, по видимому, впервые рассматривалась автором в [332]. Дальнейшее изложение настоящего раздела следует указанной работе.

Пусть известно, что α – действительное иррациональное число, определяемое равенством

$$\alpha = \sum_{n=0}^{\infty} \left(\frac{u_1}{(dn+x_1)^s} + \cdots + \frac{u_m}{(dn+x_m)^s} \right) b^{-n} = \sum_{n=0}^{\infty} \sum_{k=1}^{m} \frac{u_k}{(dn+x_k)^s} b^{-n} \quad (2.43)$$

так, что при $s = 1$ число α удовлетворяет равенству (2.2).

Пусть для некоторого натурального числа r задано рациональное приближение к числу α

$$s_r(\alpha) = \sum_{n=0}^{r} a_n b^{-n}, \quad \text{где} \quad |\alpha - s_r(\alpha)| < b^{-r} \quad \text{и} \quad s_r(\alpha) \in \mathbb{Q}.$$

Рассматриваемая задача заключается в определении неизвестных натуральных, попарно различных величин $x_1, \ldots, x_m$, если известны натуральные числа b, m, d, s, r и рациональные числа $s_r(\alpha)$ и $u_1, \ldots, u_m$.

Поскольку неизвестные величины попарно различны, то мы будем дополнительно считать, что выполнены неравенства

$$x_1 < x_2 < \cdots < x_m. \quad (2.44)$$

Предположим, что существует некоторая натуральная константа d такая, что $x_m \leqslant d$ (для чисел, определяемых равенством (2.2), это предположение выполнено).

В этом случае значения неизвестных $x_1, \ldots, x_m$ могут быть найдены простым перебором. При этом, общее количество опробуемых наборов $(x_1, \ldots, x_m)$ не превосходит d^m. Далее мы будем считать, что значения $x_1, \ldots, x_m$ не ограничены.

§ 2.4.1.1. Вывод оценок для неизвестных параметров

Верна следующая теорема, см. [332].

Теорема 2.3. *Определим последовательность действительных чисел* α_k

$$\alpha_1 = s_r(\alpha), \quad \alpha_k = \alpha_{k-1} - u_{k-1}\xi_{k-1} \quad \text{для} \quad k = 2, \ldots, m,$$

где величины $\xi_1, \ldots, \xi_{m-1}$ *удовлетворяют равенствам*

$$\alpha = \sum_{i=1}^{m} u_i \xi_i, \quad \xi_i = \sum_{n=0}^{\infty} \frac{b^{-n}}{(dn + x_i)^s}, \quad i = 1, \ldots, m. \tag{2.45}$$

Если для r *выполнены условия:*

1. *величина* $s_r(\alpha)$ *отлична от нуля,*
2. *выполнено неравенство*

$$u_m > \frac{2(b-1)(dr + x_m)^s}{b^r \left(1 - b^{-r}\right)}, \tag{2.46}$$

3. *выполнено неравенство*

$$\sum_{i=1}^{m} u_i < (b-1)(dr)^s b^r, \tag{2.47}$$

то для всех индексов $k = 1, \ldots, m$ *выполнены неравенства*

$$\left(\frac{u_k}{\alpha_k}\right)^{\frac{1}{s}} < x_k < \left(\frac{b}{\alpha_k(b-1)} \sum_{i=k}^{m} u_i\right)^{\frac{1}{s}} \tag{2.48}$$

Для доказательства приведенной теоремы нам потребуются следующие утверждения.

Лемма 2.7. *Для любого индекса* $r = 0, 1, \ldots$ *выполнено равенство*

$$\alpha - s_r(\alpha) = b^{-r}\left(\delta_r + \gamma_r\right),$$

где

$$\gamma_r = \sum_{n=r+1}^{\infty} \sum_{i=1}^{m} \frac{u_i}{(dn + x_i)^s} b^{-n}, \tag{2.49}$$

а величина $\delta_r \in \mathbb{Q}$ *определяется условиями*

$$\delta_r = b^r \sum_{i=1}^{m} \sum_{i=0}^{r} \frac{u_i}{(dn + x_i)^s} - s_r(\alpha), \quad 0 < \delta_r < 1. \tag{2.50}$$

Доказательство данной леммы проводится аналогично доказательству леммы 2.5 и приведено в работе [330].

Лемма 2.8. *Для определенных равенством (2.49) величин* γ_r *верна оценка*

$$0 < \gamma_r < \frac{1}{(b-1)(dr)^s b^r} \sum_{i=1}^{m} u_i, \quad r = 0, 1, \ldots$$

Доказательство. Неравенство $0 < \gamma_r$, очевидно, выполнено в силу определения величины γ_r. Далее, в силу неравенств (2.44), а также, из условия $n \geqslant r+1$, следует выполнение неравенства $\frac{1}{dr} > \frac{1}{dn+x_i}$ для всех $i = 1, \ldots, m$. Тогда

$$\gamma_r = \sum_{n=r+1}^{\infty} \sum_{i=1}^{m} \frac{u_i}{(dn + x_i)^s} b^{-n} < \frac{1}{(dr)^s} \sum_{i=1}^{m} u_i \sum_{n=r+1}^{\infty} b^{-n}.$$

Учитывая равенства

$$\sum_{n=r+1}^{\infty} b^{-n} = b^{-r} \sum_{n=1}^{\infty} b^{-n} = \frac{b^{-r}}{(b-1)}, \tag{2.51}$$

получаем утверждение леммы. □

Доказательство теоремы 2.3. Согласно (2.45) и (2.49), для величины α_k выполнено равенство

$$\alpha_k = s_r(\alpha) - \sum_{i=1}^{k-1} u_i \xi_i = \alpha - b^{-r} \left(\gamma_r + \delta_r)\right) - \sum_{i=1}^{k-1} u_i \xi_i = \sum_{i=k}^{m} u_i \xi_i - b^{-r} \left(\delta_r + \gamma_r\right).$$

Поскольку неизвестная x_k принимает наименьшее значение среди величин x_k, $x_{k+1}, \ldots, x_m$, а величины b, δ_r, γ_r положительны, то

$$\alpha_k < \sum_{i=k}^{m} u_i \xi_i = \sum_{i=k}^{m} u_i \sum_{n=0}^{\infty} \frac{b^{-n}}{(dn + x_k)^s} < \sum_{i=k}^{m} u_i \sum_{n=0}^{\infty} \frac{b^{-n}}{x_k^s} = \frac{b}{x_k^s (b-1)} \sum_{i=k}^{m} u_i.$$

Полученное неравенство дает нам оценку сверху для величины x_k.

С другой стороны, запишем равенство

$$\alpha_k = \sum_{i=k}^{m} u_i \sum_{n=0}^{\infty} \frac{b^{-n}}{(dn+x_i)^s} - b^{-r}\left(\delta_r + \gamma_r\right) =$$
$$= \sum_{i=k}^{m} \frac{u_i}{x_i^s} + \sum_{i=k}^{m} u_i \sum_{n=1}^{\infty} \frac{b^{-n}}{(dn+x_i)^s} - b^{-r}\left(\delta_r + \gamma_r\right) =$$
$$= \sum_{i=k}^{m} \frac{u_i}{x_i^s} + \Delta_1 - \Delta_2. \quad (2.52)$$

В начале, оценим величину Δ_1.

$$\Delta_1 = \sum_{i=k}^{m} u_i \sum_{n=1}^{\infty} \frac{b^{-n}}{(dn+x_i)^s} > \sum_{i=k}^{m} u_i \sum_{n=1}^{r} \frac{b^{-n}}{(dn+x_i)^s} >$$
$$> \sum_{i=k}^{m} u_i \sum_{n=1}^{r} \frac{b^{-n}}{(dn+x_m)^s} > \frac{1}{(dr+x_m)^s} \sum_{i=k}^{m} u_i \sum_{n=1}^{r} b^{-n}.$$

Учитывая (2.51), получим равенство

$$\sum_{n=1}^{r} b^{-n} = \sum_{n=1}^{\infty} b^{-n} - \sum_{n=r+1}^{\infty} b^{-n} = \frac{1-b^{-r}}{(b-1)},$$

из которого следует оценка

$$\Delta_1 > \frac{(1-b^{-r})}{(b-1)(dr+x_m)^s} \sum_{i=k}^{m} u_i.$$

Теперь, учитывая (2.46) и (2.47), а также утверждения доказанных ранее лемм, получаем цепочку неравенств

$$\Delta_1 > \frac{u_m\,(1-b^{-r})}{(b-1)(dr+x_m)^s} > 2b^{-r} > b^{-r}(1+\varepsilon) > \Delta_2,$$

где $\varepsilon = \frac{1}{(b-1)(dr)^s b^r} \sum_{i=1}^{m} u_i$. Следовательно, $\Delta_1 - \Delta_2 > 0$. Подставляя это неравенство в (2.52), получаем оценку

$$\alpha_k = \sum_{i=k}^{m} \frac{u_i}{x_i^s} + \Delta_1 - \Delta_2 > \sum_{i=k}^{m} \frac{u_i}{x_i^s} > \frac{u_k}{x_k^s},$$

из которой следует утверждение теоремы. □

Утверждение теоремы позволяет в явном виде выписать верхние и нижние оценки на величину неизвестных $x_1, \ldots, x_m$. Данные оценки верны, как следует из (2.46), только для r, удовлетворяющих неравенству

$$x_m < \left(\frac{u_m b^r (1 - b^{-r})}{2(b-1)} \right)^{\frac{1}{s}} - dr. \tag{2.53}$$

На практике, нам неизвестно значение x_m. Поэтому, фиксировав r, удовлетворяющее неравенству (2.47), мы получим оценку сверху на величину возможных решений исходной задачи. С другой стороны, правая часть неравенства (2.53), при $r \to \infty$, ведет себя как $O\left(b^{\frac{r}{s}}\right)$, следовательно, для любого значения x_m найдется такой индекс r, что неравенство (2.53) будет выполнено.

§ 2.4.1.2. Алгоритм поиска неизвестных

Алгоритм поиска неизвестных $x_1, \ldots, x_m$ заключается в следующем. Используя (2.48), необходимо вычислить интервал для величины x_1. Для каждого целого числа x_1 в указанном интервале определить величину

$$\alpha_2 = \alpha_1 - u_1 \sum_{n=0}^{\infty} \frac{b^{-n}}{(dn + x_1)^s}$$

и интервал для возможных значений величины x_2. Далее, для каждого целого значения x_2 из найденного интервала определить величину

$$\alpha_3 = \alpha_2 - u_2 \sum_{n=0}^{\infty} \frac{b^{-n}}{(dn + x_2)^s}$$

и интервал возможных значений для x_2. Заметим, что для x_2 также должна выполняться оценка снизу $x_1 < x_2$.

Продолжая аналогичным образом, найти все возможные наборы неизвестных значений $x_1, \ldots, x_m$. Для каждого найденного набора вычислить последовательность $a_0, a_1, \ldots$ и сравнить полученные значения с заданными. В случае совпадения, закончить алгоритм.

Приведем пример, иллюстрирующий описанные выше вычисления. Зафиксируем $b = 256$ и рассмотрим действительное число вида (2.2)

$$\alpha = \sum_{n=0}^{\infty} \left(\frac{1}{(4n + x_1)} + \frac{1}{(4n + x_2)} + \frac{1}{(4n + x_3)} \right) 256^{-n},$$

для которого $d = 4$, $m = 3$, $s = 1$ и $u_1 = u_2 = u_3 = 1$. Известно, что начальные коэффициенты в представлении $\alpha = \sum_{n=0}^{\infty} a_n 256^{-n}$ имеют вид

$$\{0, 7, 12, 235, 161, 143, 245, 159, 92, 205, 168, 97, 219, \ldots\}.$$

Выберем $r = 5$, согласно (2.53), этого достаточно для определения величин x_1, x_2, x_3 не превосходящих 2.156×10^9. Построим рациональное приближение

$$s_5(\alpha) = \frac{30281539983}{1099511627776} = 0.0275409001761.$$

Воспользовавшись неравенством (2.48) при $\alpha_1 = s_5(\alpha)$, получаем неравенства $37 \leqslant x_1 \leqslant 109$. Вычисляя для каждого значения x_1 в указанном интервале, величину $\alpha_2 = \alpha_1 - \xi_1$, найдем 73 интервала для возможных значений величины x_2. Так, при $x_1 = 37$ неизвестная x_2 удовлетворяет неравенству $2247 \leqslant x_2 \leqslant 4511$, а при $x_1 = 109$ интервал возможных значений для x_2 пуст.

Используя аналогичные соображения, для каждой пары x_1, x_2 найдем интервал для возможных значений x_3. Общее количество найденных троек, удовлетворяющих неравенствам (2.48), равно 286605.

Для отсева ложных значений мы используем следующее рассуждение. Если неизвестные x_1, x_2, x_3 принимают истинные значения, то, согласно (2.50), должно выполняться неравенство

$$\alpha(x_1, x_2, x_3) - \sigma_5 < 256^{-5},$$

где $\alpha(x_1, x_2, x_3) = \sum_{n=0}^{5} \left(\frac{1}{(4n+x_1)} + \frac{1}{(4n+x_2)} + \frac{1}{(4n+x_3)} \right) 256^{-n}$. Вычислив величины $\alpha(x_1, x_2, x_3)$ для найденных троек x_1, x_2, x_3, мы получили 211 троек, для которых выполнялось указанное неравенство.

В завершение, для каждой такой тройки было найдено преставление $\alpha(x_1, x_2, x_3) = \sum_{n=0}^{7} c_n 256^{-n}$ и, сравнивая полученные коэффициенты и коэффициенты $a_1, \ldots, a_7$, было найдено искомое решение

$$x_1 = 54, \quad x_2 = 122, \quad x_3 = 1381.$$

Описанные выше вычисления были произведены на ЭВМ, время[4] вычислений составило 4.87 сек.

Отметим, что время вычислений существенным образом зависит от величин x_1, x_2, x_3. Так, для определения неизвестных значений $x_1 = 122$, $x_2 = 1245$, $x_3 = 1381$, при тех же параметрах b, d, s, m и u_1, u_2, u_3, программе потребовался 1 час 32 минуты. В процессе поиска было перебрано 365263502 возможных троек, из которых 9169647 удовлетворяли неравенству (2.50).

[4]Вычисления производились на ноутбуке HP EliteBook с процессором Intel Core i5 CPU M 560, тактовой частотой 2.67GHz и 4Gb оперативной памяти.

§ 2.4.2. Восстановление неизвестных коэффициентов с использованием целочисленных соотношений

Описанный выше алгоритм целесообразно применять в случае, когда неизвестные $x_1, \ldots, x_m$ принимают неограниченные значения. В случае, когда известно ограничение на неизвестные $x_1, \ldots, x_m$, можно предложить поход, основанный на поиске целочисленных соотношений. Возможность применения данного подхода вытекает в силу существования представлений чисел из рассматриваемых нами классов (2.2) и (2.6) в виде линейных форм от действительных чисел с рациональными коэффициентами.

Напомним следующее определение.

Определение 2.5. *Пусть d натуральное число, числа $\alpha, \xi_0, \ldots, \xi_{d-1}$ попарно различные, не все одновременно равные нулю действительные числа.*

Мы будем называть целые, не все одновременно равные нулю числа $c_0, \ldots, c_d$ целочисленным соотношением, если

$$c_0\xi_0 + \cdots c_{d-1}\xi_{d-1} + c_d\alpha = 0.$$

Ясно, что линейные соотношения существуют только для линейно зависимых действительных чисел $\alpha, \xi_0, \ldots, \xi_{d-1}$ и

$$\alpha = -\sum_{k=0}^{d-1} \frac{c_k}{c_d}\xi_k.$$

Рассмотрим число α вида (2.2) и, также, как и в разделе 2.3.2.2, запишем его в виде

$$\alpha = \sum_{n=0}^{\infty} \left(\frac{u_1}{dn + x_1} + \cdots + \frac{u_m}{dn + x_m} \right) b^{-n} = \\ = \sum_{k=1}^{m} u_k \left(\sum_{n=0}^{\infty} \frac{b^{-n}}{dn + x_k} \right) = \sum_{k=0}^{d-1} v_k \xi_k,$$

где $\xi_k \in \mathbb{R}$, $v_k \in \mathbb{Q}$, $k = 0, \ldots, d-1$, и

$$\xi_k = \sum_{n=0}^{\infty} \frac{b^{-n}}{dn + (k+1)}, \quad v_k = \begin{cases} u_{k+1}, & \text{если } x_{k+1} = k+1, \\ 0, & \text{иначе.} \end{cases}$$

В таком случае, задача определения неизвестных значений $x_1, \ldots, x_m$ сводится к определению индексов, на которых располагаются ненулевые

значения вектора $v_0, \ldots, v_{d-1}$. При этом, величины ξ_k являются константами, не зависящими от значения величины α.

Аналогично, рассмотрим число α вида (2.6) и, также, как и в разделе 2.3.2.3, воспользуемся (2.13) и запишем его в виде

$$\alpha = \sum_{n=0}^{\infty} \frac{x_n}{n!} = \sum_{k=0}^{d-1} x_k \xi_k,$$

где d период последовательности рациональных чисел $(x_n)_{n=0}^{\infty}$, числа x_0, $\ldots$, x_{d-1} образуют период и

$$\xi_k = \sum_{n=0}^{\infty} \frac{1}{(dn+k)!}, \quad k = 0, \ldots, d-1,$$

фиксированные константы, значения которых не зависят от величины α.

Таким образом, задача восстановления неизвестных коэффициентов действительных иррациональных чисел из рассматриваемых нами классов может быть сведена к задаче определения неизвестных целых коэффициентов $c_0, \ldots, c_d$, удовлетворяющих равенству

$$c_0\xi_0 + \cdots + c_{d-1}\xi_{d-1} + c_d s_r(\alpha) = 0, \tag{2.54}$$

при известных натуральных значениях r, d,

$$s_r(\alpha) = \sum_{n=0}^{r} a_n b^{-n}, \quad \text{где} \quad |\alpha - s_r(\alpha)| < b^{-r} \quad \text{и} \quad s_r(\alpha) \in \mathbb{Q},$$

и сколь угодно точных рациональных приближений к действительным числам $\xi_0, \ldots, \xi_{d-1}$.

Для чисел вида (2.6) найденные в (2.54) значения дадут искомый результат

$$x_k = -\frac{c_k}{c_d}, \quad k = 0, \ldots, d-1.$$

Для чисел вида (2.2) необходимо будет провести дополнительное сравнение найденных значений с известными значениями $u_1, \ldots, u_m$.

Задача поиска целочисленных соотношений является хорошо известной и имеющей длительную историю, ведущую отсчет от расширенного алгоритма Эвклида. Случай произвольного натурального значения d исследовался большим числом авторов, включая Якоби, Пуанкаре, Минковского, Перрона и т.д. Первым алгоритмом, для которого получена строгая оценка трудоемкости является алгоритм Фергюссона и Фуркада, см. [88]. Данный алгоритм является первым в ряду алгоритмов, трудоемкость которых оценивается функцией, экспоненциальной по величине d

и логарифмической от величины нормы разыскиваемого целочисленного соотношения. К этому ряду алгоритмов стоит отнести LLL [142], HJLS [197] и PSLQ [86] алгоритмы. Краткое сравнение указанных алгоритмов может быть найдено в работе [45].

Для поиска линейного соотношения, удовлетворяющего (2.54), целесообразно использовать модифицированную версию PSLQ алгоритма, изложенную в работе Фергюссона, Бейли и Арно [87].

В отличие от LLL алгоритма, имеющего широкий спектр приложений, алгоритм PSLQ предназначен только для поиска линейных соотношений. Данный алгоритм представляет собой итерационное преобразование матрицы специального вида, в результате которого один или несколько столбцов с фиксированными индексами дают искомое линейное соотношение. Поскольку алгоритм PSLQ хорошо известен, мы не будем давать его детальное описание, а приведем верхнюю оценку числа итераций алгоритма.

Напомним, что эвклидовой нормой вектора $c = (c_0, \ldots, c_d)$, состоящего из действительных значений, называется величина

$$N(c) = \sqrt{\sum_{k=0}^{d} c_k^2}.$$

Верна следующая теорема, см [87, Теорема 2].

Теорема 2.V. *Пусть $\gamma > \sqrt{\frac{4}{3}}$ – произвольное действительное число и τ действительное число, удовлетворяющее равенству*

$$\tau = \frac{1}{\sqrt{\frac{1}{4} + \frac{1}{\gamma^2}}}.$$

Пусть $\xi = (\xi_0, \ldots, \xi_d)$ вектор, содержащий линейно зависимые действительные числа, не все одновременно равные нулю, а $N(c) \in \mathbb{R}$ – минимально возможное значение нормы линейного соотношения. Тогда алгоритм PSLQ найдет какое-либо линейное соотношение не более, чем за

$$\frac{d(d+1)}{2} \cdot \frac{\ln(\gamma^d N(c))}{\ln \tau} \tag{2.55}$$

итераций.

Отметим, что величина γ является параметром алгоритма, используется при поиске линейных соотношений и может существенно влиять на найденные значения.

Полученная в теореме 2.V оценка числа итераций алгоритма PSLQ не позволяет сделать вывод о точности рациональных приближений, которые должны быть использованы при практических вычислениях с величинами $\xi_0, \ldots, \xi_d$ и α. В работе [12] была высказана гипотеза о том,

что точность рациональных приближений должна составлять порядка $(d+1)\log_{10} N(c)$ десятичных знаков. Проведенные нами эксперименты показывают, что данная величина оказывается достаточной.

Алгоритм восстановления неизвестных параметров действительных чисел вида

$$\alpha = \sum_{k=0}^{d-1} x_k \xi_k,$$

основан на том, что мы можем использовать дополнительную информацию о величинах $x_0, \ldots, x_{d-1}$ для определения величины $N(c)$ – нормы линейного соотношения. Предположим, нам известны[5] две константы $z_0, z_1 \in \mathbb{N}$ такие, что

$$0 < x_k = \frac{u_k}{p_k} < z_0, \quad 0 < p_k < z_1, \quad k = 0, \ldots d-1,$$

где $\frac{u_k}{p_k}$ – несократимая рациональная дробь. Тогда, из равенства

$$c_0\xi_0 + \ldots c_{d-1}\xi_{d-1} + c_d s_r(\alpha) = 0,$$

можно сделать вывод, что $\frac{u_k}{p_k} = -\frac{c_k}{c_d}$ и

$$c_d = \mathbf{НОК}(p_0, \ldots, p_{d-1}) < \prod_{k=0}^{d-1} p_k < z_1^d,$$

а также

$$|c_k| = c_d \frac{u_k}{p_k} < z_0 z_1^d.$$

Последние два неравенства дают нам оценку на величину ожидаемой нормы целочисленного решения

$$N(c) = \sqrt{\sum_{k=0}^{d} c_k^2} < \sqrt{d z_0^2 z_1^{2d} + z_1^{2d}} < z_0 z_1^q \sqrt{d+1},$$

а также оценку погрешности при вычислении величин $\xi_0, \ldots, \xi_{d-1}$.

[5]Константы z_0, z_1, как правило, определяются конкретной реализацией алгоритма выработки псевдослучайной последовательности, см., например, раздел 2.6. Также отметим, что знание констант z_0, z_1 позволяет реализовать простой, но существенно более медленный алгоритм поиска величин $x_0, \ldots, x_{d-1}$, основанный на переборе всех возможных, взаимно простых пар натуральных чисел u_k, p_k в указанных выше границах.

Алгоритм 2.3: Алгоритм определения рациональных элементов периодической последовательности $(x_k)_{k=0}^{\infty}$, определяющих действительное число вида $\alpha = \sum_{k=0}^{\infty} \frac{x_k}{k!}$, по заданному рациональному приближению к α.

Вход : Основание системы счисления $b > 1$, период последовательности $d \geqslant 1$ и рациональное приближение $s_r(\alpha) = \sum_{n=0}^{r} a_n b^{-n}$, а также величины z_0, z_1.

Выход : Значения $x_0, \ldots, x_{d-1}$.

1 Определить $N = z_0 z_1^d \sqrt{d+1}$ и $r = \lceil (d+1)\log_b(N) \rceil$.

2 Вычислить значения констант $\xi_0, \ldots, \xi_{d-1}$ с точностью до b^{-r}.

3 Определить $\gamma = 2$ и $\tau = \sqrt{2}$.

4 Применить PSLQ-алгоритм с параметрами γ и τ к вектору $\xi_0, \ldots, x_{d-1}, s_r(\alpha)$ и найти $c_0, \ldots, c_{d-1}$ такие, что $c_0\xi_0 + \cdots + c_d s_r(\alpha) = 0$.

5 Определить $x_k = -\frac{c_k}{c_d}$ для всех $k = 0, \ldots, d-1$.

Для иллюстрации предложенного алгоритма, приведем следующий пример из работы [338]. Зафиксируем в качестве основания системы счисления $b = 10$ и рассмотрим число α вида (2.6)

$$\alpha = \sum_{k=0}^{\infty} \frac{x_k}{k!} = \sum_{k=0}^{d-1} x_k \xi_k,$$

где $(x_k)_{k=0}^{\infty}$ периодическая последовательность с периодом $d = 16$. При этом рациональные значения x_k удовлетворяют равенству

$$x_k = \frac{u_k}{p_k}, \quad k = 0, \ldots, 15,$$

где u_k натуральное число, не превосходящее 256, а $p_0, p_1, \ldots, p_{15}$ последовательность подряд идущих простых чисел, начиная с $p_0 = 257$. В этом случае рациональные числа x_k несократимы и удовлетворяют неравенствам

$$\frac{1}{349} \leqslant x_k \leqslant \frac{256}{257} < 1.$$

Величины

$$\xi_k = \sum_{n=0}^{\infty} \frac{1}{(16n+k)!}, \quad k = 0, \ldots, 15,$$

не зависят от α и принимают следующие значения:

```
0 1.0000000000000477947733238738529743858753019302950195619701694503707512320804118 ...
1 1.0000000000000028114572543455207631990607463665277357362986040211079908933869997 ...
2 0.5000000000000001561920696858622646221670306581088553970412722804623710544833633 ...
3 0.1666666666666666748873019132909963836227446794685337347973265566572347258062496 ...
4 0.0416666666666666670776984289978831525144684167305469907640903078736461388525551 ...
5 0.0083333333333333333529062743967245945641809803614405552193339609508460509971787 ...
6 0.0013888888888888888897785680281339462175637805451023407635020627982504595682333 ...
7 0.0001984126984126984127370944001190052530755818667526241010656401018168713604158 ...
```

```
 8 0.00002.480158730158730158891332487268341993635030183200591251701216282773451590 7 ...
 9 0.000002.755731922398589065320201425242433799217680484483744208732766074394097353 ...
10 0.0000002.75573192239858906528052788503083703985648136497447536272139778922598687 ...
11 0.00000002.505210838544171877514394528407967423653636557035856658645267714436810 6 ...
12 0.000000002.087675698786809897924288921357213069164494407556972964638709623462973 ...
13 0.0000000001.605904383682161459940368713304139564965726926890118466776976473662 49 ...
14 0.00000000001.14707455977297247138554679663109215718765032813418642644338974870 91 ...
15 0.000000000009.764716373181981647590234811082962396210473021448068298969839921769 ...
```

Для определения точности, с которой указанные величины должны быть определены, воспользуемся полученной выше оценкой величины $N(c)$. Для искомого линейного соотношения $c = (c_0, \ldots, c_{16})$, удовлетворяющего равенству

$$c_0\xi_0 + \cdots + c_{15}\xi_{15} + c_{16}\alpha = 0,$$

коэффициент c_{16} должен быть равен

$$c_{16} = 227 \cdot 263 \cdots 347 \cdot 349 = 406973141155791893749222226060268511021 9.$$

Поскольку при $k = 0, \ldots, 15$ все величины $c_k < c_{16}$, то

$$N(c) = \sqrt{\sum_{k=0}^{16} c_k^2} < c_{16}\sqrt{17}.$$

Тогда, для чисел $\xi_0, \ldots, \xi_{15}, \alpha$ мы будем использовать рациональные приближения, содержащие в своей записи

$$\lceil 17 \cdot \log_{10} 406973141155791893749222226060268511021 9 \rceil = 2805$$

десятичных знаков.

Для проведения вычислений было выбрано конкретное значение α, записываемое в виде десятичной дроби следующим образом

```
0.290412155877151691931103601184138977688518884733756330516356483806936952862834 2 ...
```

Для определения числа α использовались следующие значения x_0, …, x_{15}

```
13/257, 52/263, 17/269, 13/271, 12/277, 19/281, 19/283, 77/293,
                                25/307, 26/311, 131/313, 14/317, 19/331, 31/337, 43/347, 2/349,
```

образующие период последовательности значений $(x_k)_{k=0}^{\infty}$.

Автором была написана программа, в основу которой была положена программная реализация алгоритма PSLQ, написанная П. Циммерманом, см. [263]. Время[6] работы программы составило менее одной секунды.

[6] Вычисления производились на ноутбуке Dell Latitude 7390 с процессором Intel Core i5-8250U, тактовой частотой 1.67GHz и 4Gb оперативной памяти.

В результате работы программы были получены следующие значения $c_0, \ldots, c_{16}$

```
-20586190019553675559299178750130313787 1
-80466172395821971387678919221041682787 6
-25719492192001718192330029156225147536 7
-19522696808211419257342763611747197945 7
-17630605393030695758089049504415964376 4
-27517756875302654737491894288772603948 1
-27323285095265180145707499276131101446 7
-10695198590101015637778195019331288514 91
-33141135273272955517037640558653787542 5
-34023478038747875361671311503430807995 4
-17033061179363814083433901474087915317 53
-17973577211927717705012968974270533609 8
-23360996018006181212191003912825080693 1
-37436698444598067377524893198422325939 7
-50431830172043375882468460289889181481 1
-23322243046177185888207577424657221262
40697314115579189374922222606026851102 19
```

которые дали нам неизвестные значения, например,

$$x_0 = \frac{|c_0|}{c_{16}} = \frac{205861900195536755592991787501303137871}{406973141155791893749222226060268511021 9} = \frac{13}{257},$$

$$x_1 = \frac{|c_1|}{c_{16}} = \frac{804661723958219713876789192210416827876}{406973141155791893749222226060268511021 9} = \frac{52}{263},$$

и так далее.

Отдельно стоит отметить то, как влияет на работу алгоритма значение параметра γ, введенного в утверждении теоремы 2.V. Результаты нескольких запусков программы сведены в следующую таблицу.

γ	число итераций	время	примечание
$\sqrt{4/3} \sim 1.1547$	17567	$0,597$ сек.	
2	7644	$0,316$ сек.	
3	5951	$0,256$ сек.	
4	5231	$0,253$ сек.	
5	4641	$0,202$ сек.	
6	4771	$0,214$ сек.	увеличение нормы

Из приведенных данных следует, что последовательное увеличение параметра γ приводит к уменьшению числа итераций алгоритма и, как следствие, к снижению времени его работы. Однако, начиная с некоторого момента, алгоритм начинает выдавать линейные соотношения, имеющие существенно большее значение нормы. К сожалению, автору не известны теоретические результаты, объясняющие подобное поведение алгоритма.

§ 2.4.3. Методы «чтения вперед»

В этом разделе мы рассмотрим следующую постановку задачи определения коэффициентов действительного иррационального числа.

Определение 2.6. *Пусть α отличное от нуля действительное иррациональное число. Под задачей «чтения вперед» мы будем рассматривать задачу нахождения коэффициентов $a_k, a_{k+1}, \ldots$ представления числа α в виде (2.1) – систематической дроби по основанию b, если известны коэффициенты разложения $a_{k-h}, \ldots, a_{k-1}$ для некоторых целых чисел k, h, удовлетворяющих неравенствам $k \geqslant h > 0$.*

Схематично, задачу «чтения вперед» можно изобразить следующим образом

$$a_0, a_1, \ldots, \underbrace{a_{k-h}, \ldots, a_{k-1}}_{\text{известно}}, \underbrace{a_k, a_{k+1}, \ldots}_{\text{надо найти}} \tag{2.56}$$

Если $k - h = 0$ и известно, что число α имеет вид либо (2.2), либо (2.6), то можно воспользоваться известным фрагментом $a_0, \ldots, a_{k-1}$ и, с помощью изложенного в предыдущем разделе алгоритма, восстановить неизвестные коэффициенты $x_1, \ldots, x_m$ числа α. После чего можно вычислить коэффициенты систематической дроби числа α с любой точностью.

Возникает вопрос, если $k > h$ и нам неизвестны начальные коэффициенты разложения числа α в систематическую дробь, можно ли тем же сам способом реализовать «чтение вперед»? Другими словами, можно ли по известному фрагменту $a_{k-h}, \ldots, a_{k-1}$ восстановить некоторое число β вида (2.2) или (2.6) такое, что

$$\left| \beta - \sum_{n=0}^{h-1} a_{k-h+n} b^{-n} \right| < b^h$$

и разложение в систематическую дробь числа β совпадает с разложением числа α?

Дадим формальное определение.

Определение 2.7. *Пусть*

$$\alpha = \sum_{k=0}^{\infty} a_n b^{-n}, \quad \gamma = \sum_{k=0}^{\infty} c_n b^{-n}, \quad a_0, c_0 \in \mathbb{Z}$$

и $0 \leqslant a_k < b$, $0 \leqslant c_k < b$ при $k > 0$ — различные действительные иррациональные числа. Мы будем говорить что их разложения в систематическую дробь совпадают, если существуют такие целые неотрицательные числа s, t такие, что

$$a_{s+k} = c_{t+k}$$

для всех индексов $k = 1, \ldots$.

Предположим, что $s \geqslant t$. Из определения 2.7 следует, что

$$\alpha - \sum_{k=0}^{s} a_k b^{-k} = \sum_{k=1}^{\infty} a_{s+k} b^{-(s+k)} = \\ = \frac{1}{b^{s-t}} \sum_{k=1}^{\infty} c_{t+k} b^{-(t+k)} = \frac{1}{b^{s-t}} \left(\gamma - \sum_{k=0}^{t} c_k b^{-k} \right),$$

тогда

$$\frac{1}{b^s} \left(\alpha b^s - \sum_{k=0}^{s} a_{s-k} b^k \right) = \frac{1}{b^{s-t} b^t} \left(\gamma b^t - \sum_{k=0}^{t} c_{t-k} b^{t-k} \right)$$

или

$$\gamma = \alpha b^{s-t} + \frac{p}{b^t}, \quad p \in \mathbb{Z}. \tag{2.57}$$

Равенство (2.57) определяет преобразование, которое сохраняет, с точностью до сдвига, последовательность коэффициентов разложения в систематическую дробь.

§ 2.4.3.1. Числа из первого множества

В (2.2) мы определили множество чисел вида

$$\gamma = \sum_{n=0}^{\infty} \sum_{k=1}^{m} \frac{z_k}{(dn + y_k)} b^{-n},$$

где $z_k \in \mathbb{Q}$ – не все одновременно равные нулю, а y_k – натуральные, попарно различные числа, удовлетворяющие неравенству $1 \leqslant y_k \leqslant d$, $k = 1, \ldots, m$.

Предположим, что числа $y_1, \ldots, y_k$ не ограничены и обозначим

$$x_k = \begin{cases} d, & \text{если} \quad y_k \equiv 0 \pmod{d}, \\ r, & \text{если} \quad y_k \equiv r \pmod{d}, \quad r > 0 \end{cases}$$

и

$$l_k = \frac{y_k - x_k}{d} \in \mathbb{Z}.$$

Запишем равенство

$$\gamma = \sum_{n=0}^{\infty}\sum_{k=1}^{m}\frac{z_k}{(dn+y_k)}b^{-n} = \sum_{n=0}^{\infty}\sum_{k=1}^{m}\frac{z_k}{((n+l_k)d+x_k)}b^{-n} =$$

$$= \sum_{k=1}^{m} z_k \sum_{n=l_k}^{\infty}\frac{b^{l_k}}{(nd+x_k)}b^{-n} =$$

$$= \sum_{n=0}^{\infty}\sum_{k=1}^{m}\frac{u_k}{(nd+x_k)}b^{-n} - \sum_{k=1}^{m}\sum_{n=0}^{l_k-1}\frac{u_k}{(nd+x_k)}b^{-n} = \beta + \frac{v}{q} \quad (2.58)$$

где $u_k = z_k b^{l_k}$, $k = 1, \ldots, m$, а $\dfrac{v}{q} \in \mathbb{Q}$ – несократимая рациональная дробь. Запишем $\beta = \sum_{n=0}^{\infty}\frac{P_\beta(n)}{Q_\beta(n)}b^{-n}$, где $P_\beta, Q_\beta \in \mathbb{Q}[n]$, $\deg Q_\beta(n) > \deg P_\beta(n)$ и

$$Q_\beta(n) = \prod_{k=1}^{m}(dn+x_k).$$

Тогда, если величины y_k попарно не сравнимы друг с другом по модулю d, то значения x_k попарно различны и мы получаем, что β отлично от нуля и удовлетворяет условиям теоремы 2.III, т.е. является действительным иррациональным числом вида (2.2).

Теорема 2.4. *Разложение в систематическую дробь действительного иррационального числа α вида (2.2) совпадает с разложением в систематическую дробь действительного иррационального числа β вида (2.2) только в случае, когда $\beta = \alpha b^s$ для некоторого целого числа s.*

Доказательство. Предположим, что утверждение теоремы не выполнено и существует некоторое число γ, разложение которого в систематическую дробь совпадает с разложением числа α. Обозначим

$$\gamma = \sum_{n=0}^{\infty}\sum_{k=1}^{m}\frac{z_k}{(dn+y_k)}b^{-n},$$

где $z_k \in \mathbb{Q}$ – не все одновременно равные нулю, а y_k – натуральные, попарно не сравнимые друг с другом по модулю d числа, $k = 1, \ldots, m$.

Тогда, согласно (2.57) и (2.58),

$$\beta + \frac{v}{q} = \gamma = \alpha b^s + \frac{p}{b^t},$$

где s, t неотрицательные целые, $p \in \mathbb{Z}$, $\frac{v}{q} \in \mathbb{Q}$, а β – действительное иррациональное число вида (2.2). Последнее равенство равносильно

$$\beta - \alpha b^s = \frac{p}{b^t} - \frac{v}{q}. \quad (2.59)$$

Пусть

$$\alpha = \sum_{n=0}^{\infty} \sum_{j=1}^{m_1} \frac{u_j}{(d_\alpha n + x_j)} b^{-n}, \quad \beta = \sum_{n=0}^{\infty} \sum_{k=1}^{m_2} \frac{v_k}{(d_\beta n + t_k)} b^{-n},$$

тогда выполнено равенство

$$\beta - \alpha b^s = \sum_{n=0}^{\infty} \left(\sum_{j=1}^{m_1} \frac{u_j}{(d_\alpha n + x_j)} - b^s \sum_{k=1}^{m_2} \frac{v_k}{(d_\beta n + t_k)} \right) b^{-n} =$$

$$= \sum_{n=0}^{\infty} \left(\frac{P(n)}{(d_\alpha n + x_1) \cdots (d_\alpha n + x_{m_1})(d_\beta n + t_1) \cdots (d_\beta n + t_{m_2})} \right) b^{-n} =$$

$$= \sum_{n=0}^{\infty} \frac{P(n)}{Q(n)} b^{-n},$$

а многочлен $Q(n) \in \mathbb{Q}[n]$ раскладывается на линейные множители и $\deg Q(n) \leqslant m_1 + m_2$ (неравенство строгое в случае выполнения равенства $d_\beta x_j = d_\alpha t_k$ для некоторых значений j, k).

Корнями многочлена $Q(n)$ являются значения, принадлежащие интервалу $[-1, 0)$, тогда согласно теореме 2.III, число $\beta - \alpha b^s$ является либо нулем, либо трансцендентным числом.

Мы получили, что в правой части (2.59) стоит рациональное число, а в левой – трансцендетное. Это возможно только в том случае, когда обе части равны нулю. Таким образом $\beta = \alpha b^s$. □

Применим доказанную теорему для ответа на поставленный в начале параграфа вопрос. Пусть действительное иррациональное число α порождает последовательность (2.56) с известным фрагментом $a_{k-h}, \ldots, a_{k-1}$.

Пусть β действительное иррациональное число, восстановленное из известного фрагмента с помощью описанного в предыдущем параграфе алгоритма. Тогда

$$\beta = \sum_{n=0}^{\infty} a_{k-h+n} b^{-n},$$

и из утверждения леммы 2.4 следует, что найдется такое целое s, что

$$0 = \beta - \alpha b^s = \beta - b^s \left(\sum_{n=0}^{k-h-1} a_n b^{-n} + b^{h-k} \beta \right),$$

тогда

$$\beta = \frac{b^s}{(1 + b^{s+h-k})} \sum_{n=0}^{k-h-1} a_n b^{-n}.$$

Поскольку в правой части полученного равенства стоит рациональное число, а в левой части – иррациональное, отличное от нуля число, то наше исходное предположение о том, что разложения в систематическую дробь чисел α и β совпадают, не выполнено.

§ 2.4.3.2. Числа из второго множества

Рассуждения, аналогичные проведенным выше, можно провести для второго класса чисел. В (2.6) мы определили множество чисел вида

$$\gamma = \sum_{n=0}^{\infty} \frac{z_n}{n!},$$

где $(z_n)_{n=0}^{\infty}$ – чисто периодическая последовательность не всех одновременно равных нулю рациональных чисел.

Предположим, что последовательность $(z_n)_{n=0}^{\infty}$ не является чисто периодической и найдутся натуральное число λ и непериодическая часть z_0, $\ldots$, $z_{\lambda-1} \in \mathbb{Q}$. Обозначим τ период последовательности $(z_n)_{n=0}^{\infty}$ и определим $r \equiv \lambda \pmod{\tau}$, тогда

$$\begin{aligned}\gamma = \sum_{n=0}^{\lambda-1} \frac{z_n}{n!} + \sum_{n=\lambda}^{\infty} \frac{z_n}{n!} = \\ = \sum_{n=0}^{\lambda-1} \frac{z_n}{n!} + \left(\sum_{n=0}^{\infty} \frac{z_{\lambda+\tau-r+n}}{n!} - \sum_{n=0}^{\lambda-1} \frac{z_{\lambda+\tau-r+n}}{n!} \right) = \\ = \sum_{n=0}^{\infty} \frac{z'_n}{n!} + \sum_{z=0}^{\lambda-1} \frac{z_n - z_{\lambda+\tau-r+n}}{n!} = \beta + \frac{q}{(\lambda-1)!}, \quad (2.60)\end{aligned}$$

где $z'_n = z_{\lambda+\tau-r+n}$ и $q \in \mathbb{Q}$. При этом, у числа β та же длина периода последовательности коэффициентов, что и у числа α. Докажем следующее утверждение.

Теорема 2.5. *Разложение в систематическую дробь действительного иррационального числа α вида (2.6) совпадает с разложением в систематическую дробь действительного иррационального числа β вида (2.6) только в случае, когда $\beta = \alpha b^s$ для некоторого целого числа s.*

Доказательство. Предположим, что утверждение теоремы не выполнено. Тогда, согласно (2.57) и (2.60), найдется γ такое

$$\beta + \frac{q}{(\lambda-1)!} = \gamma = \alpha b^s + \frac{p}{b^t},$$

где s, t неотрицательные целые, $p \in \mathbb{Z}$, $q \in \mathbb{Q}$, а β – действительное иррациональное число вида (2.6). Последнее равенство равносильно

$$\beta - \alpha b^s = \frac{p}{b^t} - \frac{q}{(\lambda - 1)!}. \tag{2.61}$$

Предположим, что правая часть равенства (2.61) представляет собой отличное от нуля рациональное число. Используя обозначения из (2.60), можно записать число, стоящее слева в равенстве (2.61), в виде

$$\beta - \alpha b^s = \sum_{n=0}^{\infty} \frac{(z'_n - b^s x_n)}{n!},$$

Данное число снова представляет собой число вида (2.6), в котором период последовательности $(z'_n - b^s x_n)_{n=0}^{\infty}$ определяется как наименьшее общее кратное периодов последовательностей $(x_n)_{n=0}^{\infty}$ и $(z_n)_{n=0}^{\infty}$. Тогда, согласно теореме 2.1, число $\beta - \alpha b^s$ иррационально и мы получаем противоречие.

Таким образом, равенство (2.61) может выполняться, если в его правой и левой части стоят нули. В этом случае, получаем равенство $\beta = \alpha b^s$ и утверждение теоремы. □

Дальнейшие рассуждения аналогичны сделанным после доказательства теоремы 2.4.

Осталось заметить, что мы показали невозможность восстановления по известному фрагменту последовательности (2.56) числа β того же вида, что и число α, породившее указанную последовательность.

Предположение о том что число β может иметь другой вид, в настоящее время, не опровергнуто. Вместе с тем, доказательство подобного факта или его опровержение связано со сложными математическими задачами.

Доказательство теорем 2.4 и 2.5 схоже, поскольку в обоих случаях происходит опровержение равенства (2.57). Из данного равенства следует, что числа α, β и 1 являются линейно зависимыми и выполнено равенство

$$b^s \cdot \alpha - 1 \cdot \beta + \frac{p}{b^t} \cdot 1 = 0, \tag{2.62}$$

в котором $1, b^s, \frac{p}{b^t}$ образуют множество рациональных, не равных одновременно нулю коэффициентов.

Предположим, что число α имеет вид (2.2) и равно, например,

$$\alpha = \sum_{n=0}^{\infty} \left(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6} \right) 16^{-n} = \pi,$$

а число β имеет вид (2.6) и равно, например,

$$\beta = \sum_{n=0}^{\infty} \frac{1}{n!} = e,$$

тогда из (2.62) следовала бы линейная зависимость чисел π и e. Вместе тем в настоящий момент не известно, см. [182], зависимы ли данные числа или нет.

Заключение к § 2.4

В § 2.4 рассмотрен ряд вопросов, относящихся к восстановлению элементов последовательностей, порождаемых коэффициентами систематических дробей действительных иррациональных чисел из заданных классов (2.2) и (2.6):

- в первом разделе параграфа доказана теорема 2.3, позволяющая получить оценки на неизвестные коэффициенты числа вида (2.2); на основе данной теоремы автором предложен алгоритм восстановления коэффициентов чисел вида (2.2) в случае, если неизвестные коэффициенты не ограничены;

- во втором разделе параграфа описан разработанный автором алгоритм восстановления неизвестных коэффициентов чисел вида (2.2) и (2.6), основанный на поиске целочисленных соотношений; приведены результаты практической реализации алгоритма на ЭВМ;

- в третьем разделе параграфа доказаны теоремы 2.4 и 2.5, позволяющие сделать вывод о невозможности применения предложенного ранее алгоритма для реализации «чтения вперед», то есть поиска действительного иррационального числа β, имеющего тот же вид, что и α, такого, что разложения чисел α и β в систематические дроби совпадают.

§ 2.5. Анализ вырабатываемых последовательностей

Пусть, как и ранее, иррациональное число α задано быстро сходящимся рядом (2.17)

$$\alpha = \sum_{n=0}^{\infty} \omega_n$$

где $\omega_n \in \mathbb{Q}$ и существует индекс $n_0 \in \mathbb{N}$ такой, что для любого индекса $n \geqslant n_0$ будет выполнено неравенство $0 < |\omega_n| < f(n)b^{-n}$ и $\lim_{n\to\infty} f(n) = 0$. Будем также считать, что действительное число α представлено в виде систематической дроби (2.1)

$$\alpha = \sum_{n=0}^{\infty} a_n b^{-n},$$

где $a_0 \in \mathbb{Z}$ и $0 \leqslant a_n < b$ для всех $n = 1, 2, \ldots$.

В настоящем параграфе мы рассмотрим числа α вида (2.2), (2.6) и исследуем вопрос о распределении последовательности коэффициентов $\{a_n\}_{n=1}^{\infty}$ разложения числа α в систематическую дробь.

§ 2.5.1. Критерий нормальности

Как следует из приведенного в § 2.2 исторического обзора для рассматриваемых нами чисел доказательство их нормальности неизвестно. Вместе с тем, теорема А.Н. Коробова, см. теорему 2.II, дает нам простой критерий проверки заданного числа на нормальность.

Напомним, что алгоритм представления числа α в виде систематической дроби основывается на равенстве (2.24)

$$\alpha_n = b\delta_{n-1} + \omega_n b^n, \quad a_n = \lfloor \alpha_n \rfloor, \quad \delta_n = \alpha_n - a_n, \quad n = 0,\ 1,\ \ldots,$$

где $\delta_n, \omega_n, \alpha_n \in \mathbb{Q}$, $a_n \in \mathbb{Z}$ и $\delta_{-1} = 0$. Напомним также, что символом s_k мы обозначаем, см. (2.25), частичную сумму

$$s_k = \sum_{n=0}^{k} a_n b^{-n}, \quad s_k \in \mathbb{Q}, \quad k = 0,\ 1,\ \ldots.$$

Обозначим

$$\gamma_k = b^k \sum_{n=1}^{\infty} \omega_{k+n}, \quad \gamma_k \in \mathbb{Q}, \quad k = 0,\ 1,\ \ldots. \tag{2.63}$$

Тогда, из утверждения леммы 2.5, следует равенство

$$\alpha - s_k(\alpha) = b^{-k}\left(\delta_k + \gamma_k\right).$$

Теперь мы можем доказать простой критерий нормальности числа α, см. [330].

Теорема 2.6. *Пусть $\alpha > 0$ иррациональное число вида (2.2) или (2.6). Тогда α является нормальным числом по основанию b, если определяемая равенством (2.24) последовательность величин $(\delta_k)_{k=0}^{\infty}$ является реализацией равновероятно распределенной на интервале $[0,1)$ случайной величины.*

Для доказательства сформулированной теоремы необходимо следующее, хорошо известное утверждение, см. [302, гл. 1].

Лемма 2.9. *Если последовательность $(\delta_k)_{k=0}^{\infty}$ равномерно распределена на интервале $[0,1)$, а последовательность $(\gamma_k)_{k=0}^{\infty}$ имеет конечный предел, то последовательность $(\delta_k+\gamma_k \mod 1)_{k=0}^{\infty}$ также является последовательностью равномерно распределенных на интервале $[0,1)$ величин.*

Доказательство теоремы 2.6. В силу того, что число α определяется равенством (2.2) или (2.6) следует, что α представимо в виде (2.17), т.е. найдется такой индекс n_0, что для всех $k > n_0$ будет выполнено

$$0 < |\omega_k| < f(k)b^{-k}$$

и $\lim_{k\to\infty} f(k) = 0$. Выберем произвольное $\varepsilon > 0$, тогда найдется индекс n_1 такой, что для всех $k > n_1$ будет выполнено $0 < f(k) < \varepsilon$.

Теперь, для любого индекса k такого, что $k > \max\{n_0, n_1\}$, мы можем записать

$$|\gamma_k| = b^k \sum_{n=1}^{\infty} |\omega_{k+n}| < b^k \sum_{n=1}^{\infty} f(n+k) b^{-(n+k)} < \varepsilon \sum_{n=1}^{\infty} b^{-n} = \frac{\varepsilon}{b-1}.$$

Следовательно, $\lim_{k\to\infty} \gamma_k = 0$ и последовательность $(\gamma_k)_{k=0}$ имеет конечный предел. Из утверждения леммы 2.9 мы делаем следующий вывод: если последовательность $(\delta_k)_{k=0}^{\infty}$ равномерно распределена на интервале $[0,1)$, то последовательность $(\delta_k + \gamma_k \mod 1)_{k=0}^{\infty}$ также равномерно распределена на этом же интервале.

Из утверждения леммы 2.5, следует равенство

$$\{\alpha b^k\} = \{s_k(\alpha) b^k + \delta_k + \gamma_k\} = \{\delta_k + \gamma_k\},$$

поскольку $b^k s_k(\alpha) \in \mathbb{Z}$.

Таким образом, если последовательность $(\delta_k+\gamma_k \mod 1)_{k=0}^{\infty}$ равномерно распределена на интервале $[0,1)$, то и последовательность дробных долей $(\{\alpha b^k\})_{k=0}^{\infty}$ также равномерно распределена на том же интервале. Теперь, из утверждения теоремы 2.II, сразу следует доказательство теоремы 2.6. □

§ 2.5.2. Методика статистического анализа

Пусть $(a_k)_{k=0}^n$ определенный для некоторого натурального n фрагмент последовательности коэффициентов представления действительного иррационального числа α вида (2.2) или (2.6) в системе счисления по основанию b.

Как было сказано ранее, см. § 2.1, стр. 138, перед использованием элементов последовательности $(a_k)_{k=0}^n$ в средствах защиты информации должен быть проведен динамический контроль, т.е. проверена статистическая гипотеза о равномерном распределении указанных элементов. Данная проверка может быть проведена двумя способами.

1. Каждый элемент последовательности $(a_k)_{k=0}^n$ может быть представлен в виде двоичного вектора фиксированной длины $\lceil \log_2 b \rceil$. После этого для двоичной последовательности, образованной векторами, соответствующими коэффициентам последовательности $(a_k)_{k=0}^n$, должна быть проверена статистическая гипотеза о равномерном распределении последовательности из единиц и нулей. Очевидно, что такая проверка даст адекватный ответ в случае, когда $b = 2^u$, $u \in \mathbb{N}$, – есть некоторая степень двойки.

   В настоящее время разработано достаточно большое количество статистических критериев для проверки гипотезы о равномерном распределении двоичных последовательностей. Как правило, такие критерии формируются авторами в группы («батареи») и применяются одновременно. Если хотя бы один из критериев отвергает гипотезу о равномерном распределении, то применение последовательности последовательности $(a_k)_{k=0}^n$ полагается нецелесообразным.

   Среди применяемых на практике наборов статистических критериев стоит отметить:

   - «батарею» тестов Д. Кнута, см. [303];
   - набор тестов, рекомендованный американским институтом национальных стандартов NIST, см. [244];
   - набор тестов, предложенный Дж. Марсальей, см. [153, 154], а также Интернет-ресурсы [49, 152];

   Обзоры указанных тестов могут быть найдены в работах [215, 240, 297].

2. Второй способ основывается на доказанной нами ранее теореме 2.9. Из утверждения теоремы следует, что нам достаточно проверить

статистическую гипотезу о равномерном распределении величин $(\delta_k)_{k=0}^{n}$, определяемых равенствами (2.24), на интервале $[0,1)$.

Для проверки данной гипотезы может быть применен ряд статистических критериев, например:

- критерий согласия А.Н. Колмогорова, см. [130];
- критерий χ-квадрат К. Пирсона, см. [192].

Обзоры и сравнение статистических критериев, позволяющих проверять гипотезу о равномерном распределении последовательностей на интервале $[0,1)$, могут быть найдены в монографиях [305, 313].

Заключение к § 2.5

В § 2.5 доказан критерий, из которого следует, что нормальность чисел вида (2.2) и (2.6) следует из равномерного распределения на интервале $[0,1)$ элементов последовательности величин $(\delta_k)_{k=0}^{\infty}$, определяемых равенствами (2.24).

Показано, что данный критерий может быть использован для проверки статистической гипотезы о равномерном распределении последовательности коэффициентов $(a_k)_{k=0}^{n}$ представления действительного иррационального числа α вида (2.2) или (2.6) в системе счисления по основанию b.

§ 2.6. Пример практического применения

В заключительном параграфе настоящей главы мы приведем пример практического применения разработанного подхода к генерации псевдослучайных последовательностей. Для этого мы рассмотрим задачу локальной аутентификации пользователя средства защиты информации.

§ 2.6.1. Локальная аутентификация пользователей

Традиционно, локальная аутентификация пользователей осуществляется при помощи пароля, см., например, [355], при этом средство защиты информации хранит в себе результат криптографического преобразования пароля. Схематично, процесс локальной аутентификации изображен на следующем рисунке.

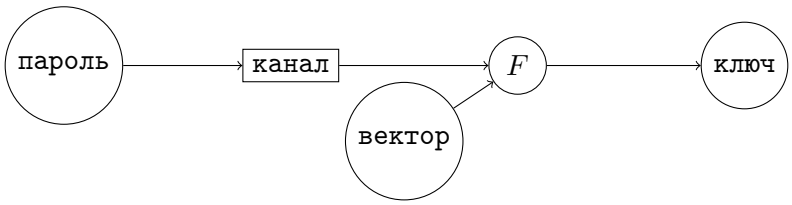


Рисунок 2.6.1. Схема локальной аутентификации.

В ходе локальной аутентификации пользователь передает пароль по каналу, не допускающему перехвата и искажений, после чего, средство защиты информации вырабатывает проверочное значение (ключ) и сравнивает выработанное значение с выработанным ранее.

Функция преобразования пароля F представляет собой отображение

$$F : \mathbb{V}_n \times \mathbb{V}_m \to \mathbb{V}_l,$$

для некоторых $n, m, l \in \mathbb{N}$.

В настоящее время для преобразования пароля используется алгоритм PDKDF2, регламентируемый рекомендациями [166, 371]. Данный алгоритм представляет собой многократное применение функции хеширования и считается криптографически стойким. Вместе с тем, он использует очень малый объем памяти, хранящей внутреннее состояние автомата, реализующего алгоритм. В связи с этим стала возможной практическая реализация атак, основанных на переборе паролей пользователей «по словарю» (тотального опробования паролей из заранее заданного множества, так называемого, словаря), использующая специальные вычислительные средства, построенные с использованием GPU (графических ускорителей) или FPGA (программируемых логических матриц), см. [78, 187, 217].

С целью защиты от подобного класса атак в конце 2012 года Ж.П. Омассон организовал конкурс PHC (Password Hashing Competition), на разработку нового алгоритма, который должен был заменить алгоритм PBKDF2. Конкурс был завершен в 2015 году. К новому алгоритму предъявлялись следующие требования, см. [191]:

- алгоритм должен представлять собой отображение

$$F : \mathbb{V}_n \times \mathbb{V}_m \times \mathbb{N} \to \mathbb{V}_l, \quad n, m, l \in \mathbb{N},$$

  где

  – $\mathbb{V}_n$ пространство двоичных векторов длины $n = 1024$, являющееся множеством возможных паролей,

  – $\mathbb{V}_m$ пространство инициализационных векторов для $m = 128$,

  – множество $\mathbb{N}$ содержит значения натурального параметра алгоритма, определяющего время работы;

- алгоритм должен быть простым, математически обоснованным и, по-возможности, использовать внутри себя минимальное число других криптографических алгоритмов;
- отображение F должно вести себя как случайное отображение, в частности, его результат должен быть статистически неотличим от случайной равномерно распределенной на $\mathbb{V}_k$ величины;
- использование оптимизированных реализаций на CPU, GPU или FPGA не должно приводить к существенному снижению времени реализации отображения F по сравнению с реализацией, применяемой в средстве защиты информации;
- алгоритм должен допускать реализацию, имеющую меры защиты от утечек по побочным каналам (временные атаки, атаки на кэш-память вычислительного средства) и, в частности, скрывать длину используемого пароля.

Разработанный автором и представленный в работе [335] алгоритм удовлетворяет перечисленным выше условиям. Поскольку к моменту окончания разработки алгоритма срок подачи заявок на конкурс истек, то алгоритм не был включен в число претендентов. В настоящее время алгоритм реализован в составе программного средства криптографической защиты информации `libakrypt`, см. [144].

§ 2.6.2. Алгоритм преобразования парольной информации

Алгоритм преобразования парольной информации представляет собой отображение

$$F(p, s, b, n, k) \to \mathbb{V}_l,$$

где

- $p \in \mathbb{V}_\infty$ – пароль, представленный в виде двоичной последовательности произвольной, отличной от нуля длины;
- $s \in \mathbb{V}_\infty$ – произвольная двоичная последовательность,
- $b \in \mathbb{N}, b > 1$ – четное натуральное число, основание используемой в алгоритме системы счисления,
- $n \in \mathbb{N}$ – количество шагов алгоритма,
- $l \in \mathbb{N}$ – длина вырабатываемой двоичной последовательности.

Алгоритм преобразования парольной информации использует регламентируемую стандартом ГОСТ Р 34.11-2012 функцию хэширования с длиной хэш-кода 512 бит, см. [281], и описывается следующим образом.

Алгоритм 2.4: Алгоритм преобразования парольной информации

Вход : Определенные выше параметры b, p, s, n, l.
Выход : Двоичная последовательность $\gamma_1, \ldots, \gamma_l$.

1 Используя ГОСТ Р 34.11-2012 вычислить двоичную последовательность $h_0, \ldots, h_{511} = \mathtt{Hash}_{512}(p)$ длины 512 бит, представляющую собой хэш-код пароля.

2 Определить последовательность из 32-х целых, неотрицательных чисел x_k

$$x_k = \sum_{i=0}^{15} h_{16*k+i} 2^i, \quad k = 0, \ldots, 31.$$

3 Определить натуральное число $c = 1$.

4 Используя ГОСТ Р 34.11-2012 вычислить двоичную последовательность $s_0, \ldots, s_{511} = \mathtt{Hash}_{512}(s||c)$, представляющую собой хэш-код входной последовательности s.

5 Определить последовательность из 32-х рациональных чисел u_k

$$u_k = \frac{1}{p_k} \mathtt{max} \left\{ 1, \sum_{i=0}^{15} s_{16*k+i} 2^i \right\}, \quad k = 0, \ldots, 31,$$

где p_k – последовательность произвольных, попарно различных простых чисел.

6 Сформировать иррациональное число α вида (2.2)

$$\alpha = \sum_{n=0}^{\infty} \left(\frac{u_0}{n2^{16} + x_0} + \cdots + \frac{u_{31}}{n2^{16} + x_{31}} \right) b^{-n}.$$

7 Используя алгоритм 2.2.1, см. стр. 157, вычислить последовательность $a_0, a_1, \ldots, a_n, a_{n+1}, \ldots, a_{n+l}$ коэффициентов числа α в системе счисления по основанию b.

8 Используя описанную в предыдущем параграфе методику проверить, что выработанная последовательность $a_1, \ldots, a_{n+l}$ статистически неотличима от реализации случайной, равновероятно распределенной на интервале $[0, b)$ величины. В случае неприятия данной гипотезы, вычислить значение $c = c + 1$ и вернуться на шаг 3.

9 Вернуть в качестве результата работы алгоритма двоичную последовательность $\gamma_1, \ldots, \gamma_l$, где величины γ_k определены равенствами

$$\gamma_k = \begin{cases} 0, & \text{если} \quad 0 \leqslant a_{n+k} < \frac{b}{2}, \\ 1, & \text{если} \quad \frac{b}{2} \leqslant a_{n+k} < b. \end{cases}$$

Добавим, что в [144] использовалось значение $b = 256$, а последовательность простых чисел p_k, участвующих на 5-м шаге в определении коэффициентов u_k, определялась следующим образом

```
 67,  71,  73,  79,  83,  89,  97, 101, 103, 107, 109, 113, 127, 131, 139, 149,
151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233.
```

Основываясь на результатах предыдущих параграфов, можно сделать вывод, что предложенный алгоритм соответствует сформулированным в начале параграфа требованиям и обладает следующими свойствами:

- последовательность $\gamma_1, \ldots, \gamma_l$ статистически неотличима от реализации равновероятно распределенной последовательности из нулей и единиц;

- восстановление неизвестных коэффициентов x_k числа α может быть сведено к перебору неизвестных значений $a_1, \ldots, a_n$ и последующему применению алгоритма из § 2.4.2; в связи с этим, величина n должна выбираться таким образом, чтобы трудоемкость опробования всех возможных значений $a_1, \ldots, a_n$ превышала трудоемкость перебора множества всех возможных паролей.

- при опробовании величин $a_1, \ldots, a_n$ алгоритм восстановления сводится к поиску целочисленных соотношений для $2^{16} + 1$ чисел

$$\alpha, \quad \xi_1 = \sum_{n=0}^{\infty} \frac{1}{n2^{16}+1}, \quad \xi_2 = \sum_{n=0}^{\infty} \frac{1}{n2^{16}+2}, \quad \ldots$$

$$\ldots, \quad \xi_{2^{16}} = \sum_{n=0}^{\infty} \frac{1}{2^{16}(n+1)},$$

  и проверке того, что коэффициенты найденных соотношений совпадают с последовательностью $1, u_1, \ldots, u_{16}$;

- использование алгоритма хэширования не увеличивает сложность определения неизвестных значений x_k, $k = 0, \ldots, 31$, но выравнивает время работы алгоритма и скрывает от нарушителя длину используемого пароля.

Заключение к § 2.6

В § 2.6 описан алгоритм преобразования парольной информации, основанный на представлении чисел вида (2.2) в системе счисления по заданному основанию. Показано, что алгоритм 2.13 удовлетворяет современным требованиям, предъявляемым к алгоритмам преобразования парольной информации и может быть использован для локальной аутентификации пользователей средств защиты информации.

глава 3

Равновероятные сжимающие отображения и их приложения

В настоящей главе излагаются результаты исследований, позволившие обосновать целесообразность применения в средствах защиты информации равновероятных сжимающих отображений, представляющих собой линейные формы от значений взаимно-однозначных функций.

В первом параграфе рассматриваются примеры равновероятных сжимающих отображений, а также приводится обзор известных результатов.

Во втором параграфе определяется новый класс ключевых сжимающих отображений и доказываются теоремы о равновероятности выходных значений построенных отображений. В третьем параграфе рассматриваются примеры практического применения данного класса отображений для реализации режима аутентифицированного шифрования.

Изложенные в настоящей главе результаты опубликованы в следующих работах автора [177, 178, 180, 272, 312, 336, 337, 339], из которых четыре работы входят в перечень рецензируемых научных изданий ВАК.

§ 3.1. Необходимые определения и обзор известных результатов

Рассмотрим непустые конечные множества — множество сообщений $\mathbb{S}$, множество ключей $\mathbb{K}$ и множество $\mathbb{A}$, элементы которого будем называть кодами целостности. Будем считать, что для указанных множеств существуют эффективно вычислимые вложения

$$\mathbb{S} \subset \mathbb{V}_\infty, \quad \mathbb{K} \subset \mathbb{V}_\infty, \quad \mathbb{A} \subset \mathbb{V}_\infty.$$

Применительно к средствам защиты информации, рассматриваются два вида сжимающих отображений, см. [161, 202, 316, 347]:

- функции хэширования – так называемые бесключевые сжимающие отображения,
- функции выработки имитовставки – сжимающие отображения, зависящие от двух аргументов – собственно сообщения $x \in \mathbb{S}$ и секретного ключа $k \in \mathbb{K}$.

Указанные отображения используются, как правило, для выработки кодов целостности, гарантирующих неизменность информации при ее хранении и/или передаче по каналам связи. В средствах защиты информации принято использовать бесключевые сжимающие отображения, удовлетворяющие следующему определению, см. [347].

Определение 3.1. *Сжимающее отображение*

$$h(x) : \mathbb{S} \to \mathbb{A}, \tag{3.1}$$

принято называть функцией хэширования, если выполнены следующие условия.

1. *Отображение h должно быть однонаправленным — для любого кода целостности $a \in \mathbb{A}$ задача вычисления прообраза функции h, т.е. определения какого-либо сообщения $x \in \mathbb{S}$ такого, что $h(x) = a$, должна являться трудноразрешимой.*

2. *Задача построения коллизии, т.е. определения двух произвольных значений $x_1, x_2 \in \mathbb{S}$ таких, что*

$$h(x_1) = h(x_2), \quad x_1 \neq x_2, \tag{3.2}$$

должна являться трудноразрешимой.

3. *Задача построения второго прообраза, т.е. задача нахождения для заданного элемента $x_1 \in \mathbb{S}$ и, соответственно, значения кода целостности $a = h(x_1)$ какого-либо элемента $x_2 \in \mathbb{S}$, удовлетворяющего условию (3.2), должна являться трудноразрешимой.*

Значение функции $h(x)$ принято называть хэш-кодом или кодом целостности сообщения x.

Принципы, лежащие в основе построения функций хэширования, изложены в работах [161, 202, 347], см. также [119, 262]. Примерами функций хэширования являются алгоритмы «SHA» [80], «Keccak» [84, 241] и «Стрибог» [281].

В 1939 году Р. Фон Мизесом был сформулирован, так называемый, «парадокс дней рождений», см. [310, 383], описывающий совпадение элементов одной выборки некоторой дискретной случайной величины, имеющей равновероятное распределение. Из результатов работ [91, 188] следует, что существует вероятностный алгоритм, основанный на данном парадоксе, который находит решение задачи о построении коллизии для любой функции хэширования с вероятностью успеха не менее $\frac{1}{2}$ и трудоемкостью, оцениваемой величиной $c\sqrt{\frac{\pi|\mathbb{A}|}{2}}$ операций вычисления функции хэширования, для некоторой действительной константы $c > 1$.

Для сжимающих отображений, результат действия которых зависит от секретного ключа, дадим следующее определение, см. [347].

Определение 3.2. *Сжимающее отображение*

$$h(k, x) : \mathbb{K} \times \mathbb{S} \to \mathbb{A}. \tag{3.3}$$

принято называть ключевой функцией хэширования или функцией вычисления имитовставки, если выполнены следующие условия.

1. *Для любого неизвестного значения $k \in \mathbb{K}$ функция $h_k(x) = h(k, x)$ должна являться функцией хэширования и удовлетворять определению 3.1.*

2. *Пусть $t \in \mathbb{N}$ и задано множество пар (x_i, a_i), где $x_i \in \mathbb{S}$, $a_i \in \mathbb{A}$, $i = 1, \ldots, t$. Тогда задача решения системы уравнений*

$$\begin{cases} h(k, x_1) = a_1, \\ \ldots \\ h(k, x_t) = a_t, \end{cases}$$

относительно неизвестного ключа имитозащиты $k \in \mathbb{K}$ должна являться трудноразрешимой.

Значение функции $h(k, x)$ принято называть имитовставкой или кодом аутентичности сообщения x, а ключ $k \in \mathbb{K}$ – ключом имитозащиты.

Максимальное значение величины t, при котором задача определения ключа имитозащиты k является трудноразрешимой, принято называть «допустимым объемом материала» для функции хэширования.

Примером функции, удовлетворяющей данному определению является алгоритм «CMAC», см. [121], принятый в качестве национального стандарта Российской Федерации, см. [283]. Допустимый объем материала для алгоритма «CMAC» регламентируется рекомендациями по стандартизации [353].

В ряде случаев требования к ключевой функции хэширования могут быть усилены следующим образом, см. [202].

Определение 3.3. *Сжимающее отображение (3.3) принято называть ключевой функцией хэширования или функцией вычисления имитовставки, если оно удовлетворяет определению 3.2 и для любого известного ключа имитозащиты $k \in \mathbb{K}$ задачи*

1. *построения коллизии,*

2. нахождения второго прообраза для функции $h_k(x) = h(k, x)$,

являются трудноразрешимыми.

Требования, содержащееся в определении 3.3, не всегда накладываются на ключевую функцию хэширования, так в работах [347, § 13.2], [383, стр. 33], данное требование считается избыточным, а работе [202] — наоборот. В случае выполнения определения 3.3 владелец секретного ключа имитозащиты не может подделать значение ключевой функции хэширования, а сама функция может быть использована для обеспечения свойства «невозможности отказа от совершенных действий» (см., далее, раздел 4.4.1, свойство 24). Примером функции, удовлетворяющей определению 3.3, является алгоритм «HMAC», см. [24, 372].

§ 3.1.1. Универсальные функции хэширования

Следуя монографии Б. Принеля, см. [202], определим отдельный класс бесключевых функций хэширования.

Определение 3.4. *Бесключевая функция хэширования (3.1) называется универсальной, если для любого $a \in \mathbb{A}$ количество значений $x \in \mathbb{S}$ таких, что $h(x) = a$ равно*

$$\left\lfloor \frac{|\mathbb{S}|}{|\mathbb{A}|} \right\rfloor + \delta_a,$$

где $\delta_a = 0$, если $|\mathbb{A}|$ делит $|\mathbb{S}|$ нацело, и $\delta_a \in \{0, 1\}$ – иначе.

Будем говорить, что универсальная функция обладает свойством равновероятности выходных значений.

В отечественных публикациях для универсальных функций также используется термин – «равновероятная» функция, см. [316, гл 5].

Свойство равновероятности выходных значений является важным при построении коллизий – из результатов работы [91] следует, что трудоемкость алгоритма построения коллизий, основанного на «парадоксе дней рождений», для универсальных функций не может быть снижена.

Для ключевых функций хэширования свойство универсальности должно рассматриваться отдельно по каждому аргументу — сжимаемому сообщению $x \in \mathbb{S}$ и ключу $k \in \mathbb{K}$.

Определение 3.5. *Ключевую функцию хэширования (3.3)*

$$h(k, x) : \mathbb{K} \times \mathbb{S} \to \mathbb{A}.$$

принято называть универсальной ключевой функцией хэширования относительно сжимаемых сообщений, если для любого фиксированного

ключа имитозащиты $k \in \mathbb{K}$ функция $h_k(x) = h(k,x)$ является универсальной бесключевой функцией хэширования в смысле определения 3.4.

При исследовании свойства универсальности относительно ключей имитозащиты будем пользоваться следующими определениями, см. работы [54, 245].

Определение 3.6. *Будем говорить, что ключевая функция хэширования (3.3) называется ε-универсальной функцией хэширования, если найдется действительное число ε такое, что $0 < \varepsilon \leqslant 1$, и для любых двух различных элементов $x, y \in \mathbb{S}$ выполнена оценка*

$$|k \in \mathbb{K} : \ h(k,x) = h(k,y)| \leqslant \varepsilon |\mathbb{K}|.$$

Если выполнено равенство $\varepsilon = |\mathbb{A}|^{-1}$, то функция $h(k,x)$ называется универсальной ключевой функцией хэширования.

Из определения 3.6 следует, что для универсальной функции хэширования, при случайном выборе ключа k, вероятность появления коллизии на двух различных сообщениях x, y не превосходит величины $|\mathbb{A}|^{-1}$.

Определение 3.7. *Мы будем говорить, что ключевая функция хэширования (3.3) называется строго ε-универсальной функцией хэширования, если выполнены следующие условия:*

1. *если для любого сообщения $x \in \mathbb{S}$ функция $h_x(k) = h(k,x)$ является универсальной бесключевой функцией хэширования в смысле определения 3.4;*

2. *для любых различных $x, y \in \mathbb{S}$ и любых $a, b \in \mathbb{A}$ количество ключей $k \in \mathbb{K}$ для которых одновременно выполнены равенства*

$$h(k,x) = a, \quad h(k,y) = b$$

не превосходит величины $\varepsilon \dfrac{|\mathbb{K}|}{|\mathbb{A}|}$, где $0 < \varepsilon \leqslant 1$.

Если выполнено равенство $\varepsilon = |\mathbb{A}|^{-1}$, то функция $h(k,x)$ называется строго универсальной ключевой функцией хэширования.

Для строго ε-равновероятных функций значение ε мажорирует величину условной вероятности появления события $h(k,y) = b$ при выполнении события $h(k,x) = a$.

Известно несколько подходов к построению ключевых функций хэширования:

- на основе алгоритмов блочного шифрования, см. [37, 121],
- криптографических функций хэширования, см. [24, 372],
- универсальных функций хэширования, см. [202, 347].

В настоящей главе рассматривается последний подход, ведущий свое начало от работ Дж. Картера и М. Вегмана [54, 256], а также Д. Стинсона [245]. Позднее подход развивался при построении ключевых функций хэширования в ряде работ, см. [11, 77, 102, 172, 254], а также в работах автора [272, 312, 336].

Приведем примеры строгих ε-равновероятных функций. Будем считать, что множества $\mathbb{S}$ и $\mathbb{K}$ совпадают и состоят из векторов длины t, т.е.

$$\mathbb{S} = \{(x_1, \dots, x_t)\}, \quad \mathbb{K} = \{(k_1, \dots, k_t)\},$$

где координаты векторов принадлежат некоторому кольцу $\mathbb{B}$. Будем считать, что $\mathbb{A}$ также является некоторым кольцом.

1. «MMH», см. [102]. Определим $\mathbb{A} = \mathbb{B} = \mathbb{Z}_{2^{32}}$ и $p = 2^{32} + 15$ простое число, тогда

$$h(k, x) \equiv \left(\left(\sum_{n=1}^{t} k_n x_n \pmod{2^{64}} \right) \pmod{p} \right) \pmod{2^{32}}.$$

2. «Square Hash», см. [77]. Пусть $w \geqslant 2$ натуральное число, p максимальное простое число меньшее, чем 2^w. Определим $\mathbb{A} = \mathbb{B} = \mathbb{Z}_{2^w}$, тогда

$$h(k, x) \equiv \sum_{n=1}^{t} (k_i + x_i)^2 \pmod{p}.$$

3. «NMH», см. [102, 256]. Пусть $t = 2l$, $l \in \mathbb{N}$. Определим $\mathbb{A} = \mathbb{B} = \mathbb{Z}_{2^{32}}$ и $p = 2^{32} + 15$ простое число, тогда

$$h(k, x) \equiv \left(\sum_{n=1}^{l} (k_{2n-1} + x_{2n-1})(k_{2n} + x_{2n}) \pmod{p} \right) \pmod{2^{32}}.$$

4. «NH» (алгоритм «UMAC»), см. [135, 254]. Пусть $w \in \mathbb{N}$, $w \geqslant 2$, $t = 2l$, $l \in \mathbb{N}$. Определим $\mathbb{B} = \mathbb{Z}_{2^w}$, $\mathbb{A} = \mathbb{Z}_{2^{2w}}$, тогда

$$h(k, x) \equiv \sum_{n=1}^{l} z_{2n-1} z_{2n} \pmod{2^{2w}},$$

где $z_n \equiv k_n + x_n \pmod{2^w}$.

5. «Badger», см. [11]. Пусть $t = 4l$, $l \in \mathbb{N}$. Определим $\mathbb{B} = \mathbb{Z}_{2^{32}}$, $\mathbb{A} = \mathbb{Z}_{2^{64}}$, тогда

$$h(k, x) \equiv \sum_{n=1}^{l} \left(z_{4n-3} z_{4n-2} + v_n\right) \pmod{2^{64}},$$

где $z_n \equiv k_n + x_n \pmod{2^{32}}$, $v_n = x_{4n-1} + 2^{32} x_{4n}$.

Отметим, что все перечисленные отображения используют операцию модульного умножения, которая обладает рядом свойств, снижающих равновероятные свойства функции $h(k, x)$. Проиллюстрируем это на примере функции «NH» (алгоритм «UMAC»). Зафиксируем произвольную пару значений $x_{2n-1}, x_{2n} \in \mathbb{Z}_{2^w}$, вычет $a \in \mathbb{Z}_{2^{2w}}$ и будем считать, что для некоторого индекса n выполнено равенство

$$\left(k_{2n-1} + x_{2n-1} \pmod{2^w}\right)\left(k_{2n} + x_{2n} \pmod{2^w}\right) \equiv a \pmod{2^{2w}}, \tag{3.4}$$

Тогда выполнены следующие свойства.

1. *Симметричность.* Если ключи k_{2n-1}, k_{2n} удовлетворяют равенству (3.4), то ключи

$$\begin{aligned} k'_{2n-1} &\equiv k_{2n} + x_{2n} - x_{2n-1} \pmod{2^w}, \\ k'_{2n} &\equiv k_{2n-1} + x_{2n-1} - x_{2n} \pmod{2^w}, \end{aligned}$$

также удовлетворяют равенству (3.4). Очевидно, что если величины x_{2n-1} и x_{2n} различны, то пары ключей (k_{2n-1}, k_{2n}) и (k'_{2n-1}, k'_{2n}) также различны.

2. *Неоднозначность разложения числа a на множители.* Пусть a есть составное число и p, q есть какая-либо пара делителей числа $a = pq$, удовлетворяющая условиям $p < 2^w$, $q < 2^w$. Тогда, с учетом первого свойства, каждой такой паре делителей p, q будет соответствовать две пары ключей k_{2n-1}, k_{2n}, для которых выполнено равенство (3.4).

   Например, для $w = 4$, $x_{2n-1} = 0$, $x_{2n} = 0$ и $a = 12$ ключевыми парами, удовлетворяющими сравнению

$$(k_{2n-1} + x_{2n-1})(k_{2n} + x_{2n}) \equiv 12 \pmod{2^8 = 256},$$

   будут $(1, 12), (12, 1), (2, 6), (6, 2), (3, 4), (4, 3)$.

3. *Существование запретных значений.* Если величина a является простым числом в интервале $2^w < a < 2^{2w}$, то она не может являться произведением двух чисел чисел, меньших 2^w.

4. *Проблема нуля.* Значение $a = 0$ достигается на значительно большем числе пар (k_{2n-1}, k_{2n}), чем все остальные значения a. Действительно, если один из сомножителей в произведении (3.4) равен нулю, то значение второго сомножителя может принимать любое допустимое значение. Так, в приведенном выше примере, равенство

$$(k_{2n-1} + x_{2n-1})(k_{2n} + x_{2n}) \equiv 0 \ \left(\mathrm{mod}\, 2^8 = 256\right)$$

достигается на следующих ключевых парах

$$\underbrace{(0,0),(0,1),\ldots,(0,15),(1,0),(2,0),\ldots,(15,0)}_{31\text{ пара}}.$$

Данное свойство, впервые, использовалось в работе [104] для определения множества, так называемых, «слабых» ключей. Позднее, автором в работе [312] был предложен алгоритм, использующий данное свойство и позволяющий строить коллизии с трудоемкостью меньшей, чем у метода, основанного на «парадоксе дней рождений».

В параграфе § 3.2 рассматривается предложенный автором класс отображений, являющийся обобщением рассмотренных функций и позволяющий строить равновероятные сжимающие отображения.

§ 3.1.2. Аутентифицированное шифрование

Одним из способов обеспечения конфиденциальности хранимой или передаваемой по каналам связи информации, является ее шифрование. Напомним следующие определения, см. [316, 347].

Определение 3.8. *Пусть $w \in \mathbb{N}$ и $x, c \in \mathbb{V}_w \subset \mathbb{S}$, $k \in \mathbb{K}$. Рассмотрим отображения*

$$E_k(x): \quad \mathbb{K} \times \mathbb{V}_w \to \mathbb{V}_w, \tag{3.5}$$

и

$$D_k(c): \quad \mathbb{K} \times \mathbb{V}_w \to \mathbb{V}_w, \tag{3.6}$$

такие, что равенство $D_k(E_k(x)) = x$ выполнено для любой пары значений $k \in \mathbb{K}$ и $x \in \mathbb{V}_w$.

Будем говорить, что пара отображений (3.5), (3.6) определяет блочный шифр, если выполнены следующие условия.

1. *При фиксированном и неизвестном значении ключа $k \in \mathbb{K}$ задача дешифрования, т.е. определения сообщения $x \in \mathbb{V}_w$ при известном значении $c = E_k(x)$ и известном наборе сообщений $x_1, \ldots, x_t$ и соответствующих им значений $c_1, \ldots, c_t$, где $c_i = E_k(x_i)$, $i = 1, \ldots, t$,*

должна являться трудноразрешимой для максимально возможного натурального значения t.

2. *Задача определения неизвестного ключа шифрования $k \in \mathbb{K}$ при известном наборе сообщений $x_1, \ldots, x_t$ и соответствующих им значений $c_1, \ldots, c_t$, где $c_i = E_k(x_i)$, $i = 1, \ldots, t$, должна является трудноразрешимой для максимально возможного натурального значения t.*

Величину w принято называть длиной блока шифрования, величину k – ключом шифрования. Максимально возможное значение параметра t принято называть «допустимым объемом материала» для блочного шифра, заданного отображениями (3.5) и (3.6).

Примерами блочных шифров служат алгоритмы «AES» [83], «Магма» и «Кузнечик» [282]. Обзор известных к настоящему времени блочных шифров может быть найден в монографии [348].

Для шифрования сообщений произвольной длины принято использовать алгоритмы, использующие блочный шифр в качестве вспомогательного преобразования, — режимы работы блочных шифров. Примерами таких алгоритмов могут служить режимы, регламентируемые в [283, 356].

В последние годы наиболее востребованными режимами работы блочных шифров являются режимы, обеспечивающие аутентифицированное шифрование, т.е. реализующие одновременный процесс шифрования и имитозащиты данных, см. [25, 170, 364], а также [316, § 8.2.4].

Определение 3.9. *Пусть $v \in \mathbb{N}$, $x, y, c \in \mathbb{S}$, $k_1, k_2 \in \mathbb{K}$ и $a \in \mathbb{A}$. Определим множество значений булевых переменных $\mathbb{V}_1 = \{\texttt{true}, \texttt{false}\}$ и рассмотрим отображения*

$$\texttt{authenc}(k_1, k_2, iv, y, x) = \{c, a\} : \quad \mathbb{K} \times \mathbb{K} \times \mathbb{V}_v \times \mathbb{S} \times \mathbb{S} \to \mathbb{S} \times \mathbb{A}, \tag{3.7}$$

$$\texttt{authdec}(k_1, k_2, iv, y, c, a) = \{x, b\} : \quad \mathbb{K} \times \mathbb{K} \times \mathbb{V}_v \times \mathbb{S} \times \mathbb{S} \times \mathbb{A} \to \mathbb{S} \times \mathbb{V}_1. \tag{3.8}$$

Будем называть k_1 – ключом шифрования, k_2 – ключом имитозащиты, x – зашифровываемым сообщением, для которого обеспечивается конфиденциальность и целостность, y – ассоциированными (дополнительными) данными, для которых обеспечивается только целостность, iv – синхропосылкой (инициализационным вектором), a – кодом целостности, c – шифрекстом.

Будем говорить, что отображение (3.7) зашифровывает сообщение x и вычисляет код аутентичности (имитовставку) сообщений x, y, а отображение (3.8) расшифровывает шифртекст c, вычисляет код аутентичности (имитовставку) сообщений x, y и, в случае совпадения вычисленного значения с аргументом a, возвращает истину (`true`*).*

Пусть $r \in \mathbb{N}$ *и заданы равенства*

$$\textbf{\textit{authenc}}(k_1, k_2, iv_i, y_i, x_i) = \{c_i, a_i\}, \quad i = 1, \ldots, r, \tag{3.9}$$

такие, что

$$\left\lceil \frac{1}{w} \sum_{i=1}^{r} \mathrm{len}_2(x_i) \right\rceil \leqslant t,$$

т.е. суммарная длина сообщений x_i *не превосходит максимального объема материала* t *для блочного шифра с длиной блока* w*.*

Будем говорить, что пара отображений (3.7), (3.8) определяет режим аутентифицированного шифрования при выполнении следующих условий.

1. *Для любых* $k_1, k_2 \in \mathbb{K}$, $iv \in \mathbb{V}_v$ *и* $y \in \mathbb{S}$ *выполнено равенство*

$$\textbf{\textit{authdec}}(k_1, k_2, iv, y, \textbf{\textit{authenc}}(k_1, k_2, iv, y, x)) = \{x, \textbf{\textit{true}}\}.$$

2. *Задача дешифрования, т.е. определения сообщения* $x \in \mathbb{S}$ *такого, что*

$$\textbf{\textit{authenc}}(k_1, k_2, iv, y, x) = \{c, a\}$$

при известных значениях $k_2 \in \mathbb{K}$, $iv \in \mathbb{V}_v$, $y, c \in \mathbb{S}$, $a \in \mathbb{A}$ *и известном наборе значений (3.9), должна являться трудноразрешимой.*

3. *Задача определения неизвестного ключа шифрования* $k_1 \in \mathbb{K}$ *при известном значении* $k_2 \in \mathbb{K}$ *и известном наборе значений (3.9), должна являться трудноразрешимой.*

4. *При фиксированных значениях* $k_1 \in \mathbb{K}$ *и* $iv \in \mathbb{V}_v$ *функция*

$$h(k_2, x||y) = \textbf{\textit{authenc}}(k_1, k_2, iv, y, x)$$

должна являться ключевой функцией хэширования (в смысле определения 3.2).

Использование двух различных ключей – шифрования и имитозащиты, позволяет отнести к режимам аутентифицированного шифрования не только специализированные алгоритмы, например, режимы [178, 259, 364], но и комбинации классических режимов шифрования и вычисления имитовставки. Стоит отметить, что достаточно часто в режимах аутентифицированного шифрования ключи шифрования и имитозащиты полагают равными, см., например, работы [100, 184, 223, 259].

В англоязычной литературе для режимов аутентифицированного шифрования принято использовать обозначение «AEAD» (Authenticated

Encryprion with Associated Data). Как представляется, впервые понятие аутентифицированного шифрования как отдельного режима, а не комбинации двух независимых преобразований, было рассмотрено в 2000 году в работе [25]. Немного позднее, был предложен ряд режимов, в частности, режимы «OCB» [185] (позднее получивший название «OCB1»), и «GCM» [158]. Следует также отметить регламентируемый в RFC3610 режим «CCM», см. [157, 257]. Обзор других режимов аутентифицированного шифрования может быть найден на сайте NIST, см. [170].

Режим «OCB1» был запатентован авторами, что не позволило ему получить какое-либо существенное применение в средствах защиты информации. Второй режим – «GCM», наоборот, был включен в ряд стандартов, см. [110, 113, 116], и получил широчайшее распространение.

Применяемая в режиме «GCM» функция выработки кода аутентификации, реализует схему Горнера вычисления значений многочлена в конечном поле; при этом коэффициенты многочлена определяются сжимаемым сообщением. Исследуя такой, называемый «полиномиальным», способ вычисления имитовставки Н. Фергюссон [89] и, позднее, М. Сааринен [218], предложили несколько атак на функцию аутентификации режима «GCM».

Отсутствие стойкого режима аутентифицированного шифрования привело к постановке в 2013 году конкурса «CAESAR» на разработку нового режима, см. [52]. Данный конкурс завершился только в 2019 году, одним из его победителей стал режим «OCB3», см. [136].

В Российской Федерации, независимо от конкурса «CAESAR», исследования по разработке режима аутентифицированного шифрования начались в 2015 году. В результате было предложено три режима:

- режим «XTSMAC», разработанный в 2016 году А. Нестеренко, см. работы [177, 178],
- режим MGM, разработанный в 2017 году В. Ноздруновым, см. работы [184, 223],
- режим «Нефрит», разработанный в 2017 году А. Бабуевой и А. Науменко, см. [271].

Первые два режима используют для вычисления имитовставки линейные формы от перестановок на множестве кодов аутентификации, и являются частными случаями рассмотренного в § 3.2 сжимающего отображения. Третий режим использует модифицированное полиномиальное преобразование. В 2019 году режим «MGM» был включен в состав рекомендаций по стандартизации Р 1323565.1.026-2019 [364], а в 2021 году — принят в качестве рекомендаций RFC 9058 [169].

§ 3.2. Равновероятные ключевые функции

Изложение этого параграфа следует работе [336].

Пусть множество кодов аутентичности $\mathbb{A}$ есть конечная аддитивная абелева группа (в практических приложениях, как правило, в качестве множества $\mathbb{A}$ выступает конечномерное векторное пространство $\mathbb{V}_w$ двоичных векторов длины $w \in \mathbb{N}$ с операцией $\oplus$ — покоординатного сложения по модулю 2).

Зафиксируем натуральные числа l, s, u и рассмотрим конечное множество $\mathbb{B}$ такое, что $|\mathbb{A}| = |\mathbb{B}|^u$. Рассмотрим множество отображений

$$\pi_n : \mathbb{V}_u(\mathbb{B}) \to \mathbb{A},\ n = 1, \dots, l,$$

задающее взаимно-однозначное соответствие между векторным пространством $\mathbb{V}_u(\mathbb{B})$ и конечным множеством $\mathbb{A}$. Если множество $\mathbb{A}$ может быть эффективно представлено в виде $\mathbb{V}_u(\mathbb{B})$, то отображения $\pi_1, \dots, \pi_l$ являются перестановками на множестве $\mathbb{A}$.

Определим множество сообщений $\mathbb{S}$ и множество ключей $\mathbb{K}$ равенствами

$$\mathbb{S} = \mathbb{V}_{lu}(\mathbb{B}) = \{(x_1, \dots, x_{lu})\}, \quad \mathbb{K} = \mathbb{V}_{slu}(\mathbb{B}) = \{(k_1, \dots, k_{slu})\},$$

где координаты $x_1, \dots, x_{lu},\ k_1, \dots, k_{slu} \in \mathbb{B}$.

Рассмотрим сжимающее отображение

$$g(k_1, \dots, k_s, x) : \mathbb{V}_{s+1}(\mathbb{B}) \to \mathbb{B}, \tag{3.10}$$

удовлетворяющее следующим свойствам.

1. При фиксированном наборе значений $k_1, \dots, k_s \in \mathbb{B}$ отображение

   $$g(k_1, \dots, k_s, x) = g(x) : \mathbb{B} \to \mathbb{B},$$

   является взаимно-однозначным отображением множества $\mathbb{B}$ в себя.

2. При фиксированном значении $x \in \mathbb{B}$ и любом значении $z \in \mathbb{B}$ уравнение $g(k_1, \dots, k_s, x) = z$ имеет ровно $|\mathbb{B}|^{(s-1)}$ различных решений, относительно неизвестных $k_1, \dots, k_s$.

3. Зафиксируем произвольные элементы $x, y \in \mathbb{B}$ и будем считать, что вычеты $z, t \in \mathbb{B}$ пробегают множество всех возможных значений. Тогда суммарное число решений системы уравнений

   $$\begin{cases} g(k_1, \dots, k_s, x) = z, \\ g(k_1, \dots, k_s, y) = t, \end{cases} \tag{3.11}$$

   относительно неизвестных $k_1, \dots, k_s$, в точности равно $|\mathbb{B}|^s$.

Отметим, что при фиксированных значениях $z, t \in \mathbb{B}$, в силу второго свойства, система сравнений (3.11) может иметь от нуля до $|\mathbb{B}|^{(s-1)}$ решений, относительно неизвестных $k_1, \ldots, k_s$.

Можно привести несколько примеров отображения (3.10), удовлетворяющих приведенным требованиям.

Пусть $\mathbb{B} = \mathbb{Z}_{2^w}$ для некоторого натурального w. При $s = 1$ отображение

$$g(k_1, x) = k_1 + x \pmod{2^w},$$

удовлетворяет сформулированным свойствам, а при $s = 2$ им удовлетворяем отображение

$$g(k_1, k_2, x) = k_1 + (k_2 \oplus x) \pmod{2^w}, \tag{3.12}$$

где операция $\oplus$ означает побитовое сложение векторов, задающих[1] двоичное представление вычетов x и k_2.

Пусть $x = (x_1, \ldots, x_{lu}) \in \mathbb{S}$, $k = (k_1, \ldots, k_{slu}) \in \mathbb{K}$. Определим сжимающее отображение $h(x, k) : \mathbb{S} \times \mathbb{K} \to \mathbb{A}$

$$h(x, k) = \sum_{n=1}^{l} \pi_n(z_{(n-1)u+1}, \ldots, z_{nu-1}, z_{nu}), \tag{3.13}$$

где

$$z_i = g(k_{s(i-1)+1}, \ldots, k_{si}, x_i),$$

для всех $i = 1, \ldots, lu$. Схематично, процесс вычисления первого слагаемого суммы (3.13) изображен на следующем рисунке.

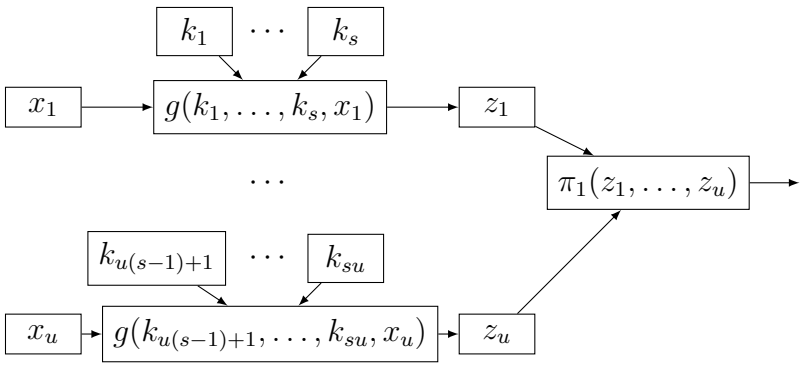


Рис. 3.1: Процесс вычисления слагаемого π_1.

Определенное в (3.13) отображение представляет собой класс ключевых функций хэширования, параметризуемый перестановками $\pi_1, \ldots, \pi_l$ и функцией g. Функции из данного класса могут рассматриваться как линейные формы от перестановок на множестве $\mathbb{A}$.

[1]Напомним, что вектор $(x_0, \ldots, x_{w-1})$ задает двоичное представление вычета $x \in \mathbb{Z}_{2^w}$, если $x = \sum_{n=0}^{w-1} x_n 2^n$.

§ 3.2.1. Свойство равновероятности относительно множества сообщений

Исследуем свойства введенного нами отображения (3.13) и покажем, что оно удовлетворяет определениям 3.5, 3.6 и 3.7.

Верна следующая теорема.

Теорема 3.1. *Пусть функция $h(x,k)$ определена равенством (3.13)*

$$h(x,k)=\sum_{n=1}^{l}\pi_n(z_{(n-1)u+1},\ldots,z_{nu-1},z_{nu}),$$

где $z_n=g(k_{s(n-1)+1},\ldots,k_{sn},x_n)$, для всех $n=1,\ldots,lu$. Тогда, для любого $a\in\mathbb{A}$ и любого $k=(k_1,\ldots,k_{slu})\in\mathbb{K}$ найдется ровно $|\mathbb{A}|^{(l-1)}$ элементов $x=(x_1,\ldots,x_{lu})\in\mathbb{S}$ таких, что $h(x,k)=a$.

Доказательство. Зафиксируем какое-либо значение $k=(k_1,\ldots,k_{su})$ и рассмотрим уравнение $\pi_1(z_1,\ldots,z_u)=a$ относительно неизвестных $x_1,\ldots,x_u$.

В силу взаимной однозначности отображения π_1 найдется ровно один вектор $(z_1,\ldots,z_u)$, удовлетворяющий данному уравнению. Далее, в силу взаимной однозначности отображения g, при фиксированных значениях $k_1,\ldots,k_{su}$ и $z_1,\ldots,z_u$ найдется ровно один вектор $(x_1,\ldots,x_u)$ такой, что

$$\begin{cases} g(k_1,\ldots,k_s,x_1) &= z_1,\\ &\ldots\\ g(k_{u(s-1)+1},\ldots,k_{su},x_u) &= z_u. \end{cases}$$

т.е., уравнение $\pi_1(z_1,\ldots z_u)=a$ имеет единственное решение, относительно неизвестных $x_1,\ldots,x_u$.

Теперь рассмотрим равенство $\sum_{n=1}^{l}a_n=a$, где

$$a_n=\pi_n(z_{(n-1)u+1},\ldots,z_{nu-1},z_{nu}),\ n=1,\ldots,l.$$

Этому равенству удовлетворяют $|\mathbb{A}|^{(l-1)}$ векторов вида

$$\left(a_1,\ldots,a_{l-1},a-\sum_{n=1}^{l-1}a_n\right), \tag{3.14}$$

где $a_1,\ldots,a_{l-1}$ принимают произвольные значения из $\mathbb{A}$. Согласно сказанному выше, каждому вектору (3.14) соотвествует единственное решение $x_1,\ldots,x_{lu}$. Теорема доказана. □

Из утверждения теоремы сразу следует, что отображение (3.13) является равновероятной ключевой функцией хэширования относительно сжимаемых сообщений, см. определение 3.5, поскольку

$$|\mathbb{A}|^{(l-1)} = \frac{|\mathbb{A}|^l}{|\mathbb{A}|} = \frac{|\mathbb{B}|^{lu}}{|\mathbb{A}|} = \frac{|\mathbb{S}|}{|\mathbb{A}|}.$$

Вероятность выбора случайного сообщения $x \in \mathbb{S}$ с заданным значением функции $h(x,k) = a$ не зависит от выбора ключа $k \in \mathbb{K}$, значения кода аутентичности $a \in \mathbb{A}$ и равна $|\mathbb{A}|^{-1}$.

§ 3.2.2. Свойство равновероятности относительно множества ключей

Теорема 3.2. *Пусть функция $h(x,k)$ определена равенством (3.13)*

$$h(x,k) = \sum_{n=1}^{l} \pi_n(z_{(n-1)u+1}, \ldots, z_{nu-1}, z_{nu}),$$

где $z_n = g(k_{s(n-1)+1}, \ldots, k_{sn}, x_n)$, для всех $n = 1, \ldots, lu$. Тогда, для любого $a \in \mathbb{A}$ и любого $x = (x_1, \ldots, x_{lu}) \in \mathbb{S}$ найдется ровно $|\mathbb{B}|^{u(sl-1)}$ элементов $k = (k_1, \ldots, k_{slu}) \in \mathbb{K}$ таких, что $h(x,k) = a$.

Доказательство. Проведем доказательство индукцией по значению величины l. Пусть $l = 1$, тогда уравнению $\pi_1(z_1, \ldots, z_u) = a$, в силу взаимной однозначности отображения π_1, удовлетворяет ровно один вектор $(z_1, \ldots, z_u) \in \mathbb{V}_u(\mathbb{B})$.

Из второго свойства отображения g следует, что каждое из уравнений

$$g(k_{s(n-1)+1}, \ldots, k_{sn}, x_n) = z_n, \quad n = 1, \ldots, u,$$

имеет ровно $|\mathbb{B}|^{(s-1)}$ решение, относительно неизвестных $k_{s(n-1)+1}, \ldots, k_{sn}$. Таким образом, общее число векторов $k = (k_1, \ldots, k_{su})$, для которых выполнено равенство $h(k,x) = a$, равно $|\mathbb{B}|^{u(s-1)}$. Для $l = 1$ утверждение теоремы выполнено.

Теперь предположим, что утверждение теоремы выполнено для всех значений $1, 2, \ldots, l-1$. Покажем, что оно выполнено и для величины l. Запишем уравнение $h(k,x) = a$ в виде

$$\pi_l(z_{u(l-1)+1}, \ldots, z_{lu}) = a - \sum_{n=1}^{l-1} a_n,$$

где

$$a_n = \pi_n(z_{(n-1)u+1}, \ldots, z_{nu-1}, z_{nu}),\ n = 1, \ldots, l-1,$$
$$z_i = g(k_{s(i-1)+1}, \ldots, k_{si}, x_i),\ i = u(l-1)+1, \ldots, lu.$$

Согласно индуктивному предположению, для любого $a' \in \mathbb{A}$ найдется $|\mathbb{B}|^{u(s(l-1)-1)}$ векторов $k = (k_1, \ldots, k_{s(l-1)u})$ таких, что

$$\sum_{n=1}^{l-1} a_n = a'.$$

Если элемент a' пробегает все возможное множество значений из $\mathbb{A}$, то элемент $a_l = a - a'$ также пробегает то же множество значений. Для каждого такого значения a_l найдется ровно один вектор $(z_{(l-1)u+1}, \ldots, z_{lu})$, удовлетворяющий равенству $\pi_l(z_{u(l-1)+1}, \ldots, z_{lu}) = a_l$.

Аналогично сказанному выше, каждому такому вектору соответствует $|\mathbb{B}|^{u(s-1)}$ наборов $k_{s(l-1)u+1}, \ldots, k_{slu}$, для которых выполнено равенство $\pi_l(z_{u(l-1)+1}, \ldots, z_{lu}) = a_l$.

Таким образом, общее число векторов $k = (k_1, \ldots, k_{slu})$, для которых выполнено равенство $h(x, k) = a$ есть

$$|\mathbb{A}| \cdot |\mathbb{B}|^{u(s(l-1)-1)} \cdot |\mathbb{B}|^{u(s-1)} = |\mathbb{B}|^{u(1+s(l-1)-1+s-1)} = |\mathbb{B}|^{u(sl-1)}.$$

Теорема доказана. □

Из доказанной теоремы следует, что число ключей $k \in \mathbb{K}$ таких, что выполнено равенство $h(k, x) = a$ в точности равно

$$|\mathbb{B}|^{u(sl-1)} = \frac{|\mathbb{B}|^{slu}}{|\mathbb{B}|^u} = \frac{|\mathbb{K}|}{|\mathbb{A}|}.$$

Таким образом, функция $h(k, x)$ удовлетворяет первому свойству определения 3.7. Легко показать, что всякая функция удовлетворяющая этому свойству также удовлетворяет и определению 3.6.

Выберем два произвольных сообщения $x, y \in \mathbb{S}$, а также произвольное $a \in \mathbb{A}$. Обозначим $\mathbb{K}_1$ множество ключей, для которых выполняется равенство $h(k, x) = a$, и $\mathbb{K}_2$ — множество ключей, для которых выполняется равенство $h(k, y) = a$. Из первого свойства определения 3.7 следует, что выполнено равенство $|\mathbb{K}_1| = |\mathbb{K}_2| = \frac{|\mathbb{K}|}{|\mathbb{A}|}$. Тогда, мощность множества $\mathbb{K}_1 \cap \mathbb{K}_2$ ключей, на которых достигается коллизия $h(k, x) = h(k, y) = a$, не превосходит $\dfrac{|\mathbb{K}|}{|\mathbb{A}|}$. Таким образом

$$|k \in \mathbb{K} :\ h(k, x) = h(k, y)| \leqslant \frac{1}{|\mathbb{A}|} \cdot |\mathbb{K}|.$$

и функция $h(k,x)$ удовлетворяет определению 3.6 при $\varepsilon = |\mathbb{A}|^{-1}$.

Из теоремы 3.2 следует, что рассматриваемая нами функция $h(k,x)$ также является и универсальной функцией хэширования, в смысле определения 3.6.

Исследуем вопрос о том, является ли введенная нами функция (3.13) строго универсальной функцией хэширования. Верна следующая лемма.

Лемма 3.1. *Пусть $n \in \mathbb{N}$ и $1 \leqslant n \leqslant l$. Выберем два произвольных вектора $(x_{(n-1)u+1}, \ldots, x_{nu})$ и $(y_{(n-1)u+1}, \ldots, y_{nu}) \in \mathbb{V}_u(\mathbb{B})$ и два произвольных элемента $a, b \in \mathbb{A}$. Тогда система уравнений*

$$\begin{cases} \pi_n(z_{(n-1)u+1}, \ldots, z_{nu-1}, z_{nu}) &= a, \\ \pi_n(t_{(n-1)u+1}, \ldots, t_{nu-1}, t_{nu}) &= b, \end{cases} \tag{3.15}$$

где

$$\begin{aligned} z_i &= g(k_{s(i-1)+1}, \ldots, k_{si}, x_i), \\ t_i &= g(k_{s(i-1)+1}, \ldots, k_{si}, y_i),\ i = (n-1)u+1, \ldots, nu, \end{aligned}$$

имеет не более $|\mathbb{B}|^{u(s-1)}$ решений относительно неизвестных $k_{su(n-1)+1}$, $\ldots$, k_{sun}.

Доказательство. Для любого $1 \leqslant n \leqslant l$, в силу взаимной однозначности отображения π_n, уравнению

$$\pi_n(z_{(n-1)u+1}, \ldots, z_{nu-1}, z_{nu}) = a$$

удовлетворяет ровно один вектор $(z_{(n-1)u+1}, \ldots, z_{nu-1}, z_{nu})$. Из свойств отображения g следует, что для любого $i = (n-1)u+1, \ldots, nu$, уравнение $g(k_{s(i-1)+1}, \ldots, k_{si}, x_i) = z_i$ имеет ровно $|\mathbb{B}|^{(s-1)}$ решение, относительно неизвестных $k_{s(i-1)+1}, \ldots, k_{si}$. Каждое из этих решений может являться решением уравнения $g(k_{s(i-1)+1}, \ldots, k_{si}, y_i) = t_i$. Следовательно, общее число решений системы (3.11)

$$\begin{cases} g(k_{s(i-1)+1}, \ldots, k_{si}, x_i) = z_i, \\ g(k_{s(i-1)+1}, \ldots, k_{si}, y_i) = t_i, \end{cases}$$

не превосходит величины $|\mathbb{B}|^{(s-1)}$. Поскольку индекс $i = (n-1)u+1, \ldots, nu$ и принимает u значений, то общее количество векторов $k_{su(n-1)+1}, \ldots, k_{sun}$, для которых выполнена система равенств (3.15), не превосходит $|\mathbb{B}|^{u(s-1)}$. Лемма доказана. □

Отметим, что полученная нами в лемме 3.1 оценка на число решений может достигать указанной величины $|\mathbb{B}|^{u(s-1)}$. Рассмотрим $\mathbb{B} = \mathbb{Z}_8$, тогда для $s = 2$, $u = 2$ и функции g, определенной равенством (3.12), система сравнений (3.11)

$$\begin{cases} k_1 + k_2 \oplus 3 \equiv 2 \pmod 8, \\ k_1 + k_2 \oplus 7 \equiv 6 \pmod 8, \end{cases}$$

имеет $8 = 8^{(2-1)}$ пар решений

$$(0,1),(1,2),(2,3),(3,4),(4,5),(5,6),(6,7),(7,0).$$

Аналогично, система

$$\begin{cases} k_3 + k_4 \oplus 4 \equiv 1 \pmod 8, \\ k_3 + k_4 \oplus 0 \equiv 5 \pmod 8, \end{cases}$$

также имеет 8 решений $(0,5),(1,4),(2,3),(3,2),(4,1),(5,0),(6,7),(7,6)$. Тогда, для $x = (3,4)$ и $y = (7,0)$ и $a, b \in \mathbb{Z}_{64}$ таких, что

$$\begin{cases} f(2,1) = a, \\ f(6,5) = b, \end{cases}$$

найдется ровно $64 = 8^{2(2-1)}$ набора решений $k_1, \ldots, k_4$.

Лемма 3.2. *Пусть в условиях леммы 3.1 элементы $a, b \in \mathbb{A}$ пробегают множество всех возможных значений, тогда суммарное число решений системы (3.15), относительно неизвестных $k_{su(n-1)+1}, \ldots, k_{sun}$ равно $|\mathbb{B}|^{us}$.*

Доказательство. Пусть для любого значения индекса i в интервале от $(n-1)u+1$ до nu пара $z_i, t_i \in \mathbb{B}$ пробегает все возможные значения, тогда, согласно третьему свойству отображения g, суммарное количество решений системы (3.11)

$$\begin{cases} g(k_{s(i-1)+1}, \ldots, k_{si}, x_i) = z_i, \\ g(k_{s(i-1)+1}, \ldots, k_{si}, y_i) = t_i, \end{cases}$$

в точности равно $|\mathbb{B}|^s$. Поскольку число решений не зависит от i, а величины $z_{(n-1)u+1}, \ldots, z_{nu-1}, z_{nu}$ и $t_{(n-1)u+1}, \ldots, t_{nu-1}, t_{nu}$ принимают значения независимо, то суммарное количество решений системы (3.15) относительно неизвестных $k_{su(n-1)+1}, \ldots, k_{sun}$, равно $|\mathbb{B}|^{us}$. □

Заметим, что полученная нами в лемме 3.2 оценка числа решений не зависит от выбора значений $(x_1, \ldots, x_{lu}), (y_1, \ldots, y_{lu}) \in \mathbb{S}$. Теперь мы можем доказать следующее утверждение.

Теорема 3.3. *Пусть функция $h(x,k)$ определена равенством (3.13). Тогда, для любых элементов $a, b \in \mathbb{A}$ и любых $x = (x_1, \ldots, k_{lu})$, $y = (y_1, \ldots, y_{lu})$ из множества $\mathbb{S}$ найдется не более $|\mathbb{B}|^{u(sl-1)}$ элементов $k = (k_1, \ldots, k_{slu})$ таких, что*

$$\begin{cases} h(x,k) = a, \\ h(y,k) = b. \end{cases} \tag{3.16}$$

Доказательство. Рассмотрим систему уравнений

$$\begin{cases} \sum_{n=1}^{l} a_n = a, \\ \sum_{n=1}^{l} b_n = b, \end{cases}$$

относительно неизвестных $a_1, \ldots, a_l, b_1, \ldots, b_l \in \mathbb{A}$. Решениями данной системы являются векторы

$$\left(a_1, \ldots, a_{l-1}, a - \sum_{n=1}^{l-1} a_n \right), \quad \left(b_1, \ldots, b_{l-1}, b - \sum_{n=1}^{l-1} b_n \right),$$

в которых элементы $a_1, \ldots, a_{l-1}, b_1, \ldots, b_{l-1}$ принимают все возможные значения.

Для произвольного индекса $n = 1, \ldots, l-1$ рассмотрим систему

$$\begin{cases} \pi_n(z_{u(n-1)+1}, \ldots, z_{un}) &= a_n, \\ \pi_n(t_{u(n-1)+1}, \ldots, t_{un}) &= b_n, \end{cases}$$

где

$$\begin{aligned} z_i &= g(k_{s(i-1)+1}, \ldots, k_{si}, x_i), \\ t_i &= g(k_{s(i-1)+1}, \ldots, k_{si}, y_i),\ i = u(n-1)+1, \ldots, un. \end{aligned}$$

Будем считать, что элементы $a_n, b_n \in \mathbb{A}$ пробегают множество всех возможных значений. Тогда, в силу леммы 3.2 для каждого индекса $n = 1, \ldots, l-1$ найдется $|\mathbb{B}|^{us}$ наборов $k_{su(n-1)+1}, \ldots, k_{sun}$, являющихся решениями указанной системы.

В силу леммы 3.1 найдется не более $|\mathbb{B}|^{u(s-1)}$ наборов $k_{su(l-1)+1}, \ldots, k_{sul}$, удовлетворяющих системе

$$\begin{cases} f(z_{u(l-1)+1}, \ldots, z_{ul}) = a - \sum_{n=1}^{l-1} a_n, \\ f(t_{u(l-1)+1}, \ldots, t_{ul}) = b - \sum_{n=1}^{l-1} b_n. \end{cases}$$

Таким образом, число решений, удовлетворяющих системе (3.16), не превосходит величины

$$|\mathbb{B}|^{us(l-1)} \cdot |\mathbb{B}|^{u(s-1)} = |\mathbb{B}|^{usl-us+us-u} = |\mathbb{B}|^{u(sl-1)}.$$

Теорема доказана. □

Пусть $\mathbb{K}' \subset \mathbb{K}$ множество ключей, для которых разрешима система (3.16). Из утверждения теоремы следует, что

$$|\mathbb{K}'| \leqslant |\mathbb{B}|^{u(sl-1)} = \frac{|\mathbb{B}|^{usl}}{|\mathbb{B}|^{u}} = \frac{|\mathbb{K}|}{|\mathbb{A}|}$$

и определенная нами функция $h(k, x)$ удовлетворяет определению 3.7 со значением $\varepsilon = 1$.

Заключение к § 3.2

В § 3.2 определен новый класс ключевых функций хэширования, представляющих собой линейные формы от перестановок на множестве кодов аутентификации. Доказано, что функции из данного класса являются:

- универсальными (равновероятными) функциями хэширования относительно сжимаемых сообщений см. теорему 3.1,
- строго универсальными функциями относительно множества ключей, см. теоремы 3.2 и 3.3.

§ 3.3. Аутентифицированное шифрование

В настоящее время в Российской Федерации стандартизирован режим аутентифицированного шифрования «MGM», см. [169, 364]. Данный режим основан на принципе[2] EtM – для шифрования данных используется режим счетчика, после чего зашифрованные данные преобразуются с помощью линейной формы для выработки кода аутентификации сообщений. Согласно Р 1323565.1.026-2019 режим «MGM» может быть изображен следующим образом.

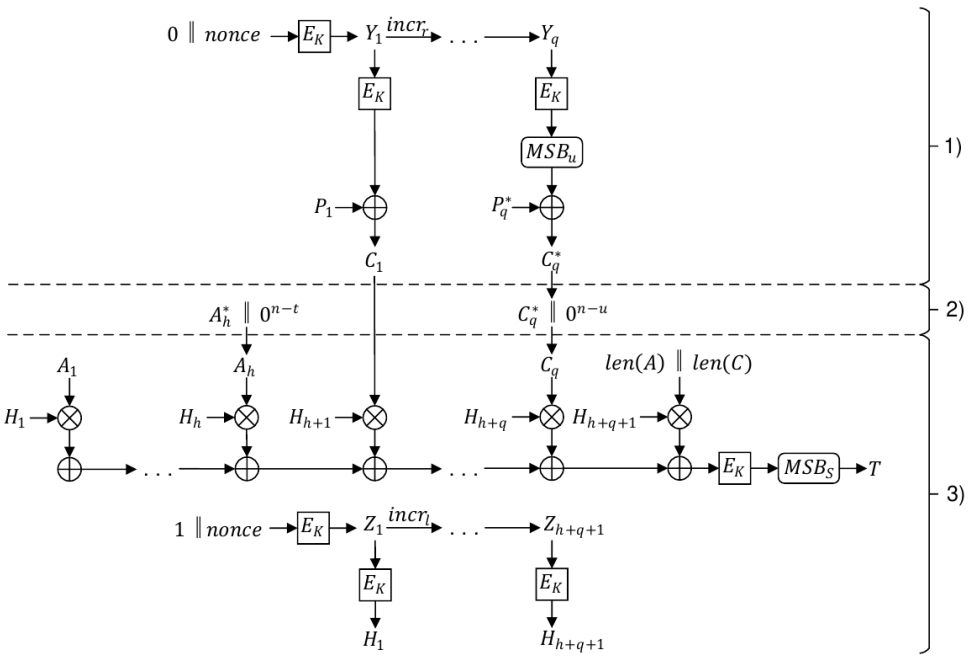


Рис. 3.2: Схема зашифрования в режиме «MGM»

[2]Принцип EtM (Encryption-then-MAC) подразумевает, что обрабатываемые данные сначала зашифровываются и только потом вычисляется код аутентификации сообщений. Это позволяет проверять вычисленный код без предварительного расшифрования данных, см. [116].

Используемая в режиме «MGM» линейная форма может быть определена равенством

$$h(x) = \sum_i H_i \otimes x_i,$$

где $\otimes$ функция умножения двух элементов конечного поля $\mathbb{F}_{2^w}$ и w длина блока для используемого алгоритма шифрования, x_i – фрагменты аутентифицируемых данных и зашифрованного текста, а H_i секретные элементы поля $\mathbb{F}_{2^w}$. Использование операции умножения приводит к появлению некоторых из рассматривавшихся нами ранее на стр. 205 особенностей, а также к нарушению свойства равновероятности.

Параллелизуемая структура режима «MGM» может быть хорошо реализована аппаратно на базе программируемых логических интегральных схем (FPGA) или графических процессоров (GPU). Однако программная реализация режима «MGM» на универсальных процессорах очень медленная, что делает этот режим неприемлемым для использования в «Интернете вещей» или в других приложениях, где используются низкоресурсные микроконтроллеры для криптографической защиты данных и аутентификации.

В этом разделе мы рассматриваем разрабатывавшийся автором на протяжении 2015-2021 годов режим аутентифицированного шифрования «XTSMAC». Данный режим обладает следующими отличительными особенностями:

- режим «XTSMAC» не только допускает эффективную аппаратную реализацию на интегральных микросхемах, но и значительно быстрее, чем режим «MGM» при программной реализации на универсальных процессорах, см. [144, 339];
- для шифрования и выработки кода аутентификации режим «XTSMAC» использует два различных ключа $k_1, k_2 \in \mathbb{V}_{256}$;
- длина вырабатываемого кода аутентификации в два раза превосходит длину блока используемого алгоритма шифрования; данная особенность может оказаться существенной для увеличения стойкости при использовании блочного шифра «Магма», см. также раздел 4.4.3.1;
- функция выработки кода аутентификации, являющаяся частью режима «XTSMAC» при фиксированных ключах k_1, k_2 обладает свойством равновероятности.

К недостаткам режима «XTSMAC» стоит отнести его достаточно высокую структурную сложность.

§ 3.3.1. Описание режима XTSMAC

При описании режима «XTSMAC» будем следовать работе [180] и рассмотрим следующие отображения.

1. Алгоритм блочного шифрования, см. определение 3.8,

$$E_k(x) : \mathbb{V}_{256} \times \mathbb{V}_w \to \mathbb{V}_w$$

где $w \in \{64, 128\}$ это длина блока алгоритма шифрования.

2. Отображение $\phi : \mathbb{V}_8 \to \mathbb{V}_8$, представляющее собой нелинейную перестановку на двоичных векторах длины 8, определяемую в стандарте ГОСТ Р 34.11-2012 [281] или RFC 6986 [70].

3. Отображение $S : \mathbb{V}_w \to \mathbb{V}_w$, определяемое равенством

$$S(x) = (\phi(x_0)|| \dots ||\phi(x_{(w/8)-1})).$$

4. Линейный оператор $L(x) : \mathbb{V}_w \to \mathbb{V}_w$, который представляет собой умножение двоичного вектора $x \in \mathbb{V}_w$ на фиксированную, обратимую матрицу $L \in GL_w(\mathbb{F}_2)$.

   Выбор матрицы L зависит от значения w — для $w = 64$ матрица L определена в [281, § 5.4], для $w = 128$ — в [282, § 4.2].

5. Отображение $\pi : \mathbb{V}_w \times \mathbb{V}_w \to \mathbb{V}_{2w}$, зависящее от четырех констант $\zeta_0, \dots, \zeta_3 \in \mathbb{V}_w$ и представляющее собой 4-х раундовую сеть Фейстеля, см. [347], с раундовой функцией $L(S(x \oplus \zeta))$. Аналитически, отображение π может быть записано следующим образом.

$$\pi(x_0||x_1) = (\underbrace{x_2 \oplus L(S(x_3 \oplus \zeta_2))}_{x_4} || \underbrace{x_3 \oplus L(S(x_4 \oplus \zeta_3))}_{x_5}) = \\ = (x_4||x_5), \quad (3.17)$$

   где

$$x_2 = x_0 \oplus L(S(x_1 \oplus \zeta_0), \quad x_3 = x_1 \oplus L(S(x_2 \oplus \zeta_1).$$

   Графическое изображение отображения π приведено на рисунке 3.3.

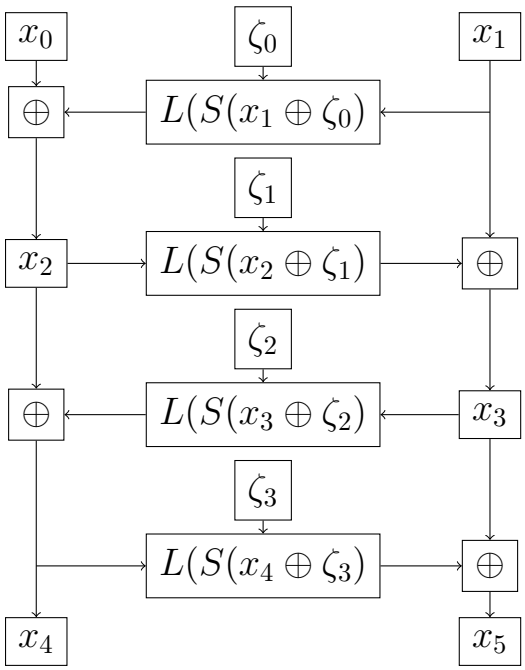


Рис. 3.3: Схема вычисления отображения π.

6. Пусть $p(x) \in \mathbb{F}_2[x]$ примитивный[3] многочлен степени $2w$, определяемый следующими равенствами, см. [242, 264].

$2w$	$p(x)$
128	$x^{128} + x^7 + x^2 + x + 1$
256	$x^{256} + x^{10} + x^5 + x^2 + 1$

Пусть $\mathbb{F}_{2^{2w}}$ конечное расширение поля $\mathbb{F}_2$, порожденное многочленом $p(x)$ и α корень многочлена $p(x)$ в $\mathbb{F}_{2^{2w}}$. Зафиксируем некоторый элемент $\gamma_{-1} \in \mathbb{F}_{2^{2w}}$ и рассмотрим отображение $G_{\gamma_{-1}} : \mathbb{N}_0 \to \mathbb{V}_{2w}$, формирующее для всех индексов $n \in \mathbb{N}_0$ последовательность элементов $\{\gamma_n\}_0^\infty$, определяемую равенством

$$\gamma_n = \gamma_{-1}\alpha^{n+1} \in \mathbb{F}_{2^{2w}},\ n = 0, 1, \ldots, \tag{3.18}$$

В дальнейшем мы будем использовать обозначение $\gamma_n = (\gamma_{n,0}||\gamma_{n,1})$.

Мы будем рассматривать открытые данные, подлежащие зашифрованию, а также ассоциированные данные, как конкатенацию фрагментов фиксированной длины

$$x = x_0||\ldots||x_{l-1}||x_l, \quad y = y_0||\ldots||y_{r-1}||y_r,$$

[3]Напомним, что неприводимый многочлен $p(x) \in \mathbb{F}_2[x]$ степени m называется примитивным, если минимальное натуральное значение e такое, что $x^e \equiv 1 \pmod{p(x)}$, равно $e = p^m - 1$, см. [316, § 6.1.3].

где

$$\mathrm{len}_2(x_0) = \cdots = \mathrm{len}_2(x_{l-1}) = \mathrm{len}_2(y_0) = \cdots = \mathrm{len}_2(y_{r-1}) = w,$$

а также $\mathrm{len}_2\, x_l \leqslant w$, $\mathrm{len}_2\, y_r \leqslant w$ и $l, r \in \mathbb{N}_0$. Кроме того, введем ограничение на общую длину открытых данных и будем считать[4], что $\mathrm{len}_2\, x \geqslant 2w$.

Разобьем входные данные на пары и определим число пар равенствами

$$l_0 = l \pmod 2 + \left\lfloor \frac{l}{2} \right\rfloor, \quad r_0 = r \pmod 2 + \left\lfloor \frac{r}{2} \right\rfloor. \tag{3.19}$$

Вместе с каждой парой блоков будет преобразовываться элемент последовательности γ_n, при этом, элементы $\gamma_0, \ldots, \gamma_{r_0-1}$ будут соответствовать парам ассоциированных данных, а элементы $\gamma_{r_0}, \ldots, \gamma_{r_0+l_0-1}$ – парам открытого текста.

Будем обозначать символом Ω_n основное нелинейное преобразование режима «XTSMAC», которое применяется к одной паре блоков входных данных и схематично представляется следующим рисунком.

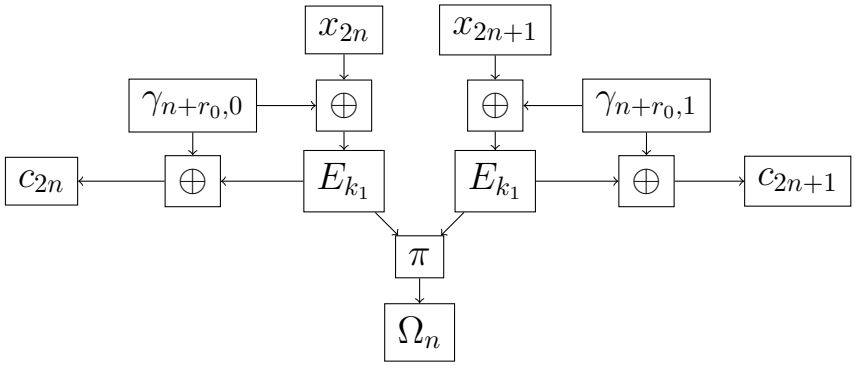


Рис. 3.4: Схема вычисления Ω_n.

На приведенном рисунке изображена одновременная выработка слагаемого, используемого для вычисления кода аутентификации

$$\Omega_n = \pi\big(E_{k_1}(x_{2n} \oplus \gamma_{n+r_0,0}) || E_{k_1}(x_{2n+1} \oplus \gamma_{n+r_0,1})\big),$$

и зашифрование пары блоков открытых данных в соответствии с равенствами

$$\begin{aligned} c_{2n} &= E_{k_1}(x_{2n} \oplus \gamma_{n+r_0,0}) \oplus \gamma_{n+r_0,0}, \\ c_{2n+1} &= E_{k_1}(x_{2n+1} \oplus \gamma_{n+r_0,1}) \oplus \gamma_{n+r_0,1}, \end{aligned} \tag{3.20}$$

для $n = 0, 1, \ldots, l_0 - 1$ и $\gamma_n \in \mathbb{F}_{2^{2w}}$ определяемого равенством (3.18). При этом, зашифрованный текст определяется равенством

$$c = (c_0 || \ \ldots \ || c_{l-1} || \, \mathrm{lsb}_{\mathrm{len}_2(x_l)}(c_l)).$$

[4]Для сообщений x, чья длина менее $2w$ бит, режим «XTSMAC» не может обеспечить равенство длин открытого и шифрованного текстов.

Данный способ зашифрования иногда называют «гамма-коммутатор-гамма» или, в англоязычной литературе, «xor-encryption-xor», см. обзор в [316, § 7.6.6].

Согласно (3.20), процедура расшифрования задается равенствами

$$\begin{array}{rcl} x_{2n} & = & D_{k_1}(c_{2n} \oplus \gamma_{n+r_0,0}) \oplus \gamma_{n+r_0,0}, \\ x_{2n+1} & = & D_{k_1}(c_{2n+1} \oplus \gamma_{n+r_0,1}) \oplus \gamma_{n+r_0,1}, \end{array}$$

для $n = 0, \dots, l_0 - 1$, из которых следует что для расшифрования исходных данных x_{2n}, x_{2n+1} нужны полные блоки c_{2n} и c_{2n+1}. Если же длина одного из последних блоков входных данных, скажем, x_{2l_0-1} меньше, чем w, то из равенства $\mathrm{len}_2(c_{2l_0-1}) = w$ следует, что длина всего шифрованного текста не будет совпадать с длиной всего открытого текста. Это приводит к необходимости реализации, так называемой, процедуры «скрадывания» шифртекста, см. [74] или [316, § 7.6.6].

Описываемое далее преобразование применяется только в случае, когда $\mathrm{len}_2(x) \not\equiv 0 \pmod{2w}$, т.е. когда длина входных данных не кратна длине двух блоков используемого алгоритма шифрования. Рассмотрим последние четыре блока открытых данных

$$x_{2l_0-4}||x_{2l_0-3}||x_{2l_0-2}||x_{2l_0-1},$$

при этом блок x_l совпадает с блоком x_{2l_0-1} при четном значении l, и совпадает с блоком $2l_0 - 2$ при нечетном значении l. Обозначим символом u число значащих бит во второй паре блоков

$$u = \mathrm{len}_2(x_{2l_0-2}||x_{2l_0-1}) = \begin{cases} w + \mathrm{len}_2(x_l), & \text{в первом случае,} \\ \mathrm{len}_2(x_l), & \text{во втором.} \end{cases}$$

Процедура «скрадывания» может быть аналитически описана следующими соотношениями

$$\begin{array}{rcl} t_0 & = & E_{k_1}(x_{2l_0-4} \oplus \gamma_{r_0+l_0-2,0}) \oplus \gamma_{r_0+l_0-2,0}, \\ t_1 & = & E_{k_1}(x_{2l_0-3} \oplus \gamma_{r_0+l_0-2,1}) \oplus \gamma_{r_0+l_0-2,1}, \\ t_2||t_3 & = & x_{2l_0-2}||x_{2l_0-1}||\,\mathrm{msb}_{2w-u}(t_0||t_1), \\ c_{2l_0-4} & = & E_{k_1}(t_2 \oplus \gamma_{r_0+l_0-1,0}) \oplus \gamma_{r_0+l_0-1,0}, \\ c_{2l_0-3} & = & E_{k_1}(t_3 \oplus \gamma_{r_0+l_0-1,1}) \oplus \gamma_{r_0+l_0-1,1}, \\ c_{2l_0-2}||c_{2l_0-1} & = & \mathrm{lsb}_u(t_0||t_1) \end{array} \tag{3.21}$$

и схематично изображена в виде следующего рисунка.

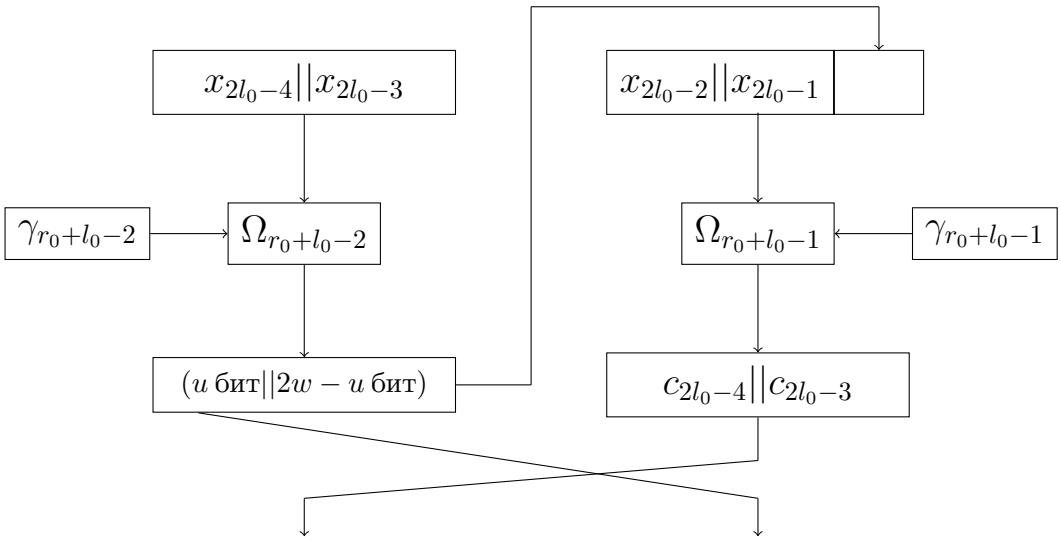


Рис. 3.5: Схема процедуры «скрадывания» зашифрованного текста.

Теперь рассмотрим натуральное число m, удовлетворяющее неравенствам $1 \leqslant m \leqslant 2w$, и определим код аутентификации длины m следующими равенствами.

$$s = (s_0||s_1) = \sum_{n=0}^{r_0-1} \pi\big(E_{k_1}(y_{2n} \oplus \gamma_{n,0})||E_{k_1}(y_{2n+1} \oplus \gamma_{n,1})\big) \oplus \sum_{n=r_0}^{r_0+l_0-1} \pi\big(E_{k_1}(x_{2(n-r_0)} \oplus \gamma_{n,0})||E_{k_1}(x_{2(n-r_0)+1} \oplus \gamma_{n,1})\big) \oplus \pi\big(E_{k_1}(\mathrm{len}_2(y) \oplus \gamma_{r_0+l_0,0})||E_{k_1}(\mathrm{len}_2(x) \oplus \gamma_{r_0+l_0,1})\big), \quad (3.22)$$

и

$$a = \mathrm{msb}_m\Big(E_{k_2}(s_0)||E_{k_2}(s_1 \oplus E_{k_2}(s_0))\Big), \quad (3.23)$$

Как видно из приведенных равенств, после вычисления значения суммы s, это значение зашифровывается на ключе K_2 в режиме простой замены с зацеплением с использованием нулевого инициализационного вектора. Окончательным кодом аутентификации служат старшие m бит вектора, полученного в результате шифрования. Графическое изображение равенств (3.22) и (3.23) приведено далее на рисунке 3.6

Полностью режим «XTSMAC» может быть описан в виде следующего алгоритма.

Алгоритм 3.1: Алгоритм $\texttt{authenc}(k_1, k_2, iv, x, y) = \{c, a\}$ аутентифицированного шифрования XTSMAC

Вход : Алгоритм блочного шифрования $E(\cdot)$ с длиной блочного шифра w, секретные ключи шифрования и аутентификации $k_1, k_2 \in \mathbb{V}_{256}$, открытые данные $x \in V_\infty$, ассоциированные данные $y \in \mathbb{V}_\infty$, инициализационный вектор $iv \in \mathbb{V}_\infty$ и m – длина кода аутентификации (в битах) и $1 \leqslant m \leqslant 2w$.

Выход : Зашифрованные данные $c \in \mathbb{V}_\infty$ такие, что $\mathrm{len}_2(c) = \mathrm{len}_2(x)$, и код аутентификации $a \in \mathbb{V}_m$.

/* *Этап инициализации* */
Используя (3.19), определить величины l_0 и r_0.
Определить в качестве $s \in \mathbb{V}_{2w}$ вектор, состоящий из одних нулей.
Если $\mathrm{len}_2(iv) < 6w$, то iv необходимо дополнить нулями в старших разрядах до длины $6w$ бит.
Используя режим простой замены с зацеплением и нулевой инициализационный вектор, см. [283, § 5.4], определить

$$(\gamma_{-1,0}||\gamma_{-1,1}||\zeta_0||\zeta_1||\zeta_2||\zeta_3) = CBC_{k_2}(\mathrm{lsb}_{6w}(iv), 0).$$

/* *Обрабатываем ассоциированные данные* */
Для всех $n = 0, \ldots, r_0 - 1$ **выполнять**
    Используя (3.18) определить $\gamma_n = (\gamma_{n,0}||\gamma_{n,1}) = \gamma_{-1}\alpha^{n+1}$.
    Определить $\Omega_n = \pi\big(E_{k_2}(y_{2n} \oplus \gamma_{n,0})||E_{k_2}(y_{2n+1} \oplus \gamma_{n,1})\big)$.
    Определить $s = s \oplus \Omega_n$.
конец
/* *Основной этап зашифрования открытых данных* */
Если $\mathrm{len}_2(x) \not\equiv 0 \ (mod\, 2w)$ **то**
    Положить $l_0 = l_0 - 2$.
конец
Для всех $n = r_0, \ldots, r_0 + l_0 - 1$ **выполнять**
    Определить $\gamma_n = (\gamma_{n,0}||\gamma_{n,1}) = \gamma_{-1}\alpha^{n+1}$.
    Определить $d_{2n} = E_{k_1}(x_{2(n-r_0)} \oplus \gamma_{n,0})$, $d_{2n+1} = E_{k_1}(x_{2(n-r_0)+1} \oplus \gamma_{n,1})$.
    Определить шифртекст $c_{2n} = d_{2n} \oplus \gamma_{n,0}$, $c_{2n+1} = d_{2n+1} \oplus \gamma_{n,1}$.
    Определить $\Omega_n = \pi\big(d_{2n}||d_{2n+1}\big)$.
    Определить $s = s \oplus \Omega_n$.
конец
Если $\mathrm{len}_2(x) \not\equiv 0 \ (mod\, 2w)$ **то**
    Положить $l_0 = l_0 + 2$.
    Используя (3.21), выполнить процедуру «скрадывания» шифртекста.
конец
/* *Этап вычисления кода аутентификации* */
Определить $\gamma_{r_0+l_0} = \gamma_{-1}\alpha^{r_0+l_0+1}$.
Определить $s = s \oplus \pi\big(E_{k_1}(\mathrm{len}_2(y) \oplus \gamma_{r_0+l_0,0})||E_{k_1}(\mathrm{len}_2(x) \oplus \gamma_{r_0+l_0,1})\big)$.
Представить $s = (s_0||s_1)$ и определить код аутентификации равенством

$$a = \mathrm{msb}_m\Big(E_{k_2}(s_0)||E_{k_2}(s_1 \oplus E_{k_2}(s_0))\Big).$$

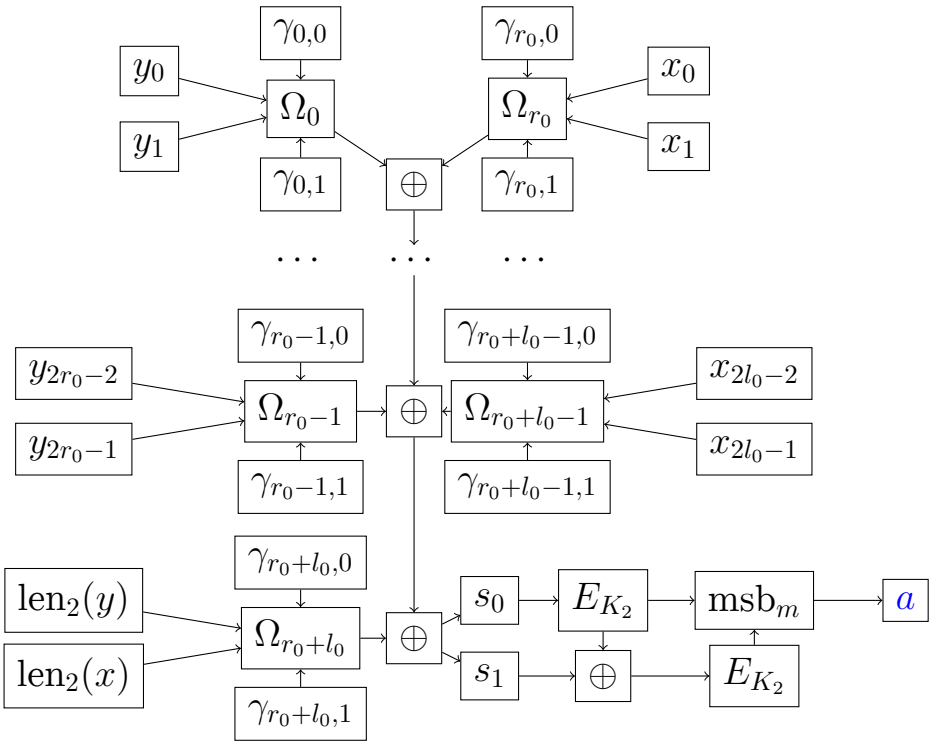


Рис. 3.6: Режим аутентификационного шифрования XTSMAC.

Сделаем несколько замечаний к приведенному выше алгоритму 3.1.

1. Все обрабатываемые данные разбиваются на пары и маркируются уникальными значениями γ_n. Выработка начального значения γ_{-1} происходит на этапе инициализации и зависит от значения iv и секретного ключа аутентификации k_2.

2. Также на этапе инициализации происходит выработка секретных значений $\zeta_0, \ldots, \zeta_3$, фактически являющихся ключами в 4-х раундовой сети Фейстеля, реализуемой преобразованием π.

3. Вычисление значений $\Omega_0, \ldots, \Omega_{r_0-1}$, соответствующих ассоциированным данным, производится с использованием ключа аутентификации k_2.

4. Если r нечетно и значение y_{2r_0-1} не определено, то оно доопределяется нулевым вектором длины w.

5. В случае, когда длина сообщения x не кратна $2w$ мы получаем, что $l_0 \geqslant 2$, следовательно, изменение величины l_0 в 14-й строке не выводит ее из области целых неотрицательных чисел. Изменение величины l_0 в 24-й строке необходимо для соответствия индексов в соотношениях (3.21).

§ 3.3.2. Исследование шифрующего преобразования

Перейдем к анализу безопасности разработанного режима аутентификационного шифрования и будем считать, что нарушитель хочет решить хотя бы одну из задач, сформулированных в определениях 3.2 и 3.9:

- определить секретные ключи k_1, k_2 или хотя бы один из них;
- определить открытый текст x, зная только зашифрованный текст c;
- для заданной пары значений y, x и кода аутентификации a, построить другую пару y_1, x_1, содержащую ассоциированные данные и открытый текст, которые дают то же значение кода аутентификации;
- для данного зашифрованного текста c построить другой зашифрованный текст c_1, который дает тот же код аутентификации.

Для достижения своих целей нарушитель может накапливать пары открытый/зашифрованный текст вместе со значениями соответствующих кодов аутентификации, модифицировать зашифрованный текст, применять повторяющиеся значения векторов инициализации и использовать легитимного пользователя в качестве оракула для зашифрования специально выбранных открытых текстов. Указанные возможности соответствуют принятым в зарубежной литературе атакам CPA, CCA и CCA2, подробнее см. в [151, гл. 14], а также далее на странице 246.

Изложенный в предыдущем разделе способ шифрования сообщений основывается на хорошо известном режиме работы блочных шифров «XTS», см. [73]. В свою очередь режим «XTS» является развитием предложенного Ф. Рогавеем подхода «xor-encryption-xor» для выработки, так называемых, «настраиваемых»[5] блочных шифров, см. [216]. Доказательство стойкости данного режима в теоретико-игровой модели может быть найдено в работах [147, 216].

Основное отличие предложенного режима от перечисленных зарубежных аналогов заключается в степени расширения поля, используемого для выработки псевдослучайной последовательности секретных значений $\gamma_{-1}, \gamma_0, \gamma_1, \ldots$. Режим «XTSMAC» использует поле $\mathbb{F}_{2^{2w}}$, в то время как режим «XTS» – поле $\mathbb{F}_{2^w}$. Причина такого увеличения заключается в следующем.

Напомним, что $\gamma_{-1} = CBC_{k_2}(\mathrm{lsb}_{2w}(iv))$, т.е. определение ключа k_2 нарушителем приводит к определению значения γ_{-1}, а согласно равенствам (3.18)

$$\alpha_n = \gamma_{-1}\alpha^{n+1}, \quad \text{и} \quad \gamma_{-1} = \alpha^{-(n+1)}\gamma_n,$$

[5] В оригинальной английской статье используется термин «tweakable block cipher».

и к определению всех элементов секретной последовательности $\gamma_{-1}, \gamma_0, \gamma_1, \ldots$. После чего, преобразование зашифрования входных данных для режима «XTSMAC» становится эквивалентно режиму простой замены, см. [72, 283]. Для определения открытого текста могут быть использованы накопленные наршителем пары открытый/зашифрованный текст и классические бесключевые методы взлома режима простой замены, см. [344, 347] или [395, гл.5].

С другой стороны, нарушитель может добиться того же результата, просто перебирая все возможные значения величины $\gamma_{-1} \in \mathbb{F}_{2^{2w}}$. Это позволяет говорить о том, что минимальная алгоритмическая сложность определения открытого текста составляет $O\big(2^{2w}\big)$ опробований величины γ_{-1}.

При использовании шифров с длиной блока $w = 128$, например шифра «Кузнечик», см. [282, § 4], такой подход эквивалентен опробованию ключа аутентификации k_2. При использовании шифров с длиной блока $w = 64$, например шифра «Магма», см. [282, § 5], алгоритмическая сложность определения открытого текста составляет $O(2^{128})$ операций опробования величины γ_{-1}, однако в ситуациях, когда определение ключа k_2 может быть реализовано другими методами с такой же сложностью, это не приводит к снижению показателей эффективности мер защиты.

§ 3.3.3. Исследование свойства равновероятности

Начнем изучение свойства равновероятности сжимающего отображения, реализуемого режимом «XTSMAC», с доказательства следующего утверждения.

Лемма 3.3. *Пусть блочный шифр $E_k(x) : \mathbb{V}_w \to \mathbb{V}_w$ является перестановкой множества $\mathbb{V}_w$ для любого фиксированного значения ключа $k \in \mathbb{K}$. Определим функцию двух аргументов $g : \mathbb{V}_w \times \mathbb{V}_w \to \mathbb{V}_w$ равенством*

$$g(\gamma_0, x) = E_k(\gamma \oplus x).$$

Тогда функция $g(\gamma, x)$ удовлетворяет условиям теоремы 3.1.

Доказательство. Для доказательства леммы необходимо проверить, что выполняются три условия, сформулированные после равенства (3.10).

При фиксированном значении $\gamma \in \mathbb{V}_w$ функция $g(\gamma, x)$ является композицией двух взаимно-однозначных отображений и, следовательно, сама является взаимно-однозначным отображением аргумента x.

Далее, фиксируем $x \in \mathbb{V}_w$ и рассмотрим уравнение $E_k(x \oplus \gamma) = z \in \mathbb{V}_w$ относительно неизвестного значения γ. В силу взаимной однозначности

найдется только одно значение $\gamma = D_k(z) \oplus x$, удовлетворяющее рассматриваемому уравнению.

Зафиксируем произвольные элементы $x, y \in \mathbb{V}_w$ и будем считать, что вычеты $z, t \in \mathbb{V}_m$ пробегают множество всех возможных значений. Тогда суммарное число решений системы уравнений

$$\begin{cases} E_k(x \oplus \gamma) = z, \\ E_k(y \oplus \gamma) = t, \end{cases}$$

относительно неизвестной γ, в точности равно 2^w. Действительно, пусть γ удовлетворяет указанной системе, тогда из равенств

$$D_k(z) \oplus x = \gamma = D_k(t) \oplus y$$

следует условие

$$z = E_k(D_k(t) \oplus x \oplus y). \tag{3.24}$$

Поскольку для каждого значения t найдется в точности одно значение z, определяемое соотношением (3.24), то количество пар z, t для которых существует решение указанной системы равно 2^t и для каждой такой пары решение единственно. □

Пусть $x \in \mathbb{V}_\infty$ произвольное сообщение, фиксируем его длину и определим величину

$$u \equiv \mathrm{len}_2(x) \pmod{2w}. \tag{3.25}$$

Если $u > 0$, то выберем произвольный элемент $\xi \in \mathbb{V}_{2w-u}$ и дополним им сообщение x до длины, кратной $2w$. Тогда, найдется натуральное число

$$v = \begin{cases} \frac{\mathrm{len}_2(x)}{2w}, & \text{если} \quad u = 0, \\ \frac{\mathrm{len}_2(x)-u}{2w} + 1, & \text{иначе}, \end{cases} \tag{3.26}$$

такое, что

$$x||\xi = x_0||x_1|| \cdots ||x_{2v-2}||x_{2v-1},$$

и $\mathrm{len}_2(x_i) = w$ для всех $i = 0, \ldots, 2v-1$. Кроме того, выполнено равенство

$$\mathrm{len}_2(x) = 2wv + (u - 2w) \cdot \left\lceil \frac{u}{2w} \right\rceil. \tag{3.27}$$

Отметим, что если в качестве x рассматривать открытые данные, к которым применяется сжимающее преобразование режима «XTSMAC», то величина u совпадает с величиной, определяемой при реализации процедуры «скрадывания» шифрованного текста, а величина v совпадает с определенной ранее в (3.19) величиной l_0. Если же вместо сообщения x рассмотреть ассоциированные данные, то значение величины v совпадет со значением r_0, также определенным в (3.19).

Рассмотрим сжимающее отображение

$$h : \mathbb{V}_{\mathrm{len}_2(x)} \times \mathbb{V}_{2w-u} \times (\mathbb{V}_w)^{2v} \to \mathbb{V}_{2w},$$
$$h(x, \xi, \gamma_{0,0}, \gamma_{0,1} \ldots, \gamma_{v-1,0}, \gamma_{v-1,1}) = \sum_{n=0}^{v-1} \pi\big(g(\gamma_{n,0}, x_{2n})||g(\gamma_{n,1}, x_{2n+1})\big), \tag{3.28}$$

где π – отображение, определяемое режимом «XTSMAC», и докажем следующее утверждение.

Лемма 3.4. *Зафиксируем некоторое значение $\xi \in \mathbb{V}_{2w-u}$, последовательность $\gamma_{0,0}, \gamma_{0,1} \ldots \in \mathbb{F}_w$ и отображение $h(x) = h(x, \xi, \gamma_{0,0}, \gamma_{0,1} \ldots)$, определяемое равенством (3.28). Определим v равенством (3.26), тогда*

1. *если $v = 1$, то для любого $a \in \mathbb{V}_{2w}$ найдется не более одного сообщения $x \in \mathbb{V}_{\mathrm{len}_2(x)}$ такого, что $h(x) = a$,*
2. *если $v \geqslant 2$, то найдется в точности $2^{\mathrm{len}_2(x)-2w}$ сообщений $x \in \mathbb{V}_{\mathrm{len}_2(x)}$ таких, что $h(x) = a$.*

Доказательство. Начнем доказательство со случая, когда $\mathrm{len}_2(x) \leqslant 2w$ и $v = 1$. Напомним, что π есть 4-х раундовая сеть Фейстеля, см. рисунок 3.3 и является взаимно-однозначным отображением множества $\mathbb{V}_w$ в себя (при фиксированных значениях $\zeta_0, \ldots, \zeta_3$). Тогда, учитывая утверждение леммы 3.3, получим, что для каждого значения a найдется в точности одно сообщение $x||\xi = x_0||x_1$ такое, что

$$\pi\big(g(\gamma_{0,0}, x_0)||g(\gamma_{0,1}, x_1)\big) = a. \tag{3.29}$$

Поскольку $v = 1$, то $u = \mathrm{len}_2(x) \leqslant 2w$ и существует в точности 2^u значений a, для которых уравнение (3.29) разрешимо относительно неизвестного значения $x \in \mathbb{V}_u$. Для остальных $2^{2w} - 2^u$ значений a уравнение (3.29) не разрешимо.

Теперь рассмотрим общий случай $v \geqslant 2$ и представим отображение $h(x)$ в виде

$$h(x) = \underbrace{\sum_{n=0}^{v-2} \pi\big(g(\gamma_{n,0}, x_{2n})||g(\gamma_{n,1}, x_{2n+1})\big)}_{h_1(x)} \oplus \underbrace{\pi\big(g(\gamma_{v-1,0}, x_{2v-2})||g(\gamma_{v-1,1}, x_{2v-1})\big)}_{h_2(x)},$$

где значения функций $h_1(x), h_2(x) \in \mathbb{V}_{2w}$.

Из утверждения леммы 3.3 следует, что функция $h_1(x)$ удовлетворяет условиям теоремы 3.1, т.е для любого $a' \in \mathbb{V}_{2w}$ найдется в точности $2^{2w(v-2)}$ сообщений $x' = x_0|| \ldots ||x_{2(v-1)-1}$ таких, что $h_1(x') = a'$.

Тогда, для любой пары x_{2v-2}, x_{2v-1} такой, что $\mathrm{msb}_u(x)||\xi = x_{2v-2}||x_{2v-1}$ найдется значение a' такое, что

$$\pi\big(g(\gamma_{v-1,0}, x_{2v-2})||g(\gamma_{v-1,1}, x_{2v-1})\big) = a \oplus a'$$

и мы можем сделать вывод о том, что найдется в точности $2^{2w(v-2)+u}$ значений x таких, что $h(x) = a$. Равенство

$$2w(v-2) + u = 2wv + u - 2w - 2w = \mathrm{len}_2(x) - 2w$$

завершает доказательство леммы. □

Следующее утверждение, см. [337], является простым следствием доказанных лемм.

Теорема 3.4. *Пусть блочный шифр $E_k(x) : \mathbb{V}_w \to \mathbb{V}_w$ является перестановкой множества $\mathbb{V}_w$ для любого фиксированного значения ключа $k \in \mathbb{K}$. Тогда, для любых ключей шифрования и имитозащиты $k_1, k_2 \in \mathbb{K}$, а также инициализационного вектора $iv \in \mathbb{V}_{6w}$ сжимающее отображение $\texttt{authenc}(k_1, k_2, iv, x, y)$, определяемое алгоритмом 3.1, обладает свойством равновероятности, т.е. для любого $a \in \mathbb{V}_{2w}$ найдется в точности*

$$2^{\mathrm{len}_2(x)+\mathrm{len}_2(y)-2w}$$

пар x, y таких, что $\texttt{authenc}(k_1, k_2, iv, x, y) = a$.

Доказательство. В начале, определим величины l_0 и r_0 равенствами (3.19) и зафиксируем последовательность секретных значений $\gamma_{0,0}, \gamma_{0,1}, \ldots, \gamma_{2(r_0+l_0),0}, \gamma_{2(r_0+l_0),1} \in \mathbb{F}_{2^w}$, однозначно определяемую секретным ключом k_2 и значением инициализационного вектора iv.

Определим в качестве ξ_y нулевой вектор длины $2w - \mathrm{len}_2(y) \pmod{2w}$, а также

$$\xi_x = \mathrm{msb}_{2w-u}(t_0||t_1), \quad u \equiv \mathrm{len}_2(x) \pmod{2w},$$

где величины $t_0, t_1 \in \mathbb{V}_w$ определены равенствами (3.21). Тогда, используя обозначение (3.28), можно записать сжимающее отображение режима «XTSMAC» в виде

$$\texttt{authenc}(k_1, k_2, iv, y, x) = CBC_{k_2}(s, 0),$$

где

$$s = h(y, \xi_y, \gamma_{0,0}, \ldots, \gamma_{r_0-1,1}) \oplus h(x, \xi_x, \gamma_{r_0,0}, \ldots, \gamma_{r_0+l_0-1,1}) \oplus \\ \pi\big(g(\gamma_{2(r_0+l_0),0}, \mathrm{len}_2(y)||g(\gamma_{2(r_0+l_0),1}, \mathrm{len}_2(x))\big)$$

Зафиксируем произвольное значение кода аутентификации $a \in \mathbb{V}_{2w}$. Поскольку при фиксированном ключе $k_2 \in \mathbb{K}$ отображение $CBC_{k_2}(s, 0)$ является взаимно-однозначным отображением множества V_{2w} в себя, то каждому значению $a \in V_{2w}$ будет соответствовать в точности одно значение суммы $s \in \mathbb{V}_{2w}$ такое, что $CBC_{k_2}(s, 0) = a$.

Для каждого $y \in \mathbb{V}_{\mathrm{len}_2(y)}$ найдется значение $s_y \in \mathbb{V}_{2w}$ такое, что

$$h(y, \xi_y, \gamma_{0,0}, \ldots, \gamma_{r_0-1,1}) = s_y$$

и уравнение

$$h(x, \xi_x, \gamma_{r_0,0}, \ldots, \gamma_{r_0+l_0-1,1}) = s_x, \tag{3.30}$$

где

$$s_x = s \oplus s_y \oplus \pi\big(g(\gamma_{2(r_0+l_0),0}, \mathrm{len}_2(y)||g(\gamma_{2(r_0+l_0),1}, \mathrm{len}_2(x))\big).$$

Поскольку $\mathrm{len}_2(x) \geqslant 2w$ то, из утверждения леммы 3.4 следует, что найдется $2^{\mathrm{len}_2(x)-2w}$ открытых данных x, для которых выполнено равенство (3.30). Следовательно, найдется в точности $2^{\mathrm{len}_2(y)+\mathrm{len}_2(x)-2w}$ пар x, y таких, что $AE(k_1, k_2, iv, x, y) = a$. □

Из утверждения теоремы сразу следует, что при случайном равновероятном выборе пар (x, y) вероятность выбрать такую пару, что $AE(k_1, k_2, iv, x, y) = a$, равна

$$\frac{2^{\mathrm{len}_2(x)+\mathrm{len}_2(y)-2w}}{2^{\mathrm{len}_2(x)+\mathrm{len}_2(y)}} = 2^{-2w}.$$

§ 3.3.4. Исследование подходов к построению коллизий

В этом разделе мы рассмотрим некоторые подходы к построению коллизий для сжимающего отображения режима «XTSMAC» и приведем объяснение применению ряда элементарных преобразований.

§ 3.3.4.1. Парадокс дней рождений

Как говорилось ранее, из результатов работ [91, 188, 202] следует, что существует вероятностный алгоритм, основанный на «парадоксе дней рождений», который находит решение задачи о построении коллизии сжимающего отображения режима «XTSMAC». Данный алгоритм заключается в фиксации некоторого натурального значения $\mathrm{len}_2(x) \geqslant 2w$, случайном выборе сообщений $x_1, x_2, \ldots \in_R \mathbb{V}_{\mathrm{len}_2(x)}$ и проверке совпадения значений

$$a_i = \texttt{authenc}(k_1, k_2, iv, x_i, y) \in \mathbb{V}_{2w}, \quad i = 1, 2, \ldots,$$

где величины k_1, k_2, iv, y фиксированы и принимают значения из своих областей определения.

Из утверждения теоремы 3.4 и равенства

$$e^x = 1 + \sum_{k=1}^{\infty} \frac{x^k}{k},$$

выполненного для действительного значения $|x| < 1$, легко получить, что вероятность p_n совпадения двух элементов указанной последовательности после n шагов алгоритма составляет

$$p_n = 1 - \prod_{k=1}^{n-1} \left(1 - \frac{k}{2^{2w}}\right) > 1 - \prod_{k=1}^{n-1} e^{-\frac{k}{2^{2w}}} = \\ = 1 - e^{-\frac{1}{2^{2w}}\left(1+\cdots+(n-1)\right)} = 1 - e^{-\frac{n(n-1)}{2^{2w+1}}}.$$

Тогда для индекса n такого, что с вероятностью не менее $\frac{1}{2}$ найдется индекс $1 \leqslant m < n$ такой, что $a_m = a_n$, выполнено неравенство

$$n^2 - n = n(n-1) > -2^{2w+1} \ln \frac{1}{2} = 2^{2w+1} \ln 2.$$

Таким образом, мы можем считать, что алгоритм построения коллизии, основанный на «парадоксе дней рождений», сможет завершить свою работу после

$$\left\lceil \frac{1 + \sqrt{1 + 2^{2w+3} \ln 2}}{2} \right\rceil = \left\lceil \frac{1}{2} + 2^w \sqrt{2 \ln 2 + \frac{1}{2^{2(w+1)}}} \right\rceil \leqslant \\ 2^w \left\lceil \frac{1}{2^{w+1}} + \sqrt{2 \ln 2 + \frac{1}{2^{2(w+1)}}} \right\rceil < 2 \cdot 2^w$$

шагов с вероятностью успеха не менее $\frac{1}{2}$.

§ 3.3.4.2. Атаки на основе перестановок блоков данных

Поскольку в основе сжимающего отображения лежит линейная форма, то нарушитель может воспользоваться свойством коммутативности операции сложения двоичных векторов. Действительно, пусть

$$\rho, \lambda : \mathbb{Z}_{l_0} \to \mathbb{Z}_{l_0},$$

две произвольные перестановки $\rho, \lambda \in S_{l_0}$, тогда из (3.22) следуют следующие равенства

$$s \oplus \pi\big(E_{k_1}(\mathrm{len}_2(y) \oplus \gamma_{r_0+l_0,0})||E_{k_1}(\mathrm{len}_2(x) \oplus \gamma_{r_0+l_0,1})\big) \oplus$$
$$\oplus \sum_{n=0}^{r_0-1} \pi\big(E_{k_1}(y_{2n} \oplus \gamma_{n,0})||E_{k_1}(y_{2n+1} \oplus \gamma_{n,1})\big) =$$
$$= \sum_{n=0}^{l_0-1} \pi\big(E_{k_1}(x_{2n} \oplus \gamma_{n+r_0,0})||E_{k_1}(x_{2n+1} \oplus \gamma_{n+r_0,1})\big) =$$
$$= \sum_{n=0}^{l_0-1} \pi\big(E_{k_1}(x_{2\rho(n)} \oplus \gamma_{\rho(n)+r_0,0})||E_{k_1}(x_{2\rho(n)+1} \oplus \gamma_{\rho(n)+r_0,1})\big) =$$
$$= \sum_{n=0}^{l_0-1} \pi\big(c_{2n} \oplus \gamma_{n+r_0,0}||c_{2n+1} \oplus \gamma_{n+r_0,1})\big) =$$
$$= \sum_{n=0}^{l_0-1} \pi\big(c_{2\lambda(n)} \oplus \gamma_{\lambda(n)+r_0,0}||c_{2\lambda(n)+1} \oplus \gamma_{\lambda(n)+r_0,1})\big), \quad (3.31)$$

где $c_0, c_1, \ldots, c_{2l_0-1}$ – блоки зашифрованного текста, определяемые равенствами (3.20).

В силу равенства (3.18) для любого индекса $n = 0, 1, \ldots$ выполнены равенства

$$(\gamma_{n,0}||\gamma_{n,1}) = \gamma_n = \gamma_{-1}\alpha^{n+1}$$

и, для любого индекса $m = 0, 1, \ldots$,

$$\gamma_m = \gamma_{-1}\alpha^{m+1} = \left(\gamma_n\alpha^{-n-1}\right)\alpha^{m+1} = \gamma_n\alpha^{m-n}.$$

Знание нарушителем одного из элементов последовательности $\gamma_{-1}, \gamma_0, \gamma_1, \ldots$ приводит к определению всех элементов последовательности и, как следствие, к построению нарушителем коллизии для сжимающего отображения режима «XTSMAC» с использованием одного из равенств (3.31).

Частный случай описанной выше атаки заключается в изучении разностей между двумя соседними значениями элементов последовательности $\gamma_{-1}, \gamma_0, \gamma_{-1}, \ldots$

Пусть x произвольное сообщение, длина которго не менее четырех полных блоков, т.е. $\mathrm{len}_2(w) \geqslant 4w$. Определим последовательность разностей

$$(\delta_{n,0}||\delta_{n,1}) = \delta_n = \gamma_n \oplus \gamma_{n+1} = (\gamma_{n,0} \oplus \gamma_{n+1,0}||\gamma_{n,1} \oplus \gamma_{n+1,1}),$$

где $n = 0, 1, \ldots$ и, используя (3.22), запишем следующие равенства

$$s \oplus \pi\big(E_{k_1}(\mathrm{len}_2(y) \oplus \gamma_{r_0+l_0,0})||E_{k_1}(\mathrm{len}_2(x) \oplus \gamma_{r_0+l_0,1})\big)\oplus$$
$$\oplus \sum_{n=0}^{r_0-1} \pi\big(E_{k_1}(y_{2n} \oplus \gamma_{n,0})||E_{k_1}(y_{2n+1} \oplus \gamma_{n,1})\big)\oplus$$
$$\oplus \sum_{n=2}^{l_0-1} \pi\big(E_{k_1}(x_{2n} \oplus \gamma_{n+r_0,0})||E_{k_1}(x_{2n+1} \oplus \gamma_{n+r_0,1})\big) =$$
$$\pi\big(E_{k_1}(x_0 \oplus \gamma_{r_0,0})||E_{k_1}(x_1 \oplus \gamma_{r_0,1})\big)\oplus$$
$$\pi\big(E_{k_1}(x_2 \oplus \gamma_{r_0+1,0})||E_{k_1}(x_3 \oplus \gamma_{r_0+1,1})\big) =$$
$$\pi\big(c_0 \oplus \gamma_{r_0,0}||c_1 \oplus \gamma_{r_0,1}\big) \oplus \pi\big(c_2 \oplus \gamma_{r_0+1,0}||c_3 \oplus \gamma_{r_0+1,1}\big) =$$
$$\pi\big(c_0 \oplus \delta_{r_0,0} \oplus \gamma_{r_0+1,0}||c_1 \oplus \delta_{r_0,1} \oplus \gamma_{r_0+1,1}\big)\oplus$$
$$\oplus \pi\big(c_2 \oplus \delta_{r_0,0} \oplus \gamma_{r_0,0}||c_3 \oplus \delta_{r_0,1} \oplus \gamma_{r_0,1}\big). \quad (3.32)$$

Из приведенных равенств следует, что для зашифрованных текстов

$$c_0, c_1, c_2, c_3, c_4, \ldots$$

и

$$c_2 \oplus \delta_{r_0,0},\ c_3 \oplus \delta_{r_0,1},\ c_0 \oplus \delta_{r_0,0},\ c_1 \oplus \delta_{r_0,1},\ c_4, \ldots$$

значения кодов аутентификации совпадают. Аналогично, из равенств (3.32), можно получить, что открытые тексты

$$x_0, x_1, x_2, x_3, x_4, \ldots$$

и

$$x_2 \oplus \delta_{r_0,0},\ x_3 \oplus \delta_{r_0,1},\ x_0 \oplus \delta_{r_0,0},\ x_1 \oplus \delta_{r_0,1},\ x_4, \ldots$$

также образуют коллизию и дают один и тот же код аутентификации.

Легко видеть, что

$$\delta_n = \gamma_n \oplus \gamma_{n+1} = \gamma_{-1}\alpha^{n+1} \oplus \gamma_{-1}\alpha^{n+2} = \gamma_{-1}\alpha^n(1 \oplus \alpha)$$

и построение коллизии эквивалентно определению величины γ_{-1} или, как было сказано выше, любого из элементов последовательности $\gamma_0, \gamma_1, \ldots$. Сложность такого подхода к построению коллизии составляет 2^{2w} операций опробования величины γ_{-1} и существенно превышает алгоритмическую сложность метода, основанного на «парадоксе дней рождений».

§ 3.3.4.3. Использование длины обрабатываемых данных

Использование последнего слагаемого

$$\pi\big(E_{k_1}(\mathrm{len}_2(y) \oplus \gamma_{r_0+l_0,0})||E_{k_1}(\mathrm{len}_2(x) \oplus \gamma_{r_0+l_0,1})\big)$$

в равенстве (3.22) позволяет указать явную зависимость кода аутентификации от длин ассоциированных и открытых данных. Это обеспечивает защиту от атак, основанных на неверной интерпретации входных данных. Действительно, если убрать данное слагаемое из суммы (3.22), то пары

$$(y||\xi, x) \quad \text{и} \quad (y, \xi||x),$$

где $\mathrm{len}_2(y) \equiv \mathrm{len}_2(x) \equiv \mathrm{len}_2(\xi) \equiv 0 \pmod{2w}$, дают одно и то же значение кода аутентификации.

Другой потенциальной атакой, защитой от которой служит явная зависимость кода аутентификации от длин входных данных, является дополнение открытых данных блоками, сумма которых представляет собой нулевой двоичный вектор, т.е. представление сообщения в виде $x||x'$, где $x' = x_{2l_0}||\cdots x_{2l_0+2t_0-1}$ для некоторого натурального $t_0 > 0$ и

$$\sum_{n=0}^{2t_0-1} \pi\big(E_{k_1}(x_{2(r_0+l_0+n)} \oplus \gamma_{r_0+l_0+n,0})||E_{k_1}(x_{2(r_0+l_0+n)+1} \oplus \gamma_{r_0+l_0+n,1})\big) = 0.$$

§ 3.3.4.4. Зашифрование значений линейной формы

В 2012 году в работе [272] автором был предложен метод построения коллизий для аддитивных отображений без завершающего зашифрования. Применительно к режиму шифрования «XTSMAC» метод заключается в следующем.

Пусть $x = x_1||x_2$, где $\mathrm{len}_2(x) = 2w$ – открытые данные и c_0, c_1 – соответствующие им зашифрованные данные, пусть ассоциированные данные не определены, т.е. $\mathrm{len}_2(y) = 0$. Предположим, что сжимающее преобразование режима «XTSMAC» не содержит завершающего шифрующего преобразования, тогда, согласно (3.22), нарушителю известно значение $s \in \mathbb{V}_{2w}$, удовлетворяющее равенству

$$s = \pi\big(E_{k_1}(x_0 \oplus \gamma_{0,0})||E_{k_1}(x_1 \oplus \gamma_{0,1})\big) \oplus \kappa = \pi\big(c_0 \oplus \gamma_{0,0}||c_1 \oplus \gamma_{0,1}\big) \oplus \kappa,$$

где $\kappa = (\kappa_0||\kappa_1) = \pi\big(E_{k_1}(\gamma_{1,0})||E_{k_1}(2w \oplus \gamma_{1,1})\big)$, а также зашифрованный текст $c_0||c_1$.

Отображение $c = (c_0||c_1) \to (s_0||s_1) = s$ имеет вид, изображенный на рисунке 3.7 (см. стр. 237) и представляет собой 4-х раундовый блочный шифр с шестью фиксированными ключами $\gamma_0, \zeta_0, \ldots, \zeta_3, \kappa$.

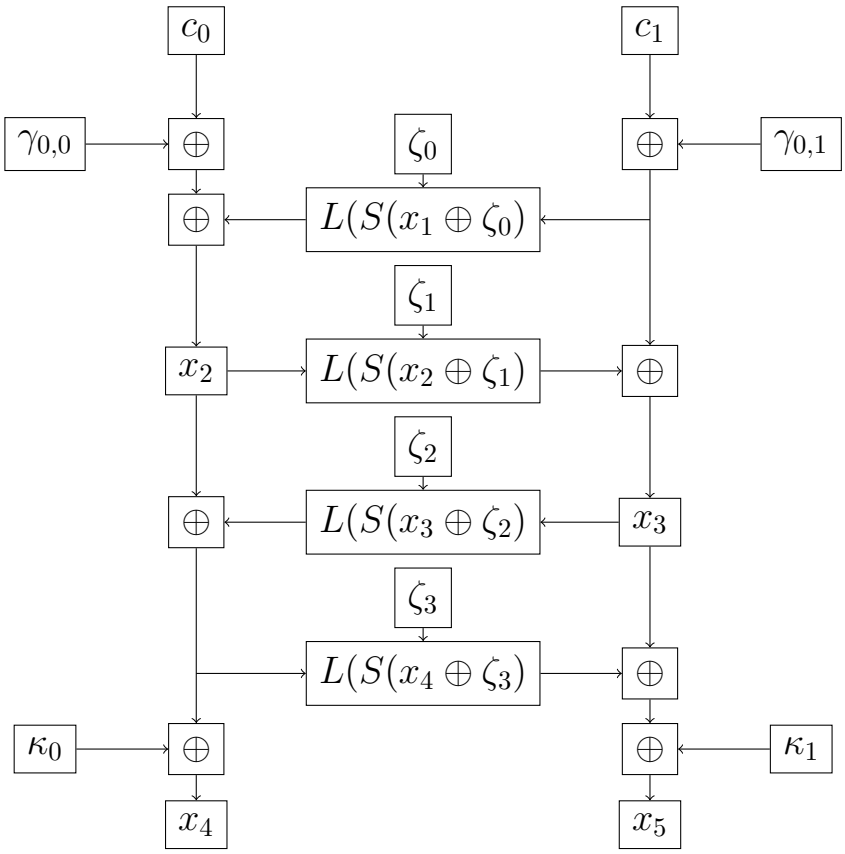


Рис. 3.7: Модифицированное отображение π.

Следовательно, нарушитель накопить необходимое количество пар c, s (при фиксированном значении вектора инициализации iv), применить линейный или разностный метод анализа блочных шифров, см. работы [34, 106, 155, 156], и определить неизвестные значения $\gamma_{0,0}$ и $\gamma_{0,1}$. После чего, он может построить коллизию описанным выше способом.

§ 3.3.4.5. Построение разностных соотношений

Идеи, лежащие в основе разностного метода анализа блочных шифров, могут быть использованы для реализации атаки на сжимающее отображение режима «XTSMAC».

В одном из ранних вариантов режима в качестве нелинейного преобразования π применялся многочлен второй степени, использовавшийся К. Гауссом для параметризации целых точек на плоскости, см. [298, 336]. Для данного многочлена в 2016 году В.И. Рудским было найдено разностное соотношение и предложен метод построения коллизии, использующий данное соотношение. Следуя работе [180], опишем обобщение атаки Рудского, применительно к определенной ранее в разделе 3.3.1 перестановке π.

Предположим, что нарушителю известны значения $\delta, \mu \in \mathbb{V}_{2w}$, где $\delta = (\delta_0||\delta_1)$, и подмножество $V_{\delta,\mu} \subset \mathbb{V}_{2w}$ такое, что для каждого $u \in V_{\delta,\mu}$ выполнено равенство

$$\pi(u) \oplus \pi(u \oplus \delta) = \mu. \tag{3.33}$$

Будем считать, что ассоциированные данные y не определены и рассмотрим блоки зашифрованного текста $c_0, c_1, c_2, c_3, \ldots, c_{2l_0-1}$, определим

$$\begin{aligned} u_{0,0} &= c_0 \oplus \gamma_{0,0} \\ u_{0,1} &= c_1 \oplus \gamma_{0,1} \\ u_{1,0} &= c_2 \oplus \gamma_{1,0} \\ u_{1,1} &= c_3 \oplus \gamma_{1,1} \end{aligned}$$

и предположим, что $(u_{0,0}||u_{0,1}), (u_{1,0}||u_{1,1}) \in V_{\delta,\mu}$, т.е.

$$\begin{aligned} \pi(u_{0,0}||u_{0,1}) &= \pi(u_{0,0} \oplus \delta_0||u_{0,1} \oplus \delta_1) \oplus \mu, \\ \pi(u_{1,0}||u_{1,1}) &= \pi(u_{1,0} \oplus \delta_0||u_{1,1} \oplus \delta_1) \oplus \mu, \end{aligned}$$

тогда, аналогично (3.32), выполнены равенства

$$\begin{aligned} s \oplus \pi\big(E_{k_1}(\gamma_{l_0,0})||E_{k_1}(\mathrm{len}_2(x) \oplus \gamma_{l_0,1})\big) \oplus \\ \oplus \sum_{n=2}^{l_0-1} \pi\big(E_{k_1}(x_{2n} \oplus \gamma_{n,0})||E_{k_1}(x_{2n+1} \oplus \gamma_{n,1})\big) = \\ = \pi\big(E_{k_1}(x_0 \oplus \gamma_{0,0})||E_{k_1}(x_1 \oplus \gamma_{0,1})\big) \oplus \\ \pi\big(E_{k_1}(x_2 \oplus \gamma_{1,0})||E_{k_1}(x_3 \oplus \gamma_{1,1})\big) = \\ = \pi\big(c_0 \oplus \gamma_{0,0}||c_1 \oplus \gamma_{0,1}\big) \oplus \pi\big(c_2 \oplus \gamma_{1,0}||c_3 \oplus \gamma_{1,1}\big) = \\ = \pi\big(u_{0,0}||u_{0,1}\big) \oplus \pi\big(u_{1,0}||u_{1,1}\big) = \\ = \pi(u_{0,0} \oplus \delta_0||u_{0,1} \oplus \delta_1) \oplus \pi(u_{1,0} \oplus \delta_0||u_{1,1} \oplus \delta_1). \end{aligned} \tag{3.34}$$

Из указанных равенств следует, что зашифрованный текст

$$c_0 \oplus \gamma_{0,0} \oplus \delta_0,\ c_1 \oplus \gamma_{0,1} \oplus \delta_1,\ c_2 \oplus \gamma_{1,0} \oplus \delta_0,\ c_3 \oplus \gamma_{1,0} \oplus \delta_1,\ \ldots c_{2l_0-1}$$

дает то же значение кода аутентификации, что и исходный зашифрованный текст.

Поскольку значения γ_0, γ_1 нарушителю неизвестны, то он не может в явном виде проверить выполнение условия $(u_{0,0}||u_{0,1}), (u_{1,0}||u_{1,1}) \in V_{\delta,\mu}$. Вместе с тем, нарушитель может выбирать значения $c_0, c_1, c_2, c_3 \in_R \mathbb{V}_w$ случайным образом и ожидать выполнение равенства (3.34) с некоторой вероятностью. Поскольку успех данной атаки зависит от мощности множества $V_{\delta,\mu}$, то нарушитель должен подобрать удовлетворяющие (3.33) значения δ, μ так, чтобы максимизировать величину $|V_{\delta,\mu}|$.

Напомним, что ранее перестановка π была определена равенством (3.17), представляющим собой 4-х раундовую сеть Фейстеля с раундовой функцией $L(S(x \oplus \zeta))$.

Поскольку разностное соотношение (3.33) зависит только от отображения ϕ (остальные преобразования линейны), необходимо опробовать все возможные значения $a, b \in \mathbb{V}_8$ и отобрать те значения, для которых мощность множества

$$V_{a,b} = \{x \in \mathbb{V}_8 : \phi(x) \oplus \phi(x \oplus a) = b\},$$

имеет максимально возможное значение.

Далее, используя элементы множеств $V_{a,b}$ в качестве координат вектора $u \in \mathbb{V}_{2w}$, можно сконструировать искомое множество $V_{\delta,\mu}$ и предъявить разностное соотношение для перестановки π. Исходя из данного способа построения вытекает оценка

$$|V_{\delta,\mu}| \leqslant |V_{a,b}|^{\frac{2w}{8}},$$

тогда вероятность выбрать два случайных вектора $u_1, u_2 \in V_{\delta,\mu}$ удовлетворяет неравенству

$$p \leqslant \left(\frac{|V_{a,b}|^{\frac{w}{4}}}{2^{2w}}\right)^2$$

В ряде работ, см., например, работы [126, 300], было показано, что для выбранного отображения ϕ максимальное значение величины $|V_{a,b}|$ не принимает значение, большее чем 8. Тогда

$$p \leqslant \left(\frac{2^{\frac{3w}{2}}}{2^{4w}}\right) = 2^{-\frac{5w}{2}} = \frac{1}{2} \cdot 2^{\left(1-\frac{5w}{2}\right)}$$

Следовательно, для того, чтобы данная атака сработала с вероятностью не менее $\frac{1}{2}$ потребуется не менее $2^{\frac{5w}{2}-1}$ попыток случайного выбора значений зашифрованного текста $c_0, c_1, c_2, c_3 \in_R \mathbb{V}_w$. Это число существенно больше, чем число шагов алгоритма, основанного на «парадоксе дней рождений».

Суммируя сказанное отметим, что разработанное сжимающее отображение режима «XTSMAC» обладает следующими свойствами:

- использование линейной формы позволяет обеспечить свойство равновероятности и, тем самым, получить точную оценку вероятности построения коллизии,
- использование неизвестной для нарушителя последовательности значений $\gamma_{-1}, \gamma_0, \gamma_1, \ldots$ позволяет защититься от атак, основанных на перестановках блоков открытых или зашифрованных данных,

- явная зависимость кода аутентификации от длин ассоциированных и открытых данных позволяет обеспечить защиту от атак, основанных на неверной интерпретации входных данных,
- зашифрование значений линейной формы обеспечивает защиту от атак, направленных на определение последних элементов последовательности $\gamma_{-1}, \gamma_0, \gamma_1, \ldots$,
- в качестве нелинейной перестановки π использован «слабый» блочный шифр, гарантирующий малую вероятность построения разностного соотношения.

§ 3.3.5. Результаты реализации на ЭВМ

В заключение параграфа приведем результаты практической реализации режима аутентифицированного шифрования «XTSMAC», подтверждающие целесообразность его применения в средствах защиты информации.

Пусть открытые и ассоциированные данные представлены в виде конкатенации из l и, соответственно, r блоков

$$x = x_0 || \ldots || x_{l-1}, \quad y = y_0 || \ldots || y_{r-1}.$$

Тогда, алгоритмическая сложность реализации режима «XTSMAC» не превосходит

$$T_1 = \big(12 + r + l\big)\epsilon + \left(\left\lceil \frac{r+1}{2} \right\rceil + \left\lceil \frac{l+1}{2} \right\rceil\right) t_1,$$

где ϵ – алгоритмическая сложность реализации операции зашифрования одного блока информации, а t_1 – сложность реализации нелинейного преобразования π и вычисления следующего элемента последовательности $\gamma_0, \gamma_1, \ldots$.

С другой стороны, алгоритмическая сложность реализации режима «MGM» составляет

$$T_2 = (3 + 2r + 2l)\epsilon + (r + l)t_2,$$

где t_2 – сложность реализации операции умножения двух произвольных элементов поля $\mathbb{F}_{2^w}$.

Полученные значения говорят о том, что режим «MGM» оказывается эффективнее только на коротких сообщениях, чья длина не превосходит[6] $8w$ бит. Если положить величины t_1, t_2 равными нулю, то легко видеть, что при $r + l \geqslant 9$ выполнено неравенство $T_1 \leqslant T_2$.

Поскольку, на практике, выполнены неравенства $0 < t_1 < t_2$, а значения t_1, t_2 зависят от конкретной аппаратной или программой реализации режима шифрования, то точная длина сообщений, на которых режим «XTSMAC» становится эффективнее, определяется экспериментально.

В рамках разработанного автором программного СКЗИ с открытыми исходными текстами, см. [144], были получены следующие показатели скорости работы различных алгоритмов аутентифицированного шифрования для шифра «Магма» (вычисления производились на персональной ЭВМ с процессором Intel (i5-8250U) и тактовой частотой 1.60GHz).

Mode	Speed, MBs	%
ecb-magma	49,411111	100
ctr-magma	48,110868	97
ofb-magma	48,040074	97
cfb-magma	47,849739	96
cbc-magma	49,273965	99
xts-magma	47,586117	95
acpkm-magma	46,089132	93
cmac-magma	48,340254	97
mgm-magma	23,424500	47
ctr-cmac-magma	24,101876	48
ctr-hmac-magma-streebog256	35,713519	72
ctr-hmac-magma-streebog512	35,713049	72
xtsmac-magma	45,877878	92

Таблица 3.1: Режимы аутентифицированного шифрования для блочного шифра «Магма».

Стандартные режимы шифрования – ecb (режим простой замены), ctr (режим гаммирования), ofb (режим гаммирования с обратной связью по

[6]Здесь стоит отметить, что если эффективность на коротких сообщениях принципиальна, то существуют структурные возможности ее снижения. Во первых, секретные значения $\zeta_0, \ldots, \zeta_3$ могут не вырабатываться из инициализационного вектора, а входить в ключевое множество режима – это позволит сэкономить четыре операции зашифрования одного блока. Кроме того, длины ассоциированных данных могут не зашифровываться как сейчас, в виде двух отдельных блоков, а помещаться в инициализационный вектор длины w бит для режима CBC, используемого для зашифрования значения линейной формы. Это позволит сэкономить еще одну операцию зашифрования. Указанные оптимизации позволят снизить преимущество режима «MGM» до сообщений длины $3w$ бит.

выходу), cfb (режим гаммирования с обратной связью по шифртексту), cbc (режим простой замены с зацеплением) и cmac (режим выработки имитовставки) определены в [283], режим «ACPKM» определен в рекомендациях [237, 356], функции хеширования семейства «Стрибог» определены в стандарте [281], функция выработки имитовставки «HMAC» – в рекомендациях [372]. Аналогичные результаты для блочного шифра «Кузнечик» приведены в таблице 3.2.

Mode	Speed, MBs	%
ecb-kuznechik	148,898152	100
ctr-kuznechik	131,583606	88
ofb-kuznechik	128,942234	86
cfb-kuznechik	129,088006	86
cbc-kuznechik	129,898637	87
xts-kuznechik	124,315330	83
acpkm-kuznechik	65,034391	43
cmac-kuznechik	127,687063	85
mgm-kuznechik	62,764779	42
ctr-cmac-kuznechik	64,808306	43
ctr-hmac-kuznechik-streebog256	67,545720	45
ctr-hmac-kuznechik-streebog512	67,537448	45
xtsmac-kuznechik	106,946651	71

Таблица 3.2: Режимы аутентифицированного шифрования для блочного шифра «Кузнечик».

Заключение к § 3.3

В § 3.3 предложен режим аутентифицированного шифрования «XTSMAC», в основу которого положен определенный в § 3.2 класс универсальных ключевых функций хеширования.

Доказана теорема о выполнении свойства равновероятности для сжимающего отображения режима «XTSMAC» при фиксированных ключах шифрования и аутентификации. Рассмотрен ряд атак и обосновано применение структурных элементов режима «XTSMAC». Приведены результаты практической реализации предложенного режима, показывающие его преимущество в скорости при программной реализации над регламентированными в Российской Федерации алгоритмами аутентифицированного шифрования.

глава 4

Вопросы взаимодействия средств защиты информации

Настоящая глава посвящена вопросам разработки и обоснования безопасности криптографических протоколов защищенного взаимодействия. Во введении к главе рассматривается принятый в Российской Федерации порядок оценки безопасности информационных систем, в состав которых входят криптографические механизмы защищенного взаимодействия.

Во втором параграфе рассматривается новый класс гибридных схем шифрования, обладающих смешанной ключевой системой. Уточняется базовая модель нарушителя и, в рамках этой модели, доказывается теорема о стойкости предложенного семейства гибридных схем относительно задач определения секретного ключа аутентификации, дешифрования и навязывания сообщений. В заключение параграфа предлагается протокол передачи ключевой информации.

В третьем параграфе предлагается новый протокол выработки общего ключа со взаимной аутентификацией субъектов взаимодействия, реализуемой с использованием электронной подписи. Доказывается теорема о стойкости данного протокола относительно задач определения общего секретного ключа, дешифрования и навязывания передаваемой в ходе выполнения протокола информации. Также рассматривается семейство криптографических протоколов, предназначенное для обеспечения защищенного взаимодействия в сетях «Интернета вещей».

В четвертом параграфе предлагается формальная модель, имитирующая криптографический протокол в виде дискретной динамической системы. В рамках данной модели формализуется перечень свойств безопасности и определяются показатели эффективности мер защиты, обеспечиваемых криптографическим протоколом. Для получения численных значений показателей эффективности мер защиты предлагается метод, использующий оценки трудоемкости компрометации криптографических преобразований, изменяющих состояния дискретной динамической системы. В завершение главы, предлагается методика проведения исследования безопасности криптографических протоколов.

Изложенные в настоящей главе результаты опубликованы в следующих работах автора [174, 181, 269, 299, 322, 327, 328, 331, 334, 340, 341, 342, 343], семь из которых входят в перечень рецензируемых научных изданий ВАК, а также использованы в программах для ЭВМ [376, 377, 378, 379].

§ 4.1. Введение

Рассматриваемые в настоящей главе криптографические механизмы предназначены для обеспечения защищенного обмена информацией с трехсторонним участием – двух средств защиты информации и доверенного центра, обеспечивающего функции аутентификации участников взаимодействия. При этом участие доверенного центра в обмене информацией может быть косвенным, т.е. без отправки и получения сообщений.

Как правило, криптографический механизм представляет из себя совокупность, состоящую из нескольких схем и протоколов. В состав такой совокупности входят:

- протокол односторонней, взаимной или многосторонней аутентификации участников информационного взаимодействия,
- протокол выработки общей для участников взаимодействия ключевой информации, действующей в рамках одной сессии информационного взаимодействия,
- транспортный протокол, предназначенный для передачи защищенной информации по каналам связи,
- процедуры выработки производной ключевой информации, контроля за временем и объемом используемой ключевой информации,
- вспомогательные протоколы, предназначенные для передачи ошибок информационного взаимодействия, квитирования абонентов, инициализации процедуры выработки нового сессионного ключа и т.п.

В ряде случаев, например в информационных системах «Интернета вещей» или в сетях беспроводной связи стандарта 5G, к указанной совокупности могут добавляться протоколы взаимодействия различных сегментов общей сети связи. Из перечисленной совокупности протоколов нас, в первую очередь, будут интересовать транспортные протоколы и протоколы аутентификации и протоколы выработки общего ключа.

Примерами стандартизированных в Российской Федерации транспортных протоколов могут служить:

- протокол ESP, входящий в семейство криптографических механизмов IPSec, см. [370],
- протокол безопасности сетевого уровня IPLir, см. [369], применяемый в средствах криптографической защиты ViPNet,

- протокол формирования защищенных записей, входящий в состав криптографического механизма безопасности транспортного уровня (TLS 1.3), см. [367],

- транспортный протокол TLP, входящий в состав криптографических механизмов взаимодействия контрольных и измерительных устройств (SP FIOT), см. рекомендации [365].

Среди зарубежных транспортных криптографических протоколов можно отметить протоколы MACSec [113], L2TP [140], DTLS [214] и т.п.

К транспортным протоколам также целесообразно отнести схемы асимметричного и гибридного шифрования, не предполагающие интерактивного обмена в процессе зашифрования сообщения. Примером таких схем могут служить классические схемы RSA [189] и NTRU [108, 171], стандартизированные [115, 363] или предлагаемые к стандартизации решения [10, 59], см. также раздел 4.2.

В большинстве случаев ключи для транспортных протоколов вырабатываются в ходе взаимодействия субъектов, после односторонней или взаимной аутентификации участников взаимодействия. Несмотря на то, что протоколы аутентификации и выработки общего ключа предназначены для обеспечения различных криптографических функций, см. [48, 355], в большинстве приложений они выполняются в виде единого многофункционального протокола[1].

Примерами протоколов, реализующих несколько криптографических функций, служат TLS [212, 367], семейство протоколов SIGMA [133], протоколы IKEv2 [120, 317], SSH [261] или WireGuard [71], схемы выработки общего ключа с аутентификацией [352], протоколы промышленного «Интернета вещей» [365, 368] и т.д.

Как показано далее в разделе 4.4, объединение протоколов аутентификации и выработки общего ключа является необходимым для обеспечения конфиденциальности и целостности передаваемой информации. Мы вводим понятие «свойства безопасности» и показываем, что необходимость выполнения протоколом заданных свойств безопасности, к которым относится, в частности, аутентификация участников взаимодействия, приводит к необходимости включения в протокол заданных фрагментов, реализующих различные криптографические функции.

Кроме того, предложенный фрагментарный подход к построению протоколов позволяет предъявить метод определения численных значений одного или нескольких *показателей эффективности* (термин определяет-

[1] В зарубежной литературе для таких протоколов принято использовать название — AKA, Authentication and Key Agreement protocol, см., например, [131].

ся в ГОСТ Р 50922-2006, см. [284, раздел 2.9]), которые позволяют оценить уровень мер защиты, обеспечиваемых криптографическим протоколом.

Напомним, что принятый в Российской Федерации порядок оценки безопасности информационных (автоматизированных) систем заключается в следующем.

1. Формируется модель угроз, создающих опасность нарушения безопасности передаваемой информации. Разработка модели угроз проводится для конкретной системы на основе базовых моделей, регламентируемых ФСТЭК, см. [387, 388], ФСБ России, отраслевыми министерствами, см. например [320], а также государственными стандартами в области защиты информации.

В настоящей работе будем основываться на базовой модели угроз, регламентируемой стандартом ГОСТ Р ИСО/МЭК 27033-1:2009 [288]. Данная модель включает в себя:

- угрозу несанкционированного доступа к передаваемой информации (нарушение конфиденциальности),
- угрозу несанкционированной передачи информации,
- угрозу несанкционированного изменения информации (нарушение целостности),
- угрозу отказа от факта отправки или приема сообщения,
- угрозу внесения вредоносного программного обеспечения,
- угрозу отказа в обслуживании или предоставлении услуг (нарушение доступности).

2. Формируется модель нарушителя, содержащая в себе совокупность возможностей, которые могут быть использованы при создании способов, подготовке и проведении атак, целью которых является реализация перечисленных ранее угроз безопасности.

Будем основываться на модели нарушителя, регламентируемой рекомендациями по стандартизации Р 1323565.1.012-2017 [355]. В данной модели инструментом реализации угроз безопасности являются проводимые нарушителем атаки на информационную систему и, в частности, на криптографические протоколы, обеспечивающие безопасность передачи информации. Каждая атака нарушителя может быть задана следующими характеристиками:

а) объектом проведения атаки, безопасность которого должна обеспечиваться в течение определенного периода времени и/или определенного этапа жизненного цикла средства защиты информации;

б) возможностями, которые могут быть использованы при создании способов, подготовке и проведении атак; каждая возможность определяется сведениями, а также техническими средствами, используемыми при создании способов, подготовке и проведении атак;

в) местом проведения атаки.

Применительно к анализу криптографических протоколов понятие объекта атаки позволяет уточнить сформулированный выше перечень угроз и включить в него, в качестве объектов атаки, открытые параметры протоколов.

Возможности нарушителя в части воздействия на канал связи, по которому происходит обмен защищаемой информацией, принято описывать расширенной моделью Долева-Яо, см. [69]. В рамках данной модели нарушитель обладает следующими возможностями:

- нарушителю известны форматы всех передаваемых сообщений;
- нарушитель может перехватить и получить содержимое любого сообщения от любого пользователя в сети связи;
- нарушитель может инициировать установление соединения с любым другим пользователем;
- нарушитель может изменять содержимое передаваемых пользователями сообщений и, в частности, посылать сообщения от имени другого пользователя;
- нарушитель может использовать все доступные ему комбинации сообщений или частей сообщений для формирования новых сообщений, в том числе и расшифровывать и зашифровывать сообщения с помощью известных ему ключей шифрования, применяя любые доступные алгоритмы;
- нарушитель является полноценным пользователем сети, обладающим корректным собственным идентификатором и допустимым множеством ключевой информации;
- нарушитель может проводить накопление всей переданной в сети связи информации, проводить ее анализ с использованием специализированных технических средств и использовать результаты анализа для компрометации криптографических схем и протоколов;

- нарушитель может организовывать одновременное выполнение некоторого числа сессий одного и того же протокола защиты информации; сессии могут выполняться одновременно для различных участников протокола, при этом нарушитель может использовать информацию, передаваемую в ходе всех выполняемых сессий протокола.

Перечисленные методы реализации угроз безопасности принято называть «активными» атаками.

Также в модели Долева-Яо нарушитель может проводить «пассивные» атаки на протокол, основанные на перлюстрации и последующем криптографическом анализе передаваемых в ходе выполнения протокола сообщений. При проведении пассивных атак предписанное спецификацией выполнение прокола не меняется — нарушитель не изменяет передаваемые сообщения, не инициирует соединений и не вмешивается в логику взаимодействия пользователей сети.

Для усложнения рассматриваемой модели будем допускать, что нарушитель может компрометировать набор долговременных ключей любого потенциального участника протокола, который не является участником атакуемой сессии выполнения протокола. Вариант нарушителя данного типа описан в [21, 69] и использовался при реализации ряда атак, например, на протокол Нидхема-Шредера [146].

3. Проводится исследование криптографических схем и протоколов, обеспечивающих безопасность сети связи; также проводится исследование входящих в состав протокола криптографических преобразований. Целью проведения указанных исследований является определение численных значений одного или нескольких показателей эффективности, которые позволяют оценить уровень защищенности информационной системы. Система считается защищенной (безопасной), если полученные в ходе исследования значения показателей эффективности попадают в заданную область, установленную нормативными, правовыми документами или требованиями по безопасности.

До последнего времени в Российской Федерации не существовало единой методологии определения показателей эффективности и их значений, применительно к криптографическим протоколам. В зарубежных публикациях предлагалось применять несколько подходов:

- базовую модель Белларе-Рогавея, см. [26], и ее модификации [27, 28, 39, 40], в которых в качестве показателя эффективности может рассматриваться вероятность нарушения формального определения безопасного протокола; получение точных численных оценок показателя эффективности в данной модели не предполагается;

- модель Конетти-Кравчука, см. [53], и ее модификации [134, 137, 159], в которых в качестве показателя эффективности выступает величина отклонения от $\frac{1}{2}$ вероятности различения двух моделей – практической модели протокола в рамках описанной выше модели нарушителя и «идеальной» модели протокола, реализующей обмен сообщениями по «идеальному» каналу связи без искажений и активного нарушителя;

- ряд подходов, предназначенных для верификации протоколов; в данных подходах показатели эффективности защиты либо не определяются, либо не могут быть явно определены – подход Долева-Яо на основе тождества слов [69], подходы на основе абстрактных автоматов [96], «логики доверия» [50], «пространства нитей» [248], логического [4, 5] и последовательного программирования [107], spi-исчисления [1] и т.п.

Среди ранних работ по анализу криптографических протоколов стоит выделить статьи Рабина [207] и Голдвассера-Микали [97]. В монографиях [48, 151] могут быть найдены более поздние обзоры зарубежных публикаций.

В отечественных работах по анализу протоколов принято использовать два похода:

- применение «практической стойкости», т.е. классического криптографического анализа для получения оценок стойкости используемых в протоколе криптографических примитивов, см. [270, 345, 347]; при данном подходе показателем эффективности служит минимальное из всех возможных значений трудоёмкости реализации известных атак на криптографические преобразования;

- применение теории «доказуемой стойкости», позволяющей исследовать безопасность протоколов в заданных вероятностных моделях поведения нарушителя с ограниченными вычислительными ресурсами; аналогично методу Канетти-Кравчука в качестве показателя эффективности в данном подходе выступает величина отклонения от $\frac{1}{2}$ вероятности различения заданных параметров моделей от случайных равновероятно распределённых величин, см., например, работы [201, 346].

В настоящей диссертационной работе предлагается модель, представляющая криптографический протокол в виде дискретной динамической системы (неавтономного автомата), фрагменты которой определяются предъявляемыми к протоколу свойствами безопасности.

§ 4.2. Схемы гибридного шифрования

На протяжении всей главы будем использовать следующие обозначения. Обозначим символом $\mathbb{B} = \{\mathit{false}, \mathit{true}\}$ булево множество, элементы которого принимают значения «истина» или «ложь», символом $\mathbb{A}$ — множество кодов аутентификации, символом $\mathbb{K}$ — множество ключей симметричных алгоритмов шифрования, символом $\mathbb{K}_a$ — множество ключей аутентификации, символом $\mathbb{K}_c$ — множество ключей проверки кода аутентификации и будем считать, что для указанных множеств существуют эффективно вычислимые вложения

$$\mathbb{A} \subset \mathbb{V}_\infty, \quad \mathbb{K} \subset \mathbb{V}_\infty, \quad \mathbb{K}_a \subset \mathbb{V}_\infty, \quad \mathbb{K}_c \subset \mathbb{V}_\infty.$$

Также будем считать, что задана функция $\boldsymbol{h} : \mathbb{K}_a \to \mathbb{K}_c$ такая, что для любого аутентификации $k_a \in \mathbb{K}_a$ найдется ключ проверки кода аутентификации $k_c \in \mathbb{K}_c$, определяемый равенством $k_c = \boldsymbol{h}(k_a)$.

Требования, накладываемые на функцию $\boldsymbol{h}$ зависят от используемой ключевой системы. При использовании симметричной ключевой системы будем считать, что выполнено равенство $\mathbb{K}_a = \mathbb{K}_c$, а $\boldsymbol{h}$ есть тривиальное отображение, не изменяющее значение своего аргумента. При использовании асимметричной ключевой системы будем считать, что функция $\boldsymbol{h}$ является однонаправленной, т.е. эффективно вычислимой функцией для которой неизвестен эффективный алгоритм обращения, см. [383, стр. 79].

Рассмотрим две функции — функцию выработки кода аутентификации $\mathit{mac} : \mathbb{K}_a \times \mathbb{V}_\infty \to \mathbb{A}$ и функцию проверки кода аутентификации $\mathit{conf} : \mathbb{K}_c \times \mathbb{V}_\infty \times \mathbb{A} \to \mathbb{B}$ такие, что

$$\mathit{conf}\,(k_c, \xi, \mathit{mac}(k_a, \xi)) = \mathit{true},$$

для любой тройки значений k_a, k_c и $\xi \in \mathbb{V}_\infty$.

Определение 4.1. *Пусть $\xi_1, \xi_2 \in \mathbb{V}_\infty$, $\xi_3 \in \mathbb{A}$ — тройка значений, первое из которых называется «заголовком» сообщения, второе – «телом» сообщения, а третье – кодом аутентификации сообщения, удовлетворяющим равенству $\xi_3 = \mathit{mac}(k_a, \boldsymbol{f}(\xi_1, \xi_2))$ для некоторого отображения $\boldsymbol{f}$, определяемого спецификацией протокола.*

Под криптографическим транспортным протоколом будем подразумевать процесс однонаправленной передачи по каналу связи от одного субъекта взаимодействия к другому тройки значений (ξ_1, ξ_2, ξ_3), а также процесс приема из канала связи указанной тройки, и проверки равенства

$$\mathit{conf}(k_c, \boldsymbol{f}(\xi_1, \xi_2), \xi_3) =^? \mathit{true}.$$

Очередность передачи значений ξ_1, ξ_2, ξ_3 определяется спецификацией транспортного протокола.

Примерами криптографических транспортных протоколов, использующих симметричную ключевую систему, могут служить следующие протоколы, рекомендованные для использования на территории Российской Федерации:

- «протокол защиты информации» ESP, входящий в семейство криптографических механизмов IPSec;

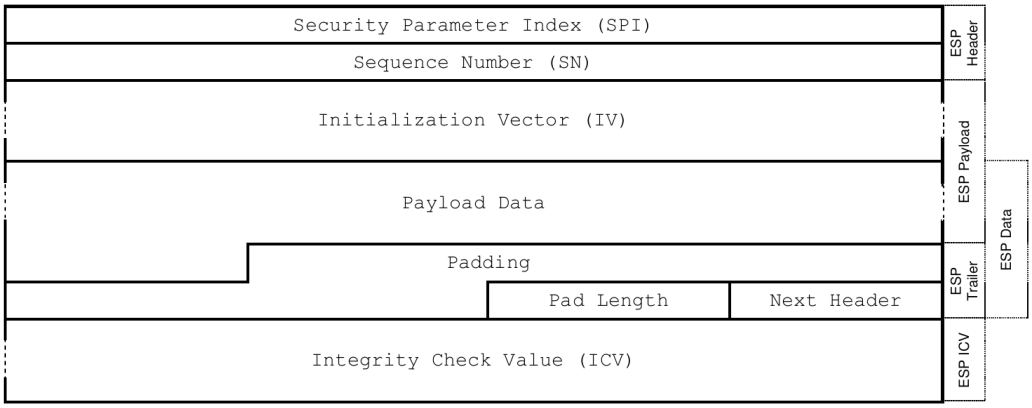


Рис. 4.1: Формат пакета протокола ESP, см. [370].

- «протокол безопасности сетевого уровня» IPLir, применяемый в средствах криптографической защиты ViPNet;

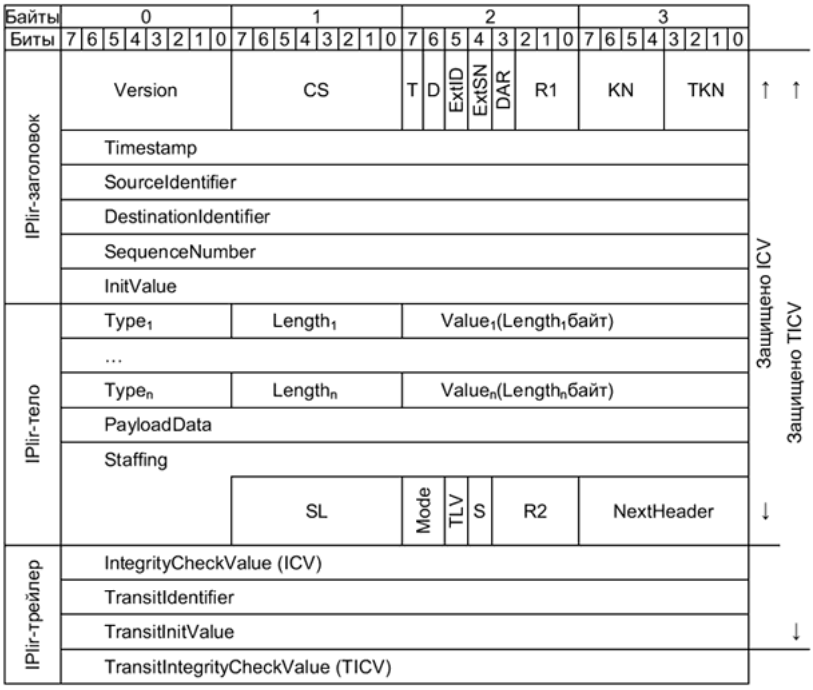


Рис. 4.2: Формат пакета протокола IPLir, см. [369].

- «протокол формирования защищенных записей», входящий в состав механизма безопасности транспортного уровня (TLS 1.3);

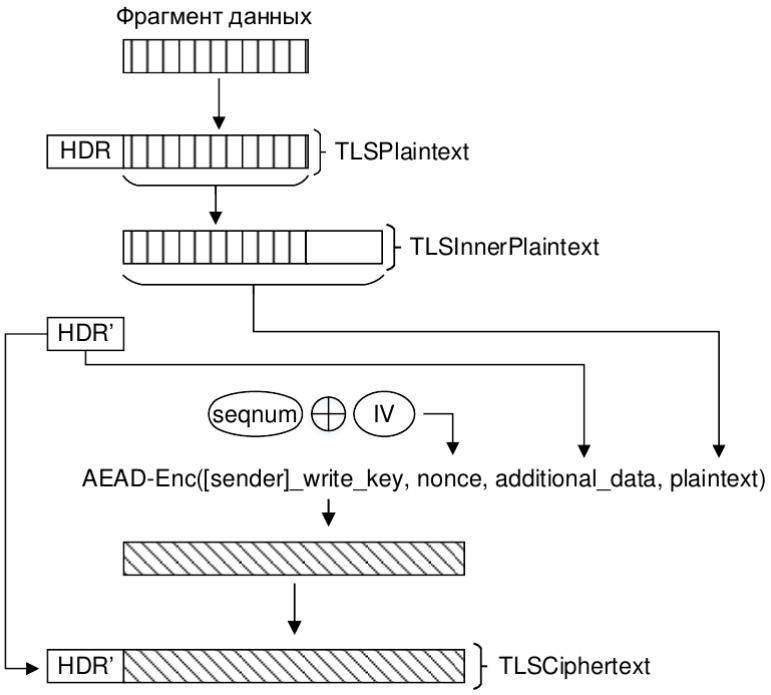


Рис. 4.3: Защищенная запись протокола TLS 1.3, см. [367].

- «транспортный протокол» TLP, входящий в состав криптографических механизмов взаимодействия контрольных и измерительных устройств (SP FIOT).

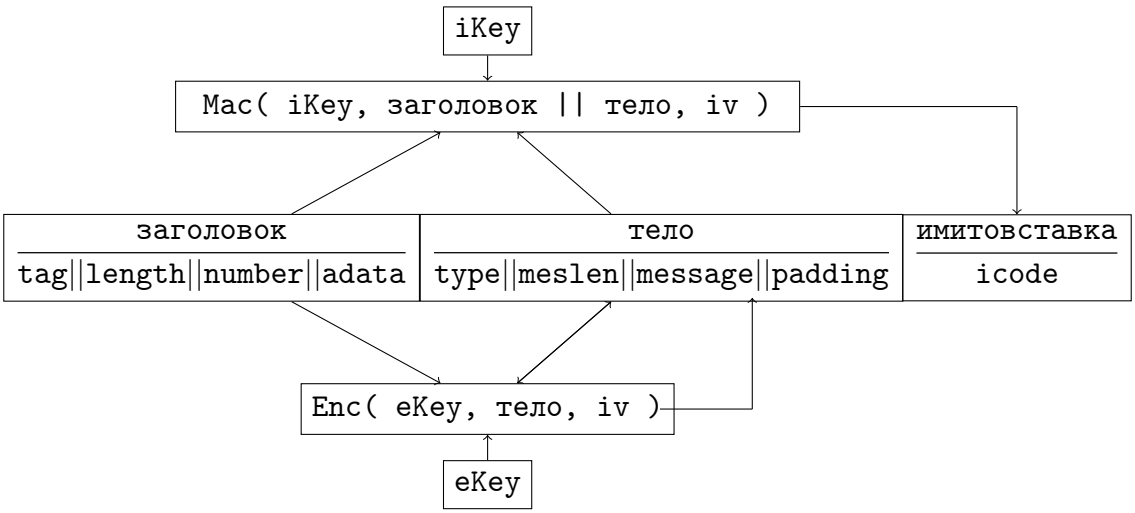


Рис. 4.4: Формат пакета транспортного протокола TLP, см. [365].

Все упомянутые протоколы выполняются после протокола выработки общего ключа, открывающего сессию защищенного взаимодействия, и обладают следующими свойствами:

- для шифрования и контроля целостности передаваемых данных используются симметричные ключи, вырабатываемые из общего сессионного ключа,
- передаваемые данные обрабатываются независимо, фрагментами малой длины; при этом длина обрабатываемого фрагмента ограничивается максимально допустимой длиной пакета протокола канального уровня,
- аутентификация отправителя и получателя данных не производится, а наследуется от протокола выработки общего ключа.

Классические асимметричные схемы обеспечивают аутентификацию получателя сообщений и выработку ключа шифрования без необходимости предварительного выполнения протокола выработки общего ключа. Основным недостатком большинства асимметричных схем является существенное увеличение длины зашифрованного сообщения по сравнению с длиной открытых данных.

Гибридные схемы позволяют избавиться от данного недостатка. Они не только наследуют свойства асимметричных схем, но и позволяют передавать данные столь же эффективно, как и в случае традиционных транспортных протоколов.

Дадим определение, которое позволит объединить в рамках одной конструкции транспортные протоколы, а также асимметричные и гибридные схемы шифрования.

Определение 4.2. *Рассмотрим защищенное взаимодействие двух субъектов A и B, в рамках которого субъект A должен передать сообщение $m \in \mathbb{V}_\infty$ субъекту B. Будем считать, что субъекты обладают идентификаторами, соответственно, $ID_A, ID_B \in \mathbb{V}_\infty$, а также определены следующие элементы ключевой системы.*

- *Пара ключей субъекта B – ключ проверки кода аутентификации $k_c \in \mathbb{K}_c$, а также ключ аутентификации $k_a \in \mathbb{K}_a$, определяемый равенством*

$$k_a = \mathbf{h}(k_c)$$

для некоторой однонаправленной функции $\mathbf{h}$. Ключ аутентификации k_a должен быть заверен в доверенном центре и однозначно связан с идентификатором ID_B.

- *Общая для двух субъектов взаимодействия ключевая информация* $k_{AB} \in \mathbb{K}$, *связанная с идентификаторами* ID_A *и* ID_B,

Тогда, под гибридной схемой взаимодействия будем понимать следующую последовательность действий субъектов A *и* B.

1. *Субъект* A, *желающий отправить сообщение* $m \in \mathbb{V}_\infty$ *субъекту с идентификатором* ID_B, *получает в доверенном центре ключ аутентификации* k_a, *после чего проверяет валидность данного ключа и его соответствие идентификатору* ID_B.

2. *Субъект* A *разбивает исходное сообщение на* $s \in \mathbb{N}$ *фрагментов* $m = m_1||\cdots||m_s$ *и для каждого фрагмента с индексом* i *выполняет следующую последовательность действий.*

   2.1) *Формируются производные ключи шифрования* $ek_i \in \mathbb{K}$ *и имитозащиты* $ik_i \in \mathbb{K}$, *а также синхропосылка* $iv_i \in \mathbb{V}_\infty$

   $$\{ek_i, ik_i, iv_i\} = \mathbf{kdf}(k_a, k_{AB}, \zeta_i, i, \ldots),$$

   где $\zeta_i \in_R \mathbb{K}$ *реализация случайной, равномерно распределенной на множестве* $\mathbb{K}$ *величины, вырабатываемая с использованием датчика случайных чисел.*

   2.2) *Формируется заголовок* $\xi_{i,1} \in \mathbb{V}_\infty$, *позволяющий субъекту* B *восстановить производные ключи.*

   2.3) *С использованием производных ключей вычисляется зашифрованный текст* $\xi_{i,2} = \mathbf{enc}(ek_i, iv_i, m_i)$, *а также код аутентификации* $\xi_{i,3} = \mathbf{mac}(ik_i, iv_i, \mathbf{f}(\xi_{i,1}, \xi_{i,2}))$ *для некоторого отображения* $\mathbf{f}$, *определяемого спецификацией протокола.*

   2.4) *Тройка значений* $(\xi_{i,1}, \xi_{i,2}, \xi_{i,3})$ *передается в канал связи.*

3. *Для каждой из полученных троек* $(\xi_{i,1}, \xi_{i,2}, \xi_{i,3})$ *субъект* B *выполняет следующую последовательность действий.*

   3.1) *С использованием ключа проверки кода аутентификации* k_c *и заголовка* $\xi_{i,1}$ *субъект* B *вырабатывает производные ключи шифрования* ek_i *и имитозащиты* ik_i, *а также синхропосылку* iv_i.

   3.2) *Выполняет проверку кода аутентификации* $\mathbf{conf}(ik_i, iv_i, \mathbf{f}(\xi_{i,1}, \xi_{i,2}), \xi_{i,3}) =^? \mathbf{true}$ *и, в случае успешной проверки, расшифровывает сообщение* $m_i = \mathbf{dec}(ek_i, iv_i, \xi_{i,2})$.

Для шифрования фрагментов данных m_i и выработки или проверки кодов аутентификации $\xi_{i,3}$ допускается использование алгоритмов аутентифицированного шифрования, см. определение 3.9.

Наличие ключей аутентификации или общей ключевой информации является опциональным и влияет на свойства безопасности, обеспечиваемые гибридной схемой взаимодействия, см. § 4.4.1.

Добавим, что механизм связывания идентификатора субъекта с его ключом аутентификации зависит от ключевой системы:

- для асимметричных ключевых систем связывание происходит путем включения идентификатора и другой аутентифицирующей информации в состав сертификата открытого ключа участника протокола,
- для симметричных систем – уникальные идентификаторы $ID_{A_1}, ID_{A_2}, \ldots ID_{A_r}$, определяемые для некоторого целого $r \geqslant 2$, используются при выработке общей ключевой информации для группы субъектов $A_1, A_2, \ldots, A_r$; примером такой схемы является ключевая система, рекомендуемая к применению в [357, 358, 365].

§ 4.2.1. Базовая схема ECISPE с шифрованием при помощи полиномиального преобразования

В статье [341] автором, совместно с А.В. Пугачевым, была предложена асимметричная схема шифрования, использующая в качестве шифрующего преобразования полином малой степени. В данной схеме применяется гибридная ключевая система, содержащая как ключи аутентификации получателя сообщения, так и общую для субъектов взаимодействия ключевую информацию. Это позволило реализовать асимметричную схему, у которой длина зашифрованного текста совпала с длиной открытых данных (без учета заголовка и контрольной суммы)[2]. Далее, мы следуем статье [341] и рассматриваем схему шифрования, а также приводим ряд

[2]Мотивация разработки схемы ECISPE и ее место среди других асимметричных схем состоит в следующем. Хорошо известно, что примерами шифров «простой замены» являются блочный шифр, имеющий симметричную ключевую систему, и схема RSA, обладающая асимметричной ключевой системой. Примерами шифров, реализующих «многозначную замену» являются поточные шифры и схема Эль-Гамаля. При этом схема Эль-Гамаля увеличивает в два раза размер шифртекста по сравнению с размером открытого текста. Помимо этого, при реализации схемы Эль-Гамаля в группе точек эллиптической кривой требуется реализация операции представления открытого текста в виде точки кривой, что также приводит к уменьшению размера открытого текста. Одним из вариантов решения этой проблемы стала схема ECIES, применяющая шифрование данных с использованием режимов работы блочных шифров. Схема ECISPE (Elliptic Curve based Integrated Scheme with Polynomial Encryption) использует для шифрования полином малой степени с коэффициентами из кольца вычетов, не связанного с используемой эллиптической кривой, и может рассматриваться как режим гаммирования для асимметричного шифра.

ее модификаций, обеспечивающих различные эксплуатационные требования.

Зафиксируем открытые параметры схемы:

- эллиптическую кривую $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и точку на этой кривой $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ такую, что $\mathtt{ord}\, P = q$, где q – простое число;
- натуральное число w, удовлетворяющее неравенству $4p - 3 < 2^w$,

а также следующие криптографические преобразования:

- функцию $\boldsymbol{bin}$, предназначенную для преобразования точки эллиптической кривой в двоичную последовательность фиксированной длины, см. [361, раздел 5.1.2];
- функцию $\boldsymbol{kdf}_n$, предназначенную для выработки производной ключевой информации и определяемую равенством

$$\boldsymbol{kdf}_n(k, \xi) = \boldsymbol{hmac}_{256}(k_1, \xi||\mu_1)|| \cdots ||\boldsymbol{hmac}_{256}(k_1, \xi||\mu_n), \tag{4.1}$$

  где вспомогательный ключ $k_1 \in \mathbb{V}_{512}$ определяется равенством

$$k_1 = \begin{cases} k||0^{512-\mathrm{len}_2(k)}, & \text{если} \quad \mathrm{len}_2(k) < 512, \\ k, & \text{если} \quad \mathrm{len}_2(k) = 512, \\ \boldsymbol{hash}_{512}(k), & \text{иначе}, \end{cases}$$

  где $\boldsymbol{hash}_{512}$ –функция бесключевого хеширования, вырабатывающая код длины 512 бит, $\boldsymbol{hmac}_{256}$ — функция выработки имитовставки, регламентируемая в [372], а $\mu_1, \ldots, \mu_n \in \mathbb{V}_\infty$ различные, фиксированные двоичные последовательности; заметим, что функция $\boldsymbol{kdf}_n$ является модификацией определенной в [372, раздел 4.5] функции диверсификации `KDF_TREE_GOSTR3411_2012_256`;
- функцию выработки кода аутентификации $\boldsymbol{mac}$ и функцию проверки кода аутентификации $\boldsymbol{conf}$ такие, что

$$\boldsymbol{conf}\,(k_c, \xi, \boldsymbol{mac}(k_a, \xi)) = \boldsymbol{true},$$

  для любой тройки значений $k_a \in \mathbb{K}_a$, $k_c \in \mathbb{K}_c$ и $\xi \in \mathbb{V}_\infty$;

В ключевую систему рассматриваемой асимметричной схемы входят:

- ключ проверки кода аутентификации $k_c \in \mathbb{F}_q^*$ субъекта B и ключ аутентификации, представленный в виде точки $Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$, определяемой равенством

$$Q = [k_c]P.$$

- общий ключ $k_{AB} \in \mathbb{F}_q^*$, связанный с идентификаторами ID_A, ID_B субъектов взаимодействия.

Далее, будем использовать краткое обозначение ECISPE для асимметричной схемы шифрования, описываемой следующей процедурой зашифрования.

Алгоритм 4.1: Базовая схема ECISPE. Процедура зашифрования.

Вход : Сообщение $m \in \mathbb{V}_w$, ключ аутентификации $Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ получателя сообщения и общий ключ $k_{AB} \in \mathbb{F}_q^*$.

Выход : Зашифрованное сообщение (ξ_1, ξ_2, ξ_3).

1 Субъект вырабатывает случайное значение $\zeta \in_R \mathbb{F}_q^*$.

2 Субъект вычисляет точки $U = [\zeta]P$, $W = [\zeta]Q$ и $S = [k_{AB}]P$. Если выполнено равенство $U = \pm S$, то субъект возвращается на 1-й шаг алгоритма. Иначе, субъект формирует заголовок $\xi_1 = ID_A || \mathtt{bin}(U)$.

3 Субъект вырабатывает производный ключ имитозащиты

$$ik = \mathtt{kdf}_1(\mathtt{bin}(W), k_{AB}) \tag{4.2}$$

и вычисляет имитовставку $\xi_3 = \mathtt{mac}(ik, \xi_1 || m)$.

4 Субъект представляет точки $S = (x_S, y_S)$ и $W = (x_W, y_W)$ в аффинной форме и определяет величины

$$\begin{cases} \mu \equiv \dfrac{y_W - y_S}{x_W - x_S} \pmod p, \\ \beta \equiv \dfrac{y_S x_W - x_S y_W}{x_W - x_S} \pmod p, \end{cases} \tag{4.3}$$

5 Субъект представляет сообщение m как вычет из кольца $\mathbb{Z}_{2^w}$ и зашифровывает его при помощи сравнения

$$\xi_2 \equiv \mathtt{f}(m) \pmod{2^w},$$

где

$$\mathtt{f}(x) = \alpha x + \beta, \quad \alpha = 2(\mu + y_W) + 1, \quad \mathtt{f}(x) \in \mathbb{Z}_{2^w}[x].$$

6 Тройка (ξ_1, ξ_2, ξ_3) отправляется в канал связи.

При получении тройки (ξ_1, ξ_2, ξ_3) субъект B должен выполнить следующую последовательность действий.

- Выделить из заголовка ξ_1 точку U и проверить, что она принадлежит подгруппе, порожденной точкой P.

- Используя ключ проверки кода аутентификации k_c, субъект B должен вычислить точку

$$W = [k_c]U = [k_c\zeta]P = [\zeta]Q,$$

и, используя (4.3), определить коэффициенты $\mu, \beta \in \mathbb{F}_p^*$.

Легко видеть, что в силу построения выполнено неравенство

$$\alpha = 2(\mu + y_W) + 1 \leqslant 4(p-1) + 1 = 4p - 3 < 2^w,$$

более того, величина α нечетна и, следовательно, обратима по модулю 2^w.

- Расшифровать сообщение $m \in \mathbb{Z}_{2^w}$, используя сравнение

$$m \equiv \boldsymbol{f}^{-1}(\xi_2) \equiv (\xi_2 - \beta)\alpha^{-1} \pmod{2^w}. \tag{4.4}$$

- Используя общий ключ k_{AB} и равенство (4.2) определить производный ключ имитозащиты ik, и проверить, что значение $\boldsymbol{mac}(ik, \xi_1||m)$ совпадает с ξ_3.

Схеме шифрования ECISPE можно дать простую геометрическую интерпретацию. Рассмотрим координаты точек S, W эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ как пары целых неотрицательных чисел. Расположим эти точки на действительной плоскости и проведём через них прямую линию, после чего изменим угол наклона прямой и рассмотрим новую прямую, пересекающую исходную прямую в точке $(0, \beta)$.

Представим открытый текст m как целое неотрицательное число, определяющее абсциссу некоторой точки T на новой прямой, тогда шифртекст ξ_2 есть ордината точки T, см. рисунок 4.5.

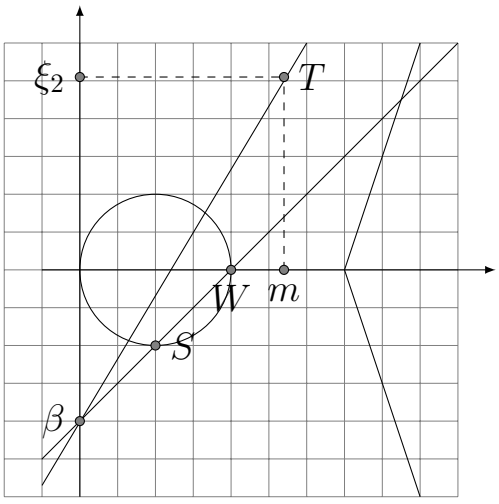


Рис. 4.5: Геометрическая интерпретация схемы ECISPE.

Переход с одной прямой на другую носит технический характер и нужен для того, чтобы угол наклона новой прямой (величина α) был обратимым элементом кольца $\mathbb{Z}_{2^w}$.

В силу случайности выбора величины $\zeta \in \mathbb{F}_q^*$, точка $W = [\zeta]Q$ принимает случайное месторасположение на плоскости и, как следствие, прямая линия, используемая для зашифрования сообщения m также является случайной.

Также заметим, что коэффициент μ может определяться не только сравнением (4.3), но и равенством (1.5), используемым для сложения точек S и W.

§ 4.2.2. Исследование безопасности схемы ECISPE

Рассмотрим несколько атак на предложенную схему асимметричного шифрования и укажем задачи, решение которых может привести к ее компрометации. Будем считать, что у нарушителя, желающего скомпрометировать схему, могут быть следующие цели:

- определение секретного ключа субъекта B,
- дешифрование переданного сообщения,
- навязывание ложного сообщения.

Мы будем использовать сформулированную выше на стр. 246 модель нарушителя, дополнительно предполагая, что нарушитель может использовать субъекта B в качестве «оракула» для расшифрования специально подобранных сообщений, т.е. нарушителю становятся известными все корректно расшифрованные сообщения, за исключением исходного сообщения, которое нарушителю необходимо дешифровать. При этом, возможности подделывать имитовставку у нарушителя нет. Такие возможности нарушителя, в англоязычной литературе, принято обозначать аббревиатурой CCA2 (расширенная атака с адаптивно подобранными открытыми текстами), более подробно см. статью [209].

Дополнительно, несколько упростим задачу нарушителю и будем предполагать, что ему известны координаты точки S, которые также можно рассматривать как долговременный ключ (знание нарушителем долговременного ключа S не приводит к возможности дешифрования переданного сообщения).

1. Сведение к задаче дискретного логарифмирования (DLP).

   Легко видеть, что сложность определения k_c — секретного ключа проверки кода аутентификации сообщения субъекта B основывается на трудоёмкости решения задачи дискретного логарифмирования в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$. Действительно, поскольку нарушителю известны все открытые параметры схемы и,

в частности, точка P порядка q, а также ключ аутентификации Q субъекта B, то секретный ключ может быть найден из уравнения

$$Q = [k_c]P, \quad k_c \in \mathbb{Z}_q^*.$$

Аналогично, если нарушителю каким-либо образом удалось определить точку W, используемую для зашифрования и расшифрования сообщений, то он может найти секретный ключ k_c из уравнения

$$W = [k_c]U, \quad k_c \in \mathbb{Z}_q^*.$$

Как было сказано ранее, см. § 1.1.2, решение задачи дискретного логарифмирования в группе точек эллиптической кривой, определённой над конечным простым полем $\mathbb{F}_p$, является сложной задачей, трудоёмкость которой оценивается величиной $\mathrm{O}(\sqrt{q})$.

2. Сведение к задаче Диффи — Хеллмана (DHP).

   Перейдём к рассмотрению вопроса о дешифровании передаваемого сообщения. Предположим, что нарушитель умеет находить решение, так называемой, задачи Диффи — Хеллмана в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, см. [151], т.е. по заданным точкам P, $U = [\zeta]P$, $Q = [k_c]P$, $P, U, Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$, нарушитель может определить точку $W \in \mathcal{E}$, удовлетворяющую равенству

   $$W = [\zeta k_c]P.$$

   Поскольку мы предполагаем, что нарушителю известна точка S, то он может воспользоваться сравнениями (4.3), определить значения $\mu, \beta \in \mathbb{F}_p$ и расшифровать передаваемое сообщение с использованием сравнения (4.4).

   Таким образом, нарушитель, который умеет решать задачу Диффи — Хеллмана в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$, может дешифровывать передаваемые сообщения. Следует отметить, что в настоящее время задача Диффи — Хеллмана в группе точек эллиптической кривой считается трудноразрешимой, а наиболее эффективный способ её решения заключается в сведении к задаче дискретного логарифмирования, см. [38].

   В случае, когда нарушитель умеет решать задачу Диффи — Хеллмана, но не знает точного значения координат точки S, он может предложить атаку для их определения.

   Действительно, пусть $l \geqslant 2$ — натуральное число, тогда, воспользовавшись субъектом B как «оракулом» расшифрования, нарушитель

может получить значения открытых текстов $m_1, \ldots, m_l \in \mathbb{Z}_m$ для некоторых произвольных, корректно расшифрованных субъектом B шифртекстов $\xi_{2,1}, \ldots, \xi_{2,l}$. После этого нарушитель может составить систему уравнений

$$\begin{cases} \mu_i \equiv \dfrac{y_{W_i} - y_S}{x_{W_i} - x_S} \pmod p, \\ \beta_i \equiv \dfrac{y_S x_{W_i} - x_S y_{W_i}}{x_{W_i} - x_S} \pmod p, \\ \xi_{2,i} \equiv (2(\mu_i + y_{W_i}) + 1)m_i + \beta_i \pmod{2^w}, \quad i = 1, \ldots, l. \end{cases} \tag{4.5}$$

Данная система состоит из $3l$ уравнений и зависит от $2l + 2$ неизвестных $\mu_1, \ldots, \mu_l, \beta_1, \ldots, \beta_l, x_S, y_S$ (значения $x_{W_1}, \ldots, x_{W_l}, y_{W_1}, \ldots, y_{W_l}$ известны нарушителю). В силу построения решение системы (4.5) существует, следовательно, нарушитель может найти это решение и определить неизвестные ему значения x_S, y_S.

3. Понятие разового ключа.

   Величина $\zeta \in \mathbb{Z}_q^*$, вырабатываемая на первом шаге алгоритма зашифрования, является *разовым* ключом, поскольку её раскрытие приводит к эффективному дешифрованию.

   Действительно, пусть (ξ_1, ξ_2, ξ_3) — шифртекст, выработанный под сообщением $m \in \mathbb{Z}_{2^w}$ с использованием величины ζ. Тогда нарушитель, зная величину ζ и открытый ключ Q субъекта B, может вычислить точку $W = [\zeta]Q$. Далее, используя (4.3) и зная долговременный ключ S, нарушитель может определить параметры $\mu, \beta \in \mathbb{F}_p^*$ и расшифровать передаваемое сообщение с использованием сравнения (4.4).

   Добавим, что если нарушитель имеет доступ к $l \geqslant 2$ различным разовым ключам $\zeta_1, \ldots, \zeta_l$, он, как и ранее, может использовать субъекта B в качестве «оракула» расшифрования и определить точное значение долговременного ключа S, решая систему сравнений (4.5).

4. Атака при шифровании на одинаковых разовых ключах.

   Рассмотрим ситуацию, при которой нарушитель перехватывает $l \geqslant 2$ шифртекстов $\xi_{2,1}, \ldots, \xi_{2,l}$, выработанных с использованием одного и того же неизвестного нарушителю разового ключа $\zeta \in \mathbb{F}_q^*$. Для этого нарушителю достаточно отобрать из множества всех перехваченных шифртекстов те, у которых совпадают координаты точки U.

   В этом случае, если нарушитель может использовать субъекта B как «оракула» для корректного расшифрования двух открытых тек-

стов, скажем m_1 и m_2, то он может при $l \geqslant 3$ эффективно дешифровать оставшиеся сообщения $m_3, \ldots, m_l$.

Действительно, при одинаковых значениях $\zeta \in \mathbb{Z}_q^*$ параметры $\mu, \beta \in \mathbb{F}_p$, определяемые равенствами (4.3), точка W и значение $\alpha \equiv 2(\mu + y_W) + 1 \pmod{2^w}$, одинаковы для всех перехваченных сообщений $\xi_{2,1}, \ldots, \xi_{2,l}$. Тогда α, β могут быть эффективно вычислены нарушителем. Для этого, ему необходимо решить систему сравнений

$$\begin{cases} \xi_{2,1} \equiv \alpha m_1 + \beta \pmod{2^w}, \\ \xi_{2,2} \equiv \alpha m_2 + \beta \pmod{2^w}, \end{cases} \tag{4.6}$$

относительно неизвестных значений α и β, после чего воспользоваться равенствами (4.4) и дешифровать открытые тексты $m_3, \ldots, m_l$.

5. Атака при шифровании на одинаковых разовых ключах алгебраически связанных текстов.

   В случае, когда нарушитель имеет только два сообщения, зашифрованных при помощи одних и тех же значений параметров α и β, описанная выше атака не реализуема. Вместе с тем предположим, что нарушителю дополнительно известны обратимые по модулю 2^w величины $a, b, c \in \mathbb{Z}_{2^w}$ такие, что неизвестные нарушителю открытые тексты m_1, m_2 удовлетворяют сравнению

$$am_1 - bm_2 \equiv c \pmod{2^m}.$$

   Тогда, из системы сравнений (4.6) следует, что

$$a\xi_{2,1} - b\xi_{2,2} \equiv \alpha(am_1 - bm_2) \equiv \alpha c \pmod{2^m}$$

   и величина α может быть определена при помощи сравнения

$$\alpha \equiv c^{-1}(a\xi_{2,1} - b\xi_{2,2}) \pmod{2^w}.$$

   Далее, выражая $m_2 \equiv (am_1 - c)b^{-1} \pmod{2^w}$, можно записать систему сравнений

$$\begin{cases} \xi_{2,1} \equiv \alpha m_1 + \beta \pmod{2^w}, \\ \xi_{2,2} \equiv \alpha(am_1 - c)b^{-1} + \beta \pmod{2^w}, \end{cases}$$

   относительно неизвестных m_1, β. Решение этой системы позволит последовательно найти открытый текст m_1, величину β и, как следствие, определить открытый текст m_2.

Изложенная атака, очевидно, является модификацией известной атаки Франклина-Ройтера на схему RSA, см. [43, 145]. Из ее существования сразу следует, что шифрование двух различных сообщений m_1, m_2 при помощи одной прямой $y = \alpha x + \beta$ приводит к компрометации схемы.

6. Атака на основе адаптивно подобранных шифртекстов.

   Известно [151], что классическая асимметричная схема Эль-Гамаля является уязвимой относительно атаки с адаптивно подобранным шифртекстом. Рассмотрим возможность применения данной атаки к рассматриваемой схеме ECISPE.

   Будем считать, что нарушитель хочет дешифровать шифртекст (ξ_1, ξ_2, ξ_3) и определить исходное сообщение m. Для этого нарушитель пользуется субъектом B как «оракулом» расшифрования и направляет ему на расшифрование два шифртекста специального вида

$$(\xi_1, \xi_{2,1} = 0, \xi_3) \quad \text{и} \quad (\xi_1, \xi_{2,2} = 1, \xi_3).$$

   Поскольку все три шифртекста имеют одно и тоже значение заголовка ξ_1, то легко заметить, что они выработаны с одним и тем же значением разового ключа ζ. Используя описанные выше рассуждения и соответствующие подобранным шифртекстам открытые тексты m_1, m_2, нарушитель может определить значения величин α и β из равенств (4.6), а именно:

$$\alpha \equiv \frac{1}{m_2 - m_1} \pmod m, \quad \beta \equiv \frac{m_1}{m_1 - m_2} \pmod m. \tag{4.7}$$

   Теперь искомая величина m определяется из сравнения

$$m \equiv (\xi_2 - \beta)\alpha^{-1} \equiv \xi_2(m_2 - m_1) + m_1 \pmod{2^w}.$$

   Отметим, что для проведения атаки значения $\xi_{2,1}$ и $\xi_{2,2}$ могут принимать произвольные отличные друг от друга значения. Значения $\xi_{2,1} = 0$ и $\xi_{2,2} = 1$ выбраны для минимизации формул (4.7).

   Однако возможность осуществления приведённой атаки возникает только в том случае, когда субъект B при расшифровании шифртекстов не проверяет код целостности сообщения. Действительно, направляемая в подобранных нарушителем шифртекстах имитовставка ξ_3 соответствует сообщению m и для корректного расшифрования она должна изменяться при замене величины ξ_2 на $\xi_{2,1}$ или $\xi_{2,2}$. Таким образом, атака на основе адаптивно подобранных шифртекстов может быть осуществлена нарушителем в одном из двух случаев:

a) Для двух произвольных, отличных друг от друга и заранее заданных значений $\xi_{2,1}, \xi_{2,2}$ нарушитель может вычислить значения имитовставок $\xi_{3,1}, \xi_{3,2}$ такие, что

$$\xi_{3,i} = \mathtt{mac}(ik, \xi_1 || m_i), \quad m_i \equiv (\xi_{2,i} - \beta)\alpha^{-1} \pmod{2^w}, \qquad (4.8)$$

для $i = 1, 2$ и неизвестного нарушителю значения производного ключа имитозащиты ik, определяемого равенством (4.2) (величины m_1, m_2 и, как следствие, величины α, β становятся известными нарушителю описанным выше способом).

b) Для двух произвольных, заранее заданных имитовставок $\xi_{3,1}, \xi_{3,2}$ нарушитель может определить два значения $\xi_{2,1}$ и $\xi_{2,2}$, такие, что выполнены сравнения (4.8) при неизвестных значениях α, β и неизвестном значении производного ключа имитозащиты ik.

Если для функции $\mathtt{mac}$ перечисленные предположения не выполняются, то схема ECISPE может считаться стойкой относительно атаки с адаптивно подобранными шифртекстами.

7. Исследование геометрических особенностей.

Легко заметить, что схема ECISPE обладает одной геометрической особенностью, которая может быть использована нарушителем для попытки компрометации схемы.

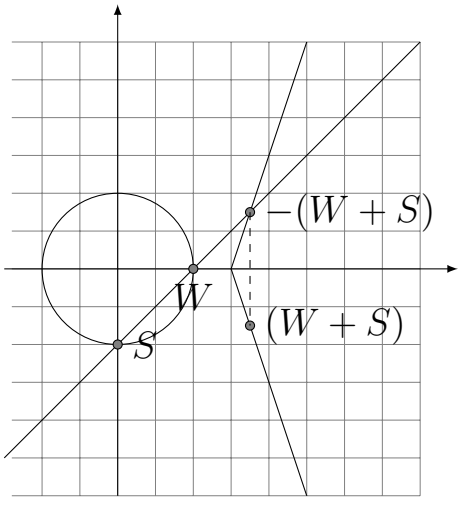


Рис. 4.6: Точки S, W и $-(W+S)$ на эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$.

Из закона сложения точек на эллиптической кривой следует, что секущая линия, соединяющая точки S и W, также содержит и третью точку W_1, принадлежащую эллиптической кривой, см. рис. 4.6,

и определяемую равенством

$$W_1 = -(W + S).$$

Поскольку точки $W, S \in \langle P \rangle$, то найдется значение ζ_0 такое, что $W_1 = [\zeta_0 k_c]P$. Таким образом, значения ζ и $\zeta_0 \equiv -(\zeta + k_{AB}k_c^{-1}) \pmod q$ приводят к вычислению одних и тех же значений μ и β.

Поскольку величина α зависит от y-координаты точки W, то точка $W_1 \neq W$ дает другое значение величины α и, как следствие, другую прямую линию, используемую для зашифрования сообщения. Случай $W_1 = W$ приводит к равенству $\zeta = \zeta_0$ и не является опасным.

8. Исследование возможности навязывания.

   Исследуем возможность реализации последней цели нарушителя – навязывания субъекту B ложного сообщения.

   Рассматриваемая схема не позволяет нарушителю зашифровать произвольное сообщение и отправить его субъекту B. Действительно, выполняя процедуру зашифрования нарушитель должен вычислить производный ключ имитозащиты ik, определяемый равенством (4.2) и зависящий от неизвестного нарушителю значения k_{AB}. Знание нарушителем точки S недостаточно, поскольку определение величины k_{AB} требует решения задачи дискретного логарифмирования $S = [k_{AB}]P$.

   Для определения неизвестного ключа имитозащиты ik нарушителю достаточно решить уравнение

$$\xi_3 = \mathtt{mac}(ik, \xi_1 || m)$$

   при известных значениях заголовка ξ_1, сообщения m и имитовставки ξ_3. Для криптографической ключевой функции хеширования задача определения ключа имитозащиты является трудноразрешимой, см. определение 3.2 на стр. 201.

   Расшифровать отправленное субъектом A сообщение (ξ_1, ξ_2, ξ_3) может только владелец ключа проверки кода аутентификации k_c — субъект B, а выполнить проверку имитовставки под расшифрованным сообщением может только владелец общей ключевой информации k_{AB}, которая однозначно связана с идентификатором ID_A отправителя сообщения. Следовательно, после проверки имитовставки субъект B получает подтверждение, что сообщение было действительно отправлено субъектом A, т.е. аутентифицирует полученное сообщение. Детальное рассмотрение свойства аутентификации сообщений проводится далее в разделе § 4.4.1.

Связь между сложными математическими задачами и целями нарушителя, изображена на рисунке 4.7. Символами DLP и DHP обозначены, соответственно, задача дискретного логарифмирования и задача Диффи-Хеллмана в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$.

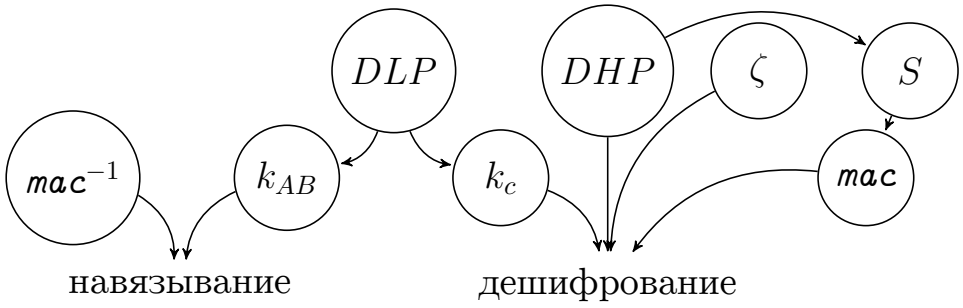


Рис. 4.7: Связь между математическими задачами и целями нарушителя.

Суммируем изложенные выше результаты в виде следующей теоремы.

Теорема 4.1. *Схема асимметричного шифрования ECISPE может считаться стойкой относительно задач определения секретного ключа проверки кода аутентификации, дешифрования и навязывания сообщений в случае, когда выполнены следующие условия:*

1. *для нарушителя являются трудоёмкими задачи дискретного логарифмирования и Диффи — Хеллмана, рассматриваемые в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$;*

2. *общая ключевая информация k_{AB} не должна быть известна нарушителю;*

3. *каждое сообщение должно шифроваться с помощью уникального разового ключа $\zeta \in \mathbb{F}_q^*$;*

4. *ключевая функция хеширования $\mathtt{mac}()$ должна удовлетворять определению 3.2 и обеспечивать невозможность решения нарушителем задач, поставленных в перечислениях 6 и 8;*

5. *расшифрованные сообщения, содержащие неверное значение имитовставки, не должны становиться известными нарушителю.*

§ 4.2.3. Модификации схемы ECISPE

Базовая схема асимметричного шифрования ECISPE допускает несколько различных модификаций, позволяющих изменить её функциональные особенности без изменения стойкости. Обозначим символом

$r \in \mathbb{N}$ – двоичную длину вырабатываемой функцией $\mathtt{mac}()$ имитовставки и рассмотрим следующий вариант схемы шифрования.

Алгоритм 4.2: Схема ECISPE с сокрытием имитовставки. Процедура зашифрования.

Вход : Сообщение $m \in \mathbb{V}_{w-r}$, ключ аутентификации $Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ получателя сообщения и общий ключ $k_{AB} \in \mathbb{F}_q^*$.

Выход : Зашифрованное сообщение $(\xi_1, \xi_2$.

1 Субъект вырабатывает случайное значение $\zeta \in_R \mathbb{F}_q^*$.

2 Субъект вычисляет точки $U = [\zeta]P$, $W = [\zeta]Q$ и $S = [k_{AB}]P$. Если выполнено равенство $U = \pm S$, то субъект возвращается на 1-й шаг алгоритма. Иначе, субъект формирует заголовок $\xi_1 = ID_A || \mathtt{bin}(U)$.

3 Субъект вырабатывает производный ключ имитозащиты

$$ik = \mathtt{kfd}_1(\mathtt{bin}(W), k_{AB})$$

и вычисляет имитовставку $\xi_3 = \mathtt{mac}(ik, \xi_1 || m)$.

4 Субъект представляет точки $S = (x_S, y_S)$ и $W = (x_W, y_W)$ в аффинной форме и определяет величины

$$\begin{cases} \mu \equiv \dfrac{y_W - y_S}{x_W - x_S} \pmod p, \\ \beta \equiv \dfrac{y_S x_W - x_S y_W}{x_W - x_S} \pmod p, \end{cases}$$

5 Субъект представляет сообщение m и имитовставку ξ_3 в виде целых натуральных чисел, после чего формирует величину $m_1 = m2^r + \xi_3$ и зашифровывает её при помощи сравнения

$$\xi_2 \equiv \mathtt{f}(m_1) \pmod{2^w},$$

где

$$\mathtt{f}(x) = \alpha x + \beta \in \mathbb{Z}_{2^w}[x], \quad \text{и} \quad \alpha = 2(\mu + y_W) + 1.$$

6 Пара значений (ξ_1, ξ_2) отправляется в канал связи.

Легко видеть, то предложенный вариант схемы ECISPE отличается от базового только тем, что значение имитовставки добавляется к исходному сообщению и зашифровывается вместе с ним. Это вызвано тем, что имитовставка может рассматриваться как некоторая информация о передаваемом сообщении m, и может быть использована нарушителем для проведения атаки на функцию выработки имитовставки $\mathtt{mac}()$ с целью определения сообщения m или ключа имитозащиты ik. Предложенная модификация позволяет говорить о неприменимости перечисленных атак на функцию $\mathtt{mac}()$. В ряде зарубежных работ такой подход принято называть «Mac-then-Encryption», см., например, сноску на стр. 218; он применяется в рекомендациях Р 1323565.1.017-2018, см. [356], или в схеме RSA, см. RFC 8017 [189].

Для возможности применения схемы ECISPE к данным произвольной длины приведем еще одну модификацию, которая минимизирует объем данных, передаваемых по каналу связи. Для реализации этой схемы потребуется еще одно криптографическое преобразование

$$\boldsymbol{g}(x) : \mathbb{V}_r \to \mathbb{V}_{w-r},$$

представляющее собой генератор псевдослучайной последовательности фиксированной длины. В качестве такого генератора может выступать преобразование, определяемое в Р 1323565.1.006-2017, см. [354], или алгоритм, предложенный ранее в разделе 2.6.2.

Обозначим $r_0 = \lceil \log_2(w) \rceil$ и рассмотрим сообщение $m \in \mathbb{V}_\infty$ сообщение, двоичная длина которого удовлетворяет неравенству $\log_2(m) \leqslant w - r - r_0$. Определим зависящую от начального значения $\xi_0 \in \mathbb{V}_\infty$ и ключа имитозащиты $ik \in \mathbb{K}$ процедуру маскирования сообщения m. В начале, определим двоичный вектор[3]

$$m_1 = m||0^{w-r-r_0-\log_2(m)}||\operatorname{len}_2(m),$$

для которого выполнено условие $\operatorname{len}_2(m_1) = w - r$. После этого, вычислим

$$\xi_3 = \boldsymbol{mac}(iv, \xi_0||m_1)$$

и определим результат процедуры маскирования

$$\boldsymbol{mask}(ik, \xi_0, m) = m_1 \oplus \boldsymbol{g}(\xi_3)||\xi_3, \tag{4.9}$$

Теперь схема асимметричного шифрования может быть записана с помощью следующего алгоритма.

[3]Напомним, что запись 0^x означает двоичную последовательность длины x, состоящую из одних нулей.

Алгоритм 4.3: Схема ECISPE для сообщений произвольной длины. Процедура зашифрования.

Вход : Сообщение $m \in \mathbb{V}_\infty$, ключ аутентификации $Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ получателя сообщения и общий ключ $k_{AB} \in \mathbb{F}_q^*$.

Выход : Зашифрованное сообщение $(\xi_1, \xi_{2,1}, \ldots, \xi_{2,n})$.

1 Субъект вырабатывает случайное значение $\zeta \in_R \mathbb{F}_q^*$.

2 Субъект вычисляет точки $U = [\zeta]P$ и $S = [k_{AB}]P$. Если выполнено равенство $U = \pm S$, то субъект возвращается на 1-й шаг алгоритма. Иначе, субъект формирует заголовок $\xi_1 = ID_A || \mathbf{bin}(U)$.

3 Субъект представляет зашифровываемое сообщение в виде $m = m_1 || \cdots || m_n$, где $\log_2(m_i) \leqslant w - r - r_0$ и определяет $k_1 = \zeta$.

4 **Для всех** $i = 1, \ldots, n$ **выполнять**

5 Субъект вычисляет точку $W_i = [k_i]Q$, после чего вырабатывает ключ имитозащиты ik и новое, секретное значение k_{i+1}, используя равенство

$$\{ik_i, k^*\} = \mathbf{kdf}_2(\mathbf{bin}(W_i), k_{AB}), \quad k_{i+1} \equiv k^* \pmod q .$$

6 Используя равенство (4.9) субъект вычисляет значение

$$m_2 = \mathbf{mask}(ik_i, \mathbf{bin}(W_i), m_i).$$

7 Субъект представляет точки $S = (x_S, y_S)$ и $W_i = (x_W, y_W)$ в аффинной форме и определяет величины

$$\begin{cases} \mu_i \equiv \dfrac{y_{W_i} - y_S}{x_{W_i} - x_S} \pmod p , \\ \beta_i \equiv \dfrac{y_S x_{W_i} - x_S y_{W_i}}{x_{W_i} - x_S} \pmod p , \end{cases}$$

8 Субъект зашифровывает сообщение m_2 при помощи сравнения

$$\xi_{2,i} \equiv \mathbf{f}_i(m_2) \pmod{2^w} ,$$

где $\mathbf{f}_i(x) = \alpha_i x + \beta_i \in \mathbb{Z}_{2^w}[x]$ и $\alpha = 2(\mu_i + y_{W_i}) + 1$.

9 **конец**

10 Последовательность значений $(\xi_1, \xi_{2,1}, \ldots, \xi_{2,n})$ отправляется в канал связи.

Изложенная модификация схемы ECISPE обладает тремя принципиальными различиями по сравнению с n-кратным применением предыдущей модификации схемы.

Первое различие очевидно — вместо n различных точек $U_1, \ldots, U_n$, передаваемых в заголовках сообщений, передается только одна точка (первая), а остальные точки вычисляются в процессе зашифрования/расшифрования сообщения. Фактически, в процессе зашифрования вырабатывается только один разовый ключ $k_1 = \zeta$, а остальные разовые ключи $k_2, \ldots, k_n$ образуют последовательность, удовлетворяющую рекуррентному соотношению

$$\{\cdot, k_{i+1}\} = \boldsymbol{kdf}_2([k_i]Q, k_{AB}), \quad i = 1, \ldots, n-1,$$

зависящему от ключа аутентификации субъекта B и общего для обоих субъектов ключа k_{AB}.

Второе различие заключается в способе вычисления имитовставки ξ_3. Если в предыдущей модификации схемы значение имитоставки вычислялось от передаваемых данных (заголовка ξ_1 и содержащейся в нем точки U), то в текущей модификации значение имитоставки вычисляется от данных (точек W_i), которые в канал связи не передаются и, следовательно, не могут быть перехвачены нарушителем. Равенство $W_1 = [k_c]U$ позволяет сделать вывод о том, что корректность проверки имитовставки под точкой W_1 влечет за собой корректность значения точки U.

Третье различие состоит в наличии преобразования маскирования исходного сообщения. Определенное равенством (4.9) преобразование скрывает от нарушителя внутреннюю структуру сообщения m_1 и не позволяет ему делать предположения о значениях некоторых его битов. Более того, поскольку вырабатываемое генератором псевдослучайной последовательности значение $\boldsymbol{g}(\xi_3)$ зависит от общего секретного ключа k_{AB} и различно для каждого фрагмента m_i, то процедура маскирования может рассматриваться как процедура предварительного поточного шифрования сообщения m_i.

В заключение раздела стоит высказать несколько критических замечаний к схеме шифрования, описываемой алгоритмом 4.3. Во-первых, схема должна реализовываться посредством канала связи, обеспечивающим очередность получения отправленных пакетов, например, посредством применения протокола TCP/IP. В противном случае, отправленные сообщения $\xi_{2,1}, \ldots, \xi_{2,n}$ могут поступить на расшифрование не в том порядке, в котором они были зашифрованы, что приведет к неверному определению точек W_i и, как следствие, ошибке при проверке имитовставки.

Во-вторых, данная модификация оптимизирована для возможности реализации эффективной процедуры расшифрования сообщения, что приводит к передаче к канал связи избыточного количества имитовставок. Оптимальной была бы отправка в канал связи всего одной имитовставки, вычисляемой от заголовка ξ_1 и всего сообщения m. Однако в этом случае, при расшифровании, пришлось бы реализовывать двухпроходную процедуру – в ходе первичной обработки поступающих из канала связи данных расшифровывать сообщение, а в ходе вторичной – вычислять имитовставку. При возникновении случайного или преднамеренного искажения зашифрованных данных такой подход не позволяет заметить ошибку до полного окончания процедуры расшифрования.

§ 4.2.4. Использование схемы ECISPE для передачи ключевой информации

Все асимметричные схемы, стойкость которых основывается на задаче дискретного логарифмирования, ведут начало от исходной схемы Эль-Гамаля [76]. В 2006 году в докладе [269] автором диссертационной работы была предложена модификация схемы Эль-Гамаля, использующая для зашифрования передаваемой информации один из режимов работы блочного шифра ГОСТ 28147-89, см. [279]. В этом же году схожая схема, названная ECIES-KEM[4], была включена в международный стандарт ISO18033-2:2006 [115, раздел 10.2].

§ 4.2.4.1. Схема ECIES с применением аутентифицированного шифрования

Приведем описание схемы из работы [269], модифицированное таким образом, чтобы удовлетворить действующим в настоящее время рекомендациям в области криптографической защиты информации.

Зафиксируем открытые параметры схемы:

- эллиптическую кривую $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и точку на этой кривой $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ такую, что $\operatorname{ord} P = q$, где q – простое число,

а также следующие криптографические преобразования:

- функцию $\boldsymbol{bin}$, предназначенную для преобразования точки эллиптической кривой в двоичную последовательность фиксированной длины, см. [361, раздел 5.1.2];
- определенную ранее равенством (4.1) функцию $\boldsymbol{kdf}_n$, предназначенную для выработки производной ключевой информации,
- режим аутентифицированного шифрования $\boldsymbol{authenc}$, см. определение 3.9 на стр. 207; в качестве такого режима могут быть использованы режимы «MGM» [364] или «XTSMAC», см. § 3.3.1.

В ключевую систему рассматриваемой асимметричной схемы входят:

- ключи субъекта B — ключ проверки кода аутентификации $k_c \in \mathbb{F}_q^*$ и ключ аутентификации, представленный в виде точки $Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$, определяемой равенством

$$Q = [k_c]P.$$

[4] англ. Elliptic Curve Integrated Encryption Scheme — Key Encapsulation Mechanism.

- опционально, общий ключ $k_{AB} \in \mathbb{V}_\infty$, связанный с идентификаторами ID_A, ID_B субъектов взаимодействия (отметим, что использование ключа k_{AB} не является обязательным и используется для аутентификации отправителя сообщения).

Алгоритм 4.4: Схема ECIES. Процедура зашифрования.

Вход : Сообщение $m \in \mathbb{V}_\infty$, ключ аутентификации $Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ получателя сообщения и, опционально, общий ключ $k_{AB} \in \mathbb{V}_\infty$.

Выход : Зашифрованное сообщение (ξ_1, ξ_2, ξ_3).

1 Субъект вырабатывает случайное значение $\zeta \in_R \mathbb{F}_q^*$.

2 Субъект вычисляет точки эллиптической кривой $U = [\zeta]P$, $W = [\zeta]Q$ и формирует заголовок $\xi_1 = ID_A || \mathbf{bin}(U)$.

3 Субъект вырабатывает производные ключи $ek, ik \in \mathbb{V}_\infty$ и, при необходимости, синхропосылку $iv \in \mathbb{V}_\infty$:

$$\{ek, ik, iv\} = \mathbf{kdf}_3(\mathbf{bin}(W), \mu || k_{AB}), \tag{4.10}$$

для некоторой константы $\mu \in \mathbb{V}_\infty$.

4 Используя режим аутентифицированного шифрования субъект зашифровывает сообщение m и вычисляет его код целостности

$$\{\xi_2, \xi_3\} = \mathbf{authenc}(ek, ik, iv, \xi_1, m)$$

5 Тройка (ξ_1, ξ_2, ξ_3) отправляется в канал связи.

Отметим, что использование идентификатора ID_A в заголовке зашифрованного сообщения (ξ_1, ξ_2, ξ_3) позволяет субъекту B идентифицировать отправителя не используя для этого данные протокола канального уровня. Более того, при использовании ключевой схемы Блома, см. работу [42] и рекомендации [358, 365], субъект B может вырабатывать общий ключ k_{AB} только после получения зашифрованного сообщения.

Для расшифрования полученного сообщения (ξ_1, ξ_2, ξ_3), субъект B должен проверить, что точка U, содержащаяся в заголовке ξ_1, принадлежит подгруппе, порожденной точкой P. После чего, используя ключ проверки кода аутентификации k_c, он должен вычислить точку

$$W = [k_c]U = [k_c\zeta]P = [\zeta]Q,$$

и, с помощью равенства (4.10), определить производные ключи ek, ik и, при необходимости, синхропосылку iv. В заключение, субъект B должен расшифровать полученное сообщение

$$\{m, b\} = \mathbf{authdec}(ek, ik, iv, \xi_1, \xi_2, \xi_3)$$

и проверить, что код целостности ξ_3 верен, что равносильно выполнению равенства $b = \mathbf{true}$.

Описываемая алгоритмом 4.4 базовая схема ECIES обладает следующими свойствами:

- целостностью и конфиденциальностью передаваемых сообщений, обеспечиваемых посредством применения режима аутентифицированного шифрования, а также сложностью решения задачи дискретного логарифмирования в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$,
- возможностью аутентификации субъектов взаимодействия, обеспечиваемой посредством использования общего секретного ключа k_{AB}
- свойством стойкости при компрометации производных ключей,

а также рядом других свойств безопасности, детальное рассмотрение которых проводится далее в § 4.4.1.

§ 4.2.4.2. Протокол передачи ключевой информации

Обе рассмотренные выше схемы ECISPE и ECIES используют для шифрования и контроля целостности передаваемой информации производные ключи, вырабатываемые в ходе выполнения процедуры зашифрования данных.

Однако, в некоторых практических ситуациях возникает необходимость шифровать информацию на ключах выработанных заранее, после чего передавать данные ключи вместе с зашифрованной информацией. Также, необходимость передачи ключевой информации от одного субъекта к другому возникает в автоматизированных системах с централизованным изготовлением ключевой информации.

Оба перечисленных случая могут быть практически реализованы при помощи еще одной схемы, включающей в себя преобразования, реализуемые схемами ECISPE и ECIES. В начале приведем описание процедуры инкапсуляции ключевой информации в передаваемое сообщение.

В дальнейшем для этой процедуры будем использовать обозначение:

$$\{\xi_1, \xi_2, \xi_3, \xi_4\} = \texttt{ecispe}(Q, k_{AB}, k_t, m, \mu),$$

где $\xi_1, \ldots \xi_4$ составные элементы шифртекста, определяемые процедурой зашифрования, $Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ — ключ аутентификации субъекта B, K_{AB} – общий для участников протокола ключ, k_t — передаваемая ключевая информация, $\mu \in \mathbb{V}_\infty$ – произвольные ассоциированные данные, которые могут быть аутентифицированы субъектом A, см. определение 3.9.

Алгоритм 4.5: Схема ECISPE с инкапсуляцией ключевой информации. Процедура зашифрования.

Вход : Сообщение $m \in \mathbb{V}_\infty$ (опционально), ключ аутентификации $Q \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ получателя сообщения, общий ключ $k_{AB} \in \mathbb{F}_q^*$, передаваемая получателю ключевая информация $k_t \in V_{w-r-r_0}$ и произвольный вектор $\mu \in V_\infty$

Выход : Зашифрованное сообщение $(\xi_1, \xi_2, \xi_3, \xi_4)$.

1 Субъект вырабатывает случайное значение $\zeta \in_R \mathbb{F}_q^*$.

2 Субъект вычисляет точки $U = [\zeta]P$ и $S = [k_{AB}]P$. Если выполнено равенство $U = \pm S$, то субъект возвращается на 1-й шаг алгоритма. Иначе, субъект формирует заголовок $\xi_1 = ID_A || \boldsymbol{bin}(U)$.

3 Субъект вычисляет точку $W = [\zeta]Q$, после чего вырабатывает ключ имитозащиты ik, используя равенство

$$ik = \boldsymbol{kdf}_1(\boldsymbol{bin}(W), k_{AB}).$$

4 Используя равенство (4.9) субъект вычисляет значение

$$m_2 = \boldsymbol{mask}(ik, \boldsymbol{bin}(W), k_t),$$

помещая в сообщение m_2 передаваемую ключевую информацию k_t.

5 Субъект представляет точки $S = (x_S, y_S)$ и $W = (x_W, y_W)$ в аффинной форме и определяет величины

$$\begin{cases} \mu \equiv \dfrac{y_W - y_S}{x_W - x_S} \pmod p, \\ \beta \equiv \dfrac{y_S x_W - x_S y_W}{x_W - x_S} \pmod p, \end{cases}$$

6 Субъект зашифровывает сообщение m_2 при помощи сравнения

$$\xi_2 \equiv \boldsymbol{f}(m_2) \pmod{2^w},$$

где $\boldsymbol{f}(x) = \alpha x + \beta \in \mathbb{Z}_{2^w}[x]$ и $\alpha = 2(\mu + y_W) + 1$.

7 Если сообщение m не определено, то в канал связи отправляется пара (ξ_1, ξ_2) и алгоритм завершает вою работу.

8 Субъект вырабатывает производные ключи $ek, ik \in \mathbb{V}_\infty$ и, при необходимости, синхропосылку $iv \in \mathbb{V}_\infty$

$$\{ek, ik, iv\} = \boldsymbol{kdf}_3(k_t, \delta),$$

для некоторой величины $\delta \in \mathbb{V}_\infty$, значение которой, в зависимости от области применения схемы, может вычисляться с помощью значений k_{AB}, $\boldsymbol{bin}(W)$ или принимать константное значение.

9 Используя режим аутентифицированного шифрования субъект зашифровывает сообщение m и вычисляет его код целостности

$$\{\xi_3, \xi_4\} = \boldsymbol{authenc}(ek, ik, iv, \mu, m).$$

10 Четверка $(\xi_1, \xi_2, \xi_3, \xi 4)$ отправляется в канал связи.

Легко видеть, что модификация схемы ECISPE, описываемая алгоритмом 4.5, последовательно добавляет в шифртекст сначала ключевую информацию k_t, а потом и сообщение m. Теперь можно описать протокол передачи ключевой информации, развивающий идеи работ [174, 328].

Будем считать, что субъект B выполняет роль инициатора протокола и запрашивает ключевую информацию у ее владельца — субъекта A. Ключевая система рассматриваемого протокола, а также его открытые параметры совпадают с ключевой системой и параметрами последней модификации схемы ECISPE.

Протокол представляет собой взаимодействие типа «запрос-ответ» и изображен на рисунке 4.8.

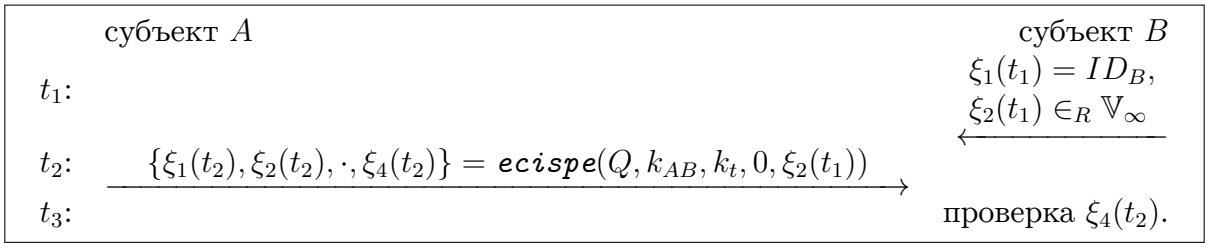


Рис. 4.8: Протокол передачи ключа с использованием схемы ECISPE.

На приведенном рисунке символы t_1, t_2, t_3 обозначают метки времени, в которые выполняются шаги протокола. В момент времени t_1 субъект B вырабатывает случайную двоичную последовательность $\xi_2(t_1)$ и отправляет ее субъекту A вместе со своим идентификатором. Эта последовательность может рассматриваться как уникальный идентификатор сессии протокола, позволяющий субъекту B убедиться в том, что полученная им ключевая информация k_t предназначена для этой сессии, а не была использована ранее.

В момент времени t_2 субъект A зашифровывает ключевую информацию k_t, а также вычисляет имитовставку $\xi_4(t_2)$ под полученным от субъекта B значением $\xi_2(t_1)$. При этом, шифрование какой-либо отличной от ключа k_t информации субъектом A не производится.

После получения величин $\xi_1(t_2), \xi_2(t_2)$ и $\xi_4(t_2)$ субъект B последовательно, расшифровывает величину $\xi_2(t_2)$ и получает ключевую информацию k_t. После чего, с ее помощью он сравнивает имитовставку под той случайной величиной $\xi_2(t_1)$, что была им отправлена в начале выполнения протокола, с полученным значением $\xi_4(t_2)$. В случае совпадения имитовставок, субъект B убеждается, что он получил верное значение.

Заключение к § 4.2

В § 4.2 предложена гибридная схема ECISPE и ряд ее модификаций, реализующих процесс шифрования с помощью полиномиального преобразования. Определена модель возможностей нарушителя и, в этой модели, доказана теорема о стойкости предложенной схемы шифрования относительно задач определения секретного ключа аутентификации, дешифрования и навязывания сообщений. Предложен протокол передачи ключевой информации, основанный на использовании рассматриваемой гибридной схемы шифрования.

§ 4.3. Протоколы выработки общего ключа с аутентификацией

В начале 2010-х годов появилась необходимость[5] в разработке и стандартизации отечественного протокола выработки общего ключа с аутентификацией участников взаимодействия. К разрабатываемому решению предъявлялись следующие требования, см. [319, 327].

- Должна обеспечиваться взаимная аутентификация сторон взаимодействия; в качестве секрета, подтверждаемого в ходе процесса аутентификации, должны выступать секретные ключи электронной подписи.

- Трудоемкость определения нарушителем общего ключа, вырабатываемого в ходе выполнения протокола, должна быть высокой и сравнимой с трудоемкостью компрометации алгоритма шифрования, для которого вырабатывается ключ.

- Трудоемкость определения секретных ключей электронной подписи быть настолько высокой, чтобы обеспечивать функционирование средства, использующего протокол, в течение достаточно большого интервала времени.

- Каждая сессия выполнения протокола должна вырабатывать уникальный общий ключ. Ситуация, при которой один или несколько общих ключей, выработанных в разных сеансах выполнения протокола, станут известны нарушителю, не должна приводить к компрометации ключей, вырабатываемых в других сеансах.

[5]Во многом это было связано с многочисленными уязвимостями повсеместно используемого в то время протокола TLS версии 1.1, а позднее и версии 1.2.

- Компрометация долговременных ключей аутентификации не должна приводить к компрометации общих ключей, выработанных ранее.
- Ни один абонент не должен иметь возможность навязать значение вырабатываемого общего ключа другому абоненту.
- После выработки общего ключа каждый из субъектов взаимодействия должен быть уверен в том, что второй субъект обладает тем же общим ключом.

Выполнение перечисленных требований должно было предотвратить возможность реализации большого числа известных атак на протоколы выработки общего ключа. Предполагалось, что в отличие от эксплуатирующейся в то время версии протокола TLS, новый протокол будет реализовываться только в группе точек эллиптической кривой и обеспечивать обязательную взаимную аутентификацию субъектов взаимодействия. Сам перечень формировался с учетом предложений, высказанных в отчете [9].

В качестве решений, удовлетворяющих сформулированным требованиям, предлагались протоколы, названные именами различных растений, см. [380]:

- протокол «Лимонник», предложенный Д.В. Матюхиным в 2011 году в докладе [319],
- протокол «Крокус», предложенный А.Ю. Нестеренко в 2012 году в докладе [331],
- семейство протоколов Эхинацея», предложенное С.В. Гребневым в 2014 году в докладе [293].

Позднее, в 2017 году, часть из перечисленных протоколов вошла в состав рекомендаций по стандартизации Р 1323565.1.004-2017 [352].

§ 4.3.1. Протокол выработки общего ключа «Крокус»

Целью выполнения протокола «Крокус» является взаимодействие двух субъектов, определяемых идентификаторами ID_A и ID_B, и однократное выполнение операции типа «запрос-ответ». Данная операция востребована, например, при реализации запросов в удаленные базы данных или при обеспечении защиты Интернет-протоколов HTTP [90] или Gemini [203].

В ходе выполнения протокола выполняется процесс выработки общего ключа, используемого в дальнейшем для генерации производных ключей

шифрования данных, передаваемых в качестве запроса и ответа на запрос. Также в ходе выполнения протокола может быть произведено согласование алгоритмов шифрования и выработки производного ключа. Изложение следует работе [327], см. также [334].

§ 4.3.1.1. Описание протокола

Зафиксируем открытые параметры протокола:

- эллиптическую кривую $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и точку на этой кривой $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ такую, что $\operatorname{ord} P = q$, где q – простое число,

а также следующие криптографические преобразования:

- функцию $\boldsymbol{bin}$, предназначенную для преобразования натуральных чисел и точек эллиптических кривых в двоичные последовательности;
- определенную ранее равенством (4.1) функцию $\boldsymbol{kdf}_n$, предназначенную для выработки производной ключевой информации,
- функцию выработки электронной подписи $\boldsymbol{mac}$ и функцию проверки электронной подписи $\boldsymbol{conf}$ такие, что

$$k_c = [k_a]P \quad \text{и} \quad \boldsymbol{conf}(k_c, \xi, \boldsymbol{mac}(k_a, \xi)) = \boldsymbol{true},$$

  для любой пары значений $k_a \in \mathbb{F}_q^*$ и $\xi \in \mathbb{V}_\infty$; считаем, что процедуры выработки и проверки электронной подписи определяются ГОСТ Р 34.10-2012[280],
- функции зашифрования $\boldsymbol{enc}$ и расшифрования $\boldsymbol{dec}$ сообщений, удовлетворяющие равенству

$$\boldsymbol{dec}\big(ek, \boldsymbol{enc}(ek, \xi)\big) = \xi$$

  для любого ключа $ek \in \mathbb{K}$ и сообщения $\xi \in \mathbb{V}_\infty$.

В ключевую систему рассматриваемой асимметричной схемы входят:

- ключи субъекта A — ключ электронной подписи $k_{aA} \in \mathbb{F}_q^*$ и ключ проверки электронной подписи

$$Q_A = [k_{aA}]P \in \mathcal{E}_{a,b}(\mathbb{F}_p),$$

- ключи субъекта B — ключ электронной подписи $k_{aB} \in \mathbb{F}_q^*$ и ключ проверки электронной подписи

$$Q_B = [k_{aB}]P \in \mathcal{E}_{a,b}(\mathbb{F}_p),$$

- сертификаты открытых ключей $\texttt{cert}(Q_A), \texttt{cert}(Q_B) \in \mathbb{V}_\infty$, заверенные подписями доверенного (удостоверяющего) центра и однозначно связанные с идентификаторами субъектов взаимодействия.

Будем интерпретировать согласуемые в ходе выполнения алгоритмы шифрования и выработки производного ключа как двоичные вектора из $\mathbb{V}_\infty$ и считать, что субъект A направляет субъекту B множество векторов $\{S_1, \ldots, S_r\}$, а субъект возвращает один вектор S_i из полученного множества, $i \in \{1, \ldots, r\}$.

Схема протокола «Крокус» приведена ниже на рисунке 4.9. Символы $t_1, \ldots t_{12}$ обозначают метки времени, в которые выполняются шаги протокола, а символы `Question` и `Answer` — отправляемый субъектом A запрос и, соответственно, возвращаемый субъектом B ответ.

Протокол «Крокус» представляет собой модификацию схемы Диффи-Хеллмана выработки общего ключа, реализуемую в группе точек эллиптической кривой, и состоит из четырех последовательно выполняемых фаз:

- фазы взаимной аутентификации,
- фазы выработки общего ключа,
- фазы подтверждения выработанного общего ключа $W \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ и
- фазы обмена зашифрованными сообщениями типа «запрос-ответ».

В ходе первой фазы субъект A, инициализирующий выполнение протокола, вырабатывает и направляет субъекту B сообщение $\xi_2(t_1)$, которое содержит перечень поддерживаемых криптографических алгоритмов, сертификат открытого ключа $\texttt{cert}(Q_A)$, а также случайное значение $\xi_1(t_1)$. Цель данного шага – запрос на аутентификацию субъекта B.

В процессе выполнения шагов протокола, помеченных метками t_2 и t_3, субъект B верифицирует полученный сертификат открытого ключа $\texttt{cert}(Q_A)$, после чего проверяет электронную подпись $\xi_4(t_1)$ под полученными данными и убеждается в том, что они действительно были сформированы субъектом, владеющим секретным ключом k_{aA}. Использование идентификатора ID_B при выработке и последующей проверке электронной подписи $\xi_4(t_1)$, позволяет субъекту B убедиться в том, что данные предназначены именно для него. Вместе с тем, субъект B пока еще не может быть уверен в том, что данные были посланы субъектом A, а не были перехвачены в ходе предыдущих сеансов и навязаны нарушителем.

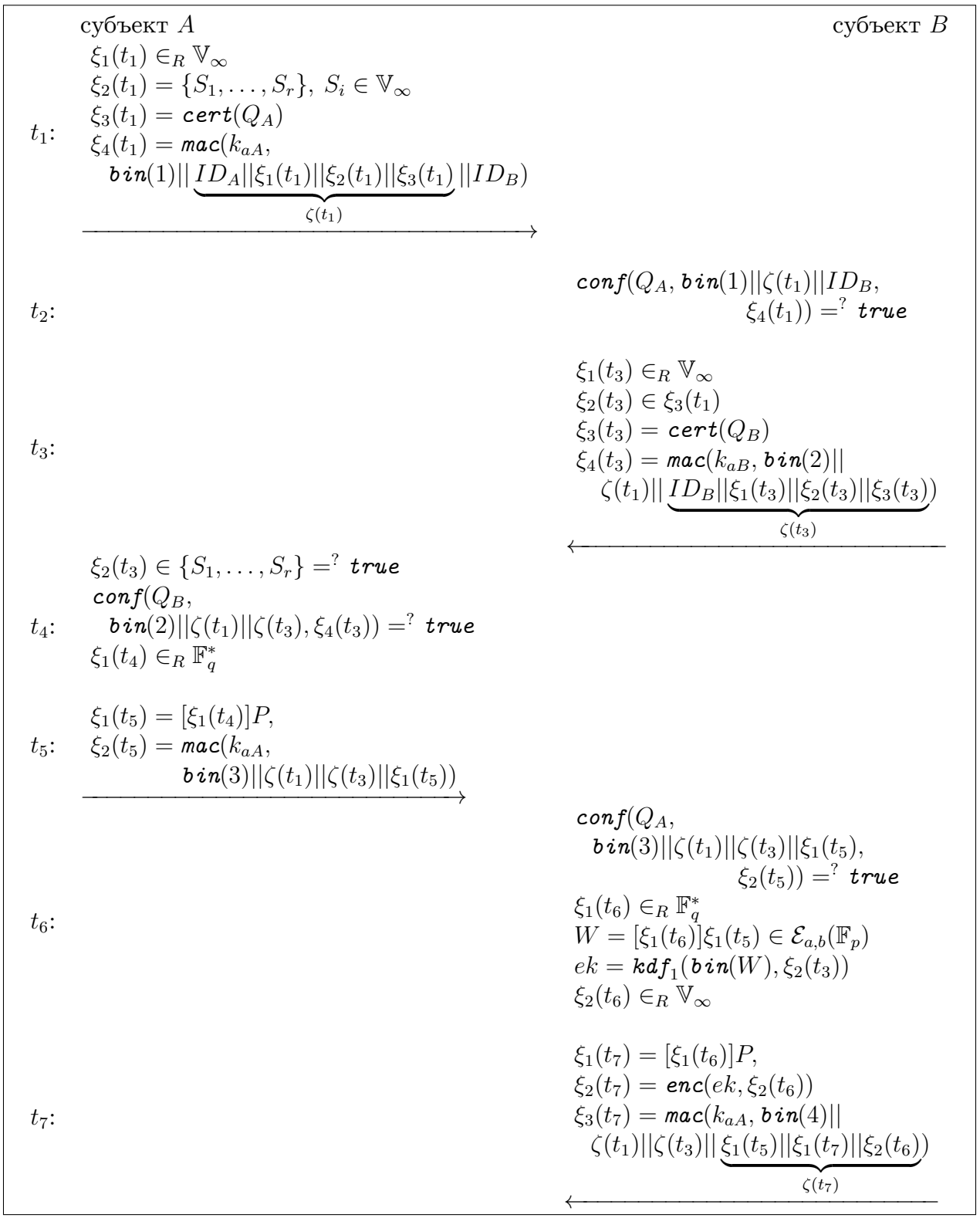


Рис. 4.9: Протокол выработки общего ключа «Крокус». Начало.

Далее, субъект B определяет множество согласуемых криптографических алгоритмов и направляет его субъекту A, а также сертификат своего открытого ключа $\mathit{cert}(Q_B)$, и случайное значение $\xi_1(t_3)$.

Теперь комбинация $ID_A||\xi_1(t_1)||ID_B||\xi_1(t_3)$ служит идентификатором

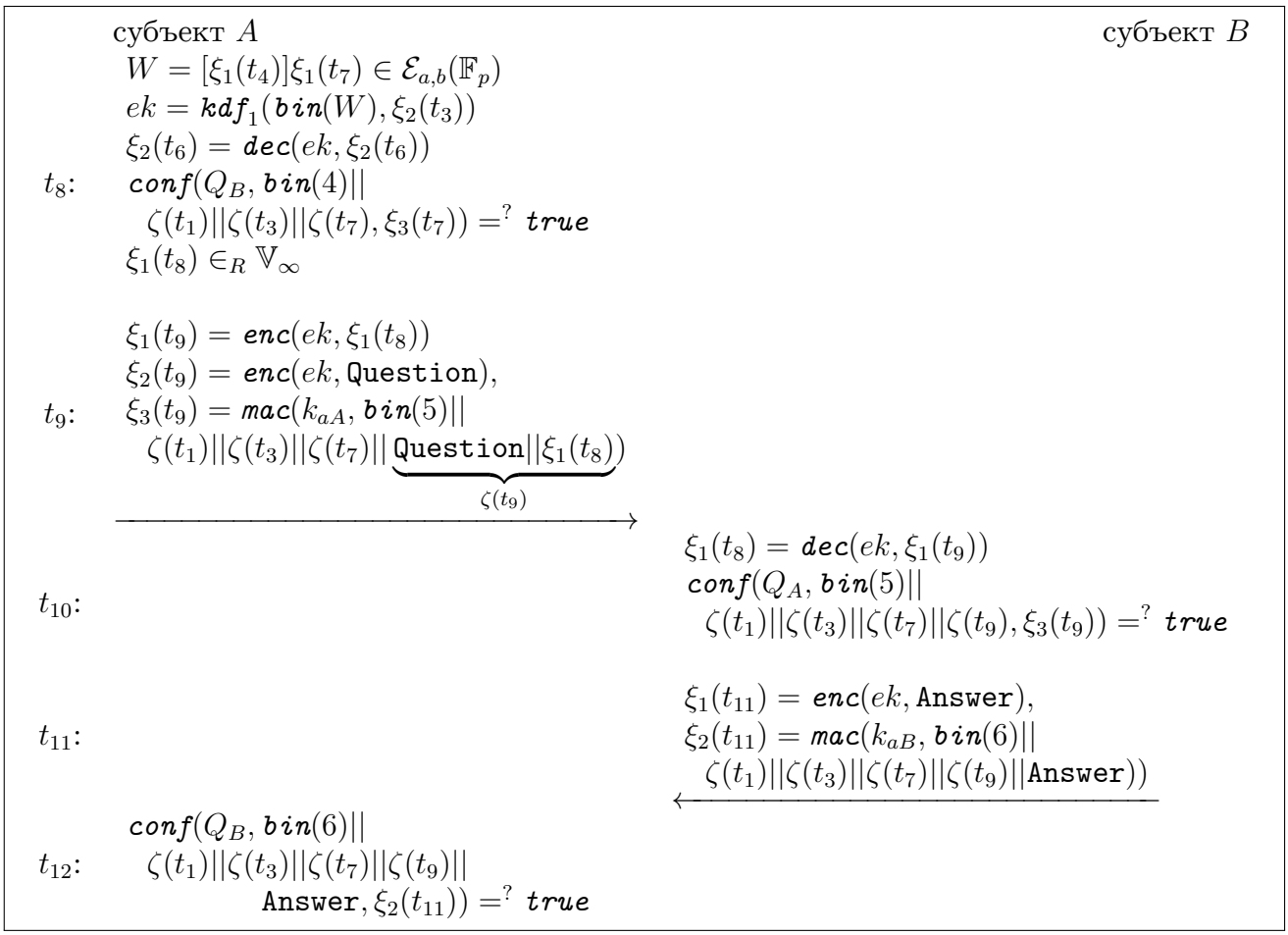


Рис. 4.10: Протокол выработки общего ключа «Крокус». Продолжение.

сессии, указывающим на то, кто из субъектов инициировал выполнение протокола, а кто на него ответил; уникальность идентификатора зависит от свойств используемых генераторов случайных чисел.

Для субъекта A первая фаза протокола завершается на шаге, помеченным меткой t_4. В этот момент субъект проверяет электронную подпись $\xi_4(t_3)$ и убеждается, что подпись выработана субъектом, владеющим секретным ключом k_{aB}. Поскольку подпись содержит данные $\xi_1(t_1)$, выработанные субъектом A в ходе текущей сессии протокола, то он успешно аутентифицирует субъекта B, а также убеждается в том, что криптографический механизм $\xi_2(t_3)$ выбран корректно.

Далее, субъект A приступает к реализации схемы Диффи-Хеллмана и отправляет субъекту B точку эллиптической кривой $\xi_1(t_5)$. На шаге, помеченным меткой t_6, субъект завершает процесс аутентификации субъекта A, после чего переходит к выработке общего ключа, в качестве которого выступает точка $W \in \mathbb{E}_{a,b}(\mathbb{F}_p)$.

В ходе выполнения шага, помеченного меткой t_7, субъект B переходит к фазе подтверждения выработанного общего ключа. Он отправляет

субъекту A точку эллиптической кривой $\xi_1(t_7)$, необходимую для выработки общего ключа, после чего отправляет случайный двоичный вектор $\xi_2(t_6)$, зашифрованный на производном ключе ek. Одновременно, при вычислении электронной подписи под передаваемыми данными, используется незашифрованное значение вектора $\xi_2(t_6)$.

Далее, субъект A, получивший перечисленные сообщения, также вычисляет общий ключ W и завершает фазу выработки общего ключа. Он расшифровывает полученное значение $\xi_2(t_7)$ и проверяет электронную подпись под полученными данными. Если подпись верна, то субъект A убеждается в том, что субъект B выработал корректный общий ключ W, после чего переходит к отправке запроса `Question`. Одновременно, так же как и субъект B, субъект A вырабатывает случайный двоичный вектор $\xi_1(t_8)$ и отправляет его в канал связи вместе с запросом в зашифрованном виде.

В заключение протокола, субъект B расшифровывает полученное от субъекта A сообщение и проверяет электронную подпись под полученными данными. Если подпись верна, то субъект B убеждается в том, что субъект A выработал тот же общий ключ W, после чего направляет в канал связи зашифрованный ответ `Answer`.

Протокол, схема которого приведена на рисунках 4.9 и 4.10, фактически представляет собой объединение протоколов выработки общего ключа и транспортного протокола, посредством которого выполняется взаимодействие по типу «запрос-ответ». По-видимому, именно присутствие излишнего функционала послужило причиной того, что протокол «Крокус» не был включен в состав рекомендаций по стандартизации Р 1323565.1.004-2017 [352].

Вместе с тем, рассматриваемый протокол иллюстрирует подход к синтезу криптографических протоколов, в рамках которого обеспечивается выполнение сразу нескольких свойств безопасности. Этот подход основан на включении в состав протокола заранее определенных фрагментов, позволяющих протоколу обеспечить заданное требование или, другими словами, обладать заданным свойством безопасности.

Так, для проведения аутентификации субъекта взаимодействия в состав протокола включена последовательность обменов сообщениями, изображенная на рисунке 4.18, для подтверждения общего ключа — включена последовательность обменов, изображенная на рисунке 4.24. Кроме того, значение вырабатываемого общего ключа не зависит от ключей аутентификации, что делает компрометацию одних ключей не опасной для других. Указанные свойства безопасности, а также ряд других свойств, подробно рассматриваются позднее в разделе 4.4.1.

§ 4.3.1.2. Исследование безопасности протокола

При исследовании безопасности протокола «Крокус» будем использовать введенную в начале главы модель нарушителя и считать, что перед нарушителем стоят следующие задачи:

- определение общего ключа W, вырабатываемого в ходе выполнения протокола,
- дешифрование передаваемых данных `Question` и `Answer`,
- навязывание ложных данных субъектам взаимодействия,
- реализация ситуации, при которой один из субъектов взаимодействия вводится в заблуждение относительно того, с кем именно осуществляется взаимодействие.

Начнем с того, что перечислим данные, которые может перехватывать нарушитель в ходе выполнения одной сессии протокола. Именно эти данные используются нарушителем при проведении пассивных атак на протокол.

- случайные двоичные векторы $\xi_1(t_1)$ и $\xi_1(t_3)$,
- множество фиксированных двоичных векторов $\xi_2(t_1)$ и вектор $\xi_2(t_3) \in \xi_3(t_1)$,
- сертификаты открытых ключей $\xi_1(t_3)$ и $\xi_3(t_3)$ субъектов взаимодействия и, как следствие, идентификаторы субъектов взаимодействия ID_A и ID_B,
- случайные точки $\xi_1(t_5)$ и $\xi_1(t_7)$ на эллиптической кривой,
- шифртексты $\xi_2(t_7)$, $\xi_1(t_9)$, $\xi_2(t_9)$ и $\xi_1(t_{11})$ соответствующие неизвестным нарушителю открытым данным.

Упрощая нарушителю задачу будем считать, то все шифртексты, передаваемые в рамках одной сессии, выработаны на одном ключе. Также будем предполагать, что для некоторых сессий взаимодействия нарушителю известны значения `Question` и `Answer`.

1. Сведение к задаче дискретного логарифмирования.

   Легко видеть, что сложность определения k_{aA}, k_{aB} — секретных ключей подписи субъектов взаимодействия, а также вырабатываемых в ходе выполнения протокола значений $\xi_1(t_4)$, $\xi_1(t_6)$ основывается на трудоёмкости решения задачи дискретного логарифмирования в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$.

Действительно, указанные значения могут быть найдены путем решения следующих уравнений

$$Q_A = [k_{aA}]P, \quad Q_B = [k_{aB}]P, \quad \xi_1(t_5) = [\xi_1(t_4)]P, \quad \xi_1(t_7) = [\xi_1(t_6)]P.$$

Умение нарушителем решать задачу дискретного логарифмирования приводит к возможности определения общего ключа W и дешифрованию передаваемой в ходе выполнения протокола информации, а также к определению секретных ключей подписи и возможности выдать себя за другого легитимного участника взаимодействия.

Как было сказано ранее, см. § 1.1.2, решение задачи дискретного логарифмирования в группе точек эллиптической кривой, определённой над конечным простым полем $\mathbb{F}_p$, является сложной задачей, трудоёмкость которой оценивается величиной $O(\sqrt{q})$.

2. Сведение к задаче Диффи — Хеллмана.

   Умение нарушителя решать задачу Диффи — Хеллмана, т.е. умение по перехваченным значениям $\xi_1(t_5) = [\xi_1(t_4)]P$ и $\xi_1(t_7) = [\xi_1(t_6)]P$ найти значение $W = [\xi_1(t_4)\xi_1(t_6)]P$, приводит к определению общего ключа и, как следствие, к дешифрованию информации, передаваемой в ходе выполнения протокола. Как уже говорилось ранее, в настоящее время задача Диффи — Хеллмана в группе точек эллиптической кривой считается трудноразрешимой, а наиболее эффективный способ её решения заключается в сведении к задаче дискретного логарифмирования.

3. Разовые ключи и атаки на датчик случайных чисел.

   Легко видеть, что значения $\xi_1(t_4)$ и $\xi_1(t_6)$ должны рассматриваться как разовые ключи, поскольку их раскрытие сразу приводит к определению величины W и, как следствие, к дешифрованию информации, передаваемой в ходе выполнения протокола.

   Еще один способ скомпрометировать протокол заключается в попытке предсказать результат действия датчика случайных чисел одного или нескольких субъектов взаимодействия. В ходе одной сессии протокола нарушитель имеет возможность наблюдать вырабатываемую датчиком последовательность фиксированной длины, а именно, значения $\xi_1(t_1)$ и $\xi_1(t_3)$. Учитывая, что нарушитель может перехватывать материал достаточно большого числа сессий, возможно, выполняемых одновременно, то оценка качества датчика является одной из первоочередных задач исследования конкретной реализации протокола.

Полагаем, что для генераторов случайных чисел должны выполняться требования, перечисленные в разделе 2.1, см. стр. 138. Более того, используемые датчики не должны изменять своих свойств при выполнении нескольких параллельных сессий выполнения протокола, а вырабатываемые в параллельных сессиях последовательности не должны коррелировать. Кроме того, согласно Р 1323565.1.012-2017 [355], реализация протокола в средстве защиты информации должна содержать встроенные средства динамического контроля, обоснования статистических свойств и уникальности вырабатываемых последовательностей.

Также, целесообразно использовать один датчик для выработки случайных октетов $\xi_1(t_1)$ или $\xi_1(t_3)$ и другой, независимый от первого датчик для выработки разовых ключей $\xi_1(t_4)$ или $\xi_1(t_6)$ и случайных значений, используемых в процессе выработки электронной подписи.

4. Подделка подписи

   Умение нарушителя подделывать электронную подпись приводит к умению навязывать значения величин $\xi_1(t_5)$ и $\xi_1(t_7)$ и, как следствие, реализовывать классическую атаку на схему Диффи-Хеллмана, называемую «человек по-середине».

   Поскольку изначально предполагается, что для выработки электронной подписи используются процедуры, регламентированные в ГОСТ Р 34.10-2012 [280], то задача её подделки является для нарушителя трудноразрешимой.

5. Атака на производные ключи и функцию вычисления производного ключа

   Ранее мы предположили, что для некоторых сессий выполнения протокола нарушителю известны не только открытые сообщения `Question` и `Answer`, но соответствующие им зашифрованные сообщения $\xi_2(t_9)$ и $\xi_1(t_{11})$. Если суммарная двоичная длина сообщений `Question` и `Answer` превышает значения, установленные в рекомендациях [353], то найдутся методы криптографического анализа, см. [139], которые позволят нарушителю определить выработанный в рамках атакуемой сессии взаимодействия производный ключ $ek = \mathit{kdf}_1(\mathit{bin}(W), \xi_2(t_3))$, где функция kdf_1 определена равенством (4.1).

   Теперь, поскольку величина ${}_2(t_3)$ известна нарушителю, вычисление величины W эквивалентно обращению функции kdf_1, т.е. решению

задачи вычисления прообраза для функции хеширования «Стрибог» [281], см. также определение 3.1 на стр. 200. Однако, если нарушителю все-таки удастся определить значение общего ключа W из атакуемого сеанса связи, то это не даст ему ни какой информации, об общих ключах, выработанных в ходе других сессий выполнения протокола (в случае, если выполнены описанные ранее требования к генерации случайных значений $\xi_1(t_4)$ и $\xi_1(t_6)$).

При анализе активных атак на криптографические протоколы наиболее правильным подходом, по мнению автора, является систематическое исследование практических приемов компрометации протоколов и разработка перечня принципов и положений, позволяющих предотвратить известные атаки. Основываясь на различных публикациях, в частности, [48, 151, 391], приведем простейшую классификацию активных атак.

- Модификация сообщений (modification attack) — самая очевидная и простая атака, основанная на подмене передаваемой в сообщении информации.

- Подмена субъекта (impersonation) — попытка подменить одного субъекта взаимодействия другим. В таких атаках нарушитель имитирует действия одного или нескольких субъектов, передавая и/или получая сообщения в ответ. В частности, нарушитель может выступать от лица одного или нескольких легитимных участников взаимодействия.

- Повторное навязывание сообщений (reply attack) — класс атак, основанных на повторном использовании переданных ранее корректных сообщений или каких-либо частей корректных сообщений. Целью атак данного класса является, как правило, попытка нарушения аутентификации одного из абонентов. Добавим, что иногда в зарубежной литературе используется другое англоязычное название атаки, а именно, freshness attack, которое может переводиться как атака на основе отсутствия новизны передаваемых данных.

- Одной из модификаций атаки с навязыванием сообщением, является атака, основанная на возврате сообщений (reflection attack) или атака отражением. Как следует из названия, нарушитель пытается вернуть абоненту отправленное им же ранее сообщение, которое может быть воспринято как истинное сообщение.

- Задержка передачи сообщений (forced delay attack) — атака, при которой перехваченное сообщение навязывается противником в более

поздний момент времени, на более позднем шаге протокола или в ходе следующих сессий взаимодействия.

- Атака с ошибочной интерпретацией сообщений (wrong interpretation attack) — класс атак, основанный на ошибочной интерпретации участниками протокола полученных сообщений или их составных частей.

- Атака с параллельными сеансами (parallel-session attack) — атака, основанная на одновременном открытии нескольких сеансов выполнения протокола с одним или несколькими абонентами, с целью использования сообщений из одной сессии в ходе выполнения другой.

- Атака на основе специально подобранных сообщений (adopted messages attack) — атака, в ходе которой нарушитель последовательно перебирает несколько специально подобранных сообщений, пытаясь по ответам одного или нескольких участников протокола, в общем случае — в разных сессиях, получить информацию, которая позволит скомпрометировать протокол.

- Временные атаки (timings attack) — частный случай атаки на основе специально подобранных сообщений, позволяющий нарушителю по времени ответа абонента на запрос получить информацию, позволяющую скомпрометировать протокол.

- Атака с известными ключами (known-key attack) — атака, при которой нарушителем используются ключи или общая ключевая информация, выработанная в ходе предыдущих или последующих, если анализ идет по перехваченной информации, сессий взаимодействия.

- Атака с известными разовыми ключами — атака, в ходе которой нарушителю становится известна секретная информация с ограниченным сроком жизни, вырабатываемая в ходе выполнения протокола. В качестве данной информации могут выступать случайные значения на шагах протокола, помеченных метками t_4, t_6 или случайные значения, вычисляемые в процессе выработки электронной цифровой подписи. Как правило, разовые ключи должны уничтожаться сразу после своего использования, однако используя активное воздействие на средство защиты или содержащееся в нем программное обеспечение, вырабатывающее разовый ключ, нарушитель может получить данную информацию. Соответствующие методы защиты должны реализовываться в обязательном порядке, выбор конкретных мер защиты должен определяться при проектировании средства защиты информации.

- Атака на параметры безопасности, в ходе которой нарушитель пытается подменить общедоступные параметры протокола, например, параметры группы точек эллиптической кривой, с целью вынудить абонента вычислить информацию, которая может быть использована для компрометации протокола. Традиционно, в этот класс атак входят атаки с навязыванием подгруппы малого порядка, подменой сертификата открытого ключа субъекта взаимодействия и т. п.

- Атака, нарушающая свойство связности передаваемого сообщения (bindings attack) — атака, использующая отсутствие проверки существования математической зависимости между различными частями сообщения или различными сообщениями, передаваемыми в ходе одной сессии выполнения протокола.

- Отказ в доступе (denial of service) — класс атак, приводящий не к компрометации протокола, а к невозможности дальнейшего функционирования средства защиты информации. На практике, наиболее простая и часто используемая атака, сводится к открытию большого числа одновременных сеансов выполнения протокола, приводящих к временному прекращению выполнения протокола.

Перечисленный список содержит лишь наиболее известные способы компрометации криптографических протоколов. Очевидно, что нарушитель может комбинировать перечисленные выше атаки, совмещая различные виды воздействий на легитимных абонентов. При этом легитимные абоненты не имеют возможности влиять на действия нарушителя.

При синтезе протокола применялись меры, обеспечивающие защиту от перечисленных типов атак нарушителя.

1. Каждое сообщение содержит в своем составе электронную подпись под передаваемыми данными. На рисунках 4.9 и 4.10 подписи обозначены символами $\xi_4(t_1)$, $\xi_4(t_3)$, $\xi_2(t_5)$, $\xi_3(t_7)$, $\xi_3(t_9)$ и $\xi_2(t_{11})$. Это позволяет не только гарантировать целостность передаваемых данных, но и аутентифицировать отправителя сообщения.

2. Все данные, которыми обмениваются субъекты взаимодействия, передаются в канал связи один раз.

3. Электронная подпись зависит не только от данных, передаваемых в канал связи, но и от дополнительных данных.

4. В ходе каждой сессии протокола вырабатывается уникальный идентификатор сессии, в качестве которого, как было сказано ранее, выступает величина $ID_A||\xi_1(t_1)||ID_B||\xi_1(t_3)$.

5. С момента своего формирования, идентификатор включается в состав данных, под которыми вычисляется электронная подпись. Это позволяет гарантировать, что передаваемые данные принадлежат сессии с указанным идентификатором.

6. К данным, для которых вычисляется электронная подпись, добавляется порядковый номер передаваемого сообщения. Это позволяет субъекту, принимающему сообщения, убеждаться в том, что последовательность принимаемых данных соответствует ожидаемой.

7. К данным, для которых вычисляется электронная подпись, добавляются все сообщения, переданные ранее в ходе текущей сессии, т.е. данные, обозначенные символами $\zeta(t_1)$, $\zeta(t_3)$, $\zeta(t_7)$ и $\zeta(t_9)$. Подобное свойство, хоть и представляется достаточно громоздким при практической реализации, позволяет гарантировать принадлежность всего множества обрабатываемых данных текущей сессии выполнения протокола.

8. На каждом шаге протокола пересылаются сообщения различной длины.

9. С точки зрения анализа алгоритмов шифрования представляется опасной ситуация, при которой один или оба абонента выполняют роль оракула, зашифровывающего случайные или специально подобранные сообщения, структура и/или содержание которых известны нарушителю — это позволяет накапливать информацию, которая может быть позднее использована для атаки на ключ шифрования, см. определение 3.8 на стр. 206.

   Протокол «Крокус» разработан таким образом, что нарушитель может перехватывать только информацию, зашифрованную при помощи неизвестного нарушителю секретного ключа. При этом открытый текст, который был зашифрован, не содержит ни каких-либо известных зависимостей, поскольку вырабатывается при помощи генератора случайных чисел.

Учитывая перечисленные меры, реализация большинства известных активных атак оказывается затруднительной. Аргументируем это утверждение.

Поскольку каждое передаваемое сообщение включает в себя электронную подпись, то для модификации передаваемых данных нарушитель должен уметь подделывать электронную подпись, что, в нашей модели нарушителя, является сложной задачей.

Подмена субъекта взаимодействия, например субъекта A, также представляется невозможной. Действительно, идентификация субъекта обеспечивается парой — идентификатор ID_A и сертификат открытого ключа $\mathtt{cert}(Q_A)$, в состав которого входят, в частности, открытый ключ Q_A и идентификатор ID_A субъекта A, заверенные электронной подписью доверенного центра.

Нарушитель имеет возможность на первом шаге протокола заменить отправленное сообщение на другое сообщение, перехваченное им в ходе выполнения другого сеанса связи и, в частности, содержащее сертификат открытого ключа другого легитимного субъекта взаимодействия. Однако, на третьем шаге, после получения ответа от субъекта B, нарушитель должен выработать электронную подпись под данными, которые выработаны субъектом B в ходе выполнения атакуемой сессии протокола. Поскольку создание электронной подписи нарушителем невозможно без знания секретного ключа, то нарушителю приходится угадывать значение подписи и, с вероятностью, близкой к единице[6], на следующем шаге, после неудачной проверки подписи, выполнение протокола завершается субъектом B. Аналогичные рассуждения позволяют говорить о невозможности подмены нарушителем субъекта B.

Поскольку модификация сообщений приводит к прекращению протокола, нарушителю остается выполнять действия связанные с навязыванием сообщений. Для предотвращения таких попыток нами введен уникальный идентификатор сессии. Данное значение известно нарушителю, однако он не может его менять, поскольку это приведет к необходимости изменения значения электронной подписи, что нарушителю недоступно. При практической реализации протокола «Крокус» размер множества возможных значений идентификаторов сессии должен быть настолько большим, чтобы нарушитель не мог позволить себе сохранить в памяти все возможные значения идентификаторов.

Для предотвращения попыток навязывания сообщений из текущей сессии в протоколе используются, во-первых, сообщения разной длины, а во-вторых, нумерация сообщений. Проверка указанных значений позволяет субъекту взаимодействия проверить, не изменена ли очередность поступления сообщений из канала связи. При этом, нарушитель не может изменять номер сообщения, поскольку он также используется при вычислении электронной подписи.

Последний класс рассматриваемых нами атак связан с изменением параметров эллиптической кривой $E_{a,b}(\mathbb{F}_p)$, в которой реализуется протокол «Крокус». Известны атаки, в которых изменение порядка группы q или

[6]Применительно к схеме ГОСТ Р 34.10-2012 [280] вероятность случайного угадывания составляет $\frac{2}{q-1}$, где q порядок подгруппы, в которой реализуется схема электронной подписи, см. так же рассуждение на стр. 342.

образующей точки P приводит к существенному снижению множества возможных значений общего ключа W и, как следствие, к возможности опробования всех возможных значений общего ключа за малое время. Для предотвращения подобного класса атак субъекты взаимодействия должны быть убеждены в том, что они вырабатывают ключ на одних и тех же параметрах эллиптической кривой, удовлетворяющих предъявляемым к ним требованиям, см. раздел 1.5.1.

К сожалению, реализовать процесс проверки корректности используемых параметров эллиптической кривой в реальном времени не предоставляется возможным, в силу высокой трудоемкости подобной проверки. Следовательно, используемые параметры должны быть вычислены заранее и храниться в используемом программном/аппаратном обеспечении. При выполнении протокола необходимо проводить проверку совпадения контрольной суммы параметров, используемых другим абонентом. Данная сумма должна входить в состав параметров безопасности S_i, согласуемых в ходе выполнения первых двух шагов шага протокола, см. рисунок 4.9.

Суммируем изложенное выше в виде следующей теоремы.

Теорема 4.2. *Протокол «Крокус» может считаться стойким относительно задач определения общего ключа, дешифрования и навязывания передаваемой в ходе выполнения протокола информации, а также возникновения ситуации, при которой один из субъектов взаимодействия вводится в заблуждение относительно того, с кем именно осуществляется взаимодействие, при выполнении следующих условий.*

1. *Для нарушителя являются трудоёмкими следующие задачи:*
   - *задача дискретного логарифмирования и*
   - *задача Диффи — Хеллмана, рассматриваемые в группе точек эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$,*
   - *задача подделки электронной подписи,*
   - *задача построения прообраза для функции выработки производного ключа $\mathtt{kdf}_n$.*
2. *Для генераторов случайных чисел должны выполняться требования, перечисленные в разделе 2.1, см. стр. 138.*
3. *Размер множества возможных значений идентификаторов сессии должен превышать объем памяти, доступный нарушителю.*

В завершение раздела сформулируем подходы к оптимизации протокола «Крокус», не снижающие его способность противостоять атакам нарушителя.

1. Допустимым является удаление возможности выбора криптографических механизмов в ходе выполнения протокола. Поскольку алгоритм электронной подписи — ГОСТ Р 34.10-2012 уже фиксирован (в противном случае невозможно было бы реализовать проверку электронной подписи), то и остальные параметры — используемый при шифровании блочный шифр, режим его использования и функцию хеширования, входящую в состав функции $\boldsymbol{kdf}_n$ — также можно зафиксировать заранее.

2. Необходимо минимизировать данные, для которых вычисляется электронная подпись. Такой способ был предложен автором еще в 2013 году в работе [334]. Суть его заключается в том, что необходимо подписывать не сами данные, которые были получены из канала связи, а значение функции хеширования, вычисленное при проверке электронной подписи. Таким образом, если вместо записи, см. шаг протокола, помеченный меткой t_2,

$$\boldsymbol{conf}(Q_A, \boldsymbol{bin}(1)||\zeta(t_1)||ID_B, \xi_4(t_1)) =^? \boldsymbol{true}$$

использовать запись,

$$h = \boldsymbol{hash}_{512}(\boldsymbol{bin}(1)||\zeta(t_1)||ID_B), \quad \boldsymbol{conf}(Q_A, h, \xi_4(t_1)) =^? \boldsymbol{true},$$

то величина $\xi_4(t_3)$ должна определяться равенством

$$\xi_3(t_4) = \boldsymbol{mac}(k_{aB}, \boldsymbol{bin}(2)||h||\zeta(t_3)).$$

На остальных шагах протокола, где вычисляется и проверяется электронная подпись, необходимо сделать аналогичные преобразования. Тем самым, мы снова получим зависимость последнего значения $\xi_2(t_{11})$ от всех данных, переданных в ходе выполнения сессии взаимодействия, однако способ определения этой зависимости будет напоминать способ контроля целостности данных, хранящихся в распределенном реестре, см. [296], и может быть более эффективно реализован на практике.

3. Перенести передачу данных с первого шага на третий, а со второго – на четвертый. Это не изменит объем передаваемых данных, однако позволит реализовать реализовать протокол в рамках четырех обменов сообщениями.

4. Можно удалить случайный двоичный вектор $\xi_1(t_9)$, передаваемый на предпоследнем шаге протокола, и использовать для подтверждения знания субъектом A общего ключа сообщение `Question`.

5. Другой оптимизацией является удаление передачи данных типа «запрос-ответ» в отдельный транспортный протокол. В этом случае, случайный двоичный вектор $\xi_1(t_9)$ необходимо оставить.

6. Для шифрования данных, передаваемых от субъекта A к субъекту B и обратно, целесообразно использовать различные производные ключи. Это позволит снизить объем информации, шифруемой на одном ключе и, как следствие, уменьшить вероятность реализации атак, направленных на определение производного ключа шифрования.

7. Процесс выработки и проверки электронной подписи может вызывать большую нагрузку на средство защиты информации в случае поддержки большого числа одновременных сессий взаимодействия. Для снижения такой нагрузки можно рассмотреть вопрос о расширении ключевой системы и переходе на симметричные ключи аутентификации.

8. В протоколе «Крокус» идентификаторы субъектов взаимодействия передаются в доступном для пассивного нарушителя виде. При модификации данного протокола можно использовать шифрование сообщений, передаваемых после выработки общего ключа. Такой подход позволяет скрыть от пассивного нарушителя идентификаторы субъектов взаимодействия, а от активного нарушителя — идентификатор субъекта, инициировавшего выполнении сессии взаимодействия.

Перечисленные модификации были учтены при разработке нового протокола, предназначенного для защищенного взаимодействия контрольных и измерительных устройств, см. [365].

§ 4.3.2. Криптографические механизмы защищенного взаимодействия контрольных и измерительных устройств

В 2019 году автором диссертационной работы, совместно с А.М. Семёновым и П.А. Лебедевым, была завершена разработка семейства криптографических протоколов, предназначенного для обеспечения защищенного взаимодействия между двумя абонентами по незащищенному каналу связи, а также для реализации каналов удаленного управления. Данное семейство протоколов получило название SP FIOT[7], см. [181, 342], и рекомендовано к использованию в Р 1323565.1.028-2019 [365].

[7] англ. Secure Protocols for «Internet Of Things».

В качестве субъектов, осуществляющих защищенное взаимодействие с помощью механизмов SP FIOT, могут выступать контрольные и измерительные устройства, объекты «Интернета вещей», в том числе технологические датчики и счетчики, камеры видеонаблюдения, удаленные сетевые маршрутизаторы, миниатюрные технические составляющие различных технологических процессов, а также произвольные субъекты автоматизированных систем, для которых необходим защищенный обмен информацией.

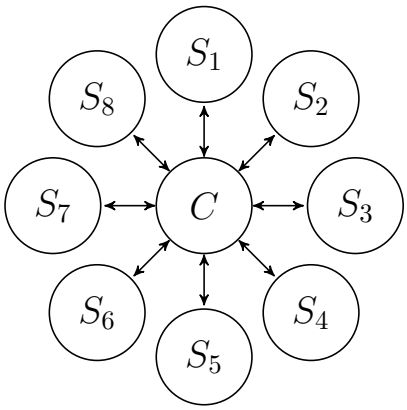


Рис. 4.11: Схема взаимодействия типа «звезда».

Семейство протоколов SP FIOT может применяться в системах типа «звезда», см. рисунок 4.11, для управления удаленными субъектами, и позволяет реализовывать на уровне приложений прикладные протоколы удаленного управления, такие как SNMP [60] или SSH [261]. Также, семейство протоколов SP FIOT может быть использовано для организации VPN-сетей и криптографических туннелей по схемам типа «точка-точка» или «звезда». Семейство включает в себя:

- протокол выработки ключей, предназначенный для выработки общей ключевой информации, уникальной для каждой сессии защищенного взаимодействия,

- протокол передачи прикладных данных, предназначенный для инкапсуляции поступающих данных, реализации контроля за процедурами выработки производной ключевой информации, состоянием сессии защищенного взаимодействия, а также процессом обработки ошибок,

- транспортный криптографический протокол, см. рисунок 4.4, предназначенный для обеспечения имитозащиты и, при необходимости,

шифрования передаваемых данных, а также защиты от атак по скрытым логическим каналам[8].

Схема стека протоколов, возникающего при использовании семейства протоколов SP FIOT, приведена на рисунке 4.12.

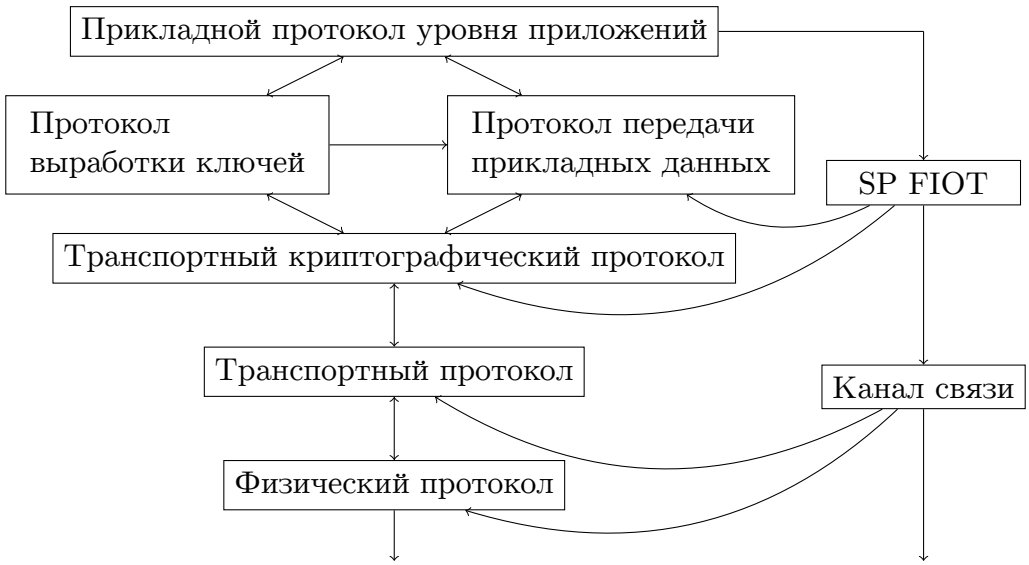


Рис. 4.12: Стек протоколов защищенного взаимодействия.

Протокол SP FIOT предназначен для обеспечения криптографической защиты информации, в то время как канал связи обеспечивает маршрутизацию и физическую доставку защищенной информации.

Отметим, что в качестве транспортного протокола канала связи могут выступать протоколы, расположенные на разных уровнях сетевой модели ISO [289], например, используемые в информационно-телекоммуникационной сети Интернет протоколы TCP [199] и UDP [200], IP [198], Ethernet [111], или протоколы Bluetoth [112] и WiFi [114], используемые для беспроводной передачи данных.

Столь большая вариативность обеспечивается за счет того, что транспортный криптографический протокол включает в себя обязательную нумерацию пакетов, отправляемых в рамках одной сессии взаимодействия, алгоритмические меры по защите от повторов и навязывания ложных пакетов, а также, при необходимости, меры по восстановлению очередности поступаемых пакетов.

[8]Соответствующее свойство безопасности вводится далее на стр. 312.

§ 4.3.2.1. Ключевая система

Для идентификации защищенного взаимодействия в рамках механизмов SP FIOT используется двухуровневая схема идентификаторов. Каждый субъект защищенного взаимодействия может обладать следующими идентификаторами:

- идентификатором первого уровня, в качестве которого выступают произвольные двоичные последовательности ID_A и ID_B, принадлежащие, соответственно, субъектам A и B, наличие у субъекта взаимодействия идентификатора первого уровня является обязательным;

  Интерпретация значения, содержащегося в идентификаторе первого уровня, должна зависеть от конкретной ситуации применения механизмов SP FIOT. Например, при реализации защищенного взаимодействия серии измерительных устройств, передающих информацию единому центру, идентификатор первого уровня может содержать информацию о производителе устройств, номере серии устройств, дате ввода серии устройств в эксплуатацию и т.п. С другой стороны, при реализации взаимодействия типа «точка-точка» идентификатор первого уровня может являться уникальным номером устройства;

- идентификатором второго уровня, в качестве которого может выступать произвольная двоичная последовательность, обозначаемая символами ID_{Ap} и ID_{Bp}, соответственно, для субъектов A и B; идентификатор второго уровня является опциональным, т.е. он может быть не определен, а, в случае определения, должен принимать уникальные значения в рамках одного субъекта взаимодействия.

  Интерпретация значения, содержащегося в идентификаторе второго уровня также должна зависеть от конкретной ситуации применения. В случае реализации защищенного взаимодействия серии измерительных устройств, передающих информацию единому центру, идентификатор второго уровня может содержать номер устройства, дату выпуска, максимальный срок эксплуатации и т.п. Данный идентификатор может иметь ограниченный временной интервал и изменяться в ходе эксплуатации устройства, например при замене содержащегося в устройстве блока криптографической защиты.

  С другой стороны, при реализации взаимодействия типа «точка-точка» идентификатор второго уровня может определять уникальный номер процесса или пользователя устройства.

Поскольку идентификаторы второго уровня являются опциональными, их применение целесообразно в рамках системы защищенного взаимодействия, поддерживающей единый формат и механизмы распознавания идентификаторов второго уровня; примером такой системы могут служит механизмы, регламентируемые в Р 1323565.1.019-2018 [358].

Столь детальное рассмотрение множества возможных идентификаторов субъектов взаимодействия вызвано существованием процедуры связывания идентификаторов с элементами клевой системы. Ключевая система семейства протоколов SP FIOT включает в себя следующие значения.

1. Ключи аутентификации субъектов взаимодействия. В качестве таких ключей могут выступать:

- предварительно распределенный, симметричный ключ $\mathtt{PSK}_{AB}$,
- пара асимметричных ключей – ключ электронной подписи и ключ проверки электронной подписи, а также сертификат ключа электронной подписи, заверенный электронной подписью удостоверяющего центра; форматы сертификатов открытого ключа должны соответствовать Р 1323565.1.023-2018 [361].

Способ формирования симметричных ключей аутентификации основан на хорошо известной схеме Р. Блома [42]. Пусть $\mathbb{F}_{2^n}$ конечное поле характеристики 2, порожденное неприводимым многочленом $p(x) \in \mathbb{F}_2[x]$ степени n. Для $n \in \{256, 512\}$, согласно [365], многочлен $p(x)$ определяется равенством

$$\begin{aligned} p(x) &= x^{256} + x^{10} + x^5 + x^2 + 1, \\ p(x) &= x^{512} + x^8 + x^5 + x^2 + 1. \end{aligned}$$

Далее, пусть $u \in \mathbb{N}$ и

$$f(x, y) = \sum_{i=0}^{u} \sum_{j=0}^{u} a_{i,j} x^i y^j \in \mathbb{F}_{2^n}[x, y]$$

многочлен, вырабатываемой центром выработки ключей с использованием СКЗИ, реализующим криптографическую функцию изготовления ключевых документов [355]. Коэффициенты многочлена $f(x, y)$ удовлетворяют равенству

$$a_{i,j} = a_{j,i}, \quad 0 \leqslant i < j \leqslant u,$$

являются мастер-ключом и не должны быть известны нарушителю и участникам взаимодействия.

Теперь, используя функцию хеширования $\boldsymbol{hash}_n$, регламентируемую ГОСТ Р 34.11-2012 [281], и функцию имитозащиты $\boldsymbol{hmac}_n$, регламентируемую Р 50.113-2016 [372], определим ключ аутентификации субъектов A и B равенством

$$\mathtt{PSK}_{AB} = \boldsymbol{hmac}_n(K_1, ID^*_{Ap}||\mathtt{0x00}||ID^*_{Bp}||\mathtt{0x00}||\mathtt{0x01}), \tag{4.11}$$

где символ * означает, что идентификатор второго уровня используется только в том случае, когда он определен, а промежуточный ключ $K_1 \in \mathbb{V}_n$ определяется равенством

$$K_1 = f\big(\boldsymbol{hash}_n(ID_A), \boldsymbol{hash}_n(ID_B)\big), \quad n \in \{256, 512\}$$

Можно заметить, что при наличии идентификаторов второго уровня ключи аутентификации $\mathtt{PSK}_{AB}$ и $\mathtt{PSK}_{BA}$ различны. Очередность использования в равенстве (4.11) идентификаторов ID_{Ap} и ID_{Bp} позволяет не только получить различные значения ключей аутентификации, но и связать направление передачи аутентифицируемой информации с ключевым значением.

С каждым ключом $\mathtt{PSK}_{AB}$ связывается свой идентификатор, обозначаемый символом ID_{PSK}. Данный идентификатор может определяться, например, равенством

$$ID_{\mathtt{PSK}_{AB}} = \boldsymbol{hash}_{256}(ID_A||ID^*_{Ap}||ID_B||ID^*_{Bp}),$$

в случае, когда необходимо скрыть информацию о субъектах взаимодействия, или равенством

$$ID_{\mathtt{PSK}_{AB}} = ID_A||ID^*_{Ap}||ID_B||ID^*_{Bp},$$

в случае, когда ключ $\mathtt{PSK}_{AB}$ определяется субъектами в ходе выполнения протокола выработки общих ключей.

2. Общая ключевая информация, которая вычисляется в ходе выполнения каждой сессии протокола выработки ключей следующим образом и представляет собой точку W, принадлежащую эллиптической кривой $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и удовлетворяющую равенству

$$W = [\xi_1\xi_2]P,$$

где величины $\xi_1, \xi_2 \in \mathbb{F}^*_q$ – случайные вычеты, вырабатываемые субъектами взаимодействия независимо друг от друга.

Также, в качестве общей ключевой информации выступает двоичный вектор $T \in \mathbb{V}_n$, определяемый равенством

$$T = \boldsymbol{hmac}_n(\boldsymbol{bin}(W), ID_A||ID^*_{Ap}||ID_B||ID^*_{Bp}||\mathtt{PSK}^*_{AB}), \tag{4.12}$$

где, как и ранее, символ * означает, что идентификатор второго уровня используется только в том случае, когда он определен.

3. Производные ключи шифрования и имитозащиты, вырабатываемые для каждой сессии протокола выработки общих ключей. Значения данных ключей зависят от двоичных последовательностей $H_1, H_3 \in \mathbb{V}_\infty$, образованных с помощью конкатенации данных, передаваемых в ходе выполнения протокола выработки общих ключей. Производные ключи формируются следующим образом.

- Для шифрования и имитозащиты данных, передаваемых от субъекта A к субъекту B в ходе выполнения протокола выработки общих ключей, используются ключи $\mathtt{eCHTK}, \mathtt{iCHTK} \in \mathbb{V}_{256}$, определяемые равенством

$$\mathtt{eCHTK}||\mathtt{iCHTK} = \mathit{hmac}_{512}\big(\mathit{hash}_{512}(\mathit{bin}(W)||\mathtt{PSK}^*_{AB}), \mathit{hash}_{512}(H_3)\big).$$

- Для шифрования и имитозащиты данных, передаваемых от субъекта B к субъекту A в ходе выполнения протокола выработки общих ключей, используются ключи $\mathtt{eSHTK}, \mathtt{iSHTK} \in \mathbb{V}_{256}$, определяемые равенством

$$\mathtt{eSHTK}||\mathtt{iSHTK} = \mathit{hmac}_{512}\big(\mathit{hash}_{512}(\mathit{bin}(W)||\mathtt{PSK}^*_{AB}), \mathit{hash}_{512}(H_1)\big).$$

Отметим, что производные ключи eCHTK, iCHTK, eSHTK, iSHTK используются только в рамках протокола выработки общих ключей и уничтожаются сразу после его завершения. Схема их выработки приведена на рисунке 4.13.

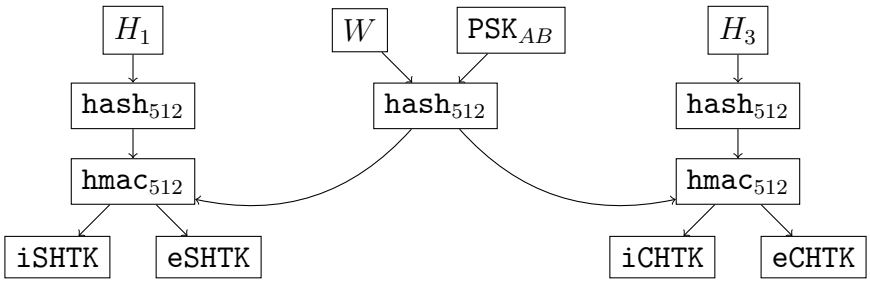


Рис. 4.13: Схема выработки ключей для шифрования и имитозащиты сообщений, передаваемых в ходе протокола выработки общих ключей.

4. Исходная ключевая информация для протокола передачи прикладных данных. Данная ключевая информация, обозначаемая символами

$\mathtt{CATS}_n, \mathtt{SATS}_n \in \mathbb{V}_{512}$, $n \in \mathbb{N}$, вырабатывается для возможности формирования производных ключей шифрования и имитозащиты информации, передаваемой с помощью транспортного криптографического протокола от субъекта A к субъекту B и, соответственно, от субъекта B к субъекту A.

Для формирования указанных векторов используются следующие соотношения.

$$\begin{aligned} \mathtt{CATS}_1 &= \mathit{hmac}_{512}(T, A_1||A_0), \\ \mathtt{SATS}_1 &= \mathit{hmac}_{512}(T, A_2||A_0), \end{aligned}$$

где величина T определена равенством (4.12) и

$$\begin{aligned} A_0 &= \mathit{hash}_{512}(H_5), \\ A_1 &= \mathit{hmac}_{512}(T, A_0), \\ A_2 &= \mathit{hmac}_{512}(T, A_1), \end{aligned}$$

где $H_5 \in \mathbb{V}_\infty$ — конкатенация всех сообщений, переданных в ходе протокола выработки общих ключей. Описанная выше последовательность действий схематично изображена на рисунке 4.14.

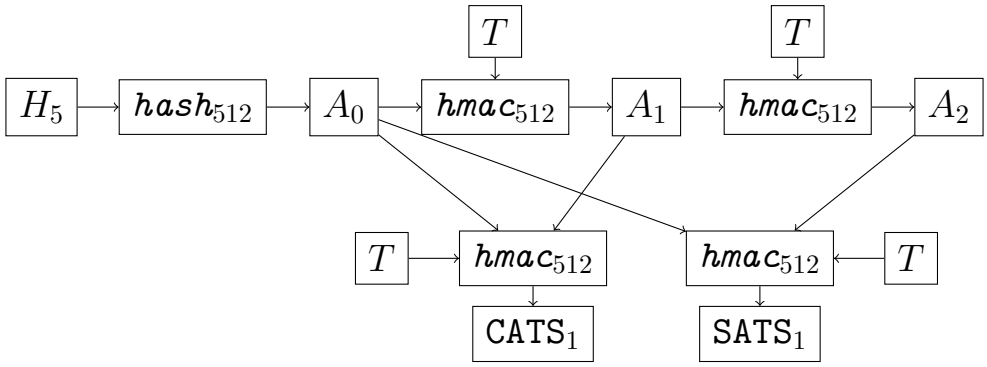


Рис. 4.14: Схема выработки исходной ключевой информации для протокола передачи прикладных данных.

В процессе обмена защищенной информацией исходная ключевая информация изменяет свое значение в соответствии со следующими соотношениями

$$\begin{aligned} \mathtt{CATS}_n &= \mathit{hmac}_{512}(T, \mathtt{CATS}_{n-1}|| \mathit{bin}(n)), \\ \mathtt{SATS}_n &= \mathit{hmac}_{512}(T, \mathtt{SATS}_{n-1}|| \mathit{bin}(n)), \quad n = 2, 3, \ldots. \end{aligned}$$

Максимально допустимое количество изменений исходной ключевой информации определяется рекомендациями Р 1323565.1.028-2018 [365].

5. Производные ключи, используемые для непосредственного шифрования и имитозащиты информации, передаваемой в ходе одной сессии

- Производные ключи шифрования обозначаются символами $\mathtt{eCFK}_{n,m}$ и $\mathtt{eSFK}_{n,m}$ и используются для шифрования информации, передаваемой от субъекта A к субъекту B и, соответственно, от субъекта B к субъекту A.

  Алгоритм выработки производных ключей шифрования зависит от выбранного алгоритма блочного шифрования и определяется равенствами

$$\begin{aligned}
\mathtt{eCFK}_{n,0} &= \mathtt{CATS}_n[0,\ldots,31],\\
\mathtt{eCFK}_{n,m} &= \mathit{acpkm}(\mathtt{eCFK}_{n,m-1}), \quad m = 1,2,\ldots,
\end{aligned}$$

$$\begin{aligned}
\mathtt{eSFK}_{n,0} &= \mathtt{SATS}_n[0,\ldots,31],\\
\mathtt{eSFK}_{n,m} &= \mathit{acpkm}(\mathtt{eSFK}_{n,m-1}), \quad m = 1,2,\ldots,
\end{aligned}$$

  где acpkm — алгоритм преобразования ключа, регламентируемый рекомендациями [356, раздел 4.1.1] следующим образом.

  Определим параметр $J = 4$ для 64-х битного блочного шифра, например «Магма», или $J = 2$ для 128-ми битного шифра, например «Кузнечик». Тогда

$$\mathtt{K}_m = \mathit{acpkm}(\mathtt{K}_{m-1}) = \mathtt{E}(\mathtt{K}_{m-1}, D_1)||\cdots||\mathtt{E}(\mathtt{K}_{m-1}, D_J),$$

  где $\mathtt{K}_m$ ключ, вырабатываемый из ключа $\mathtt{K}_{m-1}$, а $D_1||\cdots||D_J$ — константная последовательность октетов, определяемая в рекомендациях [356] следующим образом:

$$\begin{aligned}
D = (&\mathtt{0x80||0x81||0x82||0x83||0x84||0x85||0x86||0x87||}\\
&\mathtt{0x88||0x89||0x8A||0x8B||0x8C||0x8D||0x8E||0x8F||}\\
&\mathtt{0x90||0x91||0x92||0x93||0x94||0x95||0x96||0x97||}\\
&\mathtt{0x98||0x99||0x9A||0x9B||0x9C||0x9D||0x9E||0x9F}),
\end{aligned}$$

  и

```
 D[0] = 0x80;
 D[1] = 0x81;
      ...
D[30] = 0x9E;
D[31] = 0x9F;
```

- Производные ключи имитозащиты обозначаются символами $\mathtt{iCFK}_{n,m}$ и $\mathtt{iSFK}_{n,m}$ и используются для выработки имитовставки для информации, передаваемой от субъекта A к субъекту B и, соответственно, от субъекта B к субъекту A.

Для алгоритма блочного шифрования «Магма» значение натурального числа $\mathtt{Ctr}$ определяется равенством

$$\mathtt{Ctr} = 18446744069414584320_{10} = \mathtt{FFFFFFFF00000000}_{16},$$

а для алгоритма «Кузнечик» — равенством

$$\begin{aligned}\mathtt{Ctr} &= 340282366920938463444927863358058659840_{10}\\ &= \mathtt{FFFFFFFFFFFFFFFF0000000000000000}_{16}.\end{aligned}$$

Тогда

$$\begin{aligned}
\mathtt{CK}_n &= \mathtt{CATS}_n[32,\ldots,63],\\
\mathtt{SK}_n &= \mathtt{SATS}_n[32,\ldots,63],\\
\mathtt{iCFK}_{n,m} &= \mathtt{E}(\mathtt{CK}_n, \mathtt{bin}(\mathtt{Ctr}+mJ))||\cdots||\mathtt{E}(\mathtt{CK}_n, \mathtt{bin}(\mathtt{Ctr}+(m+1)J-1)),\\
\mathtt{iSFK}_{n,m} &= \mathtt{E}(\mathtt{SK}_n, \mathtt{bin}(\mathtt{Ctr}+mJ))||\cdots||\mathtt{E}(\mathtt{CK}_n, \mathtt{bin}(\mathtt{Ctr}+(m+1)J-1)),
\end{aligned}$$

где величина J определена выше.

Описанные выше алгоритмы выработки производных ключей, используемых для защиты информации, передаваемой от субъекта A к субъекту B, изображены на рисунке 4.15. Аналогичная картина верна для ключей, вырабатываемых для защиты информации, передаваемой в обратном направлении.

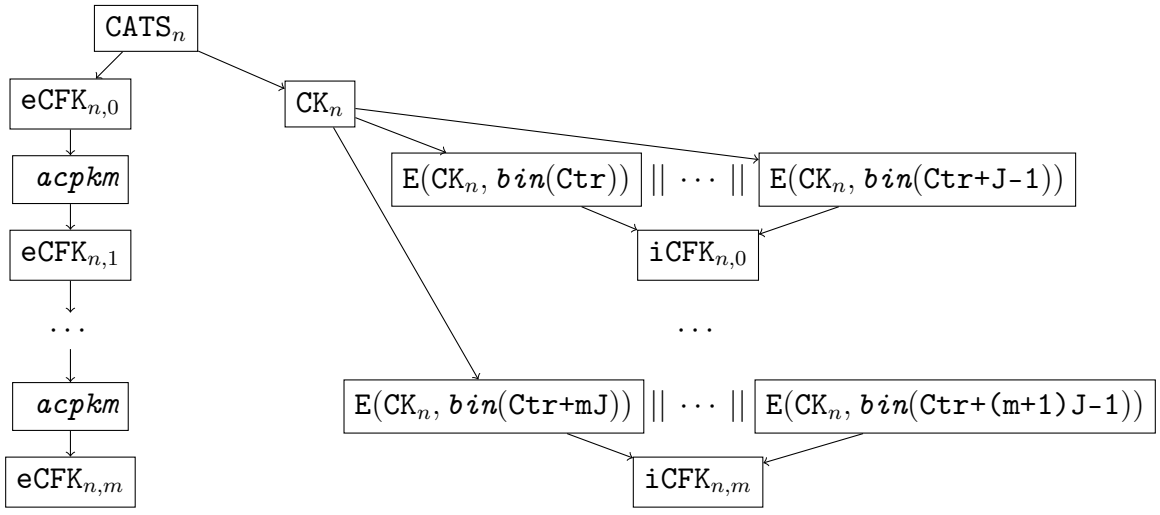


Рис. 4.15: Схема выработки производных ключей.

Максимально допустимое количество производных ключей шифрования и имитозащиты определяется рекомендациями по стандартизации Р 1323565.1.028-2018 [365]. Столь сложная ключевая система позволяет достичь нескольких целей.

- Обеспечивается возможность связать ключи аутентификации, как симметричные, так и асимметричные, с идентификаторами субъектов взаимодействия, например, с уникальными номерами устройств, DNS именами, сетевыми адресами, используемыми при маршрутизации пакетов, т.е. привязать ключевую систему к конкретной архитектуре сети взаимодействия.

- Обеспечивается независимость вырабатываемой общей ключевой информации от ключей аутентификации. Это снижает число применений ключа аутентификации и позволяет увеличить срок его эксплуатации[9].

- Реализуется двухэтапная процедура выработки производных ключей – на первом этапе реализуется медленное преобразование с помощью функции $hmac_{512}$, а на втором – быстрое преобразование, реализуемое с помощью алгоритма блочного шифрования.

  Такая процедура позволяет не только эффективно реализовывать в средствах криптографической защиты процедуру генерации производных ключей, но и обеспечивать защиту значительного объема информации. Согласно спецификации семейства протоколов SP FIOT максимально допустимое число пакетов, которые могут быть переданы в рамках одной сессии взаимодействия от одного субъекта к другому, не превышает 2^{40}. Если предположить, что контрольное устройство будет отправлять одно сообщение в секунду, то ресурса ключевой системы, выработанной в ходе одной сессии взаимодействия, достаточно для работы устройства на протяжении более 30000 лет. Столь большой ресурс позволяет использовать семейство протоколов SP FIOT не только для защиты низкоскоростных каналов связи в Интернете вещей, но и для защиты высокоскоростных магистральных каналов связи.

§ 4.3.2.2. Протокол выработки общих ключей

Приведем теперь описание протокола выработки общих ключей, входящего в семейство протоколов SP FIOT. В рамках данного протокола формируются сообщения и ключевая информация, после чего сформированные сообщения передаются транспортному криптографическому протоколу, где вычисляется имитовставка, а сами данные, при необходимости, зашифровываются и передаются в канал связи.

[9]Согласно рекомендациям Р 1323565.1.012-2017 [355] срок действия долговременной ключевой информации определяется действующими требованиями по безопасности и указывается в ТТЗ на разработку средства криптографической защиты информации.

Как и ранее, будем обозначать символами $\xi_1(t_i), \xi_2(t_i), \ldots$ данные, входящие в состав сообщения, передаваемого в момент времени, помеченный меткой t_i. Также, будем считать что определены открытые параметры протокола:

- эллиптическую кривую $\mathcal{E}_{a,b}(\mathbb{F}_p)$ и точку на этой кривой $P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ такую, что $\mathtt{ord}\, P = q$, где q – простое число,

а также следующие криптографические преобразования:

- функция $\boldsymbol{bin}$, предназначенная для преобразования натуральных чисел в двоичные последовательности;

- режим аутентифицированного шифрования $\boldsymbol{authenc}$, см. определение 3.9, задаваемый парой отображений

$$\begin{aligned} \boldsymbol{authenc}(k_1, k_2, iv, y, x) &= \{c, a\}, \\ \boldsymbol{authdec}(k_1, k_2, iv, y, c, a) &= \{x, b\}, \end{aligned}$$

  где k_1, k_2 – ключи шифрования и имитозащиты, iv – синхропосылка, x – зашифровываемое сообщение, c – шифрекст, y – ассоциированные данные и a – имитовставка.

Схема протокола выработки общих ключей с аутентификацией при помощи симметричного ключа $\mathtt{PSK}_{AB}$ приведена на рисунке 4.16. Протокол представляет из себя обмен четырьмя сообщениями, первые два из которых передаются в незашифрованном виде и предназначены для выработки общей ключевой информации, а вторые два – передаются в зашифрованном виде и предназначены для подтверждения выработанного ключа. При этом, аутентификация субъекта B производится в момент времени, помеченный меткой t_7, а аутентификация субъекта A – в момент времени t_9.

Следует отметить, что в момент времени, помеченный меткой t_3, субъект B убеждается в том, что сообщение $(\xi_1(t_2), \xi_2(t_2), \xi_3(t_2))$ действительно выработано субъектом A и не изменено при передаче по каналу связи, однако он не может убедиться в том, что это сообщение выработано в момент текущей сессии взаимодействия. Поэтому субъект B ждет ответа и завершает процесс аутентификации субъекта A в момент завершения протокола выработки общих ключей.

Исследование стойкости изображенного на рисунке 4.16 протокола проводилось при проведении работ по его стандартизации и может быть найдено в работе [340]. Дальнейшая формализация и моделирование протокола проводились в работах [343, 381].

	субъект A		субъект B
t_1:	$\xi_1(t_1) \in_R \mathbb{F}_q^*$		
t_2:	$\xi_1(t_2) \in_R \mathbb{V}_\infty$ $\xi_2(t_2) = [\xi_1(t_1)]P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ $\xi_3(t_2) = ID_{\mathtt{PSK}_{AB}}$ $\{\emptyset, \xi_4(t_2)\} = \boldsymbol{authenc}(\emptyset, \mathtt{PSK}_{AB},$ $\boldsymbol{bin}(1), \xi_1(t_2)\|\xi_2(t_2)\|\xi_3(t_2), \emptyset)$	$\longrightarrow$	
t_3:			$\xi_3(t_2) \to \mathtt{PSK}_{AB}$ $\{\emptyset, b\} = \boldsymbol{authdec}(\emptyset, \mathtt{PSK}_{AB},$ $\boldsymbol{bin}(1), \xi_1(t_2)\|\xi_2(t_2)\|\xi_3(t_2), \emptyset, \xi_4(t_2))$ $b =^? \boldsymbol{true}$ $\xi_1(t_3) \in_R \mathbb{F}_q^*$
t_4:			$\xi_1(t_4) \in_R \mathbb{V}_\infty$ $\xi_2(t_4) = [\xi_1(t_3)]P \in \mathcal{E}_{a,b}(\mathbb{F}_p)$ $\{\emptyset, \xi_3(t_4)\} = \boldsymbol{authenc}(\emptyset, \mathtt{PSK}_{AB},$ $\boldsymbol{bin}(2), \xi_1(t_4)\|\xi_2(t_4), \emptyset)$
		$\longleftarrow$	
t_5:			$H_1 = \xi_1(t_2)\|\xi_2(t_2)\|\xi_3(t_2)\|\xi_1(t_4)\|\xi_2(t_4)$ $W = [\xi_1(t_3)]\xi_2(t_2)$ $\mathtt{eSHTK}\|\mathtt{iSHTK} = \boldsymbol{hmac}_{512}\big($ $\boldsymbol{hash}_{512}(\boldsymbol{bin}(W)\|\mathtt{PSK}_{AB}), \boldsymbol{hash}_{512}(H_1)\big)$
t_6:			$\{\xi_1(t_6), \xi_2(t_6)\} = \boldsymbol{authenc}\big(\mathtt{eSHTK}, \mathtt{iSHTK},$ $\boldsymbol{bin}(3), \emptyset, \boldsymbol{hash}_{256}(H_1)\big)$
		$\longleftarrow$	
t_7:	$W = [\xi_1(t_1)]\xi_2(t_4)$ $\mathtt{eSHTK}\|\mathtt{iSHTK} = \boldsymbol{hmac}_{512}\big($ $\boldsymbol{hash}_{512}(\boldsymbol{bin}(W)\|\mathtt{PSK}_{AB}), \boldsymbol{hash}_{512}(H_1)\big)$ $\{\xi_1(t_7), b\} = \boldsymbol{authdec}(\mathtt{eSHTK}, \mathtt{iSHTK}$ $\boldsymbol{bin}(3), \emptyset, \xi_1(t_6), \xi_2(t_6))$ $b =^? \boldsymbol{true}$		
t_8:	$H_3 = H_1\|\xi_1(t_7)$ $\mathtt{eCHTK}\|\mathtt{iCHTK} = \boldsymbol{hmac}_{512}\big($ $\boldsymbol{hash}_{512}(\boldsymbol{bin}(W)\|\mathtt{PSK}_{AB}), \boldsymbol{hash}_{512}(H_3)\big)$ $\{\xi_1(t_8), \xi_2(t_8)\} = \boldsymbol{authenc}\big(\mathtt{eCHTK},$ $\mathtt{iCHTK}, \boldsymbol{bin}(4), \emptyset, \boldsymbol{hash}_{256}(H_3)\big)$	$\longrightarrow$	
t_9:			$\mathtt{eCHTK}\|\mathtt{iCHTK} = \boldsymbol{hmac}_{512}\big($ $\boldsymbol{hash}_{512}(\boldsymbol{bin}(W)\|\mathtt{PSK}_{AB}), \boldsymbol{hash}_{512}(H_3)\big)$ $\{\xi_1(t_9), b\} = \boldsymbol{authdec}(\mathtt{eCHTK},$ $\mathtt{iCHTK}, \boldsymbol{bin}(4), \emptyset, \xi_1(t_8), \xi_2(t_8)\big)$ $b =^? \boldsymbol{true}$ $H_5 = H_3\|\xi_1(t_9)$

Рис. 4.16: Семейство SP FIOT. Протокол выработки общих ключей.

Заключение к § 4.3

В параграфе § 4.3 предложен новый протокол выработки общего ключа «Крокус» со взаимной аутентификацией субъектов взаимодействия, реализуемой с использованием электронной подписи. Доказана теорема о стойкости протокола «Крокус» относительно задач определения общего ключа, дешифрования и навязывания передаваемой в ходе выполнения протокола информации.

Предложено семейство криптографических протоколов «SP FIOT», развивающее идеи, заложенные в протокол «Крокус», и предназначенное для обеспечения защищенного взаимодействия в сетях «Интернета вещей».

Отметим, что доказанные в этом и предыдущем параграфах теоремы носят неконструктивный характер, т.е. не позволяют в явном виде предъявить численные значения параметров, необходимых для оценки степени защиты, реализуемой криптографическим протоколом. Метод, позволяющий получать численные оценки таких параметров, рассматривается в последнем параграфе диссертационной работы.

§ 4.4. Методика оценки безопасности криптографических протоколов

В последнем параграфе диссертационной работы излагается подход к оценке безопасности криптографических протоколов. В первом разделе параграфа вводится понятие «свойства безопасности», позволяющее связать между собой угрозы безопасности и возможные атаки нарушителя. Приводится классификация свойств безопасности и их взаимосвязь между собой.

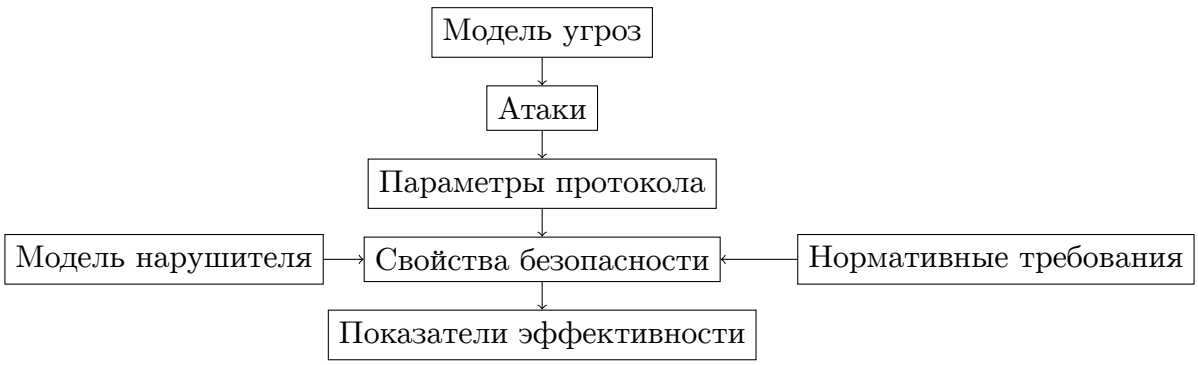


Рис. 4.17: Схема вычисления показателей эффективности.

Во втором разделе вводится формальная модель криптографического протокола и с ее помощью моделируются базовые свойства безопасности. В третьем разделе дается формальное определение показателей эффективности защиты и предлагается способ определения их численных значений.

В заключительном, четвертом разделе приводится методика оценки безопасности криптографических протоколов, сложившаяся при проведении исследований ряда отечественных стандартизированных решений [317, 352, 357, 358, 365, 367, 370, 373], фрагменты методики рассматривались ранее в работах [334, 340, 380]. Дальнейшее изложение параграфа следует работе [343].

§ 4.4.1. Свойства безопасности

При формировании требований к криптографическим алгоритмам, всегда определяются свойства, которыми эти алгоритмы должны обладать. Например, согласно [347, глава 2, стр. 56], для обеспечения безопасности информационного взаимодействия должны выполняться свойства:

- конфиденциальности,
- целостности,
- аутентификации,
- невозможности отказа от авторства.

Другим примером служит стандарт ГОСТ Р 34.10-2012, см. [280], где также определяется перечень свойств, которыми должна обладать электронная подпись:

- осуществление контроля целостности передаваемого подписанного сообщения,
- доказательное подтверждение авторства лица, подписавшего сообщение,
- защита сообщения от возможной подделки.

Также, свойства безопасности формулировались ранее в диссертационной работе в разделах 4.2.4 и 4.3.

Впервые, применительно к криптографическим протоколам, свойства безопасности рассматривались в работе Белларе-Рогавея [26], а позднее

расширялись в RFC 3552 [213] и в рамках проекта AVISPA, см. [9]. Также свойства безопасности рассматривались в ГОСТ Р ИСО/МЭК 27033-1:2011, см. [288, раздел 7.3], и в работах [380, 391]. Во всех перечисленных случаях свойства безопасности определялись качественно, без детального уточнения – в чем же именно заключается указанное свойство.

Будем использовать следующее формальное определение.

Определение 4.3. *Пусть* $0 < \pi_0 \leqslant 1$ *заданное действительное число. Под свойством безопасности будем подразумевать свойство протокола, прямо или косвенно обеспечивать невозможность реализации заданной угрозы с вероятностью, превышающей значение* π_0.

Значение π_0 может принимать значения 0.5, 0.1, 0.01, 0.001, ... и т.п. Оно определяет вероятность нарушения рассматриваемого свойства безопасности и позволяет вывести из рассмотрения атаки с ничтожной вероятностью успеха, например, случайное угадывание зашифрованного текста. Точное значение величины π_0 может определяться действующими требованиями по безопасности, моделью угроз безопасности передаваемой информации или рассчитываться с использованием риск-ориентированного подхода.

Расширим перечень из отчета [9] и будем применять при анализе протоколов следующие свойства безопасности. Формальное определение указанных свойств будет дано в следующем разделе.

С 1. *Свойство аутентификации субъекта (участника протокола) другим субъектом (участником протокола)* заключается в подтверждении одним субъектом подлинности другого субъекта, участвующего в выполнении протокола, а также получении гарантии того, что субъект, подлинность которого подтверждается, действительно принимает участие в выполнении текущей сессии протокола.

Свойство аутентификации субъекта может быть как односторонним, так и взаимным. В последнем случае свойство должно выполняться для всех участвующих во взаимодействии субъектов. Данное свойство содержится в [9, свойство G1], [213, п. 2.1.3], а также в [391, раздел 3].

С 2. *Свойство аутентификации сообщения* заключается в подтверждении подлинности источника сообщения и целостности передаваемого сообщения.

Подлинность источника сообщения означает, что протокол должен обеспечивать гарантии того, что полученное сообщение или его часть были созданы субъектом взаимодействия в ходе выполнения

текущей сессии протокола в некоторый момент времени, предшествующий получению сообщения.

Фактически, в рамках данного свойства сообщение однозначно связывается со своим источником (субъектом, отправившим сообщение), а выполнение свойства гарантирует, что сообщение не было искажено, в частности, подделано нарушителем, при передаче по каналам связи. Данное свойство содержится в [9, свойство G2], [213, п. 2.1.2] и в [391, раздел 3].

С 3. *Свойство целостности сообщений* заключается в том, что получатель сообщения обладает возможностью проверить, что полученные им в процессе информационного взаимодействия данные (или их часть) не были модифицированы, уничтожены и являются теми же самыми данными, что отправил отправитель сообщения. Данное свойство содержится в [213, п. 2.1.2], а также в [391, раздел 2].

С 4. *Свойство защиты от повторов* заключается в том, что один раз корректно принятое субъектом взаимодействия сообщение не должно быть принято повторно. В зависимости от протокола данное свойство может сформулировано в виде одного из следующих требований:

- должна быть обеспечена гарантия того, что сообщение выработано в рамках текущей сессии протокола,
- должна быть обеспечена гарантия того, что сообщение выработано в рамках заданного интервала времени,
- сообщение не было принято ранее.

В отечественной литературе данное свойство часто называют *свойством невозможности навязывания ложных сообщений*, подразумевая под этим защиту как от повторного принятия истинных сообщений, так и от подделанных нарушителем сообщений (свойство С2). Данное свойство содержится в [9, свойство G3], а также в [391, раздел 3].

С 5. *Свойство неявной аутентификации получателя* заключается в том, что протокол должен обладать средствами, гарантирующими, что отправленное сообщение может быть прочитано только теми субъектами, для которых оно предназначено. Только законные авторизованные субъекты должны иметь доступ к данной информации, многоадресным сообщениям или групповому взаимодействию. Данное свойство содержится в [9, свойство G4].

С 6. *Свойство групповой аутентификации* заключается в том, что законные авторизованные члены заранее определенной группы субъектов могут аутентифицировать источник и содержание информации или группового сообщения. Сюда также входят протоколы, в которых субъекты группового взаимодействия не доверяют друг другу. Данное свойство содержится в [9, свойство G5], а также в [391, раздел 3].

С 7. *Свойство аутентификации субъекта (участника протокола) доверенной третьей стороной.* В протоколах, явно реализующих взаимодействие субъектов с доверенной третьей стороной, данное свойство эквивалентно первому из перечисленных нами свойств.

В случае использования инфраструктуры открытых ключей данное свойство может выполняться косвенно, путем заверения открытых ключей участников взаимодействия электронной подписью удостоверяющего (доверенного) центра; при этом привязка аутентификации субъекта к какой-либо сессии протокола не может быть обеспечена. Данное свойство содержится в [9, свойство G6].

С 8. *Свойство конфиденциальности ключа* предполагает, что в ходе информационного взаимодействия значение ключа не может стать известным нарушителю, а также легитимным пользователям информационной системы, для которых данный ключ не предназначен. Данное свойство может применяться как к исходной ключевой информации, так и к производным сессионным ключам.

С 9. *Свойство аутентификации ключа* предполагает, что один из субъектов взаимодействия получает подтверждение того, что никакой другой субъект, кроме заранее определенного второго субъекта и, возможно, доверенного центра, не может обладать секретным ключом, выработанных в ходе выполнения протокола. Данное свойство содержится в [9, свойство G7], а также в [391, раздел 3].

С 10. *Свойство подтверждения ключа* заключается в том, что один из субъектов взаимодействия получает подтверждение того, что второй субъект (или группа субъектов) действительно обладает заданным секретным ключом и/или имеет доступ к информации, необходимой для выработки заданного секретного ключа. Данное свойство содержится в [9, свойство G8], а также в [391, раздел 3].

С 11. *Свойство стойкости при компрометации производных ключей* состоит в том, что компрометация производных ключей, т.е. ключей

используемых непосредственно для шифрования и имитозащиты передаваемой информации, не приводит к нарушению других свойств безопасности как в рамках текущей, так и в других сессиях протокола, в частности, к компрометации производных ключей, выработанных ранее или планируемых к выработке в дальнейшем.

В литературе данное свойство часто называют защитой от «чтения вперед/назад» или используют термин «perfect forward secrecy». Данное свойство содержится в [9, свойство G9], а также в [391, раздел 3].

С 12. *Свойство стойкости при компрометации ключа аутентификации* состоит в том, что компрометация долговременного ключа аутентификации не приводит к нарушению конфиденциальности информации, переданной до момента компрометации ключа, а в случае пассивного нарушителя – и к нарушению конфиденциальности информации, передаваемой после завершения текущей сессии протокола. В литературе данное свойство иногда называют защитой от «чтения назад».

С 13. *Свойство формирования новых ключей* заключается в том, что протокол обладающий данным свойством, позволяет формировать уникальные сессионные и/или производные ключи для каждой сессии протокола. Данное свойство содержится в [9, свойство G10].

С 14. *Свойство защиты от навязывания ключевых значений* гарантирует, что ни один из субъектов взаимодействия не может навязать значение общего секретного, сессионного или производного ключа по своему выбору другому субъекту взаимодействия.

С 15. *Свойство защиты от навязывания параметров безопасности* гарантирует, что используемые в ходе выполнения протокола или согласуемые на этапе установления соединения параметры безопасности не могут быть навязаны нарушителем.

В качестве параметров безопасности могут выступать наборы используемых криптографических преобразований, численные параметры алгоритмов и алгебраических структур, в которых выполняется протокол, случайные значения, вырабатываемые в ходе выполнения протокола и т.п. Данное свойство содержится в [9, свойство G11].

С 16. *Свойство конфиденциальности* заключается в том, что данные, передаваемые в ходе информационного взаимодействия, не могут стать известными нарушителю и/или легитимным участникам, для

которых они не предназначены. Данное свойство содержится в [9, свойство G12], [213, п. 2.1.1], а также в [347, глава 2], [391, раздел 3]. Легко видеть, что нарушение свойства конфиденциальности ключевой информации (свойство С8) приводит к нарушению конфиденциальности передаваемых данных.

С 17. *Свойство инвариантности отправителя* заключается в том, что на протяжении выполнения всего протокола получатель сообщений сохраняет уверенность в том, что источник сообщения остался тем же, что и источник, с которым было начато взаимодействие (сессия протокола). Данное свойство содержится в [9, свойство G16].

С 18. *Свойство анонимности субъекта (участника протокола)* состоит в том, что нарушитель, осуществляющий перехват сообщений, не должен иметь возможность связать сообщения одного из субъектов взаимодействия с самим субъектом или его идентификатором. Данное свойство содержится в [9, свойство G13].

С 19. *Свойство анонимности субъекта для других участников* заключается в том, каждый субъект взаимодействия не должен иметь возможность узнать реальную личность других субъектов, а должен взаимодействовать с их псевдонимом или случайным идентификатором. Данное свойство содержится в [9, свойство G14].

С 20. *Свойство защищенности от атак «отказ в обслуживании»* подразумевает, что реализующее протокол средство защиты информации обеспечивает алгоритмические, технические и организационно-штатные меры защиты от указанного типа атак. Данное свойство содержится в [9, свойство G15].

Теоретическое исследование протокола может лишь проверить наличие алгоритмических мер, обеспечивающих защиту от данного класса атак, а также наличие эксплуатационной документации, содержащей описание технических и организационно-штатных мер защиты. В рамках предлагаемой методики представляется возможным получить лишь тривиальное численное значение показателя эффективности для данного свойства.

С 21. *Свойство защищенности от утечек по скрытым (логическим) каналам* подразумевает, что протокол содержит реализацию алгоритмических мер защиты от атак реализуемых нарушителем путем применения непредусмотренных коммуникационных каналов передачи информации. Отметим, что современные транспортные протоколы,

например [365, 370], содержат в себе ряд мер, предназначенных для обеспечения данного свойства.

Классификация угроз безопасности, реализуемых с использованием скрытых каналов, модель нарушителя и перечень мер защиты информационной системы от атак с использованием скрытых каналов должны разрабатываться на основе стандартов [285, 286]. Получение численных оценок показателей эффективности мер защиты от скрытых логических каналов выходит за рамки настоящей диссертационной работы. Отдельные результаты в данном направлении получены в работах [278, 304, 318], см. также [306].

С 22. *Свойство защищенности от KCI-атак.* Под KCI[10]-атакой (атакой имперсонификации при компрометации долговременного секретного ключа) понимается атака, при выполнении которой нарушитель, получивший доступ к долговременному секретному ключу субъекта взаимодействия, может выдать себя перед ним за любого другого субъекта взаимодействия в рамках текущей или будущей сессии выполнения протокола. Свойство считается выполненным, если KCI-атака невыполнима. Данное свойство описано в [39].

С 23. *Свойство защищенности от UKS-атак.* Под UKS-атакой понимается, последовательность действий нарушителя, в результате которой законные авторизованные субъекты в процессе информационного взаимодействия вырабатывают общий ключ, но один из субъектов считает, что он выработал общий ключ с третьим субъектом (навязанным нарушителем в ходе выполнения протокола). При этом компрометации общего ключа как таковой не происходит, но нарушается требование аутентификации субъектов взаимодействия. Свойство считается выполненным, если подобная ситуация невозможна. Данное свойство описано в [41, 68].

С 24. *Свойство невозможности отказа от совершенных действий* представляет собой свойство информационной системы обеспечивать возможность проследить за всеми действиями участника взаимодействия. Согласно Р 1323565.1.012-2017, см. [355, раздел 6.1.14], данное свойство должно обеспечиваться средством криптографической защиты информации, реализующим криптографический протокол. Данное свойство, также, содержится в [9, свойство G17].

С 25. *Свойство доказательства происхождения* заключается в неоспоримом доказательстве отправки сообщения. Данное свойство содержится в [9, свойство G18].

[10]KCI-атака это сокращение английского термина «Key Compromise Impersonation attack».

С 26. *Свойство доказательства доставки* заключается в неоспоримом доказательстве получения сообщения. Данное свойство содержится в [9, свойство G19].

С 27. *Свойство целостности множества состояний (криптографическое связывание состояний)* заключается в том, что все субъекты взаимодействия после выполнения протокола (или его части) в рамках одной сессии связи имеют одинаковое представление обо всех субъектах этой сессии и выполняемых ими ролях, а также о состоянии выполнения протокола. Данное свойство описано в рекомендациях [51, 79, 105].

Можно провести классификацию свойств безопасности по объектам применения, влияющим на безопасность исследуемого криптографического протокола, см. таблицу 4.1.

Объект применения	Свойства безопасности
Аутентификация субъектов	С1, С2, С5, С6, С7, С9
Целостность	С3, С27
Ключевая система	С8, С10, С11, С12, С13, С14
Субъект взаимодействия	С17, С18, С19, С24
Атаки нарушителя	С4, С15, С20, С21, С22, С23
Защищаемые данные	С16, С25, С26

Таблица 4.1: Свойства безопасности по объектам применения.

Отметим, что для большинства используемых на практике криптографических протоколов все перечисленные свойства безопасности не могут обеспечиваться одновременно. Примерная классификация свойств безопасности, которые могут обеспечиваться протоколами с различными целевыми назначениями, приведена в таблице 4.2, см. также [391].

Класс протоколов	Свойства безопасности
Протоколы обеспечения целостности сообщения	С4, С10, С13, С3, С22, С23
Протоколы на основе цифровой подписи	С1, С2, С8, С9, С11, С12, С17, С22, С23
Протоколы на основе цифровой подписи вслепую	С1, С5, С11, С12, С19, С22, С23
Протокол односторонней аутентификации	С1, С2, С8, С9, С11, С12, С17, С22, С23
Протокол взаимной аутентификации	С1, С2, С6, С8, С9, С11, С17, С22, С23
Протокол групповой аутентификации	С1, С2, С6, С9, С11, С12, С17, С22, С23
Протоколы конфиденциальной передачи	С13, С15, С16, С3, С22, С23,
Протоколы распределения ключей	С1, С2, С8, С9, С22, С23
Протоколы выработки общего ключа	С1, С2, С4, С8, С9, С10, С11, С22, С23

Таблица 4.2: Свойства безопасности по целевому назначению.

Следует отметить, что на практике сложно отнести криптографический протокол к тому или иному классу, поскольку в большинстве случаев протоколы обеспечивают выполнение свойств, характерных для нескольких целевых функций.

Свойство	Зависимость
C1 - аутентификации участника протокола другим участником	Базовое
C2 - аутентификации сообщения	C1, C3
C3 - целостности сообщений	Базовое
C4 - защиты от повторов	Базовое
C5 - неявной аутентификации получателя	C1, C9
C6 - групповой аутентификации	C1, C9
C7 - аутентификации субъекта доверенной третьей стороной	C1
C8 - конфиденциальности ключа	Базовое
C9 - аутентификации ключа	C1, C2, C10, C15
C10 - подтверждения ключа	Базовое
C11 - стойкости при компрометации производных ключей	C13
C12 - стойкости при компрометации ключа аутентификации	C13
C13 - формирования новых ключей	C15
C14 - защиты от навязывания ключевых значений	C1, C3
C15 - защиты от навязывания параметров безопасности	C1, C2, C3
C16 - конфиденциальности	C3, C9, C10
C17 - инвариантности отправителя	C1, C9
C18 - анонимности субъекта	Базовое
C19 - анонимности субъекта для других участников	Базовое
C20 - защищенности от атак «отказ в обслуживании»	Базовое
C21 - защищенности от утечек по скрытым (логическим) каналам	Базовое
C22 - защищенности от KCI-атак	C1, C9, C10, C12, C13, C14
C23 - защищенности от UKS-атак	C1, C9, C10, C15, C27
C24 - невозможности отказа от совершенных действий	C25, C26, C27
C25 - доказательства происхождения	C1, C2, C9
C26 - доказательства доставки	C9, C10
C27 - целостности множества состояний	C1, C17, C22, C23

Таблица 4.3: Зависимости между свойствами безопасности.

Для построения формальной модели свойств безопасности полезно разбить сформулированные выше свойства на два больших класса:

- базовые свойства, выполнение которых зависит от сложности решения математических задач, используемых в криптографических преобразованиях,
- производные свойства, являющиеся комбинацией базовых и других производных свойств.

Зависимость между свойствами безопасности представлена в таблице 4.3. Отметим также, что в криптографических механизмах, представляющих собой совокупность нескольких протоколов, свойства безопасности могут наследоваться. Например, транспортный протокол, реализующий только шифрование и имитозащиту передаваемой информации, сам по себе не обеспечивают свойство аутентификации субъектов взаимодействия, однако он может его наследовать в случае использования ключевой информации, предварительно выработанной в ходе протокола выработки ключей с аутентификацией участников.

§ 4.4.2. Формальная модель протокола и моделирование свойств безопасности

В этом разделе предлагается способ моделирования криптографических протоколов в виде дискретной динамической системы. В рамках данной системы каждый субъект взаимодействия представляется в виде неавтономного автомата, множество состояний которого определяется ключевой системой и спецификацией протокола, функции перехода в новое состояние определяются криптографическими преобразованиями, а внешние воздействия – получаемыми от нарушителя и легитимных субъектов сообщениями, интерпретируемыми как реализации некоторых случайных величин.

В рамках данной модели формализуются перечисленные ранее свойства безопасности и формируются фрагменты протокола, наличие которых обязательно для выполнения того или иного свойства.

Определение 4.4. *Обозначим символом $\mathbb{V}^*_\infty$ множество двоичных последовательностей произвольной конечной длины, включающее в себя последовательность длины ноль, которую мы обозначим символом $\emptyset$.*

Пусть $a = (\alpha, \sigma)$ некоторая абстрактная ячейка памяти. Мы будем говорить, что ячейка характеризуется:

1) *значением $\alpha \in \mathbb{V}^*_\infty$ и считаем, что неопределенному значению ячейки a соответствует символ $\emptyset$;*

2) *подтверждением $\sigma \in \mathbb{B}$ и считаем, что значение `true` соответствует подтвержденному значению, а `false` – неподтвержденному значению ячейки a.*

Понятие «подтверждения» ячейки вводится для того, чтобы формализовать уверенность субъекта, владеющего ячейкой $a = (\alpha, \sigma)$ в том, значение α, содержащееся в подтвержденной ячейке истинно, а не вычислено ошибочно, подделано и/или навязано нарушителем.

Определение 4.5. *Пусть* $\{t_k\}_0^\infty$ *монотонно возрастающая последовательность натуральных чисел, где индекс* $k \in \mathbb{N}_0$ *принимает значения* $k = 0, 1, \ldots, k_{max}$ *для некоторого натурального значения* k_{max}*, определяемого спецификацией протокола.*

Для субъекта A *будем называть его состоянием в момент времени* t_k *множество ячеек памяти*

$$A(t_k) = \{a_1, \ldots, a_{n(A)} : \; a_i = \big(\alpha_i(t_k), \sigma_i(t_k)\big)\},$$

значения и подтверждения которых могут изменяться с изменением момента времени. Количество ячеек памяти $n(A)$ *зависит от роли субъекта и определяется спецификацией протокола.*

У различных субъектов точные значения временных меток t_k могут отличаться. Можно считать, что время t_0 это время начальной инициализации состояния субъекта, а $t_1, t_2, \ldots$ времена отправки и получения сообщений из канала связи.

Как указывалось ранее, в транспортных протоколах, реализуется только процедура отправки (получения) сообщений, а значение величины k_{max} может быть, формально, не ограничено. Однако существующие в Российской Федерации требования по ограничению объема зашифровываемой на одном ключе информации, см. Р 1323565.1.012-2017 [355], накладывают ограничения на число передаваемых сообщений и, как следствие, на количество возможных состояний k_{max}.

Определение 4.6. *Мы будем говорить, что модель протокола определена, если для каждого субъекта:*

- *определено множество ячеек памяти, образующих изменяемое в ходе выполнения протокола состояние;*
- *определено количество возможных состояний,*

а также, в соответствии со спецификацией протокола, определены функции перехода из одного состояния в другое, позволяющие однозначно определить значение и подтверждение каждой ячейки памяти, т.е. для всех $k = 0, 1, \ldots, k_{max}$ *и всех* $i = 1, \ldots, n(A)$ *определены:*

1) целые неотрицательные числа l_k*,*

2) отображения

$$\boldsymbol{var}_{i,k}(x_1, \ldots, x_{n(A)+l_k}) : (\mathbb{V}_\infty^*)^{n(A)+l_k} \to \mathbb{V}_\infty^*,$$

$$\boldsymbol{conf}_{i,k}(x_1, \ldots, x_{n(A)+l_k}) : (\mathbb{V}_\infty^* \times \mathbb{B})^{n(A)} \times (\mathbb{V}_\infty^*)^{l_k} \to \mathbb{B},$$

такие, что

$$\alpha_i(t_{k+1}) = \boldsymbol{var}_{i,k}(\alpha_1(t_k), \ldots, \alpha_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)), \tag{4.13}$$

$$\sigma_i(t_{k+1}) = \boldsymbol{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)), \tag{4.14}$$

где $i = 1, \ldots, n(A)$*, а значения* $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$ *рассматриваются как реализации* l_k *случайных величин, принимающих значения из* $\mathbb{V}^*_\infty$ *в момент времени* t_k*.*

Полагаем, что в подавляющем большинстве случаев введенные нами отображения $\boldsymbol{var}_{i,k}$ и $\boldsymbol{conf}_{i,k}$ будут задаваться тривиальными соотношениями

$$\begin{array}{lcl} l_k & = & 0, \\ \alpha_i(t_{k+1}) & = & \boldsymbol{var}_{i,k}(\alpha_1(t_k), \ldots, \alpha_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \quad = \quad \alpha_i(t_k), \\ \sigma_i(t_{k+1}) & = & \boldsymbol{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \quad = \quad \sigma_i(t_k), \end{array}$$

при $k \in \{1, \ldots, k_{max}\}$. В данном случае аргументы функций $\boldsymbol{var}_{i,k}$ и $\boldsymbol{cont}_{i,k}$, отличные от $\alpha_i(t_k)$ и, соответственно, $\sigma_i(t_k)$ являются несущественными, т.е. не влияют на возвращаемое значение.

Для остальных случаев поясним смысл, который вкладывается в введенные случайные значения $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$.

1) Случай $l_k = 0$ описывает автономное изменение состояния, которое субъект выполняет без какого-либо влияния извне. Такое изменение состояния может использоваться для детализации спецификации протокола, например, для изменения или подтверждения состояний элементов ключевой системы.

2) В ряде протоколов для аутентификации субъектов взаимодействия или выработки общей ключевой информации требуется генерация случайных значений; именно эти значения выступают в качестве величин $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, изменяющих состояние участника протокола (для подавляющего числа протоколов в этом случае можно считать, что $l_k = 1$).

3) Во всех протоколах субъект взаимодействия обрабатывает данные, поступающие из канала связи и рассматриваемые нами как случайные величины $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, принимающие значения в своей области определения.

Функции $\boldsymbol{conf}_{i,k}$ предназначены для подтверждения того, что значение $\alpha_i(t_k)$ является истинным, а величины, использованные для определения или формирования значения $\alpha_i(t_k)$, не были искажены или навязаны нарушителем в процессе обмена информацией по каналам связи.

Примером функции подтверждения могут служить функции проверки имитовставки или электронной подписи, которые позволяют гарантировать истинность подтверждаемых значений при помощи криптографических преобразований. При этом допускается, что одна функция $\mathtt{conf}_{i,k}$ может подтверждать истинность нескольких ячеек $\alpha_{i_1}(t_k), \alpha_{i_2}(t_k), \ldots$, если все они одновременно являются аргументами функции $\mathtt{conf}_{i,k}$, например, при проверке имитовставки проверяется истинность как сообщения, так и используемого секретного ключа.

Далее всегда предполагается, что одним из аргументов функции $\mathtt{conf}_{i,k}$ является некоторая ключевая информация, определяемая перед началом протокола (исходная ключевая информация) или вырабатываемая в ходе его выполнения.

Определение 4.7. *Зафиксируем некоторый индекс* $i \in \{1, \ldots, n(A)\}$ *и момент времени* t_k *такой, что* $k \in \{1, \ldots, k_{max}\}$.

1) *Будем говорить, что значение ячейки* $a_i = (\alpha_i(t_k), \sigma_i(t_k))$ *подтверждено в момент времени* t_k*, если* $\sigma_i(t_k) = \mathtt{true}$.

2) *Будем говорить, что значение ячейки* a_i *подтверждено косвенно, если* $\sigma_i(t_k) = \mathtt{false}$ *и значение* $\alpha_i(t_k)$ *определено равенством*

$$\alpha_i(t_k) = \mathtt{var}_{i,k-1}(\alpha_{i_1}(t_{k-1}), \ldots, \alpha_{i_{s_i}}(t_{k-1})), \quad i_1, \ldots, i_{s_i} \in \{1, \ldots, n(A)\}$$

т.е. зависит только от существенных значений $\alpha_{i_1}(t_{k-1}), \ldots, \alpha_{i_{s_i}}(t_{k-1})$ *таких, что*

$$\sigma_{i_1}(t_{k-1}) = \cdots = \sigma_{i_{s_i}}(t_{k-1}) = \mathtt{true}.$$

Можно предположить, что криптографический протокол является безопасным для субъекта A в момент времени t_k, где $k \in \{1, \ldots, k_{max}\}$, если значение всех ячеек памяти состояния $A(t_k)$ является либо подтвержденным с использованием криптографических преобразований или выработано самим субъектом, либо подтверждено косвенным образом. Однако, как мы покажем позднее, это предположение является необходимым, но не достаточным условием безопасности протокола.

В отличие от большинства других подходов к моделированию криптографических протоколов, предложенная модель ориентирована не на поиск возможных действий нарушителя и построение атак на протокол; модель ориентирована на поиск и построение графа зависимостей между всеми ячейками состояния субъекта, позволяющими проследить состояние ячеек памяти и подтвердить их «истинность» начиная с некоторого шага выполнения протокола (момента времени t_k).

В случае, если спецификация протокола допускает существование неподтверждаемых переменных, то поиск возможных атак должен производиться с использованием автоматических верификаторов таких, как Avispa [9], Scyther [65] или Proverif [204].

§ 4.4.2.1. Свойство аутентификации субъекта

Напомним, что с 2020 года вопросы идентификации и аутентификации субъектов взаимодействия в Российской Федерации должны решаться с учетом ГОСТ Р 58833-2020, см. [287]. Согласно данному стандарту при взаимодействии сторон с целью доступа к информации должны быть выполнены следующие процедуры:

1) первичная идентификация, в ходе которой регистрирующей стороной (доверенным центром) субъекту доступа должен присваиваться уникальный идентификатор $ID \in \mathbb{V}_\infty$;

2) вторичная идентификация, целью которой является опознавание субъекта доступа, т.е. предъявление субъектом присвоенного ранее идентификатора ID при попытке доступа к информации; выполнение вторичной идентификации производится субъектом, предоставляющим доступ к информации – в нашем случае, другим участником взаимодействия;

3) аутентификация субъекта доступа, в которую должны входить действия по проверке подлинности субъекта доступа, а также принадлежности субъекту доступа предъявленного идентификатора и аутентификационной информации.

Аутентификация субъекта доступа, согласно [287], может осуществляться с использованием нескольких факторов:

- фактора знания определенной информации, например, знания секретного ключа или пароля,
- фактора владения определенным предметом,
- биометрическим фактором, описывающим определенные характеристики аутентифицируемого субъекта.

Поскольку при разработке криптографических протоколов принято использовать только фактор знания ключевой информации, необходимо дополнить положения стандарта [287] и ввести в использование понятие секретного ключа аутентификации, однозначно связанного с уникальными идентификатором субъекта, см. разъяснение на стр. 255.

Определение 4.8. *Будем говорить, что криптографический протокол обеспечивает свойство аутентификации субъекта B, выполняемой субъектом A (свойство C1), если:*

1) *с субъектом B связан идентификатор ID_B,*

2) *для субъекта B определены – ключ аутентификации $k_a \in \mathbb{K}_a$ и ключ проверки кода аутентификации $k_c \in \mathbb{K}_c$, однозначно связанные с идентификатором ID_B,*

3) *для некоторого натурального m определены функции выработки кода аутентификации $\mathtt{mac} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ и проверки кода аутентификации $\mathtt{conf} : \mathbb{K}_c \times \mathbb{V}_\infty^* \times \mathbb{V}_m \to \mathbb{B}$ такие, что*

$$\mathtt{conf}(k_c, \xi, \mathtt{mac}(k_a, \xi)) = \mathtt{true},$$

для любой тройки значений k_a, k_c и $\xi \in \mathbb{V}_\infty^*$,

4) *субъекту A известны идентификатор ID_B субъекта B и подтвержденное значение ключа проверки кода аутентификации k_c,*

5) *в состав протокола входит следующая последовательность[11] шагов:*

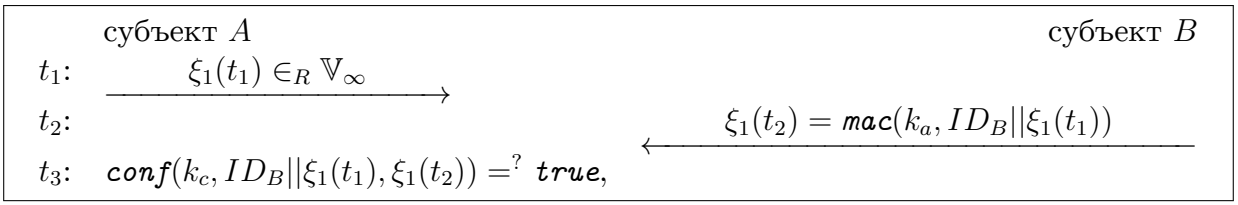


Рис. 4.18: Протокол аутентификации субъекта.

6) *истинно условие проверки на шаге t_3.*

Данное определение согласуется как с симметричными так и асимметричными ключевыми системами – в качестве функции $\mathtt{mac}$ могут быть использованы как алгоритмы выработки имитовставки, так и алгоритмы выработки электронной подписи. Для общности изложения далее будем считать, что $\mathtt{mac}$ есть отображение с конечным числом аргументов, существенно зависящее от ключа аутентификации K_a, идентификатора субъекта B и случайного значения $\xi_1(t_1)$, а также, удовлетворяющее требованиям, предъявляемым к ключевым криптографическим функциям хеширования, см. [383] и определение 3.2.

[11] Напомним, что символом $\in_R$ обозначается выбор случайного элемента из заданного множества.

Начальное множество состояний субъекта A, выполняющего процесс аутентификации субъекта B, может быть определено следующим образом

$$A(t_0) = \{a_1(t_0) = (k_c, \texttt{true}),\ a_2(t_0) = (ID_B, \texttt{false}),\ a_3 = (\emptyset, \texttt{false})\},$$

где ячейка a_1 соответствует ключу проверки кода аутентификации субъекта B, ячейка a_2 – идентификатору субъекта B, а ячейка a_3 – вырабатываемому субъектом A случайному значению, используемому при проверке кода аутентификации.

В следующие моменты времени состояние субъекта A имеет вид:

$$\begin{aligned} A(t_1) &= \{a_1(t_1) = (k_c, \texttt{true}), a_2(t_1) = (ID_B, \texttt{false}), a_3(t_1) = (\xi_1(t_1), \texttt{true})\}, \\ A(t_2) &= \{a_1(t_2) = (k_c, \texttt{true}), a_2(t_2) = (ID_B, \sigma(t_2)), a_3(t_2) = (\xi_1(t_1), \texttt{true})\}, \end{aligned}$$

где

$$\sigma(t_2) = \texttt{conf}(K_c, ID_B || \xi_1(t_1), \xi_1(t_2)),$$

или может быть описано следующими нетривиальными функциями перехода

$$\alpha_3(t_1) = \xi_1(t_1),\ \sigma_3(t_1) = \texttt{true},\ \sigma_2(t_2) = \texttt{conf}(\alpha_1(t_2), \alpha_2(t_2) || \alpha_3(t_2), \xi_1(t_2)).$$

Сделаем несколько замечаний к определению свойства аутентификации субъекта взаимодействия.

1) В рассмотренном выше протоколе, см. рисунок 4.18, ключ проверки кода аутентификации k_c является исходной ключевой информацией для субъекта A. Поскольку именно эта информация обеспечивает аутентификацию субъекта B, то ее значение должно быть подтверждено до начала выполнения протокола, например, с помощью организационно-технических мер, с помощью удостоверяющего центра или в рамках другого протокола.

2) Из определения следует, что свойство аутентификации субъекта выполнено только для протоколов, включающих в себя взаимодействие субъектов (отправку и получение сообщений). Транспортные протоколы, предусматривающие только отправку сообщений от одного субъекта к другому субъекту, данному свойству не удовлетворяют. Вместе с тем, использование в таких протоколах ключевой информации, владелец которой аутентифицирован ранее иным способом, позволяет говорить о наследовании транспортным протоколом свойства С1.

3) Включение идентификатора ID_B в состав сообщения

$$\xi_1(t_2) = \mathtt{mac}(K_a, ID_B||\xi_1(t_1))$$

является принципиальным при использовании симметричной ключевой системы. В случае исключения идентификатора ID_B не представляется возможным предъявить алгоритмический способ различения того, кто же из субъектов взаимодействия является автором пары сообщений $\xi_1(t_1)$, $\mathtt{mac}(K_a, \xi_1(t_1))$ (это следует из совпадения ключей аутентификации у обоих субъектов). Для асимметричной ключевой системы исключение идентификатора ID_B не является критичным, поскольку субъекты имеют различные ключи аутентификации.

В завершение необходимо отметить, что на настоящий момент времени в Российской Федерации действует только морально устаревший стандарт ГОСТ Р ИСО/МЭК 9594-8-98, см. [289], регламентирующий процедуры аутентификации с использованием фактора знания секретного ключа. При этом, сформулированному выше определению 4.8 соответствует лишь часть процедур «строгой» аутентификации из [289], см. раздел 10. Более современные стандарты серии ISO/IEC 9798 (части 1–6), в большинстве своем, соответствуют определению 4.8. Протоколы из ISO/IEC 9798-5:2009 и ISO/IEC 9798-6:2010 могут, формально, не соответствовать рассматриваемому определению и, в случае необходимости их применения на территории Российской Федерации, должны пройти дополнительное исследование на соответствие рассматриваемой модели.

§ 4.4.2.2. Свойство целостности сообщений

В большинстве не криптографических протоколов для защиты от случайных искажений данные передаются вместе со своими кодами целостности, выработанными с помощью сжимающих отображений таких, как Fletcher16 [92], CRC32 [193] и т.п., а также бесключевых криптографических функций хэширования, например, функции «Стрибог», см. [281]. Используемая нами модель нарушителя делает применение таких функций бесполезным для защиты от преднамеренных искажений, а для обеспечения целостности, также как и в разделе 4.4.2.1, приходится использовать сжимающие преобразования, зависящие от секретного ключа.

Определение 4.9. *Будем говорить, что криптографический протокол обеспечивает свойство целостности сообщения $M \in \mathbb{V}^*_\infty$, отправляемого субъектом B субъекту A (свойство С3), если*

1) *для субъекта B определены – ключ аутентификации $k_a \in \mathbb{K}_a$ и ключ проверки кода аутентификации $k_c \in \mathbb{K}_c$,*

2) *субъекту A известно подтвержденное значение ключа проверки кода аутентификации субъекта k_c,*

3) *для некоторого натурального m определены функции выработки кода аутентификации $\mathtt{mac} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ и проверки кода аутентификации $\mathtt{conf} : \mathbb{K}_c \times \mathbb{V}_\infty^* \times \mathbb{V}_m \to \mathbb{B}$ такие, что*

$$\mathtt{conf}\,(k_c, M, \mathtt{mac}(k_a, M)) = \mathtt{true},$$

для любой тройки значений $k_a, k_c \in \mathbb{K}$, $M \in \mathbb{V}_\infty^*$,

4) *в состав протокола входит следующая последовательность шагов:*

	субъект A		субъект B
			$\xi_1(t_1) = M \in \mathbb{V}_\infty^*$,
t_1:			$\xi_2(t_1) = \mathtt{mac}(k_a, M)$
		$\longleftarrow$	
t_2:	$\mathtt{conf}(k_c, \xi_1(t_1), \xi_2(t_1)) =^?\ \mathtt{true}$,		

Рис. 4.19: Протокол подтверждения целостности сообщения.

5) *истинно условие проверки на шаге t_2.*

Начальное множество состояний субъекта A, выполняющего процесс получения сообщений от субъекта B, может быть определено следующим образом

$$A(t_0) = \{a_1(t_0) = (k_c, \mathtt{true}),\ a_2(t_0) = (\emptyset, \mathtt{false}),\ a_3 = (\emptyset, \mathtt{false})\},$$

где ячейка a_1 соответствует ключу проверки кода аутентификации субъекта B, ячейка a_2 – получаемому из канала связи сообщению, а ячейка a_3 – коду целостности получаемого сообщения. В следующий момент времени состояние субъекта A имеет вид

$$A(t_1) = \{a_1(t_1) = (k_c, \mathtt{true}),\ a_2(t_1) = (\xi_1(t), \sigma(t_1)),\ a_3(t_1) = (\xi_2(t_1), \sigma)\},$$

где

$$\sigma(t_1) = \mathtt{conf}(K_c, \xi_1(t_1), \xi_2(t_1)).$$

Как и в случае свойства аутентификации субъекта, выполнение свойства целостности существенным образом зависит от того, подтверждена ли исходная ключевая информация – ключ проверки кода целостности k_c.

§ 4.4.2.3. Свойство аутентификации сообщения

Свойство аутентификации сообщения (свойство C2) является производным и следует из комбинации свойств аутентификации субъекта (см. свойство C1, раздел 4.4.2.1) и целостности передаваемых сообщений (см. свойство C3, раздел 4.4.2.2).

Определение 4.10. *Криптографический протокол обеспечивает выполнение свойство аутентификации субъектом A сообщения $M \in \mathbb{V}_\infty^*$, отправленного субъектом B (свойство C2), если в условиях определений 4.8 и 4.9*

1) протокол содержит последовательность шагов, изображенную на рисунке 4.19,

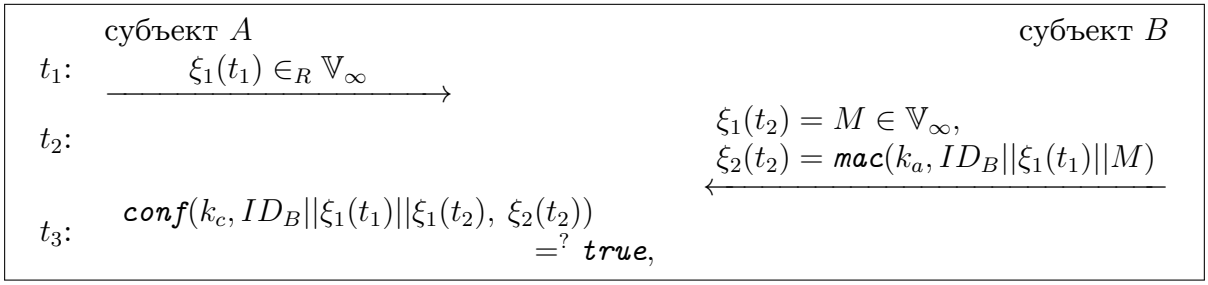


Рис. 4.20: Протокол аутентификации сообщения.

2) истинно условие проверки на шаге t_3.

В данном определении совмещены процедуры проверки кода аутентификации, выполняемые для проверки свойства аутентификации субъекта B и свойства целостности сообщения M.

Следует отметить, что для выполнения свойства аутентификации нескольких отправляемых субъектом B сообщений (с целью снижения объема передаваемой в канал связи информации) сообщение $\xi_1(t_1)$ может отправляться единожды, а значение $ID_B||\xi_1(t_1)$ заменяться на результат применения некоторого сжимающего отображения, известного обоим субъектам A и B.

§ 4.4.2.4. Свойство защиты от навязывания параметров безопасности

Свойство защиты от навязывания параметров безопасности (свойство C15) также является производным и основано на выполнении следующих свойств:

- аутентификации субъекта (свойство C1),
- целостности сообщений (свойство C3) и, как следствие,
- аутентификации сообщения (свойство C2).

Достаточно часто в рамках одной сессии протокола между субъектами взаимодействия осуществляется согласование криптографических параметров, прямо или косвенно влияющих на безопасность передаваемых данных. В качестве таких параметров могут выступать алгоритмы блочного шифрования, функции хеширования, параметры циклической абелевой группы, в которых производится выработка производных ключей и т.п. Вмешательство нарушителя в процесс согласования параметров может привести к компрометации вырабатываемой в ходе выполнения протокола ключевой информации. Наиболее ярким примером являются атаки на понижение версии протокола TLS 1.2, см. [6].

Определение 4.11. *Криптографический протокол обеспечивает выполнение свойства защиты от навязывания параметров безопасности $P \in \mathbb{V}_\infty$ субъекту A (свойство C15), если сообщение, в котором передаются параметры безопасности для субъекта A удовлетворяет свойству аутентификации сообщения (свойство C2).*

На практике согласование параметров безопасности выполняется в режиме «запрос-ответ», т.е. субъект A формирует перечень доступных для него параметров, а субъект B выбирает из них те, что будут использоваться в дальнейшем. Интерпретируя параметры безопасности как строки из $\mathbb{V}_\infty$ получаем, что субъект A направляет субъекту B множество строк $\{S_1, \ldots, S_r\}$, а субъект возвращает одну строку S_i, $i \in \{1, \ldots, r\}$, из полученного множества. Тогда будем считать, что протокол, обеспечивающий свойство защиты от навязывания параметров безопасности, должен содержать следующую последовательность шагов:

В данном протоколе используется определяемая, как и ранее, исходная ключевая информация k_a, k_c. Субъект A сначала проверяет, что выбранный субъектом B параметр безопасности $\xi_1(t_2)$ принадлежит запрашиваемому множеству, а после проверяет истинность кода аутентификации сообщения, содержащего $\xi_1(t_2)$. Отметим, что функции `mac` и `conf` должны быть зафиксированы до начала выполнения протокола, изображенного на рисунке 4.21.

Примером практической реализации описанного механизма может служить этап установления соединения в протоколе TLS 1.3, см. [212, 367], в котором все критически важные параметры передаются при помощи сообщения, подписанного электронной подписью субъекта B.

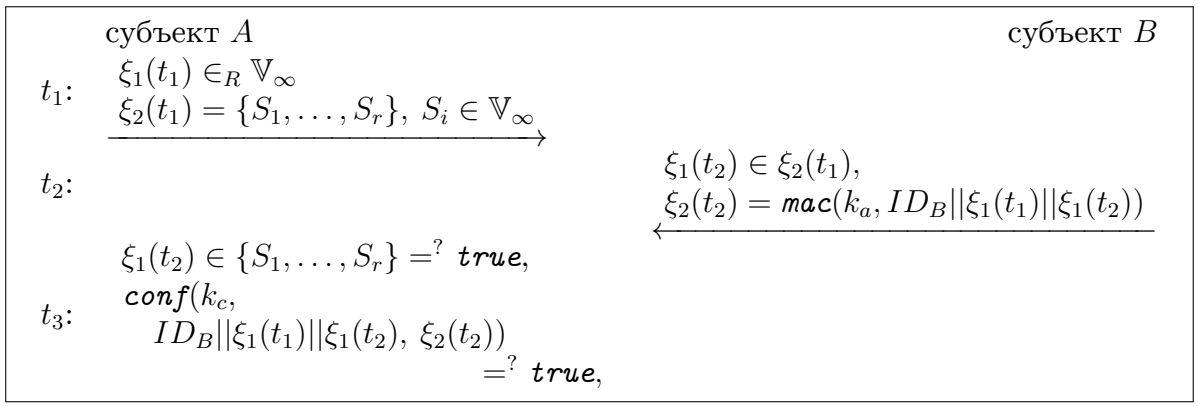


Рис. 4.21: Протокол защиты от навязывания параметров безопасности.

§ 4.4.2.5. Свойства подтверждения и аутентификации ключа

Как мы видели ранее, для обеспечения свойств безопасности необходимо использование подтвержденной исходной ключевой информации k_c. В случае использования асимметричной ключевой системы, подтверждение ключа k_c производится путем проверки электронной подписи удостоверяющего центра, выдавшего сертификат ключа k_c. В случае симметричной ключевой системы подтверждение обеспечивается организационно-штатными мерами доставки ключевой информации до субъектов взаимодействия.

Для производной ключевой информации, вырабатываемой в рамках криптографического протокола, требуется обеспечить подтверждение непосредственно в ходе взаимодействия субъектов.

Пусть $k_A \in \mathbb{K}_a$ значение, выработанное субъектом A, а $k_B \in \mathbb{K}_a$ значение, выработанное субъектом B. Теперь субъект A должен получить подтверждение того, что значение k_A совпадает со значением k_B, т.е. проверить, что для некоторой функции f выполнено равенство

$$\mathit{f}(k_A, M) =^? \mathit{f}(k_B, M), \quad M \in \mathbb{V}_\infty,$$

в котором правая часть вычислена субъектом B, а левая часть – субъектом A. В качестве функции f может выступать, например, режим блочного шифрования или алгоритм выработки имитовставки.

Наиболее простой протокол, реализующий подтверждение субъектом A ключа K_A, изображен на рисунке 4.22.

В данном протоколе субъект B, с использованием подтверждаемого значения k_B, преобразует случайное сообщение $\xi_1(t_1)$, а субъект A проверяет корректность результата преобразования с использованием подтверждаемого значения k_A. Сделаем ряд замечаний к изображенному на рисунке 4.22 протоколу.

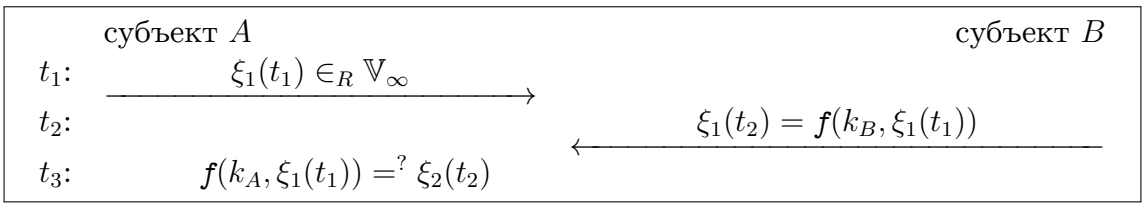


Рис. 4.22: Простой протокол подтверждения ключа.

- Легко видеть, что сообщение $\xi_1(t_2)$ не удовлетворяет свойству аутентификации сообщений. Субъект A получает подтверждение, что он вычислил значение k_A правильно, но того субъекта, что ему это подтверждение направил, субъект A идентифицировать не может. Тем самым, у нарушителя появляется потенциальная возможность навязать субъекту A ложное значение ключа k_A.

- Другим недостатком предложенного протокола является возможность накопления нарушителем пар открытый/шифрованный текст $\xi_1(t_1), \boldsymbol{f}(k_B, \xi_1(t_1))$ и их использование, в дальнейшем, либо для реализации алгоритмических методов определения секретного значения k_B, либо для последующего навязывания ложных, но корректно зашифрованных значений. Это замечание приводит к необходимости использовать на этапе подтверждения ключа преобразование $\boldsymbol{f}$, отличное от того, что будет использовано в дальнейшем при взаимодействии субъектов.

Для исправления второго замечания можно модифицировать протокол, изображенный на рисунке 4.22, и обмениваться только зашифрованными с помощью преобразования $\boldsymbol{f}$ сообщениями. Пример такой модификации изображен на рисунке 4.23.

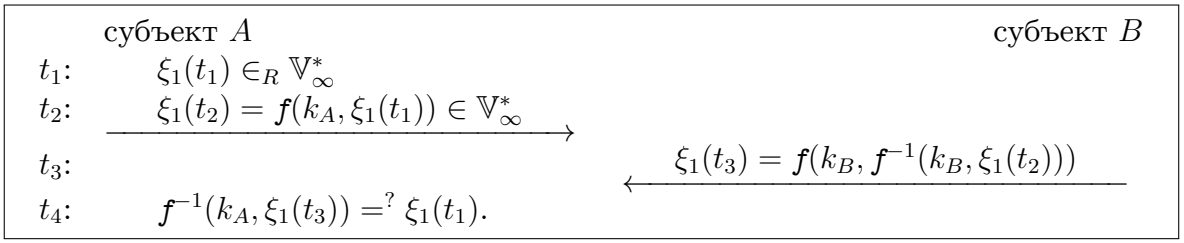


Рис. 4.23: Уязвимый протокол подтверждения ключа.

В данном протоколе нарушитель не может накапливать пары открытый/шифрованный текст $\xi_1(t_1), \boldsymbol{f}(k_A, \xi_1(t_1))$, поскольку значение $\xi_1(t_1)$ не передается субъектом A в канал связи. Также предполагается, что субъект B с помощью подтверждаемого ключа K_B сначала расшифровывает

случайное сообщение $\xi_1(t_1)$ а потом повторно зашифровывает его (такую последовательность действий субъекта B можно назвать «расшифрование и зашифрование»).

Однако протокол уязвим, поскольку нарушитель может реализовать атаку «отражением сообщений» – перехватить сообщение

$$\xi_1(t_2) = f(k_A, \xi_1(t_1))$$

и направить его обратно субъекту A вместо сообщения $\xi_1(t_3)$. После этого условие проверки на шаге t_4 всегда будет истинно, поскольку $f^{-1}(k_A, f(k_A, \xi_1(t_1))) = \xi_1(t_1)$.

Реализация атаки «отражением сообщений» становится возможной в силу следующих причин.

1) Построим формальную модель состояний субъекта A. Начальное множество состояний может быть определено следующим образом

$$A(t_0) = \{a_1(t_0) = (k_A, \mathtt{false}),\ a_2(t_0) = (\emptyset, \mathtt{false})\},$$

где ячейка a_1 соответствует подтверждаемому ключу k_A, а ячейка a_2 – вырабатываемому случайному сообщению. Тогда последовательность состояний субъекта A описывается следующими нетривиальными функциями

$$\alpha_2(t_1) = \xi_1(t_1),\ \sigma_2(t_1) = \mathtt{true},\ \sigma_1(t_4) = \big(f(\alpha_1(t_3), \xi_1(t_3)) =^{?} \alpha_2(t_3)\big).$$

Легко видеть, что содержимое ячейки a_1 подтверждается значением функции $f(\alpha_1(t_3), \xi_1(t_3)) =^{?} \alpha_2(t_3)$, не зависящим от какой-либо исходной ключевой информации.

2) С точки зрения субъекта A значение $\xi_1(t_3)$ рассматривается как реализация некоторой случайной величины. При этом ожидается, что вероятность угадывания нарушителем случайного значения $\xi_1(t_3)$ такого, что $\sigma_1(t_4)$ примет истинное значение, будет минимальной. Вместе с тем, значение $\xi_1(t_3) = \xi_1(t_2)$ и передается субъектом A в открытом виде. Это позволяет нарушителю перехватить его, отправить обратно субъекту A и с вероятностью единица быть уверенным в том, что $\sigma_1(t_4) = \mathtt{true}$.

Протокол 4.23 иллюстрирует сделанное нами ранее высказывание (см. примечание к определению 4.7 на стр. 319) о том, что подтверждение всех ячеек состояния субъекта является лишь необходимым условием безопасности протокола. Дополнительно должны рассматриваться вероятности

подделки поступающих из канала связи значений, а также, в общем случае, и трудоемкости алгоритмов подделки.

Защитой от атаки «отражением сообщений» является применение некоторого известного обоим субъектам A и B преобразования $\boldsymbol{h}$ к неизвестному для нарушителя сообщению $\xi_1(t_1)$, т.е. вычисление равенства

$$\xi_1(t_3) = \boldsymbol{f}(k_B, \boldsymbol{h}(\boldsymbol{f}^{-1}(k_B, \xi_1(t_2))))$$

(такую последовательность действий субъекта B можно назвать «расшифрование, преобразование и зашифрование»).

Если преобразование $\boldsymbol{h}$ отлично от преобразования $\boldsymbol{f}$, является однонаправленным для нарушителя и не позволяет ему по значению $\xi_1(t_3) = \boldsymbol{h}(x)$ определить значение аргумента x, то повторное применение преобразования $\boldsymbol{f}$ представляется излишним. В качестве однонаправленного преобразования $\boldsymbol{h}$ может выступать, например, бесключевая функция хеширования.

Определение 4.12. *Будем говорить, что криптографический протокол обеспечивает для субъекта A свойство подтверждения факта обладания субъектом B ключа $k_B \in \mathbb{K}_a$ (свойство С10), если:*

1) *субъект A обладает ключом $k_A \in \mathbb{K}_a$, для которого подтверждается выполнение равенства $k_A = k_B$,*

2) *определена зависящая от секретного ключа функция*

$$\boldsymbol{f} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$$

для которой определена обратная функция $\boldsymbol{f}^{-1} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$ такая, что для любого сообщения $M \in \mathbb{V}_\infty$ равенство

$$\boldsymbol{f}^{-1}(k_A, \boldsymbol{f}(k_B, M)) = M$$

справедливо, когда $k_A, k_B \in \mathbb{K}_a$ и $k_A = k_B$,

3) *задана однонаправленная функция $\boldsymbol{h} : \mathbb{V}_\infty \to \mathbb{V}_m$, определенная для некоторого $m \in \mathbb{N}$,*

4) *в состав протокола входит следующая последовательность шагов:*

5) *истинно условие проверки на шаге t_4.*

Следует отметить, что условие существования (эффективно вычислимого субъектом B) обратного преобразования $\boldsymbol{f}^{-1}$ является необходимым, так как в противном случае вычисление сообщения $\xi_1(t_1) = f^{-1}(k_B, \xi_1(t_2))$ невозможно.

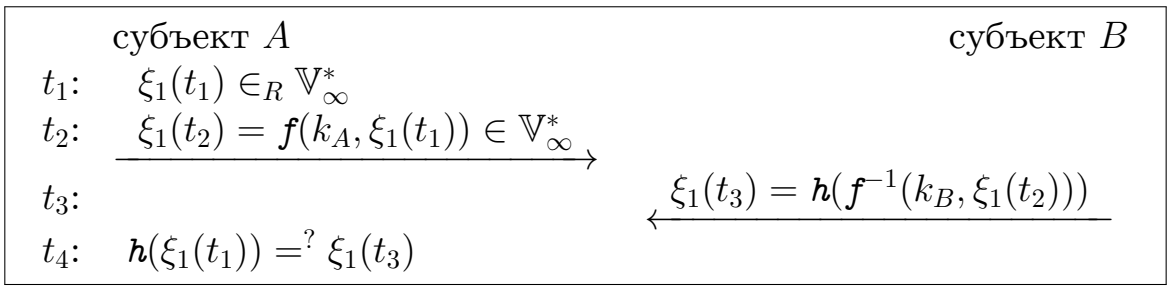


Рис. 4.24: Протокол подтверждения ключа.

Используя в качестве преобразования $\boldsymbol{h}$ функцию вычисления кода аутентификации $\boldsymbol{mac}$ можно добиться зависимости подтверждения ключа от исходной ключевой информации. В этом случае необходимо использовать равенство

$$\xi_1(t_3) = \boldsymbol{mac}(k_a, \boldsymbol{f}^{-1}(k_B, \xi_1(t_2)))).$$

Учитывая также, что сообщение $\xi_1(t_3)$ должно удовлетворять свойству аутентификации сообщений (свойство С2), дадим еще одно определение.

Определение 4.13. *Будем говорить, что криптографический протокол обеспечивает для субъекта A свойство аутентификации принадлежащего субъекту B ключа $k_B \in \mathbb{K}_a$ (свойство С9), если:*

1) *с субъектом B связан идентификатор ID_B,*

2) *для субъекта B определены – ключ аутентификации $k_a \in \mathbb{K}_a$ и ключ проверки кода аутентификации $k_c \in \mathbb{K}_c$, однозначно связанные с идентификатором ID_B,*

3) *субъекту A известны идентификатор ID_B субъекта B и подтвержденное значение ключа проверки кода аутентификации k_c,*

4) *субъект A обладает ключом $k_A \in \mathbb{K}_a$, для которого подтверждается выполнение равенства $k_A = k_B$,*

5) *для некоторого натурального m определены функции выработки кода аутентификации $\boldsymbol{mac} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ и проверки кода аутентификации $\boldsymbol{conf} : \mathbb{K}_c \times \mathbb{V}_\infty^* \times \mathbb{V}_m \to \mathbb{B}$ такие, что*

$$\boldsymbol{conf}(k_c, M, \boldsymbol{mac}(k_a, M)) = \boldsymbol{true},$$

для любой тройки значений k_a, k_c и $M \in \mathbb{V}_\infty^*$,

6) *определена зависящая от секретного ключа функция $\boldsymbol{f} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$ для которой определена обратная функция $\boldsymbol{f}^{-1} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$*

такая, что для любого сообщения $M \in \mathbb{V}_\infty$ *равенство*

$$\boldsymbol{f}^{-1}(k_A, \boldsymbol{f}(k_B, M)) = M$$

справедливо, когда $k_A, k_B \in \mathbb{K}_a$ *и* $k_A = k_B$,

7) в состав протокола входит последовательность шагов, изображенная на рисунке 4.24,

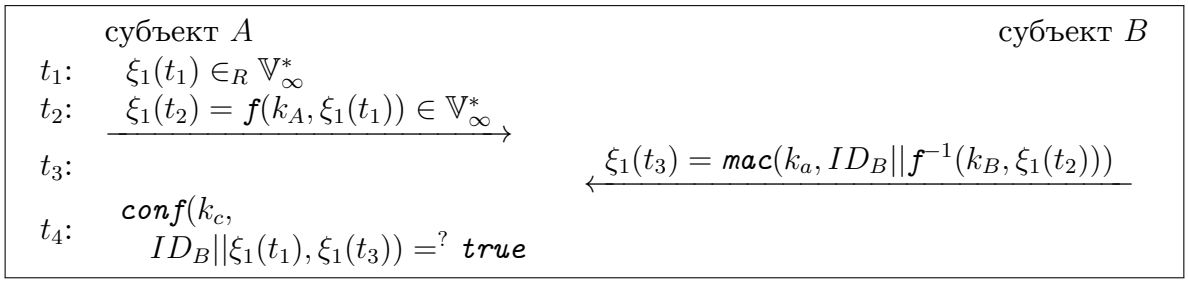


Рис. 4.25: Протокол аутентификации ключа.

8) истинно условие проверки на шаге t_4.

Отметим, что формально следуя определению 4.10 можно было бы определить множество отправляемых субъектом B сообщений следующим образом

$$\begin{array}{l} \xi_1(t_3) = \boldsymbol{mac}(k_B, \boldsymbol{f}^{-1}(k_B, \xi_1(t_2))), \\ \xi_2(t_3) = \boldsymbol{mac}(k_a, ID_B||\xi_1(t_2)||\xi_1(t_3)). \end{array} \longleftarrow$$

Вместе с тем, предложенный на рисунке 4.25 вариант достигает той же цели с меньшей длиной данных, передаваемых в канал связи.

Также отметим, что если функции $\boldsymbol{f}$, $\boldsymbol{mac}$ и $\boldsymbol{conf}$ согласуются в ходе выполнения протокола, то для протокола должно быть выполнено свойство защиты от навязывания параметров (свойство С15).

§ 4.4.2.6. Свойство конфиденциальности ключа

Свойство конфиденциальности ключа является базовым и применяется как к исходной ключевой информации, так и к сессионным (производным) ключам, вырабатываемым в ходе выполнения протокола.

Определение 4.14. *Пусть заданы* $k_a \in \mathbb{K}_a$ *– исходная ключевая информация,* $\boldsymbol{h} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ *некоторая однонаправленная функция и* $\{\xi_{i_k}(t_k), \boldsymbol{h}(k_a, \xi_{i_k}(t_k))\}$ *множество пар, перехваченных нарушителем в ходе выполнения одной или нескольких сессий протокола.*

Будем говорить, что протокол обеспечивает конфиденциальность исходной ключевой информации (свойство С8), если нарушитель не может определить значение k_a с вероятностью большей, чем величина π_0 (см. определение 4.3) и трудоемкостью, не превосходящей некоторое значение Q_0.

Легко видеть, что данное определение может быть расширено на случай нескольких однонаправленых функций $h_1, h_2, \ldots$ и т.д.

Отличие производной ключевой информации от исходной заключается в том, что она вырабатывается в ходе выполнения протокола. В случае, когда выработка происходит с использованием значений, передаваемых между субъектами взаимодействия, протокол должен реализовывать механизмы защиты передаваемых значений от подделки и навязывания нарушителем.

Определение 4.15. *Пусть заданы $k_a \in \mathbb{K}_a$ – исходная ключевая информация, отображение*

$$\boldsymbol{var}\colon \mathbb{K}_a \times \mathbb{V}_\infty^* \times \cdots \times \mathbb{V}_\infty^* \to \mathbb{K}_a,$$

используемое для выработки производной ключевой информации, и однонаправленная функция

$$\boldsymbol{h}\colon \mathbb{K} \times \mathbb{V}_\infty^* \to \mathbb{V}_m.$$

Будем говорить, что протокол обеспечивает конфиденциальность производной ключевой информации k_A (свойство С8), если выполнены следующие условия.

1) *Величина k_A, в общем виде, определяется равенством*

$$k_A = \boldsymbol{var}(K_a, \xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}}),\ \beta_{j_1}, \ldots, \beta_{j_r}), \tag{4.15}$$

   где величины $\beta_{j_1}, \ldots, \beta_{j_r}$ определены равенствами

$$\xi_{j_s}(t_{k_{j_s}}) = \boldsymbol{h}(\beta_{j_s}), \quad s = 1, \ldots, r,$$

   а величины $\xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}})$ и $\xi_{j_1}(t_{k_{j_1}}), \ldots, \xi_{j_r}(t_{k_{j_r}})$ передаются между субъектами взаимодействия в ходе выполнения протокола (данные величины могут перехватываться нарушителем).

2) *В равенстве (4.15) либо переменные $k_a, \xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}})$, либо переменные $\beta_{j_1}, \ldots, \beta_{j_r}$ могут являться несущественными.*

3) *Если переменные $\xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}})$ являются существенными, то они должны передаваться в составе сообщений, для которых выполнено свойство аутентификации сообщений (свойство С2, см. раздел 4.4.2.3).*

4) Если переменные $\beta_{j_1}, \ldots, \beta_{j_r}$ являются существенными, то они не могут быть определены нарушителем с вероятностью большей, чем величина π_0 и трудоемкостью, не превосходящей некоторое значение Q_0.

5) После выработки производной ключевой информации k_A она должна использоваться таким образом, чтобы удовлетворять определению 4.14.

§ 4.4.2.7. Свойство конфиденциальности

Согласно ГОСТ Р ИСО/МЭК 27033-1:2011, см. [288], угроза нарушения конфиденциальности передаваемой информации является одной из основных угроз при обеспечении безопасности сетей связи. Однако свойство конфиденциальности (свойство С16) не является базовым и выполняется только при условии выполнения совокупности рассмотренных ранее свойств.

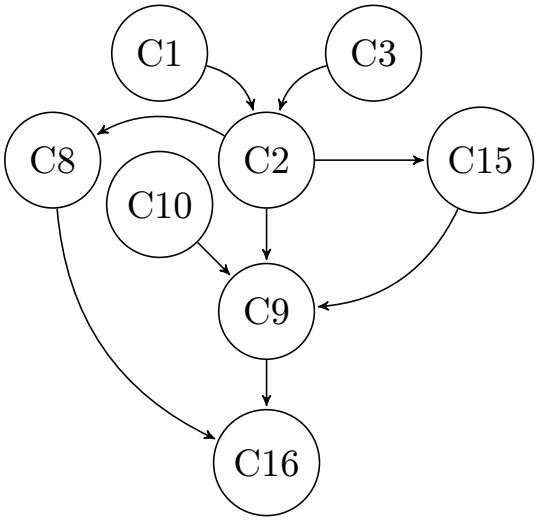


Рис. 4.26: Схема зависимостей свойства конфиденциальности.

Во-первых, используемый для шифрования информации ключ, являющийся, как правило, производной ключевой информацией, должен быть неизвестен нарушителю, т.е. удовлетворять свойству конфиденциальности ключа (свойство С8).

Во-вторых, этот ключ должен удовлетворять свойству аутентификации ключа (свойство С9). Это позволит получающему сообщения субъекту быть уверенным в том, что он не только получает сообщения зашифрованные на том самом ключе, который используется для их расшифрования, но и в том, что отправителем этих сообщений является аутентифицированный субъект взаимодействия.

В-третьих, для зашифрования информации передаваемой в информационных системах, попадающих под действие нормативного регулирования, допускается применять только алгоритмы, входящие в национальную систему стандартизации Российской Федерации.

Схема зависимостей свойства конфиденциальности приведена выше на рисунке 4.26.

В заключение этого раздела рассмотрим ряд свойств, нарушение которых может привести к появлению негативных эффектов, не предполагаемых спецификацией криптографического протокола.

§ 4.4.2.8. Свойство целостности множества состояний

С целью защиты от атак, использующих для компрометации одной сессии протокола данные, перехваченные в ходе выполнения другой сессии, рассмотрим свойство целостности множества состояний (свойство С27). Пусть

$$A(t_0) = \{a_1, \ldots, a_{n(A)} : \; a_i = \big(\alpha_i(t_0), \sigma_i(t_k)\big)\}$$

начальное состояние субъекта А. Будем считать, что для некоторого натурального $n_1(A)$ такого, что $1 \leqslant n_1(A) < n(A)$, ячейки $a_1, \ldots, a_{n_1(A)}$ содержат исходную ключевую информацию, а также любые другие значения, подтвержденные до начала выполнения протокола, т.е.

$$\sigma_i(t_0) = \texttt{true}, \quad i = 1, \ldots, n_1(A).$$

Также будем считать, что для некоторого натурального $n_2(A)$ такого, что $n_1(A) < n_2(A) < n(A)$, ячейки $a_{n_1(A)+1}, \ldots, a_{n_2(A)}$ содержат случайные значения, вырабатываемые субъектом A с использованием датчика случайных чисел (ДСЧ), т.е. найдутся такие временные метки t_{k_i} такие, что

$$\sigma_i(t_{k_i}) = \texttt{true}, \quad i = n_1(A) + 1, \ldots, n_2(A).$$

Определение 4.16. *Будем говорить, что криптографический протокол удовлетворяет свойству целостности множества состояний (свойство С27) для субъекта А, если найдется такая временная метка t_{k_0}, что для всех $t_k \geqslant t_{k_0}$ будет выполнено*

$$\sigma_i(t_k) = \texttt{conf}(a_1(t_{k-1}), \ldots, a_{n_2(A)}(t_{k-1}), \ldots), \quad i = n_2(A)+1, \ldots, n(A), \quad (4.16)$$

и зависимость от $a_1(t_{k-1})$, …, $a_{n_2(A)}(t_{k-1})$ является существенной.

Будем говорить, что криптографический протокол удовлетворяет строгому свойству целостности множества состояний для субъекта А, если одновременно с условием (4.16) выполнено

$$\alpha_i(t_k) = \texttt{var}(\ldots, \alpha_{n_1(A)+1}(t_{k-1}), \ldots, \alpha_{n_2(A)}(t_{k-1}), \ldots), \quad i = n_2(A)+1, \ldots, n(A). \quad (4.17)$$

и зависимость от $\alpha_{n_1(A)+1}(t_{k-1})$, …, $\alpha_{n_2(A)}(t_{k-1})$ является существенной.

Если криптографический протокол удовлетворяет данному определению, то субъект A может удостовериться в том, что каждая из ячеек памяти его состояния, вырабатываемая в ходе выполнения протокола, подтверждается с использованием значений, которые не могут быть навязаны нарушителем.

Существенная зависимость от значений $\alpha_{n_1(A)+1}, \ldots, \alpha_{n_2(A)}$, вырабатываемых с использованием ДСЧ, позволяет говорить о том, что данные значения выработаны непосредственно в ходе выполнения протокола, т.е. в реальном времени, и не могут быть продублированы в ходе выполнения другой сессии протокола. Невозможность дублирования случайных значений должна обеспечиваться используемым датчиком случайных чисел.

§ 4.4.2.9. Свойство защищенности от KCI-атак

Рассмотрим свойство защищенности от KCI-атак (свойство С22). Данные атаки реализуются в случае компрометации исходной ключевой информации (долговременного ключа) одного из легальных субъектов, или в случае определения нарушителем исходной ключевой информации (нарушения свойства С8, см. раздел 4.4.2.6).

В качестве примера такой атаки рассмотрим протокол MTI(C0), см. [48], и приведем пример построения KCI-атаки для данного протокола.

Пусть q – нечетное простое число, $G = \langle g \rangle$ – циклическая абелева группа, порождаемая элементом g порядка q. Будем считать, что в группе G решение задачи дискретного логарифмирования имеет высокую трудоемкость.

Субъекты A и B обладают парами асимметричных ключей (закрытый и открытый), соответственно, $k_a \in \mathbb{F}_q^*$, $k_{cA} = g^{k_a}, k_{cA} \in G$ и $k_b \in \mathbb{F}_q^*$, $k_{cB} = g^{k_b}, k_{cB} \in G$. Будем считать, что открытые ключи k_{cA}, k_{cB} известны обоим субъектам взаимодействия, а их значения подтверждены до начала выполнения протокола. Схема работы протокола MTI(C0) представлена на рисунке 4.27.

Рассмотрим реализацию KCI-атаки на протокол MTI(C0) в рамках предположения, что нарушитель C знает закрытый ключ k_a участника A, также нарушителю известны открытые ключи субъектов k_{cA} и k_{cB} Нарушитель пытается выдать себя за субъекта B перед субъектом A.

При реализации рассмотренной атаки нарушитель может сформировать сообщение $\xi_1(t_4)$ от лица субъекта B (при условии знания ключа k_a) таким образом, что субъект A ничего не заподозрит и будет думать, что ключ выработан с субъектом B, а на самом деле он будет выработан с нарушителем.

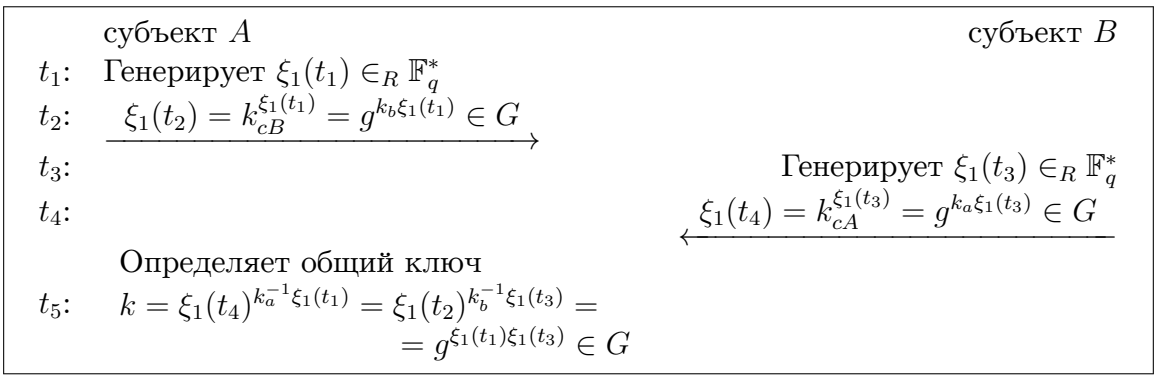


Рис. 4.27: Протокол MTI(C0).

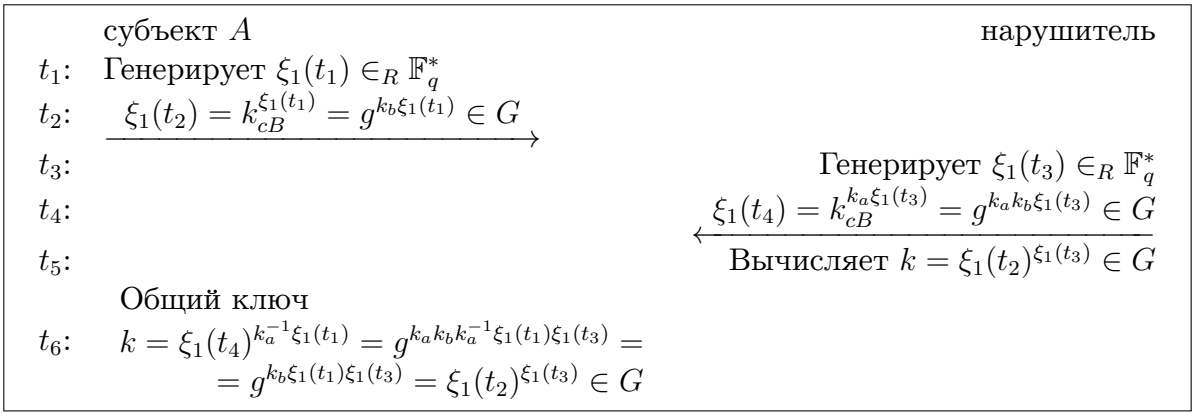


Рис. 4.28: KCI-атака на протокол MTI(C0).

В качестве основных методов противодействия KCI-атакам можно выделить следующие:

- уникальность сессионных ключей – ключи для каждой сессии протокола должны генерироваться независимо;
- запрет на использование ключей аутентификации при формировании производных ключей, см., например, равенство (4.17);
- обязательная аутентификация субъекта, с которым производится взаимодействие, а также аутентификация выработанного сессионного ключа.

Поскольку KCI-атаки нацелены на эксплуатацию недостатков ключевой системы протокола и механизмов аутентификации, то можно говорить о защищенности протокола от данного класса атак только в случае выполнения следующих свойств: C1 (свойство аутентификации субъекта), C9 (свойство аутентификация ключа), C10 (свойство подтверждения

ключа), C12 (свойство стойкости при компрометации ключа аутентификации), C13 (свойство формирования новых ключей) и C14 (свойство защиты от навязывания ключевых значений).

§ 4.4.3. Определение показателей эффективности защиты информации

В настоящее время в Российской Федерации принято оценивать стойкость средств криптографической защиты информации с точки зрения «практической стойкости», т.е. путем оценки трудоемкости известных аналитику методов решения математических задач, решение которых приводит к компрометации используемых криптографических преобразований и алгоритмов. Изложение принятой методологии оценки стойкости, с разной долей детализации, может быть найдено в работах [270, 316, 345, 347]. При этом, минимальная трудоемкость и вероятность успешного решения рассматриваемых задач выступают в качестве показателей эффективности защиты. Представляется естественным распространить эти же показатели и на анализ криптографических протоколов.

В соответствии с определенной выше моделью нарушителя, оценка возможности компрометации криптографического протокола может осуществляться помощи следующих подходов.

1) При помощи «пассивных» атак, т.е перехвата, накопления и последующего анализа перехваченной информации. В рамках применяемой в работе модели нарушителя криптографического протокола такие атаки сводятся к обращению однонаправленных функций, т.е. к решению сложных математических задач. Для каждой из таких задач рассматривается некоторое множество алгоритмов, находящих решение задачи с вероятностью π и трудоемкостью Q. Отбрасывая алгоритмы с ничтожной вероятностью успеха, меньшей чем некоторое заранее фиксированное значение π_0, мы можем выбрать алгоритм с наименьшей трудоемкостью. Именно такой алгоритм и считается наилучшим алгоритмом компрометации криптографического протокола при проведении «пассивных» атак.

2) При помощи «активных» атак, сводящихся к навязыванию одному или нескольким субъектам ложных значений, поступающих из канала связи; цель такого навязывания состоит в том, что бы заставить легитимного субъекта сделать ложный вывод о том, что значения одной или нескольких ячеек его памяти являются истинными (подтвержденными).

Навязываемые значения могут вычисляться нарушителем как случайным образом, так и с использованием методов, применяемых при реализации «пассивных» атак. В первом случае мы считаем, что трудоемкость выработки навязываемых значений ничтожна и основную роль при анализе играет вероятность π принять ложное значение за истинное. При этом мы считаем, что число попыток навязывания ограничено только временем действия исходной ключевой информации и спецификацией протокола (если спецификация содержит подобные ограничения). Если полученное после проведения исследования значение вероятности π, принимает значение меньшее, чем заранее фиксированное значение π_0, то мы отбрасываем такой способ компрометации протокола как маловероятный.

Во втором случае, когда нарушитель вырабатывает навязываемые значения путем решения сложных математических задач, в качестве вероятности успеха π естественно принять величину вероятности успеха алгоритма, имеющего наименьшую трудоемкость реализации Q.

Описанные подходы позволяют получать единообразные численные значения показателей эффективности защиты, в качестве которых мы будем использовать минимально допустимую вероятность успеха алгоритма компрометации криптографического протокола π_0 и минимальную трудоемкость Q_0 алгоритма компрометации, имеющего вероятность успеха большую или равную π_0.

Дадим более формальное описание сказанного. Зафиксируем метку времени t_k и для некоторого индекса $i \in \{1, \ldots, n(A)\}$ рассмотрим переменную $a_i = (\alpha_i(t_k), \sigma_i(t_k))$ и подтверждение значения $\alpha_i(t_k)$, определяемое, согласно (4.14), равенством

$$\sigma_i(t_k) = \mathit{conf}_{i,k}(a_{i_1}(t_{k-1}), \ldots, a_{i_{n_k}}(t_{k-1}),\ \xi_1(t_{k-1}), \ldots, \xi_{l_k}(t_{k-1})) \in \mathbb{B},$$

в котором $a_{i_1}, \ldots, a_{i_{n_k}}$ подтвержденные[12] ячейки, чьи значения существенным образом влияют на значение $\sigma_i(t_k)$.

Рассмотрим значения $\xi_1(t_{k-1}), \ldots, \xi_{l_k}(t_{k-1})$ как реализацию случайных величин, принимающих значения в своей области определения, тогда значение $\sigma_i(t_k)$ есть реализация случайной величины, принимающей значения на множестве $\mathbb{B}$.

Если величины $\xi_1(t_{k-1}), \ldots, \xi_{l_k}(t_{k-1})$ выработаны легитимным субъектом взаимодействия, то значение $\sigma_i(t_k)$ есть реализация случайной величины, принимающей значение true с вероятностью единица. Если же рассматриваемые величины выработаны нарушителем, т.е. нарушителем

[12]Имеется ввиду выполнение равенств $\sigma_{i_j}(t_{k-1}) = \mathit{true}$ для всех $j = 1, \ldots, n_k$.

реализуется атака направленная на подделку или навязывание ложных значений $\xi_1(t_{k-1}), \ldots, \xi_{l_k}(t_{k-1})$, то вероятность того, что значение $\sigma_i(t_k)$ будет равно $\boldsymbol{true}$, отлична от единицы.

Определим символом

$$\pi_{i,k_{max}} = \begin{cases} 0, & \text{если} \quad l_k = 0, \\ P(\sigma_i(t_k) = \boldsymbol{true}) & \text{иначе}, \end{cases} \tag{4.18}$$

вероятность принять случайный вектор $\xi_1(t_{k-1}), \ldots, \xi_{l_k}(t_{k-1})$ в качестве значения, подтверждающего значение $\alpha_i(t_k)$ в момент времени t_k. Будем говорить, что величина $\pi_{i,k}$ определяет вероятность принять ложное значение $\alpha_i(t_k)$ за истинное.

Поскольку значение каждой из существенных переменных $a_i(t_{k-1})$ должно быть подтверждено, то для них также могут быть определены вероятности $\pi_{i,k-1}$ принять ложное значение $\alpha_i(t_{k-1})$ за истинное. После этого мы, аналогично, должны определить вероятности $\pi_{i,k-1}$, $\pi_{i,k-2}$ и так далее.

Таким образом, для переменной $a_i(t_k)$ можно построить граф зависимостей её подтверждения $\sigma_i(t_k)$ и разместить на ребрах данного графа значения вероятностей принять ложное значение за истинное. Пример такого графа изображен на рисунке 4.29.

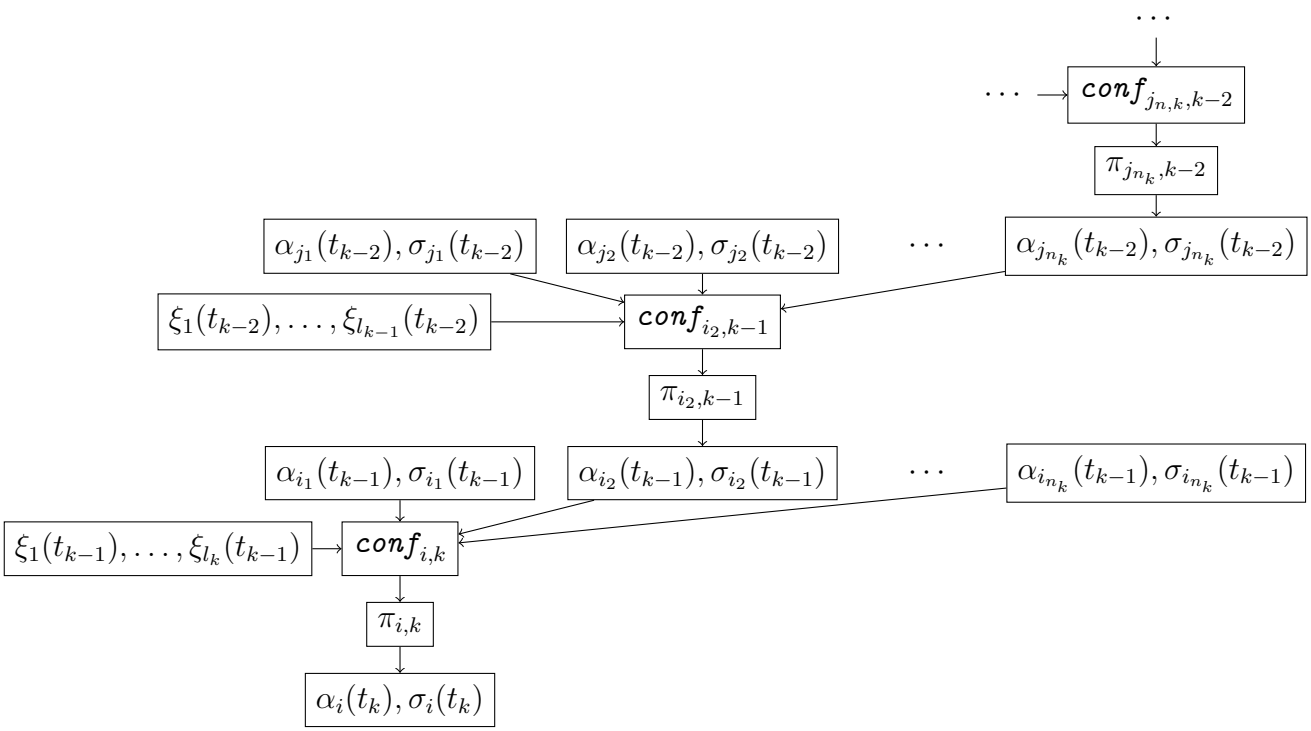


Рис. 4.29: Граф зависимостей подтверждения ячейки $a_i(t_k)$.

Теперь, выбирая в качестве значения k максимально возможное значение k_{max} и рассматривая в данном графе все пути, приводящие к значению

ячейки $a_i(t_{k_{max}})$, можно определить величину

$$\pi = \max_{i=1,\ldots,n(A)} \left(\max_{L_{i,j}} \left(1 - \prod_{\pi_{i,k} \in L_{i,j}} (1 - \pi_{i,k}) \right) \right), \quad j \in \mathbb{N}, \tag{4.19}$$

где $L_{i,1}, L_{i,2}, \ldots$ – множество путей в графе, приводящих к значению ячейки $a_i(t_{k_{max}})$.

Определение 4.17. *Величину π, определяемую равенством (4.19), будем называть вероятностью успешной компрометации криптографического протокола.*

Введенное нами значение π очевидным образом зависит от распределений случайных величин $\xi_1(t_{k_{max}-1}), \ldots, \xi_{l_k}(t_{k_{max}-1}), \ldots, \xi_1(t_1), \ldots, \xi_{l_1}(t_1)$, т.е. от способа проведения нарушителем атаки на анализируемый криптографический протокол.

Необходимо рассмотреть два основных способа проведения атак — генерации указанных значений случайным образом или с помощью вычислений, направленных на решение ряда математических задач. При этом, в случае задачи дискретного логарифмирования – трудоемкость решения весьма высока, а в случае реализации атак «отражением сообщений», см., например, рисунок 4.23, — трудоемкость решения минимальна. Рассмотрим указанные способы подробнее.

§ 4.4.3.1. Случайное угадывание

Рассмотрим принимаемые субъектом A величины $\xi_1(t_{k-1})$, …, $\xi_{l_k}(t_{k-1})$ как двоичные последовательности фиксированной длины и будем считать, что они равномерно распределены. Тогда

$$P\{\xi_j(t_{k-1}) = v\} = 2^{-\operatorname{len}_2(\xi_j(t_{k-1}))},$$

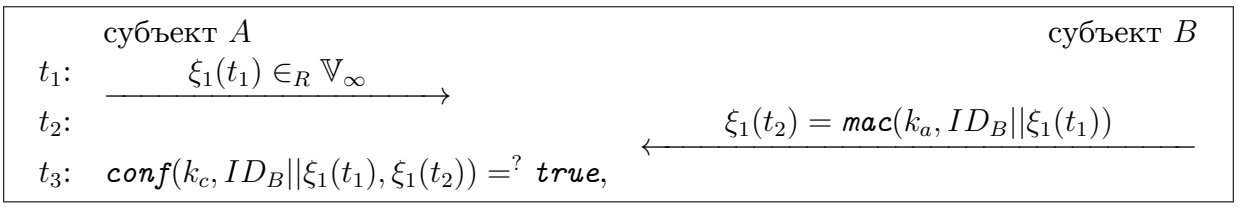
где $j = 1, \ldots, l_k$ и $v \in \mathbb{V}_{\operatorname{len}_2(\xi_j(t_{k-1}))}$).

Такая ситуация возникает в случае, когда величины $\xi_1(t_{k-1})$, …, $\xi_{l_k}(t_{k-1})$ вырабатываются субъектом B с использованием секретного ключа, не известного нарушителю. В этом случае нарушитель просто угадывает значения, выбирая их случайным образом из области определения. В этом случае вероятность

$$\pi_{i,k} = P\{\boldsymbol{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_{k-1}), \ldots, \xi_{l_k}(t_{k-1})) = \boldsymbol{true}\} = \\ = 2^{-\sum_{j=1}^{l_k} \operatorname{len}_2(\xi_j(t_{k-1}))}.$$

определяет вероятность однократного навязывания значения $\alpha_i(t_k)$.

В качестве примера рассмотрим изображенный на рисунке 4.18 протокол аутентификации.



Для ложной аутентификации нарушителю нужно предъявить значение $\xi_1(t_2)$ такое, чтобы у субъекта A выполнялось равенство

$$\mathit{conf}(k_c, \alpha_1(t_1)||\alpha_2(t_1), \xi_1(t_2)) = \mathit{true}$$

при $\alpha_1(t_1) = ID_B$, $\alpha_2(t_1) = \xi_1(t_1)$ (значение $\xi_1(t_1)$ отправляется субъектом A в канал связи и доступно нарушителю).

Поскольку нарушителю неизвестен секретный ключ k_a, то он выбирает значение кода аутентификации случайным образом. Если в качестве функции mac используется алгоритм выработки электронной подписи, регламентированный ГОСТ Р 34.10-2012 [280], и используется эллиптическая кривая с порядком группы точек q бит, где q – нечетное простое число, то вероятность однократного нарушения свойства аутентификации может быть оценена величиной $\pi_{1,3} = \frac{2}{q-1}$ (множитель 2 возникает из-за того, что в алгоритме выработки электронной подписи используется только x-координата точки эллиптической кривой и две точки — (x, y) и $(x, -y)$ дают одинаковое значение подписи).

Рассмотрим другой пример, возникающий при исследовании транспортных криптографических протоколов, Пусть субъект A принимает от субъекта B аутентифицируемые сообщения M_1, M_2, ... (см. свойство безопасности С2, а также раздел 4.4.2.3).

В соответствии с определением 4.1 будем считать, что M_i это конкатенация сообщения с его заголовком. Обозначим символом $\mathit{enc}(k_e, M)$ алгоритм зашифрования сообщения M на секретном ключе шифрования k_e, и символом $\mathit{dec}(k_e, M)$ — алгоритм расшифрования. Также будем считать, что субъект B и производные ключи шифрования k_e и имитозащиты k_i были ранее аутентифицированы субъектом A, например, в процессе выполнения протокола выработки общих ключей.

Тогда, схематичное представление транспортного протокола, реализованного по принципу[13] «MtE», может быть изображено следующим образом.

Будем считать, что для выработки кода аутентификации $\xi_2(t_k)$ используется алгоритм с длиной кода n бит. Применяя предложенный выше способ можно оценить вероятность навязывания нарушителем субъекту A одного ложного сообщения в течение заданного интервала времени.

[13]Принцип «MtE» (Mac-then-Encrypt) подразумевает, что для открытых данных сначала вычисляется код аутентификации, после чего данные зашифровываются и передаются в канал связи, см. [116], а также сноску на стр. 218.

	субъект A	субъект B
t_k:		$\xi_1(t_k) = \texttt{enc}(k_e, M_k)$, $\xi_2(t_k) = \texttt{mac}(k_i, M_k)$ $\longleftarrow$
t_{k+1}:	$\alpha_1(t_{k+1}) = \texttt{dec}(k_e, \xi_1(t_k))$, $\sigma_1(t_{k+1}) = \texttt{conf}(k_i, \alpha_1(t_{k+1}), \xi_2(t_k))$	

Рис. 4.30: Транспортный протокол передачи и приема сообщений.

Обозначим:

- символом V_1 пропускную способность канала связи, по которому передаются сообщения — на практике могут использоваться значения 100Мб/сек, 1Гб/сек, 100 Гб/сек. и т.п.,
- символом
$$l_{min} = \min\left(\mathrm{len}_2(M_1), \mathrm{len}_2(M_2), \ldots\right) > 0$$
минимально возможную длину (в битах) сообщений $M_1, M_2, \ldots$ (как правило, величина l_{min} определяется спецификацией транспортного протокола),
- символом V_2 скорость генерации нарушителем случайных двоичных векторов длины l_{min} бит.

За время T (измеряемое в секундах) нарушителем может быть отправлено субъекту A не более, чем

$$m = \left\lceil \frac{T \cdot \min\{V_1, V_2\}}{n + l_{min}} \right\rceil$$

сообщений. Тогда, согласно равенству (4.19), получаем, что для $m \leqslant 2^n$ вероятность навязывания ложного сообщения равна

$$\pi = 1 - \left(1 - \frac{1}{2^n}\right)^m = \\ = \frac{m}{2^n} - \frac{m(m-1)}{2^{2n}} + \frac{m(m-1)(m-2)}{2^{3n}} + \ldots + \frac{1}{2^{mn}} < \frac{m}{2^n},$$

где 2^{-n} вероятность случайного угадывания значения кода аутентификации. При $m > 2^n$ считаем, что $\pi = 1$.

Следует также добавить, что трудоемкость навязывания одного ложного сообщения за время T может быть оценена величиной $m\tau$, где τ — трудоемкость генерации одного случайного вектора длины n бит.

§ 4.4.3.2. Применение вычислительных алгоритмов

Пусть, как и ранее,

$$\sigma_i(t_k) = \boldsymbol{conf}_{i,k}(a_{i_1}(t_{k-1}), \ldots, a_{i_{n_k}}(t_{k-1}),\ \xi_1(t_{k-1}), \ldots, \xi_{l_k}(t_{k-1})) \in \mathbb{B}.$$

Для подделки значений $\xi_1(t_{k-1}), \ldots, \xi_{l_k}(t_{k-1})$, получаемых субъектом A из канала связи, нарушитель может использовать подход, отличный от случайного выбора значений.

Обозначим символом $\Xi_{i,k}$ множество передаваемых по каналу связи значений

$$\Xi_{i,k} = \{\xi_1(t_1), \ldots, \xi_{l_1}(t_1),\ \xi_2(t_1), \ldots, \xi_{l_2}(t_2), \ldots, \xi_1(t_k), \ldots, \xi_{i-1}(t_k)\}. \quad (4.20)$$

Тогда

$$\Xi_{1,1} \subset \Xi_{2,1} \subset \cdots \subset \Xi_{l_1,1} \subset \cdots \subset \Xi_{1,k_{max}} \subset \cdots \subset \Xi_{i,k_{max}}.$$

Фактически, мы ввели упорядочивание на величинах $\xi_i(t_k)$ и каждое из указанных множеств $\Xi_{i,k}$ получается добавлением в предыдущее одной величины $\xi_i(t_k)$.

Без ограничения общности будем считать, что значение $\xi_i(t_k)$ определяется субъектом B при помощи равенства

$$\xi_i(t_k) = \boldsymbol{f}(\Xi^{(1)}_{i,k}, B_{i,k}, \Gamma_{i,k}), \quad (4.21)$$

в котором:

- $\Xi^{(1)}_{i,k} \subseteq \Xi_{i,k}$ – множество переданных ранее по каналу связи значений, которые известны нарушителю,
- $B_{i,k} = \{\beta_1, \ldots, \beta_{r_{i,k}}\}$, где $r_{i,k} \in \mathbb{N}$ – множество значений, удовлетворяющих равенствам
$$\xi_u(t_v) = \boldsymbol{h}_i(\beta_i) \in \Xi_{i,k}$$
для некоторых $u, v \in \mathbb{N}$, $v \leqslant k$, и однонаправленных отображений $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{r_{i,k}}$ (в общем случае, используемые однонаправленные отображения могут зависеть от исходной ключевой информации, а значения $\xi_u(t_v)$ могут не принадлежать множеству $\Xi^{(1)}_{j,k}$, т.е. не передаваться по каналам связи),
- $\Gamma_{i,k} = \{\gamma_1, \ldots, \gamma_{s_{i,k}}\}$ – множество, быть может пустое, произвольно формируемых субъектом B значений, где $s_{i,k} \in \mathbb{N}_0$.

Отметим, что преобразование $\boldsymbol{f}$ может быть составным и включать в себя выработку производной ключевой информации, используемой не только при выработке значения $\xi_i(t_k)$, но и других значений, вырабатываемых позднее.

Для иллюстрации введенных обозначений рассмотрим изображенный на рисунке 4.31 вариант протокола Диффи-Хеллмана. Данный протокол реализуется в циклической абелевой группе $G = \langle g \rangle$, порождаемой элементом g порядка q, где q – нечетное простое число. Как и ранее, будем считать, что в группе G решение задачи дискретного логарифмирования имеет высокую трудоемкость.

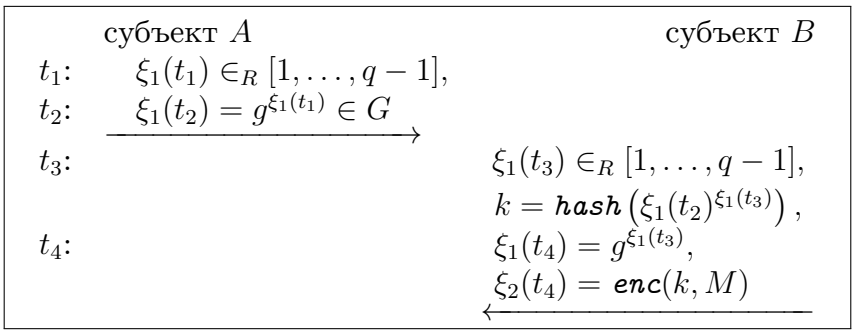


Рис. 4.31: Протокол Диффи-Хеллмана с передачей зашифрованного сообщения.

В приведенном примере субъект A принимает от субъекта B не только $\xi_1(t_4)$ — элемент группы G, необходимый для формирования общего ключа k, но и сообщение M, зашифрованное на общем ключе k.

В соответствии с (4.21) значение $\xi_2(t_4)$ может быть представлено в виде

$$\xi_2(t_4) = \boldsymbol{f}\left(\Xi^{(1)}_{2,4} = \{\xi_1(t_2)\},\ B_{2,4} = \{\xi_1(t_3)\},\ \Gamma_{2,4} = \{M\}\right),$$

где отображение $\boldsymbol{f}$ представляет собой процедуру зашифрования сообщения M, а однонаправленное отображение $\boldsymbol{h}_1$ определяется равенством

$$\xi_1(t_4) = g^{\xi_1(t_3)}$$

и представляет собой операцию возведения в степень в группе G (отметим, что изображенный на рисунке 4.31 протокол является иллюстрацией равенства (4.21) и не безопасен, поскольку общий ключ k может быть навязан субъекту A с помощью «атаки по-середине», также нарушено указанное нами ранее свойство целостности передаваемого сообщения M, см. раздел 4.4.2.2).

Вернемся к равенству (4.21). Нарушитель, которому известны используемые отображения $\boldsymbol{f}, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_{r_{i,k}}$, может попытаться определить значения β_i, используя равенства

$$\beta_i = \boldsymbol{h}_i^{-1}(\xi_u(t_v)),$$

и навязать субъекту A ложные значения $\gamma_1, \ldots, \gamma_{s_{i,k}}$.

Однонаправленные отображения $h_1, \ldots, h_{r_k}$, как правило, выбираются при синтезе протокола таким образом, чтобы вычисление обратного отображения являлось сложной математической задачей. К таким задачам могут быть отнесены задача дискретного логарифмирования в группе точек эллиптической кривой (см. определение 1.2 на стр. 53), задача построения коллизии или обращения функции хеширования (см. определение 3.1 на стр. 200), задача определения секретного ключа алгоритма блочного шифрования, (см. определение 3.8 на стр. 206) и т.п.

Для каждой из перечисленных задач могут быть рассмотрены методы ее решения, характеризуемые трудоемкостью метода $q_{i,k}$ и вероятностью успеха $p_{i,k}$, для которой выполнено неравенство $p_{i,k} \geqslant \pi_0$ (значение π_0, как указывалось ранее, позволяет вывести из рассмотрения алгоритмы с ничтожной вероятностью успеха).

Тогда вероятность навязывания ложных значений $\gamma_1, \ldots, \gamma_{s_{i,k}}$ или, другими словами, компрометации ячейки памяти $a_i(t_k)$, может быть определена[14] равенством

$$\pi_{i,k} = \prod_{j=1}^{r_{i,k}} p_{j,k},$$

и трудоемкостью

$$Q_{i,k} = \sum_{j=1}^{r_{i,k}} q_{j,k}.$$

Далее, вероятность π успешной компрометации всего криптографического протокола определяется с использованием равенства (4.19). При этом, общая трудоемкость компрометации протокола будет определяться величиной

$$Q = \sum_{\pi_{i,k} \in L} Q_{i,k}$$

где L – путь в графе, которому соответствует максимальное значение вероятности компрометации протокола.

Сделаем следующее замечание. Определенное равенством (4.20) множество $\Xi_{i,k}$ состоит из значений, передаваемых в ходе выполнения одной сессии выполнения протокола. Однако, согласно принятой нами модели нарушителя, для навязывания значений $\gamma_1, \ldots, \gamma_{s_{i,k}}$ нарушитель может использовать значения, передаваемые во всех сессиях, выполнявшихся до момента проведения атаки. Это необходимо учитывать при определении вероятности и трудоемкости обращения значений однонаправленных функций.

[14]Здесь мы предполагаем, что обращение функций $h_1, \ldots, h_{r_{i,k}}$ происходит независимо. Если же такое предположение не может быть выполнено, то величина $\pi_{i,k}$ может быть определена равенством $\pi_{i,k} = \max_{j=1,\ldots,r_{i,k}}\{p_{j,k}\}$.

Теперь мы можем дать определение безопасного криптографического протокола. Будем считать, что допустимые значения показателей эффективности защиты π_0 и Q_0 заданы и, согласно рекомендациям Р 1323565.1.012-2017 [355], определены действующими требованиями по безопасности для каждого класса средств защиты информации.

Определение 4.18. *Мы будем говорить, что протокол обладает подтвержденным состоянием субъекта A в момент времени t_{k_A}, где $k_A \in \{1, \ldots, k_{max}\}$, если значение всех ячеек памяти состояния $A(t_{k_A})$ является подтвержденным явно или косвенно, т.е.*

$$\sigma_i(t_{k_A}) = \texttt{true}, \quad i = 1, \ldots, n(A).$$

Мы будем говорить, что протокол является безопасным для субъекта A начиная с некоторого момента времени t_{k_A}, где $k_A \in \{1, \ldots, k_{max}\}$, если

1) *Для всех $k \geqslant k_A$ протокол обладает подтвержденным состоянием $A(t_k)$ субъекта A,*

2) *Выполнен одно из двух утверждений:*

   a) *вероятность успешной компрометации протокола π удовлетворяет неравенству $\pi < \pi_0$,*

   b) *трудоемкость компрометации протокола Q удовлетворяет неравенству $Q > Q_0$.*

Мы будем называть протокол безопасным начиная с некоторого момента времени t_k, если он является безопасным для каждого участвующего в информационном взаимодействии субъекта $A, B, \ldots$ начиная с момента времени $t_{K_A}, t_{K_B}, \ldots$, соответственно, и $t_k \geqslant \max\{t_{K_A}, t_{K_B}, \ldots\}$.

Отметим, что результаты расчета точных значений показателей эффективности π и Q для протокола IKEv2, см. [317], были опубликован в работе [343], а для протокола SP FIOT, см. [365], – в работе [381].

§ 4.4.4. Методика оценки безопасности

Теперь мы можем привести основной результат § 4.4: методику оценки безопасности созданного ранее или вновь разрабатываемого криптографического протокола.

Область применения методики: криптографические протоколы выработки общего ключа, а также транспортные криптографические протоколы, предназначенные для передачи конфиденциальной информации между субъектами взаимодействия.

Исходными данными для проведения исследования являются:

- класс средств защиты, в которых предполагается использование исследуемого протокола (в случае, если класс средств не определен, анализ должен проводиться для максимального класса средства защиты, см. рекомендации Р 1323565.012-2017 [355]);
- модель угроз и модель нарушителя, относительно которых оценивается эффективность защиты, обеспечиваемой исследуемым протоколом (в случае, если модель угроз не определена, должна использоваться модель согласно [288]; если не определена модель нарушителя, то должна использоваться модель, приведенная в рекомендациях Р 1323565.012-2017 для выбранного класса средств защиты информации);
- допустимые значения показателей эффективности защиты π_0 и Q_0, определенные действующими требованиями по безопасности для выбранного класса средств защиты информации (если такие требования существуют);
- спецификация протокола, в соответствии с которой предполагается его практическая реализация;
- условия практической эксплуатации протокола – к таким данным могут относится сведения о пропускной способности канала связи, объеме и допустимом времени передачи конфиденциальной информации, сроках смены ключевой информации, допустимом числе ложных попыток аутентификации и т.п.

В результате проведения исследования будут получены:

- перечень свойств безопасности, обеспечиваемых исследуемым протоколом (допускается ситуация, при которой данный перечень свойств может оказаться пустым);
- численные значения показателей эффективности защиты π и Q;
- если заданы допустимые значения показателей эффективности защиты π_0 и Q_0, то заключение о безопасности или небезопасности исследуемого протокола.

Последовательность исследований должна состоять из следующих шагов.

1. Необходимо построить формальную модель протокола, описывающую состояния каждого субъекта взаимодействия и преобразования, применяемые к ячейкам памяти и поступающим из канала связи данным, т.е.:

- определить используемые в протоколе криптографические преобразования;
- определить ключевую и криптографически опасную информацию, в частности, определить процедуры выработки сессионной и производной ключевой информации;
- определить множество ячеек памяти, образующих состояния участвующих во взаимодействии субъектов; в качестве значений, помещаемых в ячейки памяти, должны выступать:
  - исходная ключевая информация;
  - случайные значения, вырабатываемые субъектами в ходе взаимодействия;
  - производная ключевая информация;
  - а также значения, отличные от указанных выше и используемые для выработки производной ключевой информации;

  дополнительно, в состояние субъекта могут включаться значения, определяемые спецификацией протокола и которые, по мнению исследователя, могут влиять на значения определяемых показателей эффективности защиты;
- определить число возможных состояний каждого субъекта (с учетом действующих требований по безопасности, накладывающих ограничения на использование ключевой информации);
- определить области допустимых значений для каждой из случайных величин $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, получаемых субъектом из канала связи или генератора случайных величин (для всех возможных значений индекса k);
- определить отображения, задающие переход субъекта из одного состояния в другое, включая функции изменения значений ячеек памяти и функции, подтверждающие эти значения.

После построения модели протокола должно быть показано, что все ячейки состояний каждого из субъектов взаимодействия должны быть

подтверждены явно или косвенно (см. определение 4.7). При синтезе нового протокола это свойство должно выполняться в обязательном порядке.

Если для разработанного ранее протокола это свойство не выполнено, то появляется возможность построения атаки на протокол, направленной на навязывание неподтвержденного значения. Для поиска таких атак должны быть применены средства автоматизированной верификации протоколов, такие как Avispa [9], Scyther [65], Proverif [204] или им подобные.

2. Необходимо рассмотреть приведенный в разделе 4.4.1 перечень свойств безопасности и удалить из него свойства, неприменимые к исследуемому протоколу в связи с выбранной моделью угроз и условиями практического применения протокола. Для оставшихся в перечне свойств безопасности должна быть проведена проверка их выполнимости в соответствии с формальными определениями, сформулированными во втором разделе параграфа, а также с учетом зависимостей, указанных в таблице 4.3.

Если для проверки выполнимости какого-либо свойства безопасности должны быть определены значения трудоемкости Q и вероятности успеха π обращения одной или нескольких однонаправленных функций, то такие значения должны определяться с использованием условий практической эксплуатации протокола, содержащихся в исходных данных для проведения исследования.

3. В обязательном порядке должен быть проведен анализ используемой ключевой информации, включающий в себя рассмотрение следующих вопросов.

- Перед началом выполнения протокола субъектам должна быть доступна исходная ключевая информация, используемая для аутентификации сторон взаимодействия (ключи аутентификации). Если используются симметричные ключи, то они должны быть предварительно распределены с использованием организационно-технических мер защиты. Если используются асимметричные пары ключей, то открытые ключи должны быть подтверждены электронной подписью удостоверяющего (доверенного) центра, а открытые ключи удостоверяющего центра должны быть доставлены субъектам с использованием организационно-технических мер защиты. Отсутствие подтвержденной исходной ключевой информации приводит к нарушению свойства аутентификации субъекта (свойство С1), и, как следствие, к нарушению большинства из рассмотренных нами ранее свойств безопасности.

- Необходимо, чтобы ключи аутентификации не использовались непосредственно для шифрования и имитозащиты передаваемой информации. В противном случае, возможно как исчерпание ресурса ключа, так и реализация нарушителем атак, использующих конфиденциальную информацию для нарушения свойства аутентификации.

- Основное требование к производной ключевой информации, используемой для шифрования и имитозащиты передаваемых сообщений, заключается в невозможности ее определения нарушителем с трудоемкостью меньшей, чем тотальное опробование всех возможных значений. Каждый ключ, как правило, представляется в виде двоичного вектора длины m бит, таким образом нарушителю необходимо опробовать 2^m ключей для компрометации сообщений. Отсюда следует, что необходимо проверить выполнимость следующих условий:

  – множество значений, которые может принимать производный ключ совпадает со множеством $\mathbb{V}_m$,

  – принимаемые производным ключом значения непредсказуемы, т.е. последовательность нескольких, выработанных в различных сессиях протокола, производных ключей $k_1, k_2, \ldots$ должна быть статистически неотличима от последовательности случайных, равновероятно распределенных на множестве $\mathbb{V}_m$ величин.

- При практическом применении средств защиты информации могут нарушаться правила эксплуатации средств, превышаться заданные ограничения на объем обрабатываемой информации или возникать уязвимости в программном обеспечении, все вместе или по отдельности приводящие к возможности практического определения нарушителем производных ключей или исходной ключевой информации. Это приводит к необходимости встраивания в криптографические протоколы мер, минимизирующих объем скомпрометированной информации. В качестве таких мер могут выступать:

  – использование односторонних функций, не позволяющих вычислять значения ключей аутентификации по значениям производных ключей,

  – использование в каждой сессии протокола уникальных, вырабатываемых случайным образом, значений, используемых для выработки производных ключей (см. свойства С12, С13),

  – использование «древовидных» структур выработки производных ключей, не позволяющих нарушителю по известному про-

изводному ключу k_n вычислить значения ключей k_{n-1} и k_{n+1} (см. свойство С11),

- Дополнительно, в рамках математических исследований должны быть проверены следующие гипотезы:
  - о малой вероятности совпадения различных производных ключей, вырабатываемых в рамках одной сессии протокола,
  - о статистической независимости последовательности производных ключей $k_1, k_2, \ldots$, вырабатываемых в различных сессиях протокола.

4. Необходимо проверить, возможно ли применение известных ранее атак для компрометации исследуемого протокола. Для этого необходимо, во-первых, подготовить базу известных атак на криптографические протоколы из рассматриваемого класса, а во-вторых, провести классификацию известных атак, проведя систематизацию по следующим принципам:

- по методам реализации атаки; к таким методам могут быть отнесены повтор или отражение передаваемых сообщений, использование задержек и перемешивание передаваемых сообщений, изменение формата передаваемых сообщений, использование сообщений из других сессий и т.п.;
- по объектам проведения атаки; в качестве объектов атаки могут выступать передаваемые данные, секретные ключи, случайные значения, вырабатываемые в ходе протокола и т.п.
- по свойствам безопасности, поскольку каждая успешно применимая атака приводит к нарушению одного или нескольких свойств безопасности;
- техническими возможностям, необходимым для проведения атаки, например, возможностями по перехвату передаваемых данных,
- местом проведения атака, т.е. разъяснением, может ли данная атака проводиться внешним нарушителем или внутренним.

Использование подобной классификации позволяет сузить перечень атак, которые могут быть применены для компрометации исследуемого протокола. Действительно, если протокол содержит явно прописанные в спецификации меры защиты от атак повтором, то такой класс атак может оказаться неприменимым. Аналогично, атака на идентификатор субъекта взаимодействия может оказаться неприменимой, если моделью

угроз определено, что данный идентификатор представляет собой общедоступную информацию. Удостоверившись в выполнении определенного ранее перечня свойств безопасности также можно отсеять часть атак на исследуемый протокол. Для оставшихся атак в ходе исследования должна быть показана невозможность их применения, либо предложен способ компрометации исследуемого протокола.

5. С использованием описанного в разделе 4.4.3 метода должны быть определены численные значения показателей эффективности защиты информации – вероятности успешной компрометации протокола π и значения трудоемкости успешной компрометации Q. Данные величины также должны быть получены для всех способов компрометации исследуемого протокола, предложенных в ходе четвертого шага исследования. После чего, согласно определению 4.18, должен быть сделан вывод о безопасности протокола. Отметим, что полнота проводимого исследования может быть достигнута только в случае выполнения всех перечисленных шагов исследования.

Следует также отметить ряд факторов, влияющих на способность криптографического протокола противостоять атакам нарушителя:

- корректность и соответствие реализации криптографических преобразований их спецификациям,
- наличие алгоритмических мер защиты от временных атак и атак, использующих утечки информации по каналам ПЭМИН,
- непредсказуемость и распределение случайных значений, вырабатываемых датчиками случайных чисел,
- алгоритмические меры защиты исходной и производной ключевой информации,
- наличие контроля целостности специального программного обеспечения, реализующего криптографический протокол.

Проверка влияния указанных факторов возможна только при анализе конкретного средства криптографической защиты и, формально, не может быть включена в приведенную методику.

Предложенная методика обладает следующими достоинствами:

- методика позволяет определить конкретные численные значения показателей эффективности защиты; следует добавить, что на настоящий момент времени автору диссертационной работы не известен какой-либо другой подход к определению значений рассматриваемых показателей защиты;

- результаты проведенного исследования могут быть использованы при сертификации средств криптографической защиты информации, реализующих исследуемый протокол.

Также можно указать ряд недостатков предложенной методики исследования:

- сложность построения графа состояний субъекта взаимодействия при большом числе обменов сообщениями в анализируемом протоколе;

- поскольку при проведении анализа рассматриваются только известные атаки на криптографические алгоритмы и протоколы, то существует вероятность, что найдется атака, имеющая сложность, меньшую чем у атак, рассмотренных в ходе исследования; таким образом, полученные значения показателей эффективности должны рассматриваться не как точные значения, а как оценки сверху;

- несогласованность с положениями теории «доказуемой стойкости», принятой в зарубежных изданиях.

Заключение к § 4.4

Параграф § 4.4 посвящен вопросам оценки безопасности криптографических протоколов.

Предложена формальная модель, имитирующая криптографический протокол в виде дискретной динамической системы. В рамках данной модели формализован перечень свойств безопасности и определены показатели эффективности мер защиты, обеспечиваемых криптографическим протоколом. Для получения численных значений показателей эффективности мер защиты применен метод, использующий оценки трудоемкости компрометации криптографических преобразований, изменяющих состояния дискретной динамической системы. В завершение, предложена методика проведения исследования безопасности криптографических протоколов.

Заключение

В диссертационной работе предлагается решение актуальной проблемы в области информационной безопасности – проблемы синтеза безопасных криптографических схем и протоколов, применяемых для обмена информацией по открытым каналам связи.

Решение этой проблемы позволило разработать ряд математически обоснованных криптографических схем и протоколов, в частности, схему электронной подписи ГОСТ Р 34.10-2012, схемы выработки общего ключа с аутентификацией на основе открытого ключа Р 1323565.1.004-2017, криптографические механизмы аутентификации и выработки ключа фискального признака для применения в средствах формирования и проверки фискальных признаков Р 1323565.1.019-2018, криптографические механизмы защищенного взаимодействия контрольных и измерительных устройств Р 1323565.1.028-2019, протокол обмена ключами в сети Интернет МР 26.2.001- 2022, а также привело к стандартизации указанных схем и протоколов в рамках отечественной системы стандартизации.

В диссертационной работе получены следующие основные результаты:

- Получена верхняя оценка числа шагов алгоритма Госпера и предложен способ применения данного алгоритма для решения задачи дискретного логарифмирования в группе точек эллиптической кривой, что позволило уточнить оценки эффективности мер защиты, реализуемых криптографическими протоколами.

- Доказана теорема о существовании алгоритма дискретного логарифмирования в группе точек эллиптической кривой, использующего информацию о мультипликативном порядке неизвестного. Получены точные оценки трудоемкости такого алгоритма и объема используемой им памяти. Описано множество «слабых» ключей, для которых предложенный алгоритм находит решение с трудоемкостью, меньшей, чем у известных ранее алгоритмов дискретного логарифмирования. Получено точное количество «слабых» ключей для эллиптических кривых, параметры которых рекомендованы Р 1323565.1.024-2019 для использования в средствах защиты информации.

- Доказана теорема о представлении натуральных чисел значениями многочленов в точках мнимого квадратичного поля, а также предложен способ вычисления кратной точки эллиптической кривой, использующий утверждение доказанной теоремы. Предложен

алгоритм вычисления явного представления эндоморфизмов эллиптических кривых. Практическая реализация данного алгоритма на ЭВМ позволила получить представление эндоморфизмов для всех эллиптических кривых, чье кольцо эндоморфизмов изоморфно порядку мнимого квадратичного поля с числом классов равным единице.

- Предъявлены усиленные, по сравнению с ГОСТ Р 34.10-2012, требования к параметрам эллиптических кривых, рекомендуемых к применению в средствах защиты информации. Предложен алгоритм построения таких эллиптических кривых и приведены явные значения параметров, доказывающие возможность достижения предъявленных требований.

- Предложен подход к выработке псевдослучайных последовательностей, основанный на представлении действительных иррациональных чисел в виде систематической дроби по произвольному основанию. Предложены специализированные алгоритмы для представления действительных чисел специального вида, а также получены верхние оценки объема памяти, необходимого для реализации предложенных алгоритмов.

- Предложены алгоритмы восстановлении неизвестных параметров действительных иррациональных чисел специального вида. Доказаны утверждения о невозможности применения предложенных алгоритмов для построения более точных рациональных приближений, что позволяет говорить о невозможности восстановления всех элементов псевдослучайной последовательности по известному фрагменту.

- Предложен метод локальной аутентификации пользователей средств защиты информации, основанный на алгоритме представления действительных чисел в виде систематической дроби по заданному основанию. Данный метод удовлетворяет ряду специальных требований, накладываемых на подобные алгоритмы, в частности, существенно затрудняет процедуру опробования паролей с использованием специальных вычислительных средств.

- Предложен новый класс ключевых функций хэширования и доказан ряд утверждений о том, что функции из данного класса являются равновероятными сжимающими отображениями. Данный класс использован для построения нового режима аутентифицированного шифрования. Результаты практической реализации предложенного

режима показывают его преимущество в скорости при программной реализации над регламентированными в Российской Федерации алгоритмами аутентифицированного шифрования.

- Предложена гибридная схема и ряд ее модификаций, реализующих процесс шифрования с помощью полиномиального преобразования. Определена модель возможностей нарушителя и, в этой модели, доказана теорема о стойкости предложенной схемы шифрования относительно задач определения секретного ключа аутентификации, дешифрования и навязывания сообщений. Полученные результаты позволили предложить протокол передачи ключевой информации, основанный на использовании рассматриваемой гибридной схемы шифрования.

- С целью обеспечения защищенного взаимодействия в сетях «Интернета вещей» предложен новый протокол выработки общего ключа со взаимной аутентификацией субъектов взаимодействия. Доказана теорема о стойкости предложенного протокола относительно задач определения общего ключа, дешифрования и навязывания передаваемой в ходе выполнения протокола информации. Модификация данного протокола, направленная на снижение числа передаваемых в ходе выполнения протокола сообщений, успешно применена для защиты каналов управления контрольными и измерительными устройствами, и стандартизирована к качестве рекомендаций Р 1323565.1.028.–2019.

- Предложена формальная модель, имитирующая криптографический протокол в виде дискретной динамической системы. В рамках данной модели формализован перечень свойств безопасности и определены показатели эффективности мер защиты, обеспечиваемые криптографическим протоколом. Разработан метод получения численных значений показателей эффективности мер защиты, использующий оценки трудоемкости компрометации криптографических преобразований, изменяющих состояния дискретной динамической системы. Предложена методика проведения исследования безопасности криптографических протоколов.

Результаты диссертации могут применяться при исследовании специальных свойств средств криптографической защиты информации.

Литература

[1] *Abadi M., Gordon A.D.* A Calculus for Cryptographic Protocols: The Spi Calculus // *Information and Computation.* — 1999. — Vol. 148. — P. 1–70.

[2] *Adamchik V., Wagon S.* A simple formula for π // *American Mathematical Monthly.* — Nov. 1997. — P. 825–855.

[3] Advances In Elliptic Curve Cryptography / Ed. by I. Blake, G. Seroussi, N. Smart. London Mathematical Society Lecture Notes. — Cambridge : University Press, 2005. — P. 298.

[4] *Aielo L.C., Massacci F.* An Executable Specification Language for Planning Attacks to Security Protocols. // IEEE Symposium. — 2000. — P. 88–102.

[5] *Aielo L.C., Massacci F.* Planning Attacks to Security Protocols: Case Studies in Logic Programming // Computational Logic: Logic Programming and Beyond. — Vol. 2407 Of LNCS. — 2002. — P. 533–560.

[6] *Alashwali E., Rasmussen K.* What's in a Downgrade? A Taxonomy of Downgrade Attacks in the TLS Protocol and Application Protocols Using TLS. — 2019. — Cryptology ePrint Archive, Report № 2019/1083. Access mode: https://eprint.iacr.org/2019/1083 (online; accessed: May 1st, 2022).

[7] An Alternate Decomposition of an Integer for Faster Point Multiplication on Certain Elliptic Curves / Y.-H. Park, S. Jeong, C. Kim, J. Lim // Public Key Cryptography. PKC 2002. — 2002. — P. 323–334.

[8] *Atkin A.O.L., Morain F.* Elliptic curves and primality proving // *Mathematics Of Computation.* — 1993. — Vol. 61. — P. 29–68.

[9] Automated Validation of Internet Security Protocols and Applications (AVISPA). Properties (Goals). — 2006. — Access mode: http://www.avispa-project.org (online; accessed: May 1st, 2022).

[10] *Aragon N., Barreto P., Bettaieb S. et al.* BIKE: Bit Flipping Key Encapsulation. — 2021. — Access mode: https://bikesuite.

org/files/v4.2/BIKE_Spec.2021.09.29.1.pdf (online; accessed: September 1st, 2022).

[11] Badger — a fast and provably secure MAC / M. Boesgaard, O. Scavenius, T. Pedersen et al. // Applied cryptography and network security, third international conference, ACNS 2005, New York, NY, USA, June 7-10, 2005. — Vol. 3531 Of Lecture Notes Of Computer Science. — Springer, 2005. — P. 176–191.

[12] *Bailey D.H.* Integer Relation Detection and Lattice Reduction. — 2000. — preprint. Access mode: http://www.davidhbailey.com/dhbpapers/pslq-cse.pdf (online; accessed: August 31th, 2019).

[13] *Bailey D.H.* A compendium of BBP-type formulas for mathematical constants. — 2013. — preprint. Access mode: http://davidhbailey.com/dhbpapers/bbp-formulas.pdf (online; accessed: August 31th, 2017).

[14] *Bailey D.H., Borwein P.B., Plouffe S.* On the rapid computation of various polylogarithmic constants // *Mathematics of Computation*. — 1997. — Vol. 66, no. 218. — P. 903–913. — Access mode: http://www.ams.org/journals/mcom/1997-66-218/S0025-5718-97-00856-9/S0025-5718-97-00856-9.pdf (online; accessed: August 31th, 2017).

[15] *Bailey D.H., Crandall R.E.* On the random character of fundamental constant expansions // *Experimental Mathematics*. — 2001. — Vol. 10, no. 2. — P. 175–190. — Access mode: https://projecteuclid.org/download/pdf_1/euclid.em/999188630 (online; accessed: August 31th, 2017).

[16] *Bajard J.-C., Didier L.-S., Kornerup P.* An RNS Montgomery Modular Multiplication Algorithm // *IEEE Trans. On Computers*. — 1998. — Vol. 47. — P. 766–776.

[17] *Baker A.* Linear forms in the logarithms of algebraic numbers // *Mathematica*. — 1968. — Vol. 15. — P. 204–216.

[18] *Barker E., Mouha N.* Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher. — 2012. — NIST Special Publication 800-67, Revision 2. Access mode: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-67r2.pdf (online; accessed: November 7th, 2021).

[19] *Barret P.* Implementing the Rivest, Shamir and Adleman Public Key Encryption Algorithm On a Standard Digital Signal Processor // Conference on the Theory and Application of Cryptographic Techniques, CRYPTO-86. — 1987. — P. 311–323. — Access mode: http://www.loper-os.org/pub/barrett_1986.pdf (online; accessed: October 31th, 2021).

[20] *Barreto P., Pereira G., Ricardini J.* A note on high-security general-purpose elliptic curves // *Cryptology ePrint Archive, Report 2013/647*. — 2013. — Access mode: http://eprint.iacr.org/2013/647 (online; accessed: March 31st, 2020).

[21] *Basin D., Cremers C.* Modeling and Analyzing Security in the Presence of Compromising Adversaries // Computer Security – ESORICS 2010. — Vol. 6345 Of Lecture Notes Of Computer Science. — Springer, 2010. — P. 340–356. — Access mode: https://link.springer.com/content/pdf/10.1007/978-3-642-15497-3_21.pdf (online; accessed: 2nd August, 2021).

[22] *Beeler M., Gosper R.W., Schroeppel R.* HACKMEM. — 1972. — Access mode: ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-239.pdf (online; accessed: Febraury 14th, 2020).

[23] *Bellard F.* π computation record. — 2010. — Access mode: http://bellard.org/pi/pi2700e9/index.html (online; accessed: August 31th, 2017).

[24] *Bellare M., Canetti R., Krawczyk H.* Keyed Hash Functions and Message Authentication // Advances in Cryptology – Crypto '96. — Vol. 1109 Of Lecture Notes Of Computer Science. — Springer, 1996. — P. 1–15.

[25] *Bellare M., Namprempre C.* Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm // Advances in Cryptology — ASIACRYPT 2000 / Ed. by T. Okamoto. — Berlin : Springer, 2000. — P. 531–545.

[26] *Bellare M., Rogaway P.* Entity authentication and key distribution // Advances in Cryptology – Crypto '93. — Vol. 773 Of Lecture Notes Of Computer Science. — Springer, 1993. — P. 232–249. — Access mode: http://cseweb.ucsd.edu/?mihir/papers/eakd.pdf (online; accessed: August 2nd, 2021).

[27] *Bellare M., Rogaway P.* Provably secure session key distribution – the three party case // 27th ACM Symposium on Theory of Computing. — ACM Press, 1995. — P. 57–66.

[28] *Bellare M., Rogaway P., Pointcheval D.* Authenticated key exchange secure against dictionary attacks // Advances in Cryptology – EUROCRYPT 2000. — Vol. 1807 Of Lecture Notes Of Computer Science. — Springer, 2000. — P. 139–155.

[29] *Bernstein D.* Curve25519: New Diffie-Hellman speed records // Public Key Cryptography – PKC 2006 / Ed. by M. Yung, Y. Dodis, A. Kiayias, T. Malkin. — Vol. 3958. — NY. : Springer, 2006. — P. 207–228. — Access mode: https://www.iacr.org/cryptodb/archive/2006/PKC/3351/3351.pdf (online; accessed: November 2nd, 2021).

[30] *Bernstein D., Lange T.* Explicit-Formulas Database. — Access mode: https://hyperelliptic.org/EFD/index.html (online; accessed: November 1st, 2021).

[31] *Bernstein D., Lange T.* Faster addition and doubling on elliptic curves // Advances in Cryptology: ASIACRYPT 2007. — Vol. 4833. — NY. : Springer, 2007. — P. 29–50. — Access mode: http://eprint.iacr.org/2007/286.

[32] *Bernstein D., Lange T.* Failures in NIST's ECC standards. — 2016. — Access mode: https://cr.yp.to/newelliptic/nistecc-20160106.pdf.

[33] *Bertrand D., Chirskii V., Yebbou Y.* Effective estimates for global relations on Euler-type series // *Ann. Fac. Sci. Toulouse*. — 2004. — Vol. XIII, no. 2. — P. 241–260.

[34] *Biham E., Shamir A.* Differential cryptanalysis of DES-like cryptosystems // *Journal Of Cryptology*. — 1991. — Vol. 4. — P. 3–72.

[35] *Billet O., Joye M.* The Jacobi Model of an Elliptic Curve and Side-Channel Analysis // Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. AAECC 2003. — 2003. — P. 34–42.

[36] *Bisson G., Sutherland A.* Computing the endomorphism ring of an ordinary elliptic curve over finite field // *Journal Of Number Theory. Special Issue: Elliptic Curve Cryptography*. — 2011. — no. 131. — P. 815–831.

[37] *Black J., Rogaway P.* CBC MACs for arbitrary-length messages: The three-key constructions // Advances in Cryptology – Crypto 2000. — Vol. 1880 Of Lecture Notes Of Computer Science. — Springer, 2000. — P. 197–215.

[38] *Blake I., Seroussi G., Smart N.* Elliptic Curves In Cryptography. — Cambridge : University Press, 1999. — Vol. 265 of *London Mathematical Society Lecture Notes*.

[39] *Blake-Wilson S., Johnson D., Menezes A.* Key agreement protocols and their security analysis // Crypography and Coding – 6th IMA Conference. — Vol. 1355 Of Lecture Notes Of Computer Science. — Springer, 1997. — P. 20–45.

[40] *Blake-Wilson S., Menezes A.* Entity authentication and authenticated key transport protocols employing asymmetric techniques // Security Protocols – 5th International Workshop. — Springer, 1998. — P. 137–158.

[41] *Blake-Wilson S., Menezes A.* Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol // Public Key Cryptography. — Vol. 1560 Of Lecture Notes Of Computer Science. — Springer, 1999. — P. 154–170.

[42] *Blom R.* Non-Public Key Distribution // Advances in Cryptology / Ed. by D. Chaum, R.L. Rivest, A.T. Sherman. — Boston, MA : Springer US, 1983. — P. 231–236.

[43] *Boneh D.* Twenty Ears Of Attacks on the RSA Cryptosystem // *Notices of the American Mathematical Society*. — 1999. — Vol. 46. — P. 203–212. — Access mode: https://www.ams.org/notices/199902/boneh.pdf (online; accessed: October 1st, 2022).

[44] *Borel E.* Lesons sur la theorie des functions. — Paris, 1914.

[45] *Borwein J.M., Lisonek P.* Applications of integer relation algorithms // *Discrete Mathematics*. — 2000. — Vol. 217. — P. 65–82.

[46] *Bos J.W., Costello C., Miele A.* Elliptic and Hyperelliptic Curves: A Practical Security Analysis // Public-Key Cryptography (PKC 2014). — Berlin, Heidelberg : Springer Berlin Heidelberg, 2014. — P. 203–220.

[47] *Bosselaers A., Govaerts R., Vandewalle J.* Comparison of three modular reduction functions // Advances in Cryptology — CRYPTO' 93. CRYPTO 1993. — 2003. — P. 175–186.

[48] *Boyd C., Mathuria A., Stebila D.* Protocols for Authentication and Key Establishment. Second Edition. — Springer, 2020. — P. 521.

[49] *Brown R.G., Eddelbuettel D., Bauer D.* Dieharder: A Random Number Test Suite, Version 3.31.1. — 2017. — Access mode: https://webhome.phy.duke.edu/~rgb/General/dieharder.php (online; accessed: January 11th, 2020).

[50] *Burrows M., Abadi M., Needham R.* A logic of authentication // *ACM Transactions on Computer Systems*. — 1990. — Vol. 8. — P. 18–36.

[51] *Bush R., Patel K., Ward D.* Extended Message Support for BGP. — 2019. — RFC 8654. Access mode: https://tools.ietf.org/html/rfc8654 (online; accessed: January 15th, 2022).

[52] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness. — 2013. — Access mode: https://competitions.cr.yp.to/caesar.html (online; accessed: 31st May, 2021).

[53] *Canetti R., Krawczyk H.* Analysis of key-exchange protocols and their use for building secure channels // Advances in Cryptology – EUROCRYPT 2001. — Vol. 2045 Of Lecture Notes Of Computer Science. — Springer, 2001. — P. 453–474. — Access mode: https://eprint.iacr.org/2001/040 (online; accessed: August 2nd, 2021).

[54] *Carter J.L., Wegman M.N.* Universal Classes of Hash Functions // *Journal Of Computer and System Sciences*. — 1979. — Vol. 18. — P. 143–154.

[55] *Champernowne D.G.* The Construction of the Decimals Normal in the Scale of Ten // *Journal Of London Mathematical Society*. — 1933. — Vol. 8. — P. 254–260.

[56] *Charles D.* Complex Multiplication Tests For Elliptic Curves. — 2004. — arXiv.org. Access mode: https://arxiv.org/abs/math/0409501 (online; accessed: October 30th, 2021).

[57] *Chirskii V., Nesterenko A.Yu.* An approach to the transformation of periodic sequences // *Discrethe Mathematics and Applications*. — 2017. — Vol. 27, no. 1. — P. 1–7. — (English translation of [392]).

[58] *Chudnovsky D.V., Chudnovsky G.V.* Sequences of numbers generated by addition in formal groups and new primality and factorization tests // *Advances in Applied Mathematics*. — 1986. — Vol. 7, no. 4. — P. 385–434. — Access mode: https://www.sciencedirect.com/science/article/pii/0196885886900230 (online; accessed: October 31th, 2021).

[59] *Albrecht M., Bernstein D., Chou T. et al.* Classic McEliece: conservative code-based cryptography. — 2020. — Access mode: https://classic.mceliece.org/nist/mceliece-20201010.pdf (online; accessed: September 1st, 2022).

[60] Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework / R. Frye, D. Levi, S. Routhier, B. Wijnen. — 2003. — RFC 3584. Access mode: https://tools.ietf.org/html/rfc3584 (online; accessed: January 15th, 2021).

[61] *Cohen H.* A Course In Computational Algebraic Number Theory. — 3rd edition. — New York : Springer, 1996. — P. 545.

[62] *Cohen H., Miyaji A., Ono T.* Efficient Elliptic Curve Exponentiation Using Mixed Coordinates // Advances in Cryptology — ASIACRYPT'98 / Ed. by A. K. Lenstra, H. W. Lenstra. — Vol. 1514 of *Lecture Notes in Computer Science*. — 1998. — P. 51–65.

[63] *Copeland A.H., Erdos P.* Note on Normal Numbers // *Bulletin Of American Mathematical Society*. — 1946. — Vol. 52. — P. 857–860.

[64] *Cox D.* Primes of the form $x^2 + ny^2$: Fermat, Class Field Theory and Complex Multiplication. — NY. : J.Wiles and Sons, 1989. — P. 363.

[65] *Cremers C.* Scyther – Semantics and Verification of Security Protocols / C. Cremers ; Eindhoven Univ. Technology. — 2006. — P. 205. — Ph.D. Thesis.

[66] *Deuring M.* Die Typen der Multiplikatorenringe elliptischer Funktionenkörper // *Abh.Math.Semin.Univ.Hambg*. — 1941. — Vol. 14. — P. 197–272.

[67] *Dewaghe L.* Remarks on the Schoof-Elkies-Atkin algorithm // *Mathematics Of Computation*. — 1998. — Vol. 67. — P. 1247–1252.

[68] *Diffie W., Oorschot P., Wiener M.* Authentication And Authenticated Key Exchanges // *Des Codes Crypt*. — 1992. — Vol. 2. — P. 107–125.

[69] *Dolev D., Yao A.C.* On the security of public key protocols // *IEEE Transactions on Information Theory*. — 1983. — Vol. 12. — P. 198–208.

[70] *Dolmatov V., Degtyarev A.* GOST R 34.11-2012: Hash Function. — 2013. — RFC 6986. Access mode: https://tools.ietf.org/html/rfc6986 (online; accessed: January 15th, 2020).

[71] *Donenfeld J.A.* WireGuard: Next Generation Kernel Network Tunnel. — 2020. — Access mode: https://www.wireguard.com/papers/wireguard.pdf (online; accessed: September 1st, 2022).

[72] *Dworkin M.* Recommendation for Block Cipher Modes of Operation. — 2001. — NIST Special Publication 800-38A. Access mode: http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf (online; accessed: November 7th, 2021).

[73] *Dworkin M.* Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices. — 2010. — NIST Special Publication 800-38E. Access mode: https://csrc.nist.gov/publications/detail/sp/800-38e/final (online; accessed: November 7th, 2021).

[74] *Dworkin M.* Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode. — 2010. — Addendum to NIST Special Publication 800-38A. Access mode: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a-add.pdf (online; accessed: November 7th, 2021).

[75] *Edwards H.M.* A normal form for elliptic curves // *Bulletin of the American Mathematical Society*. — 2007. — P. 393–422.

[76] *ElGamal T.* A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms // *IEEE Transactions Information Theory*. — 1985. — Vol. 31. — P. 469–472.

[77] *Etzel M., Patel S., Ramzan Z.* Square Hash: Fast Message Authentication via Optimized Universal Hash Functions // Advances in Cryptology – Crypto 99. — Vol. 1666 Of Lecture Notes Of Computer Science. — Springer, 1999. — P. 234–251.

[78] Evaluation of Standardized Password-Based Key Derivation against Parallel Processing Platforms / M. Dürmuth, T. Güneysu, M. Kasper et al. — Vol. 7459. — 2012. — 09. — P. 716–733. — Access mode: https://link.springer.com/content/pdf/10.1007%2F978-3-642-33167-1_41.pdf (online; accessed: January 15th, 2020).

[79] Extensible Authentication Protocol (EAP) / B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson. — 2004. — RFC 3748. Access mode: https://tools.ietf.org/html/rfc3748 (online; accessed: January 15th, 2021).

[80] FIPS PUB 180-2. Secure Hash Standard (SHA-2). — 2002.

[81] FIPS PUB 186-4. Digital Signature Standard. — 2013. — Access mode: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf (online; accessed: March 21st, 2021).

[82] FIPS PUB 186-5. Digital Signature Standard (Draft). — 2019. — Access mode: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5-draft.pdf (online; accessed: November 1st, 2021).

[83] FIPS PUB 197. Advanced Encryption Standard. — 2001. — Access mode: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf (online; accessed: May 16th, 2021).

[84] FIPS PUB 202. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. — 2015. — Access mode: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf (online; accessed: March 21st, 2021).

[85] Faster elliptic-curve discrete logarithms on FPGAs / D.J. Bernstein, S. Engels, T. Lange et al. // *Cryptology ePrint Archive, Report 2016/382*. — 2016. — Access mode: https://eprint.iacr.org/2016/382 (online; accessed: March 1st, 2020).

[86] *Ferguson H.R.P., Bailey D.H.* A Polynomial Time, Numerically Stable Integer Relation Algorithm. — 1992. — Access mode: https://www.davidhbailey.com//dhbpapers/pslq.pdf (online; accessed: October 21th, 2019).

[87] *Ferguson H.R.P., Bailey D.H., Arno S.* Analysis of PSLQ, an integer relation finding algorithm // *Mathematics Of Computation*. — 1999. — Vol. 68, no. 225. — P. 351–369.

[88] *Ferguson H.R.P., Forcade R.W.* Generalization of the Euclidean algorithm for real numbers to all dimensions higher than two // *Bulletin (New Series) of the American Mathematical Society*. — 1979. — no. 1. — P. 912–914.

[89] *Ferguson N.* Authentication weaknesses in GCM. — 2005. — Access mode: https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/comments/cwc-gcm/ferguson2.pdf (online; accessed: 31st May, 2021).

[90] *Fielding R., Reschke J.* Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. — 2014. — RFC 7230. Access mode: https://tools.ietf.org/html/rfc7230 (online; accessed: January 15th, 2022).

[91] *Flajolet P., Odlyzko A.M.* Random mapping statistics // Advances in Cryptology: Proc. Eurocrypt'89. — Vol. 434. — NY. : Springer, 1990. — P. 329–354.

[92] *Fletcher J.G.* An Arithmetic Checksum for Serial Transmissions // *IEEE Transactions on Communications*. — 1982. — Vol. 30. — P. 247–252.

[93] *Fürer M.* Faster Integer Multiplication // *SIAM Journal On Computing*. — 2009. — Vol. 39. — P. 979–1005.

[94] *Galbraith S.D., Lin X., Scott M.* Endomorphisms for faster elliptic curve cryptography on general curves // *Journal Of Cryptology*. — 2011. — Vol. 24. — P. 446–469.

[95] *Gallant R.P., Lambert R.J., Vanstone S.A.* Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms // Advances in Cryptology – CRYPTO 2001. — 2001. — P. 190–200. — Access mode: https://www.iacr.org/archive/crypto2001/21390189.pdf (online; accessed: November 1st, 2021).

[96] *Gargantini A., Roccobene E.* Encoding abstract state mashines in PVS // ASM 2000: International Workshop on Abstract State Machines. — Vol. 1912 of LNCS. — 2000. — P. 303–322.

[97] *Goldwasser S., Micali S.* Probabilistic encryption // *Journal of Computer and System Sciences*. — 1984. — Vol. 28. — P. 270–299.

[98] *Gordon D.* Discrete Logarithms in F_p Using the Number Field Sieve // *SIAM J. Discrete Math*. — 1993. — Vol. 6. — P. 124–138.

[99] *Grebnev S. V.* Optimizing memory cost of multi-scalar multiplication // *Матем. вопр. криптогр.* — 2016. — Vol. 7. — P. 53–60.

[100] *Günther F., Thomson M., Wood C.A.* Network Working Group. Internet-Draft. Usage Limits on AEAD Algorithms. — 2021. — Access mode: https://cfrg.github.io/draft-irtf-cfrg-aead-limits/draft-irtf-cfrg-aead-limits.html (online; accessed: May 31th, 2021).

[101] *Hadano T.* Conductor of Elliptic Curves with Complex Multiplication and Elliptic Curves with Prime Conductor // *Proc. Japan Acad.* — 1975. — Vol. 51. — P. 92–95.

[102] *Halevi S., Krawczyk H.* MMH: Software Message Authentication in the Gbit/second Rates // Proceedings Of Fast Software Encryption. — Vol. 1267 Of Lecture Notes Of Computer Science. — Springer, 1997. — P. 172–189.

[103] *Hancl J., Tijdeman R.* On the irrationality of factorial series II // *Journal of Number Theory*. — 2010. — Vol. 130. — P. 595–607. — Access mode: http://www.math.leidenuniv.nl/~tijdeman/hantijd4.pdf (online; accessed: August 31th, 2017).

[104] *Handschuh H., Preneel B.* Key-Recovery Attacks on Universal Hash Function based MAC Algorithms // Proceedings of 28th Annual International Cryptology Conference, Crypto 2008, Santa Barbara, CA, USA, August 17-21, 2008. — Vol. 5157 Of Lecture Notes Of Computer Science. — Springer, 2008. — P. 144–161.

[105] *Hartman S., Wasserman M., Zhang D.* Extensible Authentication Protocol (EAP) Mutual Cryptographic Binding. — 2013. — RFC 7029. Access mode: https://tools.ietf.org/html/rfc7029 (online; accessed: January 15th, 2020).

[106] *Heys H.M.* A Tutorial On Linear and Differential cryptanalisis // *Cryptologia*. — 2002. — Vol. 26, no. 3. — P. 189–221.

[107] *Hoare C.A.R.* Communicating Sequential Processes. — Prentice Hall International, 1985.

[108] *Hoffstein J., Piper J., Silverman H.* NTRU: A ring-based public key cryptosystem // Algorithmic Number Theory, ANTS-III. — Vol. 1423 Of Lecture Notes Of Computer Science. — Springer, 1998. — P. 267–288.

[109] *Husemöller D.* Elliptic Curves. — 2 edition. — New-York : Springer-Verlag, 2004. — Access mode: https://web.math.rochester.edu/people/faculty/doug/otherpapers/Husemoller.pdf (online; accessed: February 14th, 2020).

[110] IEEE 1619.1-2018 – IEEE Standard for Authenticated Encryption with Length Expansion for Storage Devices. — 2018.

[111] IEEE 802-2001 – IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture. — 2001.

[112] IEEE 802.15.3-2016 - IEEE Standard for High Data Rate Wireless Multi-Media Networks. — 2016. — Access mode: https://standards.ieee.org/ieee/802.15.3/6211/ (online; accessed: September 1st, 2022).

[113] IEEE 802.1AE-2018 – IEEE Standard for Local and metropolitan area networks-Media Access Control (MAC) Security. — 2018.

[114] IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. — 2020. — Access mode: https://standards.ieee.org/ieee/802.11/7028/ (online; accessed: September 1st, 2022).

[115] ISO/IEC 18033-2:2006. Information technology. Security techniques. Encryption algorithms — Part 2: Asymmetric ciphers. — 2006.

[116] ISO/IEC 19772:2009. Information technology. Security techniques. Authenticated encryption. — 2009.

[117] Implementation of RSA Algorithm Based on RNS Montgomery Multiplication / H. Nozaki, M. Motoyama, A. Shimbo, S. Kawamura // Cryptographic Hardware and Embedded Systems – CHES 2001. — 2001. — P. 364–376.

[118] Improved Algorithms for Efficient Arithmetic on Elliptic Curves Using Fast Endomorphisms / M. Ciet, T. Lange, F. Sica, J.J. Quisquater // Advances in Cryptology — EUROCRYPT 2003. — 2003. — P. 388–400.

[119] *Indesteege S.* Analysis and Design of Cryptographic Hash Functions : Doctoral dissertation. — Katholieke Universiteit Leuven / S. Indesteege. — 2010. — Access mode: https://www.esat.kuleuven.be/cosic/publications/thesis-171.pdf (online; accessed: July 20th, 2020).

[120] Internet Key Exchange Protocol Version 2 (IKEv2) / C. Kaufman, P. Hoffman, Y. Nir et al. — 2014. — RFC 7296. Access mode: https://tools.ietf.org/html/rfc7296 (online; accessed: January 15th, 2020).

[121] *Iwata T., Kurosawa K.* OMAC: One-Key CBC MAC // Fast Software Encryption / Ed. by Thomas Johansson. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2003. — P. 129–153. — Access mode: https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/proposed-modes/omac/omac-ad.pdf (online; accessed: May 1st, 2020).

[122] *Jolley L.* Summation of series. — London : Chapman and Hall LTD, 1925.

[123] *Joux A., Lercier R.* Improvements to the general Number Field Sieve for Discrete Logarithms in Prime Fields // *Mathematics Of Computation.* — 2003. — Vol. 72, no. 242. — P. 953–967.

[124] *Kaya Koc C., Acar T., Kaliski B.S.* Analyzing and comparing Montgomery multiplication algorithms // *IEEE Micro.* — 1996. — Vol. 16, no. 3. — P. 26–33.

[125] *Kim J.* On the security of the block cipher GOST suitable for the protection in U-business services // *Personal and Ubiquitous Computing volume.* — 2013. — P. 1429–1435.

[126] *Kiryukhin V.A.* Exact maximum expected differential and linear probability for 2-round Kuznyechik // *Математические вопросы криптографии.* — 2010. — Т. 10. — С. 107–116.

[127] *Knapp A.* Elliptic Curves. — New Jersey : Princeton University Press, 1992. — В русском переводе: *Кнэпп Э.* Эллиптические кривые. – М.:Факториал Пресс, 2004.

[128] *Koblitz N.* Elliptic Curve Cryptosystems // *Mathematics Of Computation.* — 1987. — no. 48. — P. 203–209.

[129] *Kohel D.* Endomorphism rings of elliptic curves over finite fields : PhD thesis of University Of California At Berkley / D. Kohel. — 1996. — P. 96. — Access mode: http://iml.univ-mrs.fr/~kohel/pub/thesis.pdf (online; accessed: October 30th, 2021).

[130] *Kolmogoroff A.N.* Sulla determinazione empirica di una legge di distribuzione // *G. Ist. Ital. attuar.* — 1933. — Vol. 4, no. 1. — P. 83–91.

[131] *Koutsos A.* The 5G-AKA Authentication Protocol Privacy // 2019 IEEE European Symposium on Security and Privacy (EuroS&P). — 2019. — P. 464–479.

[132] *Koyama K., Tsuruoka Y.* Speeding up elliptic cryptosystems using a signed binary window method // CRYPTO-92. — 1992. — P. 345–347.

[133] *Krawczyk H.* SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols // Advances in Cryptology - CRYPTO 2003. — 2003. — P. 400–425. — Access mode: 10.1007/978-3-540-45146-4_24.

[134] *Krawczyk H.* HMQV: A high-performance secure Diffie–Hellman protocol // Advances in Cryptology - CRYPTO 2005. — Vol. 3621 Of

Lecture Notes in Computer Science. — 2005. — P. 546–566. — Access mode: https://eprint.iacr.org/2005/176.

[135] *Krovetz T.* Message authentication on 64-bit architectures // Selected areas in cryptography, 13th international workshop, SAC 2006, Montreal, Canada, August 17-18, 2006. — Vol. 4356 Of Lecture Notes in Computer Science. — 2007. — P. 327–341.

[136] *Krovetz T., Rogaway P.* The Software Performance of Authenticated-Encryption Modes // Fast Software Encryption – FSE 2011. — 2011. — Access mode: https://www.cs.ucdavis.edu/~rogaway/papers/ae.pdf (online; accessed: 26th April, 2011).

[137] *LaMacchia B.A., Lauter K., Mityagin A.* Stronger security of authenticated key exchange // Provable Security, First International Conference, ProvSec 2007. — Vol. 4787 Of Lecture Notes in Computer Science. — 2007. — P. 1–16.

[138] *Lagarias J.C.* On the normality of arithmetical constants // *Experimental Mathematics*. — 2001. — Vol. 10, no. 3. — P. 355–368. — Access mode: http://emis.ams.org/journals/EM/expmath/volumes/10/10.3/Lagarias.pdf (online; accessed: August 31th, 2017).

[139] *Lavrikov I.V., Shishkin V.A.* How much data may safely processed on one key in different modes? // *Математические вопросы криптографии*. — 2019. — Vol. 10, no. 2. — P. 125–134.

[140] Layer Two Tunneling Protocol "L2TP" / W. Townsley, A. Valencia, A. Rubens et al. — 1999. — RFC 2661. Access mode: https://tools.ietf.org/html/rfc2661 (online; accessed: January 15th, 2021).

[141] *Lehmer D.H.* Euler constants for arithmetical progressions // *Acta Arithmetica*. — 1975. — Vol. 27. — P. 125–142.

[142] *Lenstra A.K., H.W. Lenstra H. W., Lovasz L.* Factoring polynomials with rational coefficients // *Mathematische Annalen*. — 1981. — Vol. 4, no. 261. — P. 515–534.

[143] *Lenstra H.W.* Factoring Integers with Elliptic Curves // *Ann. Math.* — 1987. — no. 126. — P. 649–673.

[144] Libakrypt: software crypto module for user space. — 2022. — (in accordance with R 1323565.1.012-2017 [355]). Access mode: https://github.com/axelkenzo/libakrypt-0.x (online; accessed: January 15th, 2022).

[145] Low exponent RSA with related messages / D. Coppersmith, M. Franklin, J. Patarin, M. Reiter // Advances in Cryptology — EUROCRYPT'98. — Vol. 1070 of *Lecture Notes in Computer Science*. — 1996. — P. 1–9.

[146] *Lowe G.* Breaking and fixing the Needham-Schroeder Public-Key Protocol using FDR // Tools and Algorithms for the Construction and Analysis of Systems. — 1996. — P. 147–166.

[147] *Lyskov M., Rivest R., Wagner D.* Tweakable Block Ciphers // *Journal Of Cryptology*. — 2011. — Vol. 24. — P. 588–613.

[148] *Lórencz R.* New Algorithm for Classical Modular Inverse // Cryptographic Hardware and Embedded Systems, CHES-2002. — 2003. — P. 57–70.

[149] *Lórencz R., Hlaváč J.* Subtraction-free Almost Montgomery Inverse algorithm // *Information Processing Letters*. — 2005. — Vol. 94, no. 1. — P. 11–14. — Access mode: https://www.sciencedirect.com/science/article/pii/S0020019004003692 (online; accessed: October 31th, 2021).

[150] Magma Computational Algebra System. — 2020. — Access mode: http://magma.maths.usyd.edu.au/magma/ (online; accessed: March 30th, 2020).

[151] *Mao W.* Modern Cryptography: Theory and Practice. — Prentice Hall, 2003. — С. 648. — В русском переводе: Мао, Венбо. Современная криптография : теория и практика [пер. с англ. и ред. Д. А. Клюшина]. - М. [и др.] : Вильямс, 2005 (ГПП Печ. Двор). - 763 с. : ил., табл.; 24 см.; ISBN 5-8459-0847-7.

[152] *Marsaglia G.* The Marsaglia Random Number CDROM, with The Diehard Battery of Tests of Randomness. — 1985. — produced at Florida State University under a grant from The National Science Foundation. Access mode: https://web.archive.org/web/20160125103112/http://stat.fsu.edu/pub/diehard/ (online; accessed: January 11th, 2020).

[153] *Marsaglia G.* A current view of random number generators // Keynote Address, Statistics and Computer Science: XVI Symposium on the Interface, Atlanta, Proceedings, Elsevier. — 1985.

[154] *Marsaglia G., Tsang W.W.* Some Difficult-to-pass Tests of Randomness // *Journal of Statistical Software, Articles*. —

2002. — Vol. 7, no. 3. — P. 1–9. — Access mode: https://www.jstatsoft.org/v007/i03 (online; accessed: January 13th, 2020).

[155] *Matsui M.* Linear Cryptanalysis Method for DES Cipher // Advances in Cryptology — EUROCRYPT '93. — Berlin : Springer, 1993. — P. 386–397.

[156] *Matsui M.* The First Experimental Cryptanalysis of the Data Encryption Standard // Advances in Cryptology — CRYPTO '94. — Berlin : Springer, 1994. — P. 1–11.

[157] *McGrew D., Bailey D.* AES-CCM Cipher Suites for Transport Layer Security (TLS). — 2012. — RFC 6655. Access mode: https://tools.ietf.org/html/rfc6655 (online; accessed: January 15th, 2021).

[158] *McGrew David A., Viega John.* The Security and Performance of the Galois/Counter Mode (GCM) of Operation // In INDOCRYPT, volume 3348 of LNCS. — Springer, 2004. — P. 343–355.

[159] *Menezes A., Ustaoglu B.* On the importance of public-key validation in the MQV and HMQV key agreement protocols // Progress in Cryptology - INDOCRYPT 2006. — Vol. 4329 Of Lecture Notes in Computer Science. — 2006. — P. 133–147.

[160] *Menezes A., Vanstone S., Okamoto T.* Reducing elliptic curve logarithms to logarithms in a finite field // Proc. 23rd ACM Symp. Theory of Computing. — 1991. — P. 80–89.

[161] *Menezes A.J., van Oorschot P.C., Vanstone S.A.* Handbook Of Apllied Cryptography. — CRC Press, 1996. — P. 816. — ISBN: 0-8493-8523-7. — Access mode: http://cacr.uwaterloo.ca/hac/ (online; accessed: August 31th, 2021 r.).

[162] *Miller V.* Use of elliptic curves in cryptography // CRYPTO-86. — Springer, 1986. — P. 417–426.

[163] *Montgomery P.L.* Modular Multiplication Without Trial Division // *Mathematics Of Computation*. — 1985. — Vol. 44, no. 170. — P. 519–521.

[164] *Montgomery P.L.* Speeding the Pollard and elliptic curve methods of factorization // *Mathematics Of Computation*. — 1987. — Vol. 48. — P. 243–264.

[165] *Morain F.* Implementation of the Atkin-Goldwasser-Kilian primality testing algorithm. — 1988. — RR-0911, INRIA. Access mode: https://hal.inria.fr/file/index/docid/75645/filename/RR-0911.pdf (online; accessed: Marh 31st, 2020).

[166] *Moriarty K., Kaliski B., Rush A.* PKCS #5: Password-Based Cryptography Specification Version 2.1. — 2017. — RFC 8018. Access mode: https://tools.ietf.org/html/rfc8018 (online; accessed: January 15th, 2020).

[167] *Müller V.* Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two // *Journal of Cryptology*. — 1998. — Vol. 11. — P. 219–234.

[168] *Müller V.* Efficient Point Multiplication for Elliptic Curves over Special Optimal Extension Fields // Public-Key Cryptography and Computational Number Theory. — 2000. — P. 197–207. — Proceedings of the International Conference organized by the Stefan Banach International Mathematical Center Warsaw, Poland, September 11-15, 2000.

[169] Multilinear Galois Mode (MGM) / S. Smyshlyaev, V. Nozdrunov, V. Shishkin, E. Griboedova. — 2021. — RFC 9058. Access mode: https://tools.ietf.org/html/rfc9058 (online; accessed: January 15th, 2022).

[170] NIST Cryptographic toolkit. Modes Development. — 2017. — Access mode: https://web.archive.org/web/20170830120738/http:/csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html (online; accessed: 31at May, 2021).

[171] *Chen C., Danba O., Hoffstein J. et al.* NTRU: Algorithm Specifications And Supporting Documentation. — 2019. — Access mode: https://ntru.org/f/ntru-20190330.pdf (online; accessed: September 1st, 2022).

[172] *Nandi M.* On the Minimum Number of Multiplications Necessary for Universal Hash Constructions. — 2013. — preprint. Access mode: https://eprint.iacr.org/2013/574.pdf (online; accessed: August 31th, 2019).

[173] *Nesterenko A. Yu.* Cycle detection algorithms and their applications // *Journal of Mathematical Sciences*. — 2012. — Vol. 182, no. 4. — P. 518–526. — (English translation of [325]).

[174] *Nesterenko A. Yu.* Key Transport Protocol Based On Hybryd Encryption Scheme // The 7th International Computer Science Symposium in Russia. Workshop «Current Trends in Cryptology», Nizhny Novgorod, Russia. — 2012. — P. 20–21.

[175] *Nesterenko A. Yu.* Constructions of elliptic curves endomorphisms // *Математические вопросы криптографии*. — 2014. — Т. 5, № 2. — С. 99–102.

[176] *Nesterenko A. Yu.* Some remarks on the elliptic curve discrete logarithm problem // *Математические вопросы криптографии*. — 2016. — Т. 7, № 2. — С. 115–120.

[177] *Nesterenko A. Yu.* A new authenticated encryption mode for arbitrary block cipher based on universal hash function // CTCrypt 2016. — 2016. — Access mode: https://ctcrypt.ru/program_2016 (online; accessed: 31st May, 2021).

[178] *Nesterenko A. Yu.* A new authenticated encryption mode for arbitrary block cipher based on universal hash function // *Математические вопросы криптографии*. — 2017. — Vol. 8, no. 2. — P. 117–130.

[179] *Nesterenko A. Yu.* Construction of strong elliptic curves suitable for cryptographic applications // *Математические вопросы криптографии*. — 2019. — Vol. 10, no. 2. — P. 135–144.

[180] *Nesterenko A. Yu.* Differential properties of authenticated encryption mode based on universal hash function (XTSMAC) // 2021 XVII International Symposium "Problems of Redundancy in Information and Control Systems"(REDUNDANCY). — 2021. — P. 39–44.

[181] *Nesterenko A. Yu., Semenov A. M.* On the practical implementation of Russian protocols for low-resource cryptographic modules // *Journal of Computer Virology and Hacking Techniques*. — 2020. — Vol. 16, no. 4. — P. 305–312.

[182] *Nesterenko Yu. V.* Algebraic Independence. — Narosa Publishing House, 2009. — P. 162. — ISBN: 8173199841.

[183] *Nivash G.* Cycle Detecting Using a Stack // *Journal Information Processing Letters*. — 2004. — Vol. 90.

[184] *Nozdrunov V.* Parallel and double block cipher mode of operation (PD-mode) for authenticated encryption // CTCrypt 2017. — 2017. — P. 36–45.

[185] OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption / P. Rogaway, M. Bellare, J. Black, T. Krovetz // ACM Conference on Computer and Communications Security 2001 - CCS 2001. — 2001.

[186] *Okeya K., Kurumatani H., Sakurai K.* Elliptic Curves with the Montgomery-Form and Their Cryptographic Applications // Public Key Cryptography. PKC 2000 / Ed. by H. Imai, Y. Zheng. — 2000.

[187] On the Weaknesses of PBKDF2 / A. Visconti, S. Bossi, H. Ragab, A.Calò // In: Cryptology and Network Security. CANS 2015. — Vol. 9476 of *Lecture Notes in Computer Science*. — Springer, 2015. — Access mode: https://hanyr.ax/files/pub/pbkdf2.pdf (online; accessed: January 15th, 2020).

[188] *Oorschot P.C., Wiener M.J.* Parallel Collision Search with Cryptanalytic Applications // *Journal of Cryptology*. — 1999. — Vol. 12. — P. 1–28.

[189] PKCS #1: RSA Cryptography Specifications Version 2.2 / K. Moriarty, B. Kaliski, J. Jonsson, A. Rush. — 2016. — RFC 8017. Access mode: https://tools.ietf.org/html/rfc8017 (online; accessed: January 15th, 2020).

[190] *Parry W.* On the β-expansions of real numbers // *Acta Mathematica Academiae Scientiarum Hungaricae*. — 1960. — Vol. 11. — P. 401–416.

[191] Password Hashing Competition. — 2015. — Access mode: https://password-hashing.net/ (online; accessed: January 15th, 2020).

[192] *Pearson K.* On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from randon sampling // *Phil. Mag.* — 1900. — Vol. V. — P. 157.

[193] *Peterson W.W., Brown D.T.* Cyclic Codes for Error Detection // *Proceedings of the IRE*. — 1961. — Vol. 49. — P. 157.

[194] *Petit C., Kosters M., Messeng A.* Algebraic Approaches for the Elliptic Curve Discrete Logarithm Problem over Prime Fields // Public-Key Cryptography (PKC 2016). — Berlin Heidelberg : Springer, 2016. — P. 3–18.

[195] *Pollard J.M.* A Monte Carlo Method for Factorisation // *BIT*. — 1975. — no. 15. — P. 331–334. — Access mode: http://pages.

cs.wisc.edu/~cs812-1/pollardrho.pdf (online; accessed: November 21st, 2021).

[196] *Pollard J.M.* Monte Carlo methods for index computation (mod p) // *Mathematics Of Computation*. — 1978. — Vol. 32, no. 143. — P. 918–924.

[197] Polynomial Time Algorithms for Finding Integer Relations Among Real Numbers / J. Hastad, B. Just, J.C. Lagarias, C.P. Schnorr // *SIAM Journal of Computing*. — 1989. — Vol. 18. — P. 859–881.

[198] *Postel J.* Internet Protocol. — 1980. — RFC 760. Access mode: https://tools.ietf.org/html/rfc760 (online; accessed: January 15th, 2021).

[199] *Postel J.* Transmission Control Protocol. — 1980. — RFC 761. Access mode: https://tools.ietf.org/html/rfc761 (online; accessed: January 15th, 2021).

[200] *Postel J.* User Datagramm Protocol. — 1980. — RFC 768. Access mode: https://tools.ietf.org/html/rfc768 (online; accessed: January 15th, 2021).

[201] Practical Significance of Security Bounds for Standardized Internally Re-keyed Block Cipher Modes / L. R. Ahmetzyanova, E. K. Alekseev, G. K. Sedov et al. // *Mat. Vopr. Kryptogr.* — 2019. — Vol. 10. — P. 31–46.

[202] *Preneel B.* Analysis and Design of Cryptographic Hash Functions : Doctoral dissertation. — Katholieke Universiteit Leuven / B. Preneel. — 1993. — Access mode: https://homes.esat.kuleuven.be/~preneel/phd_preneel_feb1993.pdf (online; accessed: July 20th, 2020).

[203] Protocol Gemini. — 2022. — Access mode: https://gemini.circumlunar.space/ (online; accessed: October 1st, 2022).

[204] Proverif: Automatic Cryptographic Protocol Verifier. — 2021. — User Manual and Tutorial. Access mode: http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf (online; accessed: May 1st, 2022).

[205] The Quest for π / D.H. Bailey, J.M. Borwein, P.B. Borwein, S. Plouffe // *Mathematical Intelligencer*. — 1997. — Vol. 19, no. 1. — P. 50–57. — Access mode: https://crd-legacy.lbl.gov/~dhbailey/dhbpapers/pi-quest.pdf (online; accessed: March 3th, 2019).

[206] RNS Montgomery reduction algorithms using quadratic residuosity / S. Kawamura, Y. Komano, H. Shimizu, T. Yonemura // *Journal of Cryptographic Engineering*. — 2019. — Vol. 9. — P. 313–331.

[207] *Rabin M.O.* Digitalized Signatures And Public Key Fuction As Intractable As Factorization. — 1979. — MTI Techreport, January 1979.

[208] *Rabin M.O.* Probabilistic algorithm for testing primality // *Journal of Number Theory*. — 1980. — Vol. 12, no. 1. — P. 128–138.

[209] Relations among notions of security for public-key encryption schemes / M. Bellare, M. Desai, A. Pointcheval, P. Rogaway // CRYPTO-98. — 1998. — P. 26–46. — Access mode: https://link.springer.com/content/pdf/10.1007/BFb0055718.pdf (online; accessed: September 31st, 2022).

[210] *Remmert R., Ullrich P.* Elementare Zahlentheorie. — Berlin : Birkhäuser, 1995. — P. 276.

[211] *Rényi A.* Representations for real numbers and their ergodic properties // *Acta Mathematica Academiae Scientiarum Hungaricae*. — 1957. — Vol. 8. — P. 477–493.

[212] *Rescorla E.* The Transport Layer Security (TLS) Protocol Version 1.3. — 2018. — RFC 8446. Access mode: https://tools.ietf.org/html/rfc8446 (online; accessed: January 15th, 2020).

[213] *Rescorla E., Korver B.* Guidelines for Writing RFC Text on Security Considerations. — 2003. — RFC 3552. Access mode: https://tools.ietf.org/html/rfc3552 (online; accessed: January 15th, 2021).

[214] *Rescorla E., Tschofenig H., Modadugu N.* The Datagram Transport Layer Security (DTLS) Protocol Version 1.3. — 2022. — RFC 9147. Access mode: https://tools.ietf.org/html/rfc9147 (online; accessed: January 15th, 2023).

[215] *Ritter T.* Randomness Tests: A Literature Survey. — 2002. — Access mode: http://www.ciphersbyritter.com/RES/RANDTEST.HTM (online; accessed: January 11th, 2020).

[216] *Rogaway P.* Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC // Advances in Cryptology – Asiacrypt 2004. — Vol. 3329 of *Lecture Notes in Computer Science*. — 2004. — P. 16–31.

[217] *Ruddick A., Yan J.* Acceleration Attacks on PBKDF2: Or, What Is inside the Black-Box of oclHashcat? // 10th USENIX Workshop on Offensive Technologies (WOOT 16). — Austin, TX : USENIX Association, 2016. — Access mode: https://www.usenix.org/conference/woot16/workshop-program/presentation/ruddick.

[218] *Saarinen M.-J.O.* Cycling attacks on GCM, GHASH and other polynomial MACs and hashes // Fast Software Encryption – FSE 2012. — 2012. — P. 216–225.

[219] *Satoh T., Araki K.* Fermat quotients and the polynomial time discrete log algorithm for anomalous curves // *Comm. Math. Univ. Sancti Pauli*. — 1998. — Vol. 47. — P. 81–92.

[220] *Savas E., Kaya Koc C.* Montgomery inversion // *J Cryptogr Eng*. — 2018. — Vol. 8. — P. 201–210.

[221] *Schönhage A. Strassen V.* Schnelle Multiplikation großer Zahlen // *Computing*. — 1971. — no. 7. — P. 281–292.

[222] *Schoof R.* Counting Points On Elliptic Curves Over Finite Fields // *J. Theorie des Nombres de Bordeaux*. — 1995. — Vol. 7, no. 1. — P. 219–254.

[223] Security of Multilinear Galois Mode (MGM) / L. Ahmetzyanova, E. Alekseev, G. Karpunin, V. Nozdrunov. — 2019. — preprint. Access mode: https://eprint.iacr.org/2019/123.pdf (online; accessed: August 31th, 2019).

[224] *Sedgewick R., Szymansky T.G., Yao A.C.* The Complexity of Finding Cycles In Pediodic Functions // *Siam Journal Of Computing*. — 1982. — Vol. 11, no. 2. — P. 376–390.

[225] *Semaev I.* Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p // *Mathematics of Computation*. — 1998. — Vol. 67, no. 221. — P. 353–356.

[226] *Semaev I.* Summation Polynomials and the Discrete Logarithm Problem on Elliptic Curves. — 2004. — Access mode: https://eprint.iacr.org/2004/031.pdf (online; accessed: April 15th, 2020).

[227] Set It and Forget It! Turnkey ECC for Instant Integration / D. Belyavsky, B.B. Brumley, J.-J. Chi-Domínguez et al. — 2021. — Access mode: https://arxiv.org/pdf/2007.11481.pdf (online; accessed: November 1st, 2021).

[228] *Shanks D.* Class number, a theory of factorization and genera // Proceedings Of Symposium Pure Mathematics. — Vol. 20. — Providence, R. I. : AMS, 1971. — P. 415–440.

[229] *Shipsey R., Swart C.* Elliptic divisibility sequences and the elliptic curve discrete logarithm problem. — 2008. — preprint. Access mode: http://eprint.iacr.org/2008/044 (online; accessed: Fenruary 21th, 2020).

[230] *Shoup V.* A Computational Introduction to Number Theory and Algebra. — 2nd edition. — Cambridge University Press, 2009. — P. 590.

[231] *Sica F., Ciet M., Quisquater J.J.* Analysis of the Gallant-Lambert-Vanstone Method Based on Efficient Endomorphisms: Elliptic and Hyperelliptic Curves // Selected Areas in Cryptography. SAC 2002. — 2003. — P. 21–36.

[232] *Silverman J.H.* The Xedni Calculus and the Elliptic Curve Discrete Logarithm Problem // *Designs, Codes and Cryptography*. — 2000. — no. 20. — P. 5–40.

[233] *Silverman J.H.* The Arithmetic Of Elliptic Curves. — 2nd edition. — Springer, 2009. — Access mode: http://www.pdmi.ras.ru/~lowdimma/BSD/Silverman-Arithmetic_of_EC.pdf (online; accessed: February 14th, 2020).

[234] *Smart N.* Elliptic Curve Cryptosystems over Small Fields of Odd Characteristic // *Journal of Cryptology*. — 1999. — Vol. 12. — P. 141–151.

[235] *Smart N.* The discrete logarithm problem on elliptic curves of trace one // *Journal of Cryptology*. — 1999. — Vol. 12. — P. 193–196.

[236] *Smart N.* The Hessian Form of an Elliptic Curve // Cryptographic Hardware and Embedded Systems – CHES 2001. — 2001. — P. 118–125.

[237] *Smyshlyaev S.* Re-keying Mechanisms for Symmetric Keys. — 2019. — RFC 8645. Access mode: https://tools.ietf.org/html/rfc8645 (online; accessed: January 15th, 2022).

[238] Solving a 112-Bit Prime Elliptic Curve Discrete Logarithm Problem on Game Consoles Using Sloppy Reduction / J.W. Bos, M.E. Kaihara, T. Kleinjung et al. // *Int. J. Appl. Cryptol.* — 2012. — Feb. — Vol. 2, no. 3. — P. 212–228. — Access mode: http://joppebos.com/files/noan112.pdf (online; accessed: March 1st, 2020).

[239] Solving a 114-Bit ECDLP for a Barreto-Naehrig Curve / T. Kusaka, S. Joichi, K. Ikuta et al. // Information Security and Cryptology – ICISC 2017. — 2018. — P. 231–244. — Access mode: https://hal.archives-ouvertes.fr/hal-01633653/file/article.pdf (online; accessed: March 1st, 2020).

[240] *Soto J.* Statistical Testing of Random Number Generators // In: Proceedings of the 22nd National Information Systems Security Conference. — 1999. — Access mode: http://csrc.nist.gov/nissc/1999/proceeding/papers/p24.pdf (online; accessed: January 11th, 2020).

[241] Sponge functions / G. Bertoni, J. Daemen, M. Peeters, G. van Assche // Ecrypt Hash Workshop. — 2007. — Access mode: https://keccak.team/files/SpongeFunctions.pdf (online; accessed: March 21st, 2021).

[242] *Stahnke W.* Primitive binary polynomials // *Mathematics Of Computation.* — 1973. — Vol. 27. — P. 977–980.

[243] *Stark H.* Class numbers of complex quadratic fields // Modular Functions of one variable I. — Vol. 320 of *Lecture Notes in Math.* — Springer-Verlag, 1973. — P. 153–174.

[244] *Rukhin A., Soto J., Nechvatal J. et al.* A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. — NIST Special Publication 800-22, Rev. 1a. — 2010. — Access mode: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf (online; accessed: January 11th, 2020).

[245] *Stinson D.R.* Universal hashing and message authentication codes // *Designs, Codes, and Cryptography.* — 1994. — Vol. 4, no. 4. — P. 369–380.

[246] Technical Guideline BSI TR-03111. Elliptic curve cryptography : Rep. / German Federal Office for Information Security : 2018. — Access mode: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03111/BSI-TR-03111_V-2-1_pdf.pdf?__blob=publicationFile&v=2. — Version 2.10.

[247] *Teske E.* On Random Walks For Pollard's Rho Method // *Mathematics of Computation.* — 2000. — Vol. 70. — P. 809–825. — Access mode: https://www.ams.org/journals/mcom/2001-70-234/

S0025-5718-00-01213-8/S0025-5718-00-01213-8.pdf (online; accessed: February 15th, 2020).

[248] *Thayer F.J., Herzog J., Guttman J. D.* Strand spaces: Proving security protocols correct // *Journal of Computer Security*. — 1999. — Vol. 7, no. 2/3. — P. 191–230.

[249] *Tijdeman R.* On irrationality and transcendency of infinite sums of rational numbers // Diophantine Equations / Ed. by N. Saradha. — New Delhi, India : Narosa Publisher, 2008. — P. 279–296. — Access mode: http://www.math.leidenuniv.nl/~tijdeman/tijsho3.pdf (online; accessed: August 31th, 2017).

[250] Topics in Computational Number Theory Inspired by Peter L. Montgomery / Ed. by J.W. Bos, A.K. Lenstra. — Cambridge University Press, 2017. — P. 266.

[251] Transcendental Infinite Sums / S.D. Adhikari, N. Saradha, T.N. Shorey, R. Tijdeman // *Indag. Math.* — 2001. — Vol. 12. — P. 1–14. — Access mode: http://www.math.leidenuniv.nl/~tijdeman/asst.ps (online; accessed: August 31th, 2017).

[252] *Tu S.J., Fischbach E.* A study on the randomness of the digits of π // *International Journal of Modern Physics C*. — 2005. — Vol. 16, no. 2. — P. 281–294.

[253] Twisted Edwards Curves / D. Bernstein, P. Birkner, M. Joye et al. // Progress in Cryptology – AFRICACRYPT 2008. — 2008. — P. 389–405. — Access mode: https://eprint.iacr.org/2008/013.pdf (online; accessed: November 1st, 2021).

[254] UMAC: Fast and Secure Message Authentication / J. Black, Halevi S., H. Krawczyk et al. // Advances in Cryptology – Crypto 99. — Vol. 1666 Of Lecture Notes Of Computer Science. — Springer, 1999. — P. 216–233.

[255] *Weber H.* Lehrbuch der Algebra. — Chelsea. New York, 1903.

[256] *Wegman M.N., Carter J.L.* New Hash Functions and their Use in Authentication and Set Equality // *Journal of Computer and System Sciences*. — 1981. — Vol. 22, no. 3. — P. 265–279.

[257] *Whiting D., Housley R., Ferguson N.* Counter with CBC-MAC (CCM). — 2003. — RFC 3610. Access mode: https://tools.ietf.org/html/rfc3610 (online; accessed: January 15th, 2021).

[258] *Wiener M.J., Zuccherato R.J.* Faster Attacks on Elliptic Curve Cryptosystems // Selected Areas in Cryptography (SAC-98). — Berlin, Heidelberg : Springer Berlin Heidelberg, 1999. — P. 190–200.

[259] *Wu H., Preenel B.* AEGIS:A Fast Authenticated EncryptionAlgorithm (v1.1). — 2016. — Access mode: https://competitions.cr.yp.to/round3/aegisv11.pdf (online; accessed: May 31th, 2021).

[260] *Yee A.J.* World π record for both desktop and supercomputer. — 2012. — Access mode: http://www.numberworld.org/y-cruncher/ (online; accessed: March 3th, 2019).

[261] *Ylonen T., Lonvick C.* The Secure Shell (SSH) Transport Layer Protocol. — 2006. — RFC 4253. Access mode: https://tools.ietf.org/html/rfc4253 (online; accessed: January 15th, 2021).

[262] *Yoshida H.* Design and Analysis of Cryptographic Hash Functions : Doctoral dissertation. — Katholieke Universiteit Leuven / H. Yoshida. — 2013. — Access mode: https://www.esat.kuleuven.be/cosic/publications/thesis-200.pdf (online; accessed: July 20th, 2020).

[263] *Zimmermann P.* Implementation of PSLQ in GMP. — 2004. — Access mode: https://members.loria.fr/PZimmermann/software/pslq-1.1.c (online; accessed: August 31th, 2019).

[264] *Zivkovic M.* A Table Of Primitive Binary Polinomials // *Mathematics of Computation*. — 1994. — 01. — Vol. 62. — P. 385–386.

[265] *Акушский И.Я., Юдицкий Д.И.* Машинная арифметика в остаточных классах. — М. : Сов. радио, 1968. — С. 439.

[266] Алгебраическая теория чисел / Под ред. Дж. Касселс, А. Фрёлих. — М. : Мир, 1969. — С. 483.

[267] *Амербаев В.М.* Теоретические основы машинной арифметики. — Алма-Ата : Наука, АН КазССР, Ин-т математики и механики, 1976. — С. 324.

[268] *Анашин В.С.* Равномерно распределенные последовательности целых p-адических чисел // *Дискретная математика*. — 2002. — Т. 14, № 4. — С. 3–64.

[269] *Аносов В.Д., Нестеренко А.Ю.* Схема асимметричного шифрования, основанная на отечественных криптографических примитивах // Материалы IX международной конференции «Интеллектуальные системы и компьютерные науки» в МГУ, 23-27 октября 2006 г., Москва, Россия. — Т. 1 (часть 1). — 2006. — С. 45–47.

[270] *Бабаш А.В., Шанкин Г.П.* Криптография / Под ред. В.П. Шерстюка, Э.А. Применко. — М. : Солон-Пресс, 2007. — С. 512.

[271] *Бабуева А.А., Науменко А.П.* О подходах к анализу схем аутентифицированного шифрования, построенных с использованием умножения в конечных полях // Рускрипто 2018. — 2018. — Доступ: https://www.ruscrypto.ru/resource/archive/rc2018/files/02_Babueva_Naumenko.pdf (дата обращения: 31st May, 2021).

[272] *Билык Т.А., Нестеренко А.Ю.* Код аутентификации сообщений на основе универсального хэширующего преобразования // *Безопасность информационных технологий.* — 2012. — № 2. — С. 38–42. — Доступ: https://bit.mephi.ru/index.php/bit/article/download/463/468 (дата обращения: 24 декабря 2022 г.).

[273] *Боревич З.И., Шафаревич И.Р.* Теория чисел. — 3-е изд. — М. : Наука, 1985. — С. 503.

[274] *Борель Э.* Вероятность и достоверность. — М. : Мир, 1969. — С. 112.

[275] *Бухштаб А.А.* Теория чисел. — М. : Просвещение, 1966. — С. 384.

[276] *Василенко О.Н.* Теоретико-числовые алгоритмы в криптографии. — М. : МЦМНО, 2003. — С. 325.

[277] *Венков Б.А.* Элементарная теория чисел. Математика в монографиях. Серия обзоров, вып.4. — М. : ОНТИ НКТП СССР, 1937. — Р. 222.

[278] *Видякин В.В.* О связи скрытых информационных каналов и субпротоколов // *Обозрение прикл. и промышл. матем.* — 2006. — Т. 13, № 1. — С. 87–88.

[279] ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. — М. : ИПК Изд-во стандартов, 1996. — С. 26.

[280] ГОСТ Р 34.10–2012 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. — М. : Стандартинформ, 2012.

[281] ГОСТ Р 34.11–2012 Информационная технология. Криптографическая защита информации. Функция хэширования. — М. : Стандартинформ, 2012.

[282] ГОСТ Р 34.12–2015 Информационная технология. Криптографическая защита информации. Блочные шифры. — М. : Стандартинформ, 2015.

[283] ГОСТ Р 34.13–2015 Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров. — М. : Стандартинформ, 2015.

[284] ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. — М. : Стандартинформ, 2006.

[285] ГОСТ Р 53113.1-2008 Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Часть 1. Общие положения. — М. : Стандартинформ, 2008. — С. 12.

[286] ГОСТ Р 53113.2-2009 Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Часть 2. Рекомендации по организации защиты информации, информационных технологий и автоматизированных систем от атак с использованием скрытых каналов. — М. : Стандартинформ, 2009. — С. 12.

[287] ГОСТ Р 58833-2020. Защита информации. Идентификация и аутентификация. Общие положения. — М. : Стандартинформ, 2010. — С. 28.

[288] ГОСТ Р ИСО/МЭК 27033-1:2011. Информационная технология. Методы и средства обеспечения безопасности. Безопасность сетей. Часть 1. Обзор и концепции. — М. : Стандартинформ, 2012. — С. 73.

[289] ГОСТ Р ИСО/МЭК 9594-8-98. Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 8. Основы аутентификации. — М. : Госстандарт, 2001. — С. 29.

[290] *Гаусс К.Ф.* Труды по теории чисел. — М. : Изд-во Академии наук СССР, 1959. — С. 978.

[291] *Герман О.Н., Нестеренко Ю.В.* Теоретико-числовые методы в криптографии. — М. : Издательский центр «Академия», 2012. — С. 272.

[292] *Градштейн И.С., Рыжик И.М.* Таблицы интегралов, сумм, рядов и произведений. — 4 изд. — М. : Физматлит, 1963. — С. 1100.

[293] *Гребнев С.В.* О возможности стандартизации протоколов выработки общего ключа // Материалы конференции «РусКрипто-2014», 25 — 28 марта 2014, Солнечногорск. — 2014. — Доступ: https://www.ruscrypto.ru/resource/archive/rc2014/files/03_grebnev.pdf (дата обращения: октябрь, 2022).

[294] *Гурвиц А.* Теория аналитических и эллиптических функций. — М. : ГТТИ, 1933. — С. 344.

[295] *Жуков А.В.* Вездесущее число π. — 5-е изд. — М. : Либроком, 2012. — С. 240.

[296] *Запечников С.В.* Системы распределенного реестра как инструмент обеспечения доверия между участниками бизнес-процессов // *Безопасность информационных технологий*. — 2019. — Т. 26, № 4. — С. 37–53.

[297] *Иванов М.А., Чугунков И.В.* Теория, практика и оценка качества генераторов псевдослучайных последовательностей. — М. : Кудиц-Образ, 2003. — С. 240.

[298] *Ильясов И.И.* К распределению простых чисел в многочленах второй степени с целыми коэффициентами // *Чебышевский сборник*. — 2013. — Т. 4. — С. 56–60.

[299] Интеграция отечественных протоколов выработки общего ключа в протокол TLS 1.3 / Гребнев С.В., Лазарева Е.В., Лебедев П.А. и др. // Труды всероссийской конференции «Компьютерная безопасность и криптография» – SIBECRYPT-18 (Абакан, 3–8 сентября 2018 г.). — 2018. — С. 62–65.

[300] *Ищукова Е.А., Бабенко Л.К., Толоманенко Е.А.* Дифференциальный анализ шифра Кузнечик // *Известия ЮФУ. Технические науки*. — 2017. — Т. 5. — С. 25–37.

[301] *Карацуба Е.А.* Быстрое вычисление трансцендентных функций // *Проблемы передачи информации*. — 1991. — Т. 27.

[302] *Кейперс Л., Нидеррайтер Г.* Равномерно распределенные последовательности. — М. : Наука, 1985. — С. 408.

[303] *Кнут Д.Э.* Искусство программирования для ЭВМ. Получисленные алгоритмы. — 3 изд. — М. : Вильямс, 2000. — Т. 2. — С. 788.

[304] *Князев А.В., Ронжин А.Ф.* Инструментальный анализ мутных протоколов // *Обозрение прикл. и промышл. матем.* — 2007. — Т. 14, № 4. — С. 577–646.

[305] *Кобзарь А.И.* Прикладная математическая статистика для инженеров и научных работников. — М. : Физматлит, 2006. — С. 816.

[306] *Когос К.Г.* Метод противодействия утечке информации по скрытым каналам, основанным на изменении длин передаваемых пакетов : Диссертация на соискание ученой степени кандидата технических наук / К.Г. Когос ; НИЯУ МИФИ. — 2015.

[307] *Козлов М.В., Прохоров А.В.* Введение в математическую статистику. — М. : Изд-во МГУ, 1987. — С. 264.

[308] *Колчин В.Ф.* Случайные графы. — 2 изд. — М. : Физматлит, 2004. — С. 206.

[309] *Коробов Н.М.* О некоторых вопросах равномерного распределения // *Известия Академии наук СССР. Серия математическая.* — 1950. — Т. 14. — С. 215–231.

[310] *Крилли Т.* 50 идей о которых нужно знать. Математика. — М. : ФантомПресс, 2014. — С. 208.

[311] *Лебедев П.А., Нестеренко А.Ю.* Арифметика эллиптических кривых с использованием графических вычислителей // *Чебышовский сборник.* — 2012. — Т. 13, № 2. — С. 91–105. — Доступ: http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=cheb&paperid=40 (дата обращения: 2 февраля 2020 г.).

[312] *Лебедев П.А., Нестеренко А.Ю.* Режим шифрования с возможностью аутентификации // *Системы высокой доступности.* — 2013. — Т. 9, № 3. — С. 6–13.

[313] *Лемешко Б., Блинов П.* Критерии проверки отклонения распределения от равномерного закона. Руководство по применению. — М. : НИЦ Инфра-М, 2015. — С. 183. — ISBN: 978-5-16-011011-0. — Доступ: https://ami.nstu.ru/~headrd/seminar/

publik_html/test_random_lection.pdf (дата обращения: 15 января 2020).

[314] *Ленг С.* Эллиптические функции. — М. : Наука, 1984.

[315] *Лидл Р., Нидеррайтер Г.* Конечные поля. — М. : Мир, 1988. — Т. 1. — С. 430.

[316] *Лось А.Б., Нестеренко А.Ю., Рожков М.И.* Криптографические методы защиты информации. Учебник для академического бакалавриата. Серия: Бакалавр. Академический курс. — 2 изд. — М. : Издательство «Юрайт», 2016. — С. 473. — ISBN: 978-5-9916-9644-9.

[317] МР 26.2.001.–2022 Информационная технология. Криптографическая защита информации. Использование российских криптографических алгоритмов в протоколе обмена ключами в сети Интернет версии 2 (IKEv2). — М. : ТК26, 2022.

[318] *Матвеев С.В.* Некоторые подходы к оценке пропускной способности скрытых каналов в IP-сетях // *Системы высокой доступности.* — 2012. — Т. 8. — С. 68–71.

[319] *Матюхин Д.В.* О некоторых свойствах схем выработки общего ключа, использующих инфраструктуру открытых ключей, в контексте разработки стандартизированных криптографических решений // Материалы конференции «РусКрипто-2011», 30 марта – 2 апреля 2011, Солнечногорск. — 2011. — Доступ: https://www.ruscrypto.ru/resource/archive/rc2011/files/02_matyukhin.pdf (дата обращения: октябрь, 2022).

[320] Министерство энергетики Российской Федерации. Базовая модель угроз безопасности информации интеллектуальной системы учета электрической энергии. — 2021. — письмо НШ-7491/07 от 29.06.2021. Доступ: https://minenergo.gov.ru/system/download-pdf/20966/158908 (дата обращения: March 30th, 2022).

[321] *Миронкин В.О.* Явные формулы для распределений характеристик итераций случайных отображений : Диссертация на соискание ученой степени кандидата физико-математических наук / В.О. Миронкин ; МГУ им. М.В. Ломоносова. — 2021. — С. 134. — Доступ: https://istina.msu.ru/dissertations/396081864/ (дата обращения: 3 ноября 2021 г.).

[322] *Нестеренко А.Ю.* Схема асимметричного шифрования с возможностью аутентификации // Труды VII Международной научно-технической конференции «Новые информационные технологии и системы», Пенза, Россия. — 2006. — С. 104–107.

[323] *Нестеренко А.Ю.* Об одном варианте метода Ленстры факторизации целых чисел // Материалы третьей международной конференции по проблемам безопасности и противодействия терроризму в МГУ 25-27 октября 2007 г., Москва, Россия. — М. : Изд-во «МЦНМО», 2008. — С. 234–240.

[324] *Нестеренко Ю.В.* Теория чисел. — М. : Академия, 2008. — С. 272.

[325] *Нестеренко А.Ю.* Алгоритмы поиска длин циклов в последовательностях и их приложения // *Фундаментальная и прикладная математика.* — 2010. — Т. 16, № 6. — С. 109–122. — Доступ: http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=fpm&paperid=1354 (дата обращения: 2 февраля 2020 г.).

[326] *Нестеренко А.Ю.* О некоторых свойствах эллиптической кривой в форме Якоби // *Чебышовский сборник.* — 2010. — Т. 11, № 1. — С. 202–208.

[327] *Нестеренко А.Ю.* Новый протокол выработки общего ключа // *Системы высокой доступности.* — 2012. — № 2. — С. 81–90.

[328] *Нестеренко А.Ю.* О криптографических протоколах удаленного управления // *Проблемы информационной безопасности. Компьютерные системы.* — 2012. — № 2. — С. 76–82.

[329] *Нестеренко А.Ю.* О разложениях некоторого класса трансцендентных чисел // Тезисы докладов X-й международной конференции «Алгебра и теория чисел: современные проблемы и приложения», Волгоград, Россия. — 2012. — С. 51–52.

[330] *Нестеренко А.Ю.* О статистических свойствах некоторых трансцендентных чисел // *Ученые записки Орловского государственного университета.* — 2012. — № 6 (часть 2). — С. 170–176.

[331] *Нестеренко А.Ю.* Об одном протоколе выработки общего ключа // Материалы конференции «РусКрипто-2012», 28-31 марта 2012, Солнечногорск. — 2012. — Доступ: https://www.ruscrypto.ru/resource/archive/rc2012/files/03_nesterenko.pdf (дата обращения: октябрь, 2019).

[332] *Нестеренко А.Ю.* Алгоритм восстановления параметров одного класса иррациональных чисел // *Известия Саратовского университета. Серия: Математика. Механика. Информатика.* — 2013. — Т. 13, № 4 (часть 2). — С. 89–93. — Доступ: http://mi.mathnet.ru/isu466 (дата обращения: 2 февраля 2020 г.).

[333] *Нестеренко А.Ю.* Алгоритм построения эндоморфизмов эллиптических кривых // Тезисы докладов XI Международной конференции «Алгебра и теория чисел: современные проблемы и приложения», Саратов, Россия, 9-14 сентября. — 2013. — С. 63–64.

[334] *Нестеренко А.Ю.* Об одном подходе к построению защищенных соединений // *Математические вопросы криптографии*. — 2013. — Т. 4, № 2. — С. 101–111.

[335] *Нестеренко А.Ю.* Об одном алгоритме развертки ключа из пароля // Материалы конференции «РусКрипто-2015», 17-20 марта 2015, Солнечногорск. — 2015. — Доступ: https://www.ruscrypto.ru/resource/archive/rc2015/files/02_nesterenko.pdf (дата обращения: октябрь, 2019).

[336] *Нестеренко А.Ю.* Об одном семействе универсальных функций хеширования // *Математические вопросы криптографии*. — 2015. — Т. 6, № 3. — С. 135–151.

[337] *Нестеренко А.Ю.* Об одном подходе к построению схем шифрования с возможностью аутентификации // *Обозрение прикладной и промышленной математики.* — 2016. — Т. 23, № 5. — С. 478–479.

[338] *Нестеренко А.Ю.* Об одном подходе к разложению иррациональных чисел // *Математические вопросы криптографии*. — 2018. — Т. 9, № 1. — С. 89–106.

[339] *Нестеренко А.Ю.* О программной реализации алгоритмов шифрования с аутентификацией // Материалы конференции «РусКрипто-2021», 23-26 марта 2021, Солнечногорск. — 2021. — Доступ: https://www.ruscrypto.ru/resource/archive/rc2021/files/02_nesterenko.pdf (дата обращения: октябрь, 2021).

[340] *Нестеренко А.Ю., Лебедев П.А., Семенов А.М.* Краткий анализ криптографических механизмов защищенного взаимодействия контрольных и измерительных устройств, Технический комитет по стандартизации «Криптографическая защита информации». Серия б/н «Криптографические исследования». — 2019. — Доступ: https:

//tc26.ru/standarts/kriptograficheskie-issledovaniya/ (дата обращения: 27 февраля 2021 г.).

[341] *Нестеренко А.Ю., Пугачев А.В.* Об одной схеме гибридного шифрования // *Прикладная дискретная математика.* — 2015. — № 4. — С. 56–71. — Доступ: http://journals.tsu.ru/engine/download.php?id=60398&area=files (дата обращения: 24 декабря 2022 г.).

[342] *Нестеренко А.Ю., Семенов А.М.* Криптографические механизмы защищенного взаимодействия контрольных и измерительных устройств // *Безопасность информационных технологий.* — 2020. — Т. 27, № 4. — С. 7–16. — Доступ: https://bit.mephi.ru/index.php/bit/article/view/1301/1205.

[343] *Нестеренко А.Ю., Семенов А.М.* Методика оценки безопасности криптографических протоколов // *Прикладная дискретная математика.* — 2022. — № 56. — С. 33–82. — Доступ: http://journals.tsu.ru/engine/download.php?id=240427&area=files (дата обращения: 8 августа 2022 г.).

[344] *Нечаев В.И.* Элементы криптографии (Основы теории защиты информации). — М. : Высшая школа, 1999. — С. 109.

[345] Об основных концепциях криптографической стойкости / И.Ф. Качалин, А.С. Кузьмин, Е.А. Суслов и др. // Тезисы XII Всероссийской школы-коллоквиума по стохастическим методам и VI Всероссийского симпозиума по прикладной и промышленной математике. — 2005. — С. 982–983. — Сочи-Дагомыс, 1-7 октября 2005 г.

[346] Обзор уязвимостей некоторых протоколов выработки общего ключа с аутентификацией на основе пароля и принципы построения протокола *SESPAKE* / Е. К. Алексеев, Л. Р. Ахметзянова, И. Б. Ошкин, С. В. Смышляев // *Математические вопросы криптографии.* — 2016. — Vol. 7, no. 4. — P. 7–28.

[347] Основы криптографии / А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин, А.В. Черемушкин. — 2 изд. — М. : Гелиос АРВ, 2002. — С. 480. — ISBN: 5-85438-025-0.

[348] *Панасенко С.П.* Алгоритмы шифрования. Специальный справочник. — СПб. : БХВ-Петербург, 2009. — С. 576. — ISBN: 978-5-9775-0319-8.

[349] *Понтрягин Л.С.* Обыкновенные дифференциальные уравнения. — М. : Наука, 1974. — С. 331.

[350] *Постников А.Г.* Арифметическое моделирование случайных процессов. Труды МИАН СССР. — М. : Изд-во АН СССР, 1960. — С. 84.

[351] Поточные шифры. Результаты зарубежной открытой криптологии. — 1997. — Доступ: https://kiwiarxiv.files.wordpress.com/2016/02/potochnye_shifry_stream_ciphers_ru_1997.pdf (дата обращения: 31 августа 2017 г.).

[352] Р 1323565.1.004.–2017 Информационная технология. Криптографическая защита информации. Схемы выработки общего ключа с аутентификацией на основе открытого ключа. — М. : Стандартинформ, 2017.

[353] Р 1323565.1.005.–2017 Информационная технология. Криптографическая защита информации. Допустимые объёмы материала для обработки на одном ключе при использовании некоторых вариантов режимов работы блочных шифров в соответствии с ГОСТ Р 34.13-2015. — М. : Стандартинформ, 2017.

[354] Р 1323565.1.006.–2017 Информационная технология. Криптографическая защита информации. Механизмы выработки псевдослучайных последовательностей. — М. : Стандартинформ, 2017.

[355] Р 1323565.1.012.–2017 Информационная технология. Криптографическая защита информации. Принципы разработки и модернизации шифровальных (криптографических) средств защиты информации. — М. : Стандартинформ, 2017.

[356] Р 1323565.1.017.–2018 Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов блочного шифрования. — М. : Стандартинформ, 2018.

[357] Р 1323565.1.018.–2018 Информационная технология. Криптографическая защита информации. Криптографические механизмы аутентификации в контрольных устройствах для автотранспорта. — М. : Стандартинформ, 2018.

[358] Р 1323565.1.019.–2018 Информационная технология. Криптографическая защита информации. Криптографические механизмы аутентификации и выработки ключа фискального признака для применения в средствах формирования и проверки фискальных признаков, обеспечивающих работу контрольно-кассовой техники, операторов

и уполномоченных органов обработки фискальных данных. — М. : Стандартинформ, 2018.

[359] Р 1323565.1.020.–2018 Информационная технология. Криптографическая защита информации. Использование криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.2). — М. : Стандартинформ, 2018.

[360] Р 1323565.1.022.–2018 Информационная технология. Криптографическая защита информации. Функции выработки производного ключа. — М. : Стандартинформ, 2018.

[361] Р 1323565.1.023.–2018 Информационная технология. Криптографическая защита информации. Использование алгоритмов ГОСТ Р 34.10-2012, ГОСТ Р 34.11-2012 в сертификате, списке аннулированных сертификатов (CRL) и запросе на сертификат PKCS #10 инфраструктуры открытых ключей X.509. — М. : Стандартинформ, 2018.

[362] Р 1323565.1.024.–2019 Информационная технология. Криптографическая защита информации. Параметры эллиптических кривых для криптографических алгоритмов и протоколов. — М. : Стандартинформ, 2019.

[363] Р 1323565.1.025.–2019 Информационная технология. Криптографическая защита информации. Форматы сообщений, защищенных криптографическими методами. — М. : Стандартинформ, 2019.

[364] Р 1323565.1.026.–2019 Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров, реализующие аутентифицированное шифрование. — М. : Стандартинформ, 2019.

[365] Р 1323565.1.028.–2019 Информационная технология. Криптографическая защита информации. Криптографические механизмы защищенного взаимодействия контрольных и измерительных устройств. — М. : Стандартинформ, 2019.

[366] Р 1323565.1.029.–2019 Информационная технология. Криптографическая защита информации. Протокол защищенного обмена для индустриальных систем. — М. : Стандартинформ, 2019.

[367] Р 1323565.1.030.–2020 Информационная технология. Криптографическая защита информации. Использование криптографических ал-

горитмов в протоколе безопасности транспортного уровня (TLS 1.3). — М. : Стандартинформ, 2020.

[368] Р 1323565.1.032.–2020 Информационная технология. Криптографическая защита информации. Использование российских криптографических механизмов для реализации обмена данными по протоколу DLMS. — М. : Стандартинформ, 2020.

[369] Р 1323565.1.034.–2020 Информационная технология. Криптографическая защита информации. Протокол безопасности сетевого уровня. — М. : Стандартинформ, 2020.

[370] Р 1323565.1.035.–2021 Информационная технология. Криптографическая защита информации. Использование российских криптографических алгоритмов в протоколе защиты информации ESP. — М. : Стандартинформ, 2021.

[371] Р 50.1.111.–2016 Информационная технология. Криптографическая защита информации. Парольная защита ключевой информации. — М. : Стандартинформ, 2016. — Доступ: https://tc26.ru/standard/rs/%D0%A0%2050.1.111-2016.pdf (дата обращения: 15 января 2020).

[372] Р 50.1.113.–2016 Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования. — М. : Стандартинформ, 2016. — Доступ: https://tc26.ru/standard/rs/%D0%A0%2050.1.113-2016.pdf (дата обращения: 15 января 2020).

[373] Р 50.1.115.–2016 Информационная технология. Криптографическая защита информации. Протокол выработки общего ключа с аутентификацией на основе пароля. — М. : Стандартинформ, 2016. — Доступ: https://tc26.ru/standard/rs/%D0%A0%2050.1.115-2016.pdf (дата обращения: 15 января 2022).

[374] *Ростовцев А.Г.* О выборе эллиптической кривой над простым полем для построения криптографических алгоритмов // *Проблемы информационной безопасности. Компьютерные системы.* — 1999. — Т. 3. — С. 37–40.

[375] *Ростовцев А.Г.* Арифметика эллиптических кривых над простыми полями без удвоения точек // *Проблемы информационной безопасности. Компьютерные системы.* — 2000. — Т. 4.

[376] Свидетельство о государственной регистрации программы для ЭВМ № 2018666094 «Библиотека криптографических механизмов защиты контрольных цифровых устройств». — 2018. — Авторы: Жуков И.Ю., Мурашов О.Н., Нестеренко А.Ю., Решетник В.В. Доступ: https://www1.fips.ru/fips_servl/fips_servlet?DB=EVM&DocNumber=2018666094 (дата обращения: 24 декабря 2022 г.).

[377] Свидетельство о государственной регистрации программы для ЭВМ № 2018666095 «Криптографические механизмы аутентификации в контрольных устройствах для автотранспорта». — 2018. — Авторы: Жуков И.Ю., Мурашов О.Н., Нестеренко А.Ю., Решетник В.В. Доступ: https://www1.fips.ru/fips_servl/fips_servlet?DB=EVM&DocNumber=2018666095 (дата обращения: 24 декабря 2022 г.).

[378] Свидетельство о государственной регистрации программы для ЭВМ № 2018666420 «Криптографические механизмы защищенного взаимодействия контрольных и измерительных устройств». — 2018. — Авторы: Жуков И.Ю., Мурашов О.Н., Нестеренко А.Ю., Решетник В.В. Доступ: https://www1.fips.ru/fips_servl/fips_servlet?DB=EVM&DocNumber=2018666420 (дата обращения: 24 декабря 2022 г.).

[379] Свидетельство о государственной регистрации программы для ЭВМ № 2018666512 «Криптографические механизмы аутентификации и выработки ключа фискального признака». — 2018. — Авторы: Жуков И.Ю., Мурашов О.Н., Нестеренко А.Ю., Решетник В.В. Доступ: https://www1.fips.ru/fips_servl/fips_servlet?DB=EVM&DocNumber=2018666512 (дата обращения: 24 декабря 2022 г.).

[380] *Семенов А.М.* Analysis of Russian key-agreement protocols using automated verification tools // *Математические вопросы криптографии*. — 2017. — Т. 8. — С. 131–142. — Доступ: http://mi.mathnet.ru/mvk229 (дата обращения: March 30th, 2020).

[381] *Семенов А.М.* Методы защищенной передачи данных для низкоресурсных вычислительных устройств : Диссертация на соискание ученой степени кандидата техических наук / А.М. Семенов ; Национальный исследовательский университет «Высшая школа экономики». — 2022. — С. 149. — Доступ: https://www.hse.ru/data/xf/933/046/1611/%D0%94%D0%B8%D1%

81%D1%81%D0%B5%D1%80%D1%82%D0%B0%D1%86%D0%B8%D1%8F%20%D0%A1%D0%B5%D0%BC%D0%B5%D0%BD%D0%BE%D0%B2.pdf (дата обращения: 26 августа 2022 г.).

[382] Семинар по комплексному умножению / Ж.-П. Серр, А. Борель, К. Ивасава, Чоула // *Математика*. — 1968. — Т. 12. — С. 55–95. — Доступ: http://mi.mathnet.ru/mat457 (дата обращения: March 30th, 2020).

[383] Словарь криптографических терминов / Под ред. Б.А. Погорелов, В.Н. Сачков. — М. : МЦМНО, 2006. — С. 88.

[384] *Степанов С.А.* Арифметика алгебраических кривых. — М. : Наука. Гл. ред. физ.-мат. лит., 1991. — С. 368.

[385] *Тихомандрицкій М.* Теорія эллиптическихъ интеграловъ и эллиптическихъ функцій. — Харьковъ : Типографія Зильберберга, 1895. — С. 449.

[386] *Торгашев В.А.* Система остаточных классов и надежность ЦВМ. — М. : Сов. радио, 1973. — С. 118.

[387] ФСТЭК России. Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных. — 2008. — в редакции 2013 г. Доступ: https://fstec.ru/component/attachments/download/289 (дата обращения: March 30th, 2022).

[388] ФСТЭК России. Руководящий документ. Методика оценки угроз безопасности информации. — 2021. — Доступ: https://fstec.ru/component/attachments/download/2919 (дата обращения: March 30th, 2022).

[389] *Феллер В.* Введение в теорию вероятностей и её приложения. — М. : Мир, 1963. — Т. 1. — С. 511.

[390] *Фихтенгольц Г.М.* Курс дифференциального и интегрального исчисления. — М. : ГИТТЛ, 1947. — Т. 2.

[391] *Черемушкин А.В.* Криптографические протоколы: основные свойства и уязвимости // *ПДМ. Приложение к № 2*. — 2009. — С. 115–150. — Доступ: mi.mathnet.ru/pdm141 (дата обращения: 31 Августа 2022 г.).

[392] *Чирский В.Г., Нестеренко А.Ю.* Об одном подходе к преобразованию периодических последовательностей // *Дискретная математика*. — 2015. — Т. 27, № 4. — С. 150–157.

[393] *Шафаревич И.Р.* Основы алгебраической геометрии. — М. : Наука, 1972.

[394] *Шидловский А.Б.* Трансцендентные числа. — М. : Наука, 1987. — С. 417.

[395] *Шнайер Б., Фергюссон Н.* Практическая криптография. — М. : Диалектика, 2005. — С. 420.

Тексты программ из главы 1

В настоящем приложении приводятся тексты программ, реализующие ряд описанных в 1-й главе алгоритмов. Все программы написаны для системы компьютерной алгебры Magma [150].

§ А.1. Эндоморфизмы эллиптических кривых

В § 1.4.2 излагается алгоритм построения эндоморфизма ϕ_α для эллиптической кривой $\mathcal{E}_{\Lambda_\tau}(\mathbb{C})$, см. алгоритм 1.6. Реализация данного алгоритма приводится ниже.

```
// ------------------------------------------------------------------------------
// Построение кривой и эндоморфизма (файл endm2.m)

// ------------------------------------------------------------------------------
// Функция строит эллиптическую кривую вида y^2 = 4x^3 - g2x - g3
// При построении коэффициенты приводятся таким образом, что
// знаменатель коэффициента g2 равняется единице, а числитель содержит
// произведение неприводимых сомножителей в первой степени (j - целое число)
// ------------------------------------------------------------------------------
WeierstrassCurveConstructFromJ := function( j )
  k := j/(j-1728); g2 := 3*k; g3 := k; t := 1;
  print "k = ", k;

  cf := Factorization(Denominator(g2));
  for i in [1 .. #cf] do
    alpha := cf[i][2] div 2;
    if IsEven( cf[i][2] ) then
     t := t * ( cf[i][1]^alpha );
    else
     t := t * ( cf[i][1]^(alpha+1));
    end if;
  end for;

  cf := Factorization(Numerator(g2));
  for i in [1 .. #cf] do
    if cf[i][2] gt 1 then
      alpha := cf[i][2] div 2;
      t := t / ( cf[i][1]^alpha );
    end if;
  end for;
```

```
  g2 := g2*t^2;
  g3 := g3*t^3;

  if Denominator(g3) gt 1 then
    cf := Factorization( Denominator(g3) );
    for i in [1 .. #cf] do
      alpha := cf[i][2] div 3;
      g3 := g3*cf[i][1]^(3*alpha);
      g2 := g2*cf[i][1]^(2*alpha);
      t := t*cf[i][1];
    end for;
  end if;

 // величины g2/t^2, g3/t^3 дадут нам исходные величины
 return [g2,g3,t];
end function;

// ------------------------------------------------------------------------
// Функция строит эллиптическую кривую вида y^2 = 4x^3 - g2x - g3
// с коэффициентами из конечного расширения L = Q(j(tau))
// ------------------------------------------------------------------------
WeierstrassCurveConstructFromJ2 := function( j )
       k := j/(j-1728);
       g2 := 3*k;
       g3 := k;
       t := 1;
       cf := Factorization( Denominator(g2));

     // удаляем квадраты из знаменателя дроби
       for i in [1..#cf] do
         alpha := cf[i][2] div 2;
         beta := ( cf[i][1] )^alpha;

         g2 := g2*beta^2;
         g3 := g3*beta^3;
         t := t*beta;
       end for;

 // величины g2/t^2, g3/t^3 дадут нам исходные величины
 return [g2,g3,t];
end function;

// ------------------------------------------------------------------------
// Строим объекты, с которыми потом будем проводить вычисления
// ------------------------------------------------------------------------
 SetDefaultRealField(RealField(4096));
 Q := RationalField();
 Qx<x> := PolynomialRing(Q);

 Params := {@ car< Integers(), Integers() > |

    < -1, 2 >,  <  -2, 1 >,  <  -3, 2 >,  <  -3, 3 >,  <  -7, 1 >,  <   -7, 2 >,
```

```
 <  -11, 1 >, <  -19, 1 >, <  -43, 1 >,  < -67, 1 >, < -163, 1 >,

   < -5, 1 >,   < -6, 1 >,  < -10, 1 >,  < -13, 1 >,  < -15, 1 >,  <  -22, 1 >,
  < -35, 1 >,  < -37, 1 >,  < -51, 1 >,  < -58, 1 >,  < -91, 1 >,  < -115, 1 >,
 < -123, 1 >, < -187, 1 >, < -235, 1 >, < -267, 1 >, < -403, 1 >,  < -427, 1 >,

  < -23, 1 >,  < -31, 1 >,  < -59, 1 >,  < -83, 1 >, < -107, 1 >,  < -139, 1 >,
 < -211, 1 >, < -283, 1 >, < -307, 1 >, < -331, 1 >, < -379, 1 >,  < -499, 1 >,
 < -547, 1 >, < -643, 1 >

@};

for Index in [ 1 .. 1 ] do
  K<Sqrt_d> := QuadraticField( Params[Index][1] );
  Zk<omg> := MaximalOrder(K);
  Ok<tau> := sub< Zk | Params[Index][2] >;
  mp := MinimalPolynomial( tau );
  print "mp = ", mp;

  D := Integers()! Discriminant( mp );
  hp := HilbertClassPolynomial(D);
  print "d = ", Params[Index][1], "(Degree: ", Degree( hp ), ")";
  print "H(x) = ", hp;

  if Degree(hp) eq 1 then
      j := Integers()!( -1*Coefficient(hp, 0));
      print "j = ", j;
      wcc := WeierstrassCurveConstructFromJ(j);
      L := Rationals();

      g2 := wcc[1];
      g3 := wcc[2];
      print "g2 = ", g2;
      print "g3 = ", g3;
      print "t = ", wcc[3];
  else
      L<j> := NumberField( hp );
      wcc := WeierstrassCurveConstructFromJ2(j);

      g2 := wcc[1];
      g3 := wcc[2];
      print "g2 = ", g2, "g3 = ", g3;

  end if;

  print "K:", K;
  print "L:", L;
  H<xi> := Compositum( K,L );
  print "H:", H;
  P<x> := PolynomialRing(H);
  print "P: ", P;

  g2 := H!g2;
```

```
g3 := H!g3;
alpha := H!tau;
print "g2(H) = ", g2;
print "g3(H) = ", g3;
print "alpha = ", alpha;

// определяем коэффициенты разложения wp(z) в ряд:  c_0, c_1, c_2, c_3
c := [ 1, 0, g2/20, g3/28 ];
N := 4 + 2*Norm(tau);
for n in [5 .. N] do
  s := 0;
  for k in [ 3 .. n-1 ] do
     s := s + c[k]*c[n+1-k];
  end for;
  c[n] := (3*s)/((n-1-3)*(2*(n-1)+1));
end for;
print "wp(z) = ", c;

// вычисляем коэффициенты wp(alpha.z) (d_0,n)
d := [ alpha^(-2), 0 ];
for n in [3 .. N] do
  d[n] := c[n]*alpha^(2*(n-1)-2);
end for;
print "wp(alpha.z) = ", d;

// определяем многочлены
Pm1 := 1;
Qm1 := 0;
P0 := d[1]*x+d[2];
Q0 := 1;
print "l[ 0 ] = ", P0;

// переходик к основному циклу
for k in [1 .. Norm(tau) -1] do

 // уменьшаем точность
  N := N -2;
 // вычисляем коэффициенты e
  em1 := d[3]-d[1]*c[3];
  if em1 eq 0 then
    print "Wrong e_{-1}";
    exit;
  end if;

  e := [ 1, ( d[4] - d[1]*c[4] )/em1 ];
  u := [ 1, -e[2] ];
  for n in [ 3 .. N ] do
    e[n] := ( d[n+2] - d[1]*c[n+2] )/em1;
    u[n] := e[n];
    for m in [ 2 .. n-1 ] do
      u[n] := u[n] + e[m]*u[n+1-m];
    end for;
    u[n] := -u[n];
```

```
  end for;

  if e[1]*u[1] +
   ( e[2] + u[2] ) +
   ( e[3] + e[2]*u[2] + u[3] ) +
   ( e[4] + e[2]*u[3] + e[3]*u[2] + u[4] ) +
   ( e[5] + e[2]*u[4] + e[3]*u[3] + e[4]*u[2] + u[5] ) +
   ( e[6] + e[2]*u[5] + e[3]*u[4] + e[4]*u[3] + e[5]*u[2] + u[6] ) ne 1 then
   print "Wrong inversion";
   exit;
  end if;

  for n in [1 .. N] do
    d[n] := u[n]/em1;
  end for;
  d[N+1] := 0;
  d[N+2] := 0;

  l := d[1]*x + d[2];
  print "l[", k, "] = ", l;

  t := P0*l + Pm1; Pm1 := P0; P0 := t;
  t := Q0*l + Qm1; Qm1 := Q0; Q0 := t;
  print "";

end for; // конец цикла по k, т.е. конец построения R(x) = P0/Q0

  Rx := P0/Q0;
  P0 := Numerator( Rx );
  Q0 := Denominator( Rx );
  xp := Denominator( Coefficient( P0, 0 ));
  for k in [1 .. Degree(P0)] do
    xp := LCM( xp, Denominator( Coefficient( P0, k )));
  end for;

  xq := Denominator( Coefficient( Q0, 0 ));
  for k in [1 .. Degree(Q0)] do
    xq := LCM( xq, Denominator( Coefficient( Q0, k )));
  end for;

  P0 := P0*LCM( xp, xq );
  Q0 := Q0*LCM( xp, xq );
  Rx := P0/Q0;

 // вариант многочленов с целыми коэффициентами
  print "Pm = ", P0;
  print "Qm = ", Q0;
  print "R(x)    =  ", Rx;

  Pd := 0;
  for k in [1 .. Degree(P0) ] do
    Pd := Pd + Coefficient( P0, k )*k*x^(k-1);
  end for;
```

```
    Qd := 0;
    for k in [1 .. Degree(Q0) ] do
      Qd := Qd + Coefficient( Q0, k )*k*x^(k-1);
    end for;
    Ry := ( Pd*Q0 - Qd*P0) / ( Q0^2 );

    S0 := Numerator( Ry );
    T0 := Denominator( Ry );
    xs := Denominator( Coefficient( S0, 0 ));
    for k in [1 .. Degree(S0)] do
      xs := LCM( xs, Denominator( Coefficient( S0, k )));
    end for;

    xt := Denominator( Coefficient( T0, 0 ));
    for k in [1 .. Degree(T0)] do
      xt := LCM( xt, Denominator( Coefficient( T0, k )));
    end for;

    S0 := S0*LCM( xs, xt );
    T0 := T0*LCM( xs, xt );
    Ry := S0/T0;

   // вариант многочленов с целыми коэффициентами
    print "";
    print "Sm = ", S0;
    print "Tm = ", T0;
    print "R'x(x) = ", Ry;

    Ry := Ry/alpha;
    if ( 4*Rx^3 - g2*Rx - g3 ) eq ( Ry^2*(4*x^3-g2*x-g3) ) then
      print "Result correct";
    else
      print "Result incorrect";
      exit;
    end if;
    print "=======================================";
    print "";
end for;
```

Для проверки предъявленных в § 1.4.3 эндоморфизмов разработана отдельная программа, содержащая явный вид эндоморфизмов для всех эллиптических кривых, перечисленных в таблице 1.4. Текст данной программы приводится ниже.

```
// ------------------------------------------------------------------------------
// Проверка корректности построенных отображений (файл endm-check.m)

// ------------------------------------------------------------------------------
// Кривая 1.
 K<xi> := QuadraticField( -1 );
 R<x> := PolynomialRing(K);
```

```
alpha := 2*xi;

g2 := 11; g3 := 7;
Rx := -(16*x^4 + 64*x^3 + 216*x^2 + 304*x + 137) / ( 16*(x+1)*(2*x+3)^2 );
Ry := (-32*x^5 - 208*x^4 - 272*x^3 - 8*x^2 + 134*x + 47 ) /
                                          ( 16*alpha*((x+1)^2)*((2*x+3)^3) );

if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
  print "Weierstrass form of curve N1: correct";
end if;

u := (x+1)*2;
A := -6; B := 1;

Ru := -(u-1)^2*(u^3+A*u^2+B*u )/(4*u^2*(u+1)^2);
Rv := -(u^5 + 3*u^4 - 30*u^3 + 30*u^2 -3*u -1)/(4*alpha*u^2*(u+1)^3);

if ( Ru^3 + A*Ru^2 +B*Ru ) eq ( (u^3+A*u^2+B*u)*Rv^2 ) then
  print "Hadano form of curve N1: correct";
end if;

// ------------------------------------------------------------------------------
// Кривая 2.
K<xi> := QuadraticField( -2 );
R<x> := PolynomialRing(K);
alpha := xi;

g2 := 30; g3 := 28;
Rx := -(2*x^2 + 4*x + 9) / ( 4*(x+2) );
Ry := (-2*x^2 - 8*x + 1 ) / ( 4*alpha*(x+2)^2 );

if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
  print "Weierstrass form of curve N2: correct";
end if;

u := (x+2)*2/3;
A := -4; B := 2;

Ru := -( u^3+A*u^2+B*u )/(2*u^2);
Rv := (2-u^2)/(2*alpha*u^2);

if ( Ru^3 + A*Ru^2 +B*Ru ) eq ( (u^3+A*u^2+B*u)*Rv^2 ) then
  print "Hadano form of curve N2: correct";
end if;

// ------------------------------------------------------------------------------
// Кривая 3.
K<xi> := QuadraticField( -3 );
R<x> := PolynomialRing(K);
alpha := xi;

g2 := 15; g3 := 11;
Rx := -(4*x^3 + 12*x^2 + 33*x + 28) / ( 3*(2*x+3)^2 );
```

```
Ry := (-8*x^3 - 36*x^2 - 6*x + 13) / ( 3*alpha*(2*x+3)^3 );

if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
  print "Weierstrass form of curve N3: correct";
end if;

u := (x+1)*2;
A := -6; B := -3;

Ru := -u*(u-3)^2/(3*(u+1)^2);
Rv := -(u^3 + 3*u^2 - 21*u + 9)/(3*alpha*(u+1)^3);

if ( Ru^3 + A*Ru^2 +B*Ru ) eq ( (u^3+A*u^2+B*u)*Rv^2 ) then
  print "Hadano form of curve N3: correct";
end if;

// ------------------------------------------------------------------------------
// Кривая 4.
K<xi> := QuadraticField( -3 );
R<x> := PolynomialRing(K);
alpha := 3/2*(xi+1);

g2 := 120; g3 := 253;
f:=(x+3)*(x^3 + (15-3*alpha)*x^2 + (57-18*alpha)*x + (62-27*alpha));
Rx := (1/(27*f^2))*( -alpha*x^9 - 18*(alpha +3)*x^8 + 9*(50*alpha -297)*x^7 +
                        6*(2021*alpha -6210)*x^6 + 9*(12538*alpha -26529)*x^5 +
                    9*(61340*alpha -85743)*x^4 + 3*(519281*alpha -349893)*x^3 +
                     36*(70243*alpha +13647)*x^2 + 9*(239966*alpha +335061)*x +
                                                    728569*alpha + 2442393 );
Ry := (-1/(27*alpha*f^3))*
      (47 + 33*x + 9*x^2 + x^3)*
 ( alpha*x^9 + 9*(9 + 2*alpha)*x^8 + 9*(81 + 40*alpha)*x^7 +
   18*x^4*(-17523 + 470*alpha) + 9*x^5*(-7227 + 1988*alpha) +
   x^6*(-3159 + 3912*alpha) - 18*x^2*(1575 + 51791*alpha) -
   3*x^3*(200151 + 77831*alpha) - 9*x*(-156789 + 166580*alpha)
                                          + 1385100 - 912277*alpha );

if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
  print "Curve N4: correct";
end if;

// ------------------------------------------------------------------------------
// Кривая 5.
K<xi> := QuadraticField( -7 );
R<x> := PolynomialRing(K);
alpha := 1/2*(xi+1);

g2 := 35; g3 := 49;
Rx :=  (- 4*(alpha +1)*x^2 - 4*(alpha +3)*x + 7*(5*alpha -7))/
                                                  (8*(2*x - alpha + 4));
Ry := -( 4*(alpha +1)*x^2 + 8*(alpha +3)*x + 7*(5*alpha - 3))/
                                        (4*alpha*(2*x -alpha + 4)^2);
```

```
if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
  print "Weierstrass form of curve N5: correct";
end if;

theta := (alpha-4)/2;
gamma := -(alpha+10)/112;

u := (x-theta)*gamma;

A := 3*theta*gamma;                      // -3*(alpha-6)/32;
B := (3*theta^2-g2/4)*gamma^2;           // -(3*alpha-2)/64
C := (theta^3-g2*theta/4-g3/4)*gamma^3; // 0

mu := (alpha-2)/16;
Ru := -( (alpha+1)*u^2 + u - mu )/(4*u);
Rv := -( (alpha+1)*u^2 + mu )/(4*alpha*u^2);

if ( Ru^3 + A*Ru^2 +B*Ru ) eq ( (u^3+A*u^2+B*u)*Rv^2 ) then
  print "Hadano form of curve N5: correct";
end if;

// ------------------------------------------------------------------------------
// Кривая 6.
K<xi> := QuadraticField( -7 );
R<x> := PolynomialRing(K);
alpha := xi;

g2 := 595; g3 := 2793;
f := 8*x^3+252*x^2+2422*x+7357;
Rx := (-1/(7*f^2))*(64*x^7 + 4032*x^6 + 193648*x^5 + 4900000*x^4 +
                        65275644*x^3 + 472046204*x^2 + 1765121561*x + 2683223144);
Ry := (1/(7*alpha*f^3))*( -512*x^9 - 48384*x^8 - 1257984*x^7 - 12757248*x^6 -
            18411456*x^5 + 695999136*x^4 + 6221339488*x^3 + 22527872304*x^2 +
                                                  34125504238*x + 11533585259 );

if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
  print "Curve N6: correct";
end if;

// ------------------------------------------------------------------------------
// Кривая 7.
K<xi> := QuadraticField( -11 );
R<x> := PolynomialRing(K);
alpha := 1/2*(xi+1);

g2 := 264; g3 := 847;

Rx := -( (alpha +2)*x^3 + 6*(alpha +5)*x^2 - 33*(4*alpha -13)*x -
                                          11*(59*alpha - 134))/(9*(x-alpha+6)^2);
Ry := -( (alpha +2)*x^3 + 9*(alpha +5)*x^2 + 33*(4*alpha -1)*x +
                                        11*(19*alpha-70))/(9*alpha*(x-alpha+6)^3);

if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
```

```
  print "Curve N7: correct";
 end if;

// ------------------------------------------------------------------------------
// Кривая 8.
 K<xi> := QuadraticField( -19 );
 R<x> := PolynomialRing(K);
 alpha := 1/2*(xi+1);

 g2 := 152; g3 := 361;
 f := 5*x^2+5*(10-alpha)*x+(114-19*alpha);
 Rx := (1/(f^2))*(-(alpha +4)*x^5-10*(alpha +9)*x^4 + 95*(2*alpha -17)*x^3 +
                + 380*(7*alpha -32)*x^2+5415*(2*alpha -7)*x + 1805*(8*alpha -23));
 Ry := (-5/(alpha*f^3))*((alpha +4)*x^6 + 15*(alpha +9)*x^5 + 209*(alpha +4)*x^4 +
                        + 19*(73*alpha -8)*x^3 + 1083*(3*alpha -13)*x^2 -
                                              361*(alpha +104)*x - 6859*(alpha +4));

 if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
  print "Curve N8: correct";
 end if;

// ------------------------------------------------------------------------------
// Кривая 9.
 K<xi> := QuadraticField( -43 );
 R<x> := PolynomialRing(K);
 alpha := 1/2*(xi+1);

 g2 := 3440; g3 := 38829;
 f := 11*x^5 - 66*(alpha-22)*x^4 - 473*(11*alpha-145)*x^3 -
    473*(320*alpha -3213)*x^2 - 20339*(95*alpha -794)*x - 1849*(4951*alpha-36037);

 Rx :=  ( - (alpha +10)*x^(11)
          - 132*(alpha +21)*x^(10)
          + 946*(20*alpha - 581)*x^9
          + 473*(10717*alpha -126382)*x^8
          + 101695*(4505*alpha -37219)*x^7
          + 6996616*(3287*alpha -21635)*x^6
          + 874577*(828781*alpha -4587030)*x^5
          + 874577*(17123327*alpha -82078613)*x^4
          + 37606811*(5410144*alpha -22883347)*x^3
          + 75213622*(23349061*alpha -88276902)*x^2
          + 1617092873*(5412111*alpha -18461213)*x
          + 1617092873*(11878584*alpha -36810755))/(f^2);
 Ry := -11*((alpha +10)*x^(15)
          + 198*(alpha +21)*x^(14)
          + 43*(560*alpha +12497)*x^(13)
          + 129*(16759*alpha +266326)*x^(12)
          + 22188*(5991*alpha +53981)*x^(11)
          + 5547*(957664*alpha +3343567)*x^(10)
          + 636056*(204783*alpha -361643)*x^9
          + 238521*(6346253*alpha -79067847)*x^8
          - 10256403*(1319376*alpha +45425861)*x^7
          - 27350408*(32732861*alpha +234409806)*x^6
```

```
          - 441025329*(40240641*alpha +107283901)*x^5
          - 294016886*(695797144*alpha +172857357)*x^4
          - 151712713176*(9706277*alpha -16452654)*x^3
          - 37928178294*(167920463*alpha -641046237)*x^2
          - 271818611107*(52619384*alpha -372703907)*x
          - 271818611107*(39005407*alpha -616133530))/(alpha*f^3);

 if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
   print "Curve N9: correct";
 end if;

// ------------------------------------------------------------------------------
// Кривая 10.
 K<xi> := QuadraticField( -67 );
 R<x> := PolynomialRing(K);
 alpha := 1/2*(xi+1);

 g2 := 29480; g3 := 974113;
 Rx := (1/578*(-xi - 33)*x^17
      + 1/17*(-19*xi - 1273)*x^16
      + 1/17*(11055*xi - 1146638)*x^15
      + 1/17*(16322875*xi - 607994649)*x^14
      + 1/34*(16035143433*xi - 401864361247)*x^13
      + 1/17*(2294319047692*xi - 44890234491893)*x^12
      + 1/34*(882685977159917*xi - 14366812033835585)*x^11
      + 1/17*(60988987693314091*xi - 855052803772988031)*x^10
      + 1/17*(628066311670928 7404*xi - 77535579280699142825)*x^9
      + 1/34*(9841737033881496 58735*xi - 10861930752377458583767)*x^8
      + 1/17*(2963095674319232 0366560*xi - 2955988235451137248 01867)*x^7
      + 1/17*(1373289869498134743859211*xi - 1248631588241467282 0731381)*x^6
      + 1/17*(4864677410409550975 0360043*xi - 40569619903960008876 4369909)*x^5
      + 1/17*(1294744171446986407502557130*xi - 995296654594084504707 2812707)*x^4
      + 1/34*(5016728176013331283822006 2351*xi - 3568534686399068141876929 61103)*x^3
      + 1/17*(3341797807188105649918969 94810*xi - 2206280523458717479850463459712)*x^2
      + 1/34*(5478632858069821978823110691157*xi - 336492625614157932557706992 10279)*x
      + 1/34*(2084097489572716360419535753 6693*xi - 1192908068739339993514494333836 61))/
    ( x^16
      + (-19*xi + 1273)*x^15
      + 1/2*(-40267*xi + 1431991)*x^14
      + (-9552726*xi + 238832756)*x^13
      + (-2719772875*xi + 53371714116)*x^12
      + (-523114720197*xi + 8532644204659)*x^11
      + 1/2*(-144660980220925*xi + 2029899412755711)*x^10
      + (-7452141868998384*xi + 92004145401482308)*x^9
      + 1/34*(-19857102477212937355*xi + 219090613097898997135)*x^8
      + (-3517173436049304 4361*xi + 3507426902432539420 51)*x^7
      + 1/34*(-5542432514406179412 0781*xi + 5037790202612671185 64541)*x^6
      + 1/17*(-9816365886984828333 28745*xi + 8185039836279751152409415)*x^5
      + 1/17*(-2612516322856227719 3195123*xi + 2008212498350112370024 77955)*x^4
      + 1/17*(-5061059463810727091 59765747*xi + 3600338433659364675221454325)*x^3
      + 1/34*(-1348461497926137887 3559444857*xi + 8904070460353569571215 8552063)*x^2
      + 1/17*(-5526522120037577830 1816893555*xi + 3395056169964720818213 06486671)*x
      + 1/289*(-3573819340594140725220323934120*xi + 2046075433498349936896 4237022661));
```

```
Ry := (1/4913*(8*xi - 25)*x^24
     + 1/578*(1881*xi - 3819)*x^23
     + 1/9826*(21960791*xi - 39125521)*x^22
     + 1/9826*(8187912751*xi - 14889622457)*x^21
     + 1/4913*(971073560946*xi - 2000024786472)*x^20
     + 1/9826*(312659738979115*xi - 794083609479177)*x^19
     + 1/9826*(34349375566164931*xi - 118283359442803441)*x^18
     + 1/9826*(2341720630729481085*xi - 13122441725860415835)*x^17
     + 1/9826*(4188953966053948791*xi - 10361801961830601138 81)*x^16
     + 1/4913*(-5717302549162724374131*xi - 24225855500160829840501)*x^15
     + 1/4913*(-8162877697984970533778 49*xi + 17427790193941262919 3951)*x^14
     + 1/9826*(-1250089092240113524015 48641*xi + 29920697033052337599 6401909)*x^13
     + 1/9826*(-6250077830601484878864624723*xi + 30816345132820075481296683691)*x^12
     + 1/9826*(-1880667017947494546005972150 31*xi
                                              + 19813340795188429388174766960 69)*x^11
     + 1/4913*(-3686597396505011950685270077 99*xi
                                             + 45460077982896270293147153507244)*x^10
     + 1/4913*(1343623416405243150420051510 36877*xi
                                            + 15085593643205360514206168635167 27)*x^9
     + 1/4913*(8530216632779892378794212360740171*xi
                                           + 33794385183108841319172261658089261)*x^8
     + 1/4913*(3157673625094910054957803357804282 05*xi
                                          + 34855315292655850833148618741015980 0)*x^7
     + 1/4913*(8174248096739623444900412335790607866*xi
                                         - 73260039682262749695987278992189325 47)*x^6
     + 1/4913*(1532269108367258203388748856617937032 99*xi
                                        - 43041197215469515030055834525824046900 7)*x^5
     + 1/9826*(4106614097660152915130997933558761507225*xi
                                       - 20948414365316237671547008361427676151059)*x^4
     + 1/9826*(3733046719834018857080886403316043942003 7*xi
                                      - 31710957881389461256558940688215778552707 3)*x^3
     + 1/9826*(2035188209237658810410604664011576504805 11*xi
                                     - 30801675215636344075256339301179915579649 09)*x^2
     + 1/9826*(4570144645442978583247004848747863836368 07*xi
                                     - 17716838410505846439213136350036842089384635)*x
     + 1/4913*(-2199192645523154375551941723722381204525 82*xi
                                     - 23049992521886724882001247970931183129692275))/
   ( x^24
     + 1/2*(-57*xi + 3819)*x^23
     + 1/2*(-96681*xi + 3345243)*x^22
     + 1/2*(-76020143*xi + 1798926349)*x^21
     + (-18514458534*xi + 335000540154)*x^20
     + 1/2*(-12587818230747*xi + 184688218970349)*x^19
     + 1/2*(-3187414989860607*xi + 39284662074719261)*x^18
     + 1/2*(-626091292752026067*xi + 6632379089727461073)*x^17
     + 1/34*(-1666501534928698109007*xi + 15410945275101981980517)*x^16
     + (-623416941811363960943 0*xi + 508702046528718601680 62)*x^15
     + 1/17*(-1109662696140280207188899 4*xi + 80495030744125128085749564)*x^14
     + 1/34*(-1930618459003325656180479693*xi + 12512415188433249785170633509)*x^13
     + 1/34*(-1403739683584290478149509940 67*xi + 81535091843487726378485941 6139)*x^12
     + 1/34*(-8558778521942231259783058138191*xi
                                              + 44621471681340532753780303642545)*x^11
     + 1/17*(-2189686955768817011389811070610 65*xi
```

```
                                                + 10248092397160756816843044904867 50)*x^10
      + 1/17*(-9386983641301351700482560156677100*xi
                                                + 39391187822564551340265156155118510)*x^9
      + 1/289*(-5706605764183315147517899690578775896*xi
                                              + 21417563431168250795108889899826378522)*x^8
      + 1/17*(-9939338599130953802952535355031170946*xi
                                              + 33232073687304944092150332843367665813)*x^7
      + 1/289*(-4095642132499108670210739709538522361744*xi
                                            + 12131452009913362095395508141756170825749)*x^6
      + 1/289*(-7991970496751951878990792886784251226899 4*xi
                                           + 20814573282392570595772175267254074061439 4)*x^5
      + 1/578*(-2449625102544244096807117006725136206571947*xi
                                            + 55537440026007453559546863459451886585316 09)*x^4
      + 1/578*(-2839345983492451580828964017231147558752749 1*xi
                                           + 55293813802506824840483732541545665048898531)*x^3
      + 1/578*(-2340061006974981293373958078090795226682548 67*xi
                                          + 38441694208269446205001095086617485760313730 9)*x^2
      + 1/578*(-1222013684851538805349370986329045088559665755*xi
                                            + 16512489281400005820661859239863649491319285 47)*x
      + 1/4913*(-2583741285980491732436474768984726003295317 6295*xi
                                          + 27677121776930525051976840552448253554988702084));

if ( 4*Rx^3 - g2*Rx - g3 ) eq ( (4*x^3-g2*x-g3)*Ry^2 ) then
  print "Curve N10: correct";
end if;
exit;
```

§ A.2. Алгоритм построения строго безопасных эллиптических кривых

Далее приводится текст программы, реализующей алгоритм генерации параметров строго безопасных эллиптических кривых, описанный ранее в § 1.5.

§ A.2.1. Текст программы

```
1  # функция проверяет, является ли число безопасным простым
2  ecIsSafePrime := function( p, ch )
3    pValue := Integers()!p;
4    if not IsPrime( pValue ) then
5      return false;
6    end if;
7    print " - простое число = ", p;
8    pValue := Integers()!((p-1)/2);
9    if not IsPrime( pValue ) then
10     return false;
11   end if;
12   print " - простое число (", ch, "-1 )/2 = ", pValue;
13   return true;
```

```
14  end function;
15
16  # функция проверяет, является ли порядок кривой безопасным простым числом
17  ecCheckSafeOrder := function( m )
18    factorSet := Factorization( Integers()!m );
19    q := factorSet[#factorSet][1];
20    c := m/q;
21    print " - кофактор c = ", m/q;
22    if not (( q gt 2^254 ) and ( q lt 2^256 )) then
23      print " ! делитель слишком мал (only ", Ceiling(Log(2, Integers()!q)), "bits )";
24      return q,c,0;
25    end if;
26    if( ecIsSafePrime(q, "q") eq true ) then
27      print " - порядок является безопасным простым\n";
28      return q,c,1;
29    else
30      print " - порядок не является безопасным простым\n";
31      return q,c,0;
32    end if;
33  end function;
34
35  # функция строит два варианта порядка группы точек эллиптической кривой
36  ecCreateOrders := function( p, d )
37    tuple := [[-1,0,0,0,0],[1,0,0,0,0]];
38    if not ecIsSafePrime(p, "p") then
39      return tuple;
40    end if;
41    Q<z> := QuadraticField(d);
42    f, s := NormEquation( Q, 4*p );
43    if not f then
44      return tuple;
45    end if;
46    tuple[1][2] := p + 1 - s[1][1];
47    print " - модуль m1 = ", tuple[1][2];
48    tuple[1][3], tuple[1][4], tuple[1][5] := ecCheckSafeOrder(tuple[1][2]);
49
50    tuple[2][2] := p + 1 + s[1][1];
51    print " - модуль m2 = ", tuple[2][2];
52    tuple[2][3], tuple[2][4], tuple[2][5] := ecCheckSafeOrder(tuple[2][2]);
53
54    return tuple;
55  end function;
56
57  #  функция строит точку эллиптической кривой заданного порядка
58  ecCreateBasePoint := function( a, b, p, q )
59    K := GF(p);
60    ec := EllipticCurve( [K!a, K!b] );
61    for x in [1..100] do
62      y := K!(x^3 + a*x + b);
63      if JacobiSymbol( Integers()!y, p ) eq 1 then
64        P := ec![x,Modsqrt( Integers()!y, p )];
65        if Order(P) eq q then
66          print " - построена точка ", P;
```

```
67          print " - порядок точки: ", Order(P);
68          return P;
69        end if;
70      end if;
71    end for;
72    return false;
73  end function;
74
75  # алгоритм построения эллиптической кривой
76  ecCreateCurve := function( p, d, index )
77    tuple := ecCreateOrders( p, d );
78    print "+! найдена комбинация:", tuple;
79    if ( tuple[1][5] eq 0 ) and ( tuple[2][5] eq 0 ) then
80      return 0;
81    end if;
82    if (d mod 4) eq 1 then
83      D := d;
84    else
85      D := 4*d;
86    end if;
87    print " - значение d =", d, "=", Factorization(d);
88    print " - фундаментальный дискриминант D =", D;
89
90   # построение инвариантов кривой
91    K := GF(p);
92    R<x> := PolynomialRing(K);
93    print " - строится многочлен Гильберта";
94
95    fp := R!HilbertClassPolynomial(D);
96
97    Deg := Degree(fp);
98    print " - степень многочлена Гильберта:", Deg;
99    if( Deg lt 500 ) then
100    return false;
101   end if;
102
103   idx := 0;
104   for jp in Roots(fp) do
105     idx := idx + 1;
106     print " - тестируем корень j =", jp[1], " индекс:", idx;
107     k := jp[1]*(K!Modinv( Integers()!(1728 - jp[1]), p ));
108     print " - определено k =", k;
109
110       if JacobiSymbol( Integers()!(-k), p) eq 1 then
111           c := Modinv( Modsqrt( Integers()!(-k), p ), p);
112           a := K!(3*k*c^2);
113           b := K!(2*k*c^3);
114           ec := EllipticCurve([a, b]);
115
116           print "\n - построена кривая:", ec, " - знак эпсилон = 1";
117           m := Order(ec);
118           print " - SEA порядок:", m;
119           localid := 0;
```

```
120             if( m eq tuple[1][2] ) then localid := 1; end if;
121             if( m eq tuple[2][2] ) then localid := 2; end if;
122             if( tuple[localid][5] eq 1 ) then
123               fprintf "ec.curves",
124                " index = %o\n p = %o\n a = %o\n b = %o\n m = %o\n q = %o\n %o\n\n",
125                                              index, p, a, b, m, tuple[localid][3],
126                                    ecCreateBasePoint( a, b, p, tuple[localid][3] );
127               fprintf "ec.table",
128                        " %o & %o & %o & %o & %o & %o & %o & 1 & %o \\\\ \\hline \n",
129                 index, 2^256-p, Abs(d), tuple[localid][1], tuple[localid][4],
130                           Deg, idx, Ceiling(Log(2, Integers()!tuple[3-localid][3]));
131               return true;
132             end if;
133
134             print " - порядок соответствует твисту";
135             a := K!(3*k*c^2);
136             b := K!(-2*k*c^3);
137             ec := EllipticCurve( [a, b]);
138             print "\n - построена кривая:", ec, " - знак эпсилон = -1";
139             m := Order(ec);
140             print " - SEA порядок:", m;
141             localid := 0;
142             if( m eq tuple[1][2] ) then localid := 1; end if;
143             if( m eq tuple[2][2] ) then localid := 2; end if;
144             if( tuple[localid][5] eq 1 ) then
145               fprintf "ec.curves",
146                " index = %o\n p = %o\n a = %o\n b = %o\n m = %o\n q = %o\n %o\n\n",
147                                              index, p, a, b, m, tuple[localid][3],
148                                    ecCreateBasePoint( a, b, p, tuple[localid][3] );
149               fprintf "ec.table",
150                       " %o & %o & %o & %o & %o & %o & %o & -1 & %o \\\\ \\hline \n",
151                 index, 2^256-p, Abs(d), tuple[localid][1], tuple[localid][4],
152                           Deg, idx, Ceiling(Log(2, Integers()!tuple[3-localid][3]));
153               return true;
154             end if;
155         else
156             print " - значение k не является квадратом";
157         end if;
158     end for;
159
160  return false;
161 end function;
162
163 myFirsrList := [
164   [ 0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff72ef, -6333787 ],
165   [ 0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff72ef, -185015326 ],
166   [ 0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffd215b, -354691387 ],
167   [ 0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffc50cf, -88727446 ],
168   [ 0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffc4ab7, -1468413670 ]
169 ];
170
171 myFoundedData := [
172   [ 0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff6e7a7, -2942326 ],
```

```
173   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffcd827b, -4711954 ],
174   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffb515bb, -2767810 ],
175   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffb07d67, -2166070 ],
176   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffacac87, -640030 ],
177   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff863417, -4683971 ],
178   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff41052b, -2965114 ],
179   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffecd6287, -5936782 ],
180   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffeb17533, -2447434 ],
181   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffea5d32f, -6494062 ],
182   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffe8be393, -1904482 ],
183   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffe59a627, -2105014 ],
184   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffe3591ab, -1449922 ],
185   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffe2e7a57, -6847462 ],
186   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffdc6f89b, -4450978 ],
187   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffca18a6f, -1235854 ],
188   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffc51c30b, -5309194 ],
189   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffc020fff, -1917790 ],
190   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffbb12ab7, -6899806 ],
191   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffbaf4c1f, -2674102 ],
192   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffb878823, -2181898 ],
193   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffb73d517, -5302246 ],
194   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffadf2c43, -2307178 ],
195   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffaa1e237, -935518 ],
196   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffa5a06d3, -6502618 ],
197   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffa4e1e5b, -2511370 ],
198   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffa4d2f87, -6317710 ],
199   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffa0ca273, -6531466 ],
200   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff9f996bf, -1645030 ],
201   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff9e1c33b, -1215034 ],
202   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff9cdab4f, -6801790 ],
203   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff9c00a1f, -2754502 ],
204   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff9b9eb77, -4675246 ],
205   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff9a883ab, -1325146 ],
206   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff9a5b927, -2245198 ],
207   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff991896b, -3847762 ],
208   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff95603e7, -2773595 ],
209   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff91c9f2b, -1329946 ],
210   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff8d216cb, -6456754 ],
211   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff867bbaf, -977566 ],
212   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff837344f, -1591774 ],
213   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff80431f7, -6810214 ],
214   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff7fc52f3, -1884010 ],
215   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff791e787, -1770334 ],
216   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff78a649f, -974830 ],
217   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff7764113, -2646898 ],
218   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff7745213, -5793539 ],
219   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff727e71f, -835054 ],
220   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff6b82ef3, -2711290 ],
221   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff69c8c87, -2111422 ],
222   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff61a8e53, -2589274 ],
223   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff60806f3, -2797339 ],
224   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff594c863, -4264714 ],
225   [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffff58ba1db, -1328986 ],
```

```
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffef0c5a3, -2763419 ],
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff532a897, -6456982 ],
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff51a7b2b, -2662211 ],
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff4f889b7, -1648331 ],
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff4c79267, -2594902 ],
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff4b65f3f, -2677739 ],
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff49d7e6f, -6125782 ],
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff476c49b, -6657706 ],
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff445b833, -2217739 ],
  [ 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff42241af, -3709294 ]
];

index := 1;
for i in myFoundedData do
  print "\n -- INDEX: ", index, " --";
  print "\n", i;
  if( ecCreateCurve( i[1], i[2], index ) eq true ) then
   index := index + 1;
  end if;
end for;
```

§ A.2.2. Результаты практических вычислений

Приведем параметры найденных эллиптических кривых $\mathcal{E}_{a,b}(\mathbb{F}_p)$, заданных сравнением

$$y^2 \equiv x^3 + ax + b \pmod{p},$$

а также значение порядка группы точек m и безопасного простого q такого, что $q|m$. Дополнительно мы приведем координаты точек $P = (x, y)$, образующих подгруппу простого порядка q.

В начале приведены параметры эллиптической кривой, модуль которой удовлетворяет равенству $p = 2^{256} - 188069$. Данная кривая была найдена с использованием стратегии перебора неограниченных значений d. Время построения параметров кривой составило 13 ч. 45 мин. с использованием процессора Intel(R) Core(TM) i5-8250U, с тактовой частотой 1.60GHz.

```
 index = 0
 p = 115792089237316195423570985008687907853269984665640564039457584007913129451867
 a = -3
 b = 30248189431475512214188672690637910310234046139542618758265309564348112627199
 m = 115792089237316195423570985008687907852988330907758208561031253359358293855963
 q = 115792089237316195423570985008687907852988330907758208561031253359358293855963
px = 2
py = 505813215244405194293364042253826382537353917741486700005654868494394106239
```

Остальные кривые были получены с использованием поиска величины d в ограниченном множестве, см. таблицу 1.7 на стр. 129.

```
 N = 1
 p = 1157920892373161954235709850086879078532699846656405640394575840079131290438 79
 a = -3
 b = 7446981130497742293783343308011296654323426666651497905404668186408759890923 0
 m = 1157920892373161954235709850086879078528024475833675140377989101452697438803 26
 q = 5789604461865809771178549250434395392640122379168375701889945507263487194016 3
px = 1
py = 1841338330403266699485463308806415316904365020253352747924925616238938127975 5

 N = 2
 p = 1157920892373161954235709850086879078532699846656405640394575840079131263310 03
 a = -3
 b = 1091545283938856828173941204606887888526302943902695950603434865879470081656 74
 m = 1157920892373161954235709850086879078537409191607392993901012612645171946509 98
 q = 5789604461865809771178549250434395392687045958036964969505063063225859732549 9
px = 2
py = 6559436936706083516732151389608215571296402593319611257946118536365901531101 2

 N = 3
 p = 1157920892373161954235709850086879078532699846656405640394575840079131247302 99
 a = -3
 b = 2313144805511080231913435995668105281855109519315142572321746200305529660843 2
 m = 1157920892373161954235709850086879078526992166274277497652706798582011482752 86
 q = 5789604461865809771178549250434395392634960831371387488263533992910057413764 3
px = 2
py = 1032631318191528701525810868474421367611285801871368050165644576457490977706 44

 N = 4
 p = 1157920892373161954235709850086879078532699846656405640394575840079131244291 59
 a = -3
 b = 9268057490824854186394479249797364185320715667018788655848523061414926084297 2
 m = 1157920892373161954235709850086879078531899186536928303482082671799088849595 26
 q = 5789604461865809771178549250434395392659495932684641517410413358995444247976 3
px = 1
py = 8271861410729594337667923052301169607273023145441970896797402551174623999985 2

 N = 5
 p = 1157920892373161954235709850086879078532699846656405640394575840079131241790 79
 a = -3
 b = 4236276030788503688989053776382260536323655408771970924831190644736722148692 4
 m = 1157920892373161954235709850086879078529894138509775060317908878302734301793 74
 q = 5789604461865809771178549250434395392649470692548875301589544391513671508968 7
px = 2
py = 1403410938384014429849683201790717414128455696493989499942679190272963015648 4

 N = 6
 p = 1157920892373161954235709850086879078532699846656405640394575840079131216578 79
 a = -3
 b = 5895097359372949471104876181202547022356588426201176205810863513816733865103 1
 m = 1157920892373161954235709850086879078530525089289636814663916093037714927150 9
```

```
  q = 385973630791053984745236616695626359510175029766321227155463869767923830905 03
 px = 3
 py = 205965177433974771078635046317975268211325215140165294580871699537154805183 37

  N = 7
  p = 115792089237316195423570985008687907853269984665640564039457584007913117123883
  a = -3
  b = 891067883433122655663357603027777392464718525404542763134830135757969052096 40
  m = 115792089237316195423570985008687907852622053566145200904921225110885498298398
  q = 578960446186580977117854925043439539263110267830726004524606125554427491491 99
 px = 9
 py = 793080291018550952185540162136247614508179946693354007953154990832197173574 98

  N = 8
  p = 115792089237316195423570985008687907853269984665640564039457584007913111864739
  a = -3
  b = 105598554324686606448243902967547806710936194579460793109384049876194488547338
  m = 115792089237316195423570985008687907853790251168837911307502650378398427214281
  q = 385973630791053984745236616695626359512634170562793037691675501261328090714 27
 px = 4
 py = 263862874584583477161982638502301582793742349153166753443682612045753645448 41

  N = 9
  p = 115792089237316195423570985008687907853269984665640564039457584007913109545607
  a = -3
  b = 363086312107663765853505097684600685735524878298415668598901341091718388618 13
  m = 115792089237316195423570985008687907853236825202202410861911484046220248883758
  q = 578960446186580977117854925043439539266184126011012054309557420231101244418 79
 px = 1
 py = 545990815028717062664290813036676690452769123158415392798539516931440203482 43

  N = 10
  p = 115792089237316195423570985008687907853269984665640564039457584007913107715379
  a = -3
  b = 997419877214260318962218352417330453170050894373299654411445138917011977943 86
  m = 115792089237316195423570985008687907853908514137917002958388358587562721006126
  q = 578960446186580977117854925043439539269542570689585014791941792937813605030 63
 px = 1
 py = 118032496538148616464119954227253991396897523837834030068148760530686367391 06

  N = 11
  p = 115792089237316195423570985008687907853269984665640564039457584007913106953007
  a = -3
  b = 161375784084888317849097910967085702667394990907887234833476470336776920640 31
  m = 115792089237316195423570985008687907853775518886756685996145524310853074122 38
  q = 578960446186580977117854925043439539268877594433783429983072762155426537061 19
 px = 4
 py = 830273976867280816195974828172207091770737186598276197551859978315108403588 79

  N = 12
  p = 115792089237316195423570985008687907853269984665640564039457584007913105253267
  a = -3
  b = 404666985415886321483128662832329024511445318990163145904825318751731670460 71
```

```
 m = 1157920892373161954235709850086879078538796576053164219275154264123081895905 18
 q = 5789604461865809771178549250434395392693982880265821096375771320615409479525 9
px = 1
py = 4949665119590062883660778991680654334463923389051938655295993619924045949561 7

 N = 13
 p = 1157920892373161954235709850086879078532699846656405640394575840079131019607 43
 a = -3
 b = 2126977297758327218667076031794803609842489409195203823759943378891309491772 3
 m = 1157920892373161954235709850086879078537627068089359888916638099356242580778 78
 q = 5789604461865809771178549250434395392688135340446799444583190496781212903893 9
px = 2
py = 7983616671185532991249551403664025978947947941159074591868413343950566429294 1

 N = 14
 p = 1157920892373161954235709850086879078532699846656405640394575840079130995962 03
 a = -3
 b = 2356329602371881148873586760743945558484714062363962549102795959464968562364
 m = 1157920892373161954235709850086879078536946762274939333575231560715532201732 86
 q = 5789604461865809771178549250434395392684733811374696667876157803577661008664 3
px = 1
py = 6795015963140790637271204621879522412134306546230782712561802099728397811627 7

 N = 15
 p = 1157920892373161954235709850086879078532699846656405640394575840079130991314 79
 a = -3
 b = 1083708602433943055316114313549782951649978964298540440388741961399385107065 02
 m = 1157920892373161954235709850086879078538870861665888979216031011720688802040 38
 q = 5789604461865809771178549250434395392694354308329444896080155058603444010201 9
px = 2
py = 5347875408342995130750439008034793281129508750089559619322755951819860834739 7

 N = 16
 p = 1157920892373161954235709850086879078532699846656405640394575840079130923480 59
 a = -3
 b = 4805282944323163141233305595330088610554127168820234778479439913728218881866 7
 m = 1157920892373161954235709850086879078526066754968844147556402203965963235008 38
 q = 5789604461865809771178549250434395392630333774844220737782011019829816175041 9
px = 4
py = 8511735287370905808654101171968469881603331249720884219905414359391505770072 3

 N = 17
 p = 1157920892373161954235709850086879078532699846656405640394575840079130731178 07
 a = -3
 b = 7387592346869170285633493182627321637405849798745464648483781890605400448120
 m = 1157920892373161954235709850086879078529088717182104091774992216249824595843 66
 q = 5789604461865809771178549250434395392645443585910520458874961081249122979218 3
px = 5
py = 7542764518055713029604642593672429477007181756948304153520496047432652622060 8

 N = 18
 p = 1157920892373161954235709850086879078532699846656405640394575840079130678894 19
 a = -3
```

```
 b = 10050798997774277555412035358709552792748909462292816525119134956143381708024 1
 m = 11579208923731619542357098500868790785260237665266147791551022168486094189964 6
 q = 57896044618658097711785492504343953926301188326330738957755110842430470949823
px = 2
py = 94297570663564679937026558907489858527753902540729614955599483347399068837713

 N = 19
 p = 11579208923731619542357098500868790785326998466564056403945758400791306266623 9
 a = -3
 b = 47175291938656847137940045251913939950300956561315333478881253243786827861464
 m = 11579208923731619542357098500868790785325843701419531484831812953426292187105 4
 q = 57896044618658097711785492504343953926629218507097657424159064767131460935527
px = 9
py = 60500759973134150187220773690735531996301760574101010071823687563820884346529

 N = 20
 p = 11579208923731619542357098500868790785326998466564056403945758400791305736466 3
 a = -3
 b = 10926036773453502677931487391343612880299547290915661457195291748438775149466 8
 m = 11579208923731619542357098500868790785262338150454193453162628510279616096439 8
 q = 57896044618658097711785492504343953926311690752270967265813142551398080482199
px = 1
py = 70186026567115213517463447685727914108820822018918711039423846488160543089 85

 N = 21
 p = 11579208923731619542357098500868790785326998466564056403945758400791305724214 3
 a = -3
 b = 52877540268648378046101045274559226039890366084767742825199422632068619998 05
 m = 11579208923731619542357098500868790785320406984482071946622063825133033330221 4
 q = 57896044618658097711785492504343953926602034922410359733110319125665166651107
px = 1
py = 74344459774423811021512481636900120527822011185306100299437458428354546013777

 N = 22
 p = 11579208923731619542357098500868790785326998466564056403945758400791305463606 7
 a = -3
 b = 52474008387703667742846535649870195540416560911857676565841459306196158163663
 m = 11579208923731619542357098500868790785394108723284132919846033510960882342943 8
 q = 57896044618658097711785492504343953926970543616420664599230167554804411714719
px = 3
py = 16767306740135133121678998262890212805867190106945206318604149511320639144876

 N = 23
 p = 11579208923731619542357098500868790785326998466564056403945758400791305334504 7
 a = -3
 b = 48055843695636093074633296328278857435669387632647108022083565207056130493267
 m = 11579208923731619542357098500868790785378616841057008525009756620997188320440 6
 q = 57896044618658097711785492504343953926893084205285042625048783104985941602203
px = 1
py = 46732678612748241096467200865664403114829426446783420938850859547804772459979

 N = 24
 p = 11579208923731619542357098500868790785326998466564056403945758400791304360249 9
```

```
  a = -3
  b = 74101573399098734643434694192589435778443494696379670317717917165770481008643
  m = 11579208923731619542357098500868790785369115943108681487185462759031770371161 4
  q = 57896044618658097711785492504343953926845579715543407435927313795158851855807
 px = 3
 py = 59284978820442007657817637987007630077025642983667247575984918640468238074913

  N = 25
  p = 11579208923731619542357098500868790785326998466564056403945758400791303958584 7
  a = -3
  b = 37413243307094081968176982834897266145877153645696246356231958778895471516840
  m = 11579208923731619542357098500868790785259546013688801467082598664484682255593 4
  q = 57896044618658097711785492504343953926297730068444007335412993322423411277967
 px = 1
 py = 85454125317474820695257828634978239806824953780839493288267254506319239792750

  N = 26
  p = 11579208923731619542357098500868790785326998466564056403945758400791303487662 7
  a = -3
  b = 99080326561424844556206797479802237624180326203382375521512222911672197106322
  m = 11579208923731619542357098500868790785280804840117196773725660987818744478143 8
  q = 57896044618658097711785492504343953926404024200585983868628304939093722390719
 px = 3
 py = 49416404477635758754671984070120938234668805870119641653501891974648610736989

  N = 27
  p = 11579208923731619542357098500868790785326998466564056403945758400791303409621 9
  a = -3
  b = 47087175485317494864685757036269171845652442265724537090588347679321398775350
  m = 11579208923731619542357098500868790785303128658560987835357861672278882657673 4
  q = 57896044618658097711785492504343953926515643292804939176789308361394413288367
 px = 1
 py = 83655894991550891046893235318374657387694535556091546884560597962575595877955

  N = 28
  p = 11579208923731619542357098500868790785326998466564056403945758400791303403507 9
  a = -3
  b = 17445425007998807975370468479744298745793037481264193657530440321751100656570
  m = 11579208923731619542357098500868790785377183311806004782051080897055795507772 6
  q = 57896044618658097711785492504343953926885916559030023910255404485278977538863
 px = 2
 py = 47879177895568779938715394443847204375805079684803453721382070361717401367686

  N = 29
  p = 11579208923731619542357098500868790785326998466564056403945758400791302980465 9
  a = -3
  b = 59440404654850459664944137278352318449129874222723620505686374538253865234099
  m = 11579208923731619542357098500868790785263134064917875866900468006671761314265 4
  q = 57896044618658097711785492504343953926315670324589379334502340033358806571327
 px = 2
 py = 10553593241597269630272183312383090288583952990334141859127621545755655306903 3

  N = 30
```

```
 p = 1157920892373161954235709850086879078532699846656405640394575840079130285564 79
 a = -3
 b = 3666253768328529795449497060633401054307747814244591644190315508984735227869 0
 m = 1157920892373161954235709850086879078525955412004645387455986247874240112036 86
 q = 5789604461865809771178549250434395392629777060023226937279931239371200560184 3
px = 2
py = 5856756515372542524242142728759731884981484736590183564555709169001436243243 3

 N = 31
 p = 1157920892373161954235709850086879078532699846656405640394575840079130269950 03
 a = -3
 b = 6609253574123298775126164187455362084704412888395507737220562280974337263795 8
 m = 1157920892373161954235709850086879078531469126119594596192586863900996974853 58
 q = 5789604461865809771178549250434395392657345630597972980962934319504984874267 9
px = 4
py = 2489054327817393251043518592336732450722561482068494270353328905269409938579 0

 N = 32
 p = 1157920892373161954235709850086879078532699846656405640394575840079130256781 59
 a = -3
 b = 1757200110067339045322481372533064501345874624637364188295489904096212799067 8
 m = 1157920892373161954235709850086879078527002884634579080721094480711692986394 86
 q = 5789604461865809771178549250434395392635014423172895403605472403558464931974 3
px = 1
py = 1010847020208227795327284069647815755745849757379121201129932945284673948619 5

 N = 33
 p = 1157920892373161954235709850086879078532699846656405640394575840079130247849 27
 a = -3
 b = 8594530379982101222950500260939897497444356640135688038761385490762905729949 9
 m = 1157920892373161954235709850086879078528621063989895008060797176949926025897 98
 q = 5789604461865809771178549250434395392643105319949475040303985884749630129489 9
px = 3
py = 7505654955467612719600310462361738779658010079314383570211280974740038406122 0

 N = 34
 p = 1157920892373161954235709850086879078532699846656405640394575840079130243838 63
 a = -3
 b = 7142466782090767055452656424959003474287008747040121823734426378320644716711
 m = 1157920892373161954235709850086879078538521998150683399514273533784500190410 38
 q = 5789604461865809771178549250434395392692609990753416997571367668922500952051 9
px = 20
py = 7693168749014458269233908664117062863368629651565998992081803489234353488927 3

 N = 35
 p = 1157920892373161954235709850086879078532699846656405640394575840079130232431 79
 a = -3
 b = 7241363450366576397602034602659975415671405187708420961896724342393813216053 3
 m = 1157920892373161954235709850086879078530166331882510789195626115743814458325 26
 q = 5789604461865809771178549250434395392650831659412553945978130578719072291626 3
px = 1
py = 6130286392822865323576951979278597472492611092431359366278910123395572770518 9
```

```
 N = 36
 p = 1157920892373161954235709850086879078532699846656405640394575840079130230602 63
 a = -3
 b = 4660374088803784751279620900456458037613556457582226550447146124742191711232 5
 m = 1157920892373161954235709850086879078526267332604751348617780646068872590759 34
 q = 5789604461865809771178549250434395392631336663023756743088903230344362953796 7
px = 8
py = 9989200880069414775010284939988072121033693522210446565324783847462000741189 8

 N = 37
 p = 1157920892373161954235709850086879078532699846656405640394575840079130217373 23
 a = -3
 b = 1128994932018396110092281694516103922377078217626780109336876319651666602173 57
 m = 1157920892373161954235709850086879078539358393643234507012248608900762906119 26
 q = 5789604461865809771178549250434395392696791968216172535061243044503814530596 3
px = 3
py = 8254394886664710421155561000905610639291702284833662786086871782183436955821 2

 N = 38
 p = 1157920892373161954235709850086879078532699846656405640394575840079130178365 19
 a = -3
 b = 4849141914608117554749044363725618020751981702634436723395080501813340217614 6
 m = 1157920892373161954235709850086879078526617322158761596658597501997364853619 29
 q = 3859736307910539847452366166956263595088724407195871988861991673324549512064 3
px = 6
py = 7848122309816080658818984472877823617367035388565081207557410250793101638521 4

 N = 39
 p = 1157920892373161954235709850086879078532699846656405640394575840079130140751 79
 a = -3
 b = 9808003997946125370608130605544036109681658041881493790327092993577977198158 5
 m = 1157920892373161954235709850086879078526242401153647942535672679622835007397 26
 q = 5789604461865809771178549250434395392631212005768239712678363398114175036986 3
px = 2
py = 7033498284595619925204016473401963402002458741077540362009355226677701178540 5

 N = 40
 p = 1157920892373161954235709850086879078532699846656405640394575840079130091906 03
 a = -3
 b = 2811621352148083592853964051945420213348082127823903696760949367117984140627 0
 m = 1157920892373161954235709850086879078532907542987278522763635005987098492713 18
 q = 5789604461865809771178549250434395392664537714936392613818175029935492463565 9
px = 1
py = 4255364119282246998585462203993912941676968803209219419496948098651388905227 0

 N = 41
 p = 1157920892373161954235709850086879078532699846656405640394575840079130022204 63
 a = -3
 b = 4877888533235153950986983928356451608311775947764107625379911842548146338252 4
 m = 1157920892373161954235709850086879078525919949943087583570099863297065660136 78
 q = 5789604461865809771178549250434395392629599749715437917850499316485328300683 9
px = 4
py = 4660889099118415126688543595352456103652010455722919673281067815326676618151 7
```

```
 N = 42
 p = 1157920892373161954235709850086879078532699846656405640394575840079129990400 79
 a = -3
 b = 1073086436720092918935925262089341792332854494088600468315491413896353738926 07
 m = 1157920892373161954235709850086879078532788404108129762110652778411430431947 34
 q = 5789604461865809771178549250434395392663942020540648810553263892057152159736 7
px = 7
py = 2703053112221661916989312860407991388487787519371557897941476186355773972024 0

 N = 43
 p = 11579208923731619542357098500868790785326998466564056403945758400791299569714 3
 a = -3
 b = 7358531467600903802588624263517167538141735584871698411336522764585237521058 9
 m = 1157920892373161954235709850086879078532549231883464281222421312043819509211 58
 q = 5789604461865809771178549250434395392662746159417321406112106560219097546057 9
px = 3
py = 4358041907477588658069302208440805422427259749254899634868860548306223987614 4

 N = 44
 p = 11579208923731619542357098500868790785326998466564056403945758400791299518129 9
 a = -3
 b = 1155576582591497902567497921445098439908651673443923732357958676595988817372 59
 m = 1157920892373161954235709850086879078533119151385951490064903760994819213292 94
 q = 5789604461865809771178549250434395392665595756929757450324518804974096066464 7
px = 1
py = 8093066339683647571475916259211636188821016140271667986479821288776377353503 6

 N = 45
 p = 11579208923731619542357098500868790785326998466564056403945758400791298820698 3
 a = -3
 b = 6959960736068434282701997198129339812327825562780864892197198698416457457461
 m = 1157920892373161954235709850086879078538348767537708644549707340507633325648 78
 q = 5789604461865809771178549250434395392691743837688543222748536702538166628243 9
px = 8
py = 7217554744422923521247035167519517146573810021095248857774220406590239898581 8

 N = 46
 p = 11579208923731619542357098500868790785326998466564056403945758400791298771471 9
 a = -3
 b = 6653173091317060689142751341944179566835088687424911077421669677874387827631 6
 m = 1157920892373161954235709850086879078537544771912021109824097125005859812378 54
 q = 5789604461865809771178549250434395392687723859560105549120485625029299061892 7
px = 1
py = 4440635709801075302503511046262670847711249830034462055977521599699524439680 4

 N = 47
 p = 11579208923731619542357098500868790785326998466564056403945758400791298639489 9
 a = -3
 b = 9036566774951182464858734122077910151827553922346082789010469526917576272239 8
 m = 1157920892373161954235709850086879078538052332148088449651602125237443038564 86
 q = 5789604461865809771178549250434395392690261660740442248258010626187215192824 3
px = 2
```

```
py = 30356706589345556902647215835251491712541138467416550469689249962375428519101

 N = 48
 p = 11579208923731619542357098500868790785326998466564056403945758400791298626817 9
 a = -3
 b = 17178155077333678469285047711588600727116291251170817315604580193784655982130
 m = 11579208923731619542357098500868790785381731220040625001208531808109686603040 9
 q = 38597363079105398474523661669562635951272437400135416670695106027032288676803
px = 3
py = 33249396398153617234939075430644200381335825584320058717858270556218893314197

 N = 49
 p = 11579208923731619542357098500868790785326998466564056403945758400791298126006 3
 a = -3
 b = 40637388098740133735229824212046009122976442486954032993380786916736871746107
 m = 11579208923731619542357098500868790785260096446190806981532611537131007052663 8
 q = 57896044618658097711785492504343953926300482230954034907663057685655035263319
px = 24
py = 45227585604526613067216385314015197332516517595209395639273295119971524443059

 N = 50
 p = 11579208923731619542357098500868790785326998466564056403945758400791297393841 9
 a = -3
 b = 90611188751456814189090775622029340168431178047912177961448804855003391250660
 m = 11579208923731619542357098500868790785388086575492585660834201028501524539737 4
 q = 57896044618658097711785492504343953926940432877462928304171005142507622698687
px = 1
py = 16889325228117755488903469665733287922037862505955137406065846150132246381 98

 N = 51
 p = 11579208923731619542357098500868790785326998466564056403945758400791297212736 7
 a = -3
 b = 10842444356109607320507386262101642020714723359775818478651565194810409019142 5
 m = 11579208923731619542357098500868790785268921937560196113152207001155359654351 8
 q = 57896044618658097711785492504343953926344609687800980565761035005776798271759
px = 4
py = 46569525281717902101559259233491126442839417652993573531380232956349728912953

 N = 52
 p = 11579208923731619542357098500868790785326998466564056403945758400791296360814 7
 a = -3
 b = 62836385125034841899261767208569644373494825391621223277934587248683139457 59
 m = 11579208923731619542357098500868790785345272760831342925603712145630569460796 6
 q = 57896044618658097711785492504343953926726363804156714628018560728152847303983
px = 1
py = 96186379998378831566388897110870548552124730367043091854962274660704866730490

 N = 53
 p = 11579208923731619542357098500868790785326998466564056403945758400791296239384 3
 a = -3
 b = 10931040661408156785346219187468650829501287815360806636107897048113876653234 6
 m = 11579208923731619542357098500868790785325587983038104861369086144479539543716 3
 q = 11579208923731619542357098500868790785325587983038104861369086144479539543716 3
```

```
 px = 1
 py = 629102427858308133843042014988656540284987881477030394795993385731755056997 42

  N = 54
  p = 11579208923731619542357098500868790785326998466564056403945758400791295484 1187
  a = -3
  b = 90209079909712717284518144813688047631678010586573755238661642974430936616872
  m = 11579208923731619542357098500868790785330480841621931005522787203108704820 2046
  q = 57896044618658097711785492504343953926652404208109655027613936015543524101023
 px = 4
 py = 51173232978127832991891211271828213717585336578142928692942992267655674256333

  N = 55
  p = 11579208923731619542357098500868790785326998466564056403945758400791295424 1499
  a = -3
  b = 22019648332028218661792302702833474461590309849997097467146329216510446556037
  m = 11579208923731619542357098500868790785264368285537484003010398957043902419 7886
  q = 57896044618658097711785492504343953926321841427687420015051994785219512098943
 px = 4
 py = 55016078567573562904495237635527442458780136020681588705916263376947200674958

  N = 56
  p = 11579208923731619542357098500868790785326998466564056403945758400791294841 0519
  a = -3
  b = 52340638462568645828091510106077304585636136633625098367994080858841516086370
  m = 11579208923731619542357098500868790785394356771183940104410553457865836573 8694
  q = 57896044618658097711785492504343953926971783855919700522052767289329182869347
 px = 2
 py = 12431225378237906713529543467019682226703859397316871906694682228384918869689

  N = 57
  p = 11579208923731619542357098500868790785326998466564056403945758400791294682 6027
  a = -3
  b = 76660543866360873577263816270207616785919723237373322006141520555839648087 41
  m = 11579208923731619542357098500868790785277823518264839945654991561865972387 3961
  q = 38597363079105398474523661669562635950926078394216133152183305206219907957987
 px = 10
 py = 37345955700412339032633054187572246783564910234546947544888915252777969595 55

  N = 58
  p = 11579208923731619542357098500868790785326998466564056403945758400791294460 1527
  a = -3
  b = 10740541259077564687644040231384353891041207477033730763215131791105868690 7066
  m = 11579208923731619542357098500868790785313366188066871995111473215238220942 9921
  q = 38597363079105398474523661669562635951044553960222906650371577384127403143307
 px = 4
 py = 36296451629168698241851246529103006864310429098019318571282804031759454737 19

  N = 59
  p = 11579208923731619542357098500868790785326998466564056403945758400791294139 2487
  a = -3
  b = 90834685951589113721519601583857206047030843784305266081924222249247338735518
  m = 11579208923731619542357098500868790785313026440119206908646094479055968170 1334
```

```
 q = 57896044618658097711785492504343953926565132200596034543230472395279840850667
px = 3
py = 11203244920158798985499841508052418792517472930850832800474952653369179515647 8

 N = 60
 p = 11579208923731619542357098500868790785326998466564056403945758400791294026527 9
 a = -3
 b = 14245201932861945675315759986481509317683583185153884602467791784660675100841
 m = 11579208923731619542357098500868790785391015035005854235735083908449647273550 1
 q = 38597363079105398474523661669562635951303383450019514119116946361498824245167
px = 1
py = 10220035225469324195855551425736979929253737194342457275778134989339721839973 6

 N = 61
 p = 11579208923731619542357098500868790785326998466564056403945758400791293863486 3
 a = -3
 b = 18471650760155359564348539929327966233865210886437209895689521663912766562167
 m = 11579208923731619542357098500868790785267127468095694722796860194576863646325 4
 q = 57896044618658097711785492504343953926335637340478473613984300972884318231627
px = 5
py = 53603951649819033149465596396274378690546765470812537635150807082497259073612

 N = 62
 p = 11579208923731619542357098500868790785326998466564056403945758400791293609692 3
 a = -3
 b = 11512001445808851744933613095893585381400064620605519069967320120553316561337
 m = 11579208923731619542357098500868790785307388106921803095110757029840247538663 8
 q = 57896044618658097711785492504343953926536940534609015475553785149201237693319
px = 2
py = 96186905714965752056282043364772801122302568860340658124175009597104813451111

 N = 63
 p = 11579208923731619542357098500868790785326998466564056403945758400791293248348 3
 a = -3
 b = 59046870414610474699354235431082734854510819690592242760958263819711559664238
 m = 11579208923731619542357098500868790785301307396403952762798316691375758268692 3
 q = 11579208923731619542357098500868790785301307396403952762798316691375758268692 3
px = 1
py = 66025523393957924010982859709953432563144190557740958945614219334128877615777

 N = 64
 p = 11579208923731619542357098500868790785326998466564056403945758400791293055838 3
 a = -3
 b = 20645970233373882322748908554664146484842427067489252849720812137642641087790
 m = 11579208923731619542357098500868790785355013009219567754020325282728634347543 8
 q = 57896044618658097711785492504343953926775065046097838770101626413643171737719
px = 3
py = 13575496027422359896361821230894392944741765739925131295323431580230196427493
```