МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

На правах рукописи

Бабуева Александра Алексеевна

Свойства безопасности схем подписи вслепую на основе уравнений Шнорра и Эль-Гамаля

2.3.6 – Методы и системы защиты информации,
информационная безопасность

ДИССЕРТАЦИЯ
на соискание ученой степени
кандидата физико-математических наук

Научный руководитель
доктор физико-математических наук
Смышляев Станислав Витальевич

Москва – 2025

Оглавление

Введение

Диссертация посвящена решению задачи получения обоснованных оценок стойкости для схем подписи вслепую. Схема подписи вслепую представляет собой криптографический механизм, определяемый алгоритмом генерации ключей, протоколом формирования подписи и алгоритмом проверки подписи. Протокол формирования подписи является интерактивным протоколом, выполняемым между подписывающей стороной (сервером) и запрашивающей стороной (пользователем, клиентом). В результате выполнения этого протокола клиент получает подпись для некоторого сообщения, при этом подписывающий не получает информации ни о сообщении, ни о сформированном значении подписи, а клиент не может сформировать корректное значение подписи без взаимодействия с подписывающим.

Актуальность темы. Схемы подписи вслепую применяются в прикладных информационных системах, в которых возникает необходимость одновременного обеспечения целостности данных и невозможности установления связи между конкретными данными и их владельцем. К таким информационным системам относятся, в том числе, системы дистанционного электронного голосования и системы электронных платежей.

Задача получения обоснованных оценок стойкости для схем подписи вслепую является важной для обеспечения защиты информации как с практической, так и с теоретической точки зрения. Так, при использовании таких схем в прикладных системах наличие обоснованных оценок позволяет определять безопасные условия эксплуатации схем, например, возможность параллельного подписания данных различными пользователями или максимальное количество подписей, сформированное с использованием схемы. Задача получения оценок стойкости предполагает исследование комбинаторных и/или алгебраических свойств схем подписи вслепую и получение строгих доказательств в математических моделях безопасности. Под моделью безопасности понимается совокупность угроз (свойств) безопасности и возможностей нарушителя, потенциально доступных ему при использовании криптографического механизма.

Модели безопасности для схем подписи вслепую. Для схем подписи вслепую традиционно рассматривают [31, 53, 68] два целевых свойства безопасности: свойство неотслеживаемости и свойство неподделываемости. Определим каждое из них, задав соответствующие модели безопасности.

При рассмотрении свойства неотслеживаемости предполагается, что нарушитель может выступать в роли подписывающего и (в зависимости от определения конкретной модели

безопасности) обладать или не обладать следующими возможностями: генерировать ключ подписи произвольным образом, навязывать клиенту сообщения для подписи, узнавать о факте завершения протокола с ошибкой в качестве выходного результата на стороне клиента. В качестве угрозы рассматривается факт получения нарушителем в результате выполнения протокола формирования подписи нетривиальной информации о паре (сообщение, подпись), сформированной на стороне клиента.

При рассмотрении свойства неподделываемости предполагается, что нарушитель может выступать в роли клиента и (в зависимости от определения конкретной модели безопасности) обладать или не обладать возможностью начинать выполнение новых сеансов протокола формирования подписи вслепую до завершения предыдущих (т.е. проводить так называемую атаку с параллельными сеансами). В качестве угрозы рассматривается факт формирования нарушителем подделки. Так же как и для классических схем подписи, для схем подписи вслепую различают свойство сильной неподделываемости (задача нарушителя — сформировать новую пару (сообщение, подпись)) и слабой неподделываемости (задача нарушителя — сформировать подпись для нового сообщения).

Модели безопасности, описывающие свойства неотслеживаемости и неподделываемости для схем подписи вслепую и предоставляющие нарушителю все указанные выше возможности, далее будем называть *расширенными моделями безопасности*.

Выбор модели безопасности для анализа стойкости конкретной схемы подписи вслепую определяется условиями ее эксплуатации, т.е. особенностями той прикладной системы, в рамках которой будет использоваться соответствующая схема. Большой интерес представляют схемы подписи вслепую, стойкие (при некоторых предположениях) в расширенных моделях безопасности, поскольку они предъявляют наименьшее число требований к высокоуровневой прикладной системе.

Вместе с тем некоторые прикладные системы предъявляют к используемым криптографическим механизмам (в том числе схемам подписи вслепую) специфичные требования. При этом соотношение данных требований с обеспечением стойкости в расширенных моделях безопасности зачастую является открытым вопросом. Одним из важных классов таких систем являются системы формирования подписи в условиях, когда ключ подписи хранится на функциональном ключевом носителе (смарт-карте). Для такого типа прикладных систем в работе [4] были введены и математически строго описаны два типа нарушителей: внешний нарушитель и нарушитель с агентом. Задача обеспечения защиты от подделки подписи такими нарушителями может быть решена, в частности, за счет использования классических схем подписи с дополнительными свойствами или схем подписи вслепую. Модели безопасно-

сти, описывающие соответствующие свойства схем подписи и схем подписи вслепую, далее будем называть *специализированными моделями безопасности*.

Формализация целевой модели безопасности заключается в формировании строгих определений безопасности путем моделирования возможностей нарушителя по взаимодействию с механизмом, целей нарушителя и его ресурсов. В рамках диссертации применяется алгоритмический подход на основе экспериментатора, подробно описанный в [7, 23] и заключающийся в построении вероятностного интерактивного алгоритма, моделирующего работу схемы в присутствии нарушителя, и определении количественной характеристики успешности нарушителя по реализации угрозы — преимущества нарушителя. Отметим, что алгоритм работы экспериментатора не зависит от конкретного алгоритма работы нарушителя. Задача получения обоснованной оценки стойкости схемы подписи вслепую в конкретной модели безопасности в рамках используемого подхода сводится либо к предъявлению конкретного алгоритма работы нарушителя с заданными ограничениями, реализующего целевую угрозу в данной модели безопасности, и оценке снизу величины его преимущества (верхняя оценка стойкости схемы в данной модели безопасности), либо к получению верхней оценки величины преимущества для любого нарушителя с заданными ограничениями в данной модели безопасности (нижняя оценка стойкости схемы в данной модели безопасности). Верхняя оценка величины преимущества нарушителя в конкретной модели безопасности в общем случае представляет собой функцию от параметров схемы и преимуществ нарушителей в моделях безопасности для некоторых базовых примитивов, на основе которых построена схема.

Существующие схемы подписи вслепую. В основе стойкости существующих схем подписи вслепую могут лежать различные вычислительно трудные задачи. Так, самой первой и широко используемой на практике схемой является схема подписи вслепую Шаума [31], стойкость которой основана на сложности задачи факторизации. Известно также большое количество схем на основе решеток [46], изогений [73, 78] и других базовых примитивов. С точки зрения практики большой интерес представляет изучение схем подписи вслепую, использующих в качестве базового блока группу точек эллиптической кривой. Это обуславливается тем, что задача дискретного логарифмирования в группе точек эллиптической кривой является одной из наиболее изученных и надежных в мировом криптографическом сообществе. Более того, все отечественные стандартизированные на настоящий момент высокоуровневые асимметричные криптографические механизмы построены именно на основе эллиптических кривых.

Как показывает история анализа схем подписи вслепую, задача синтеза стойкой схемы на основе эллиптических кривых является вовсе не тривиальной. Как и в случае классических схем подписи, большинство схем подписи вслепую такого типа построены на основе одного

из двух уравнений подписи (или их незначительных модификаций):

- уравнение Шнорра [74];
- уравнение Эль-Гамаля [45].

Схемы обоих типов требуют осуществления как минимум трех пересылок между подписывающей и запрашивающей стороной в процессе формирования подписи. Поэтому при рассмотрении свойства неподделываемости для этих схем в общем случае необходимо учитывать возможность нарушителя проводить атаку с параллельными сеансами. Анализ безопасности относительно этой возможности приводит к интересным результатам: необходимым условием стойкости ряда схем подписи вслепую на основе уравнений Шнорра и Эль-Гамаля, в отличие от классических схем подписи Шнорра и Эль-Гамаля, является сложность новых нестандартных задач в группе точек эллиптической кривой [75]. Диссертационная работа посвящена методам получения обоснованных оценок стойкости для схем подписи вслепую такого типа.

Схемы подписи вслепую на основе уравнения Шнорра. Классической схемой подписи вслепую на основе уравнения Шнорра является схема подписи вслепую Шнорра, предложенная в 1996 году в работе [68]. Анализ стойкости этой схемы был проведен в 2001 году в работе [75]. Свойство неподделываемости (сильной неподделываемости в модели с параллельными сеансами) было доказано в предположении сложности задачи ROS (Random inhomogeneities in a Overdetermined Solvable system of linear equations) [75] относительно ограниченного множества нарушителей (в генерической модели [63] со случайным оракулом [22]). Позже в работе [42] было построено доказательство свойства неподделываемости относительно более широкого множества нарушителей (в модели с алгебраической группой [41] и случайным оракулом) в предположении сложности задач ROS и OMDL (One-More Discrete Logarithm, введена в [21]).

Задача ROS на протяжении 20 лет считалась сложной. В 2002 году она была сведена к обобщенной задаче дней рождения [83], для решения которой может быть использован алгоритм Вагнера [83] с субэкспоненциальной сложностью. В 2020 году был предложен полиномиальный алгоритм решения задачи ROS [24], который позволил построить полиномиальную атаку, приводящую к нарушению свойства неподделываемости для схемы подписи вслепую Шнорра в случае, если нарушитель имеет возможность открыть $\ell \geqslant \lceil \log q \rceil$ параллельных сеансов протокола формирования подписи с подписывающим, где q — простой порядок подгруппы группы точек эллиптической кривой.

Оказалось [24], что аналогичная атака применима не только к схеме подписи вслепую

Шнорра, но и к ряду других схем на основе эллиптических кривых: схеме Окамото-Шнорра [69], схеме Абе [15], обеспечивающей частичную неотслеживаемость, а также схеме Брандса [26], используемой в системе подтверждения персональных данных без их разглашения (anonymous credentials) U-Prove [66]. При этом для схемы Брандса атака позволяет строить подделки только при некоторых ограничениях на подписываемые в рамках атаки сообщения (для их формирования должен использоваться один и тот же «номер аккаунта» пользователя). Схема Брандса, в свою очередь, построена на основе схемы подписи вслепую Шаума-Педерсена [32]. Для схемы Шаума-Педерсена в литературе не представлено ни атак, ни строгого математического обоснования свойства неподделываемости, поэтому вопрос ее стойкости в расширенных моделях безопасности оставался открытым. Таким образом, актуальной задачей является получение верхних и/или нижних оценок стойкости этой схемы.

С момента публикации ROS атаки в литературе было предложено несколько схем подписи вслепую на основе уравнения Шнорра, для которых данная атака неприменима. Первым примером такой схемы является схема Clause Blind Schnorr, предложенная в 2020 году в работе [42]. Свойство неподделываемости для этой схемы обосновано в модели с алгебраической группой и случайным оракулом в предположении сложности задач OMDL и MROS (Modified ROS problem). Однако задача MROS введена только в 2020 году и почти не изучена, для нее отсутствуют нижние оценки трудоемкости ее решения. Позже в 2022 году в работе [81] была предложена схема подписи вслепую Tessaro-Zhu. Авторы этой схемы предложили модифицировать уравнение подписи Шнорра путем добавления в него еще одного случайного элемента для защиты от атак типа ROS. Свойство неподделываемости этой схемы математически строго обосновано в модели с алгебраической группой и случайным оракулом в предположении сложности задачи дискретного логарифмирования. Более того, в 2023 году в работе [34] была предложена пороговая схема подписи вслепую Snowblind, которая построена на основе схемы Tessaro-Zhu и позволяет уменьшить размер подписи (даже в случае одного подписывающего), свойство неподделываемости этой схемы было обосновано в тех же предположениях, что и для схемы Tessaro-Zhu. Наконец, в работе [30] была предложена схема подписи вслепую, которая построена на основе схемы Tessaro-Zhu и для которой удалось обосновать свойство неподделываемости в предположении сложности задачи CDH (Computational Diffie-Hellman) в модели со случайным оракулом, т.е. без использования модели с алгебраической группой. Также можно отметить схему подписи вслепую Абе на основе уравнения Шнорра, предложенную в работе [13] в 2001 году. Для этой схемы ROS атака также оказалась неприменимой, однако обоснование стойкости этой схемы, представленное в оригинальной работе [13], содержало ошибки [64]. Более того, исправленное обоснование, предложенное в работе [54] в 2020

году, также содержит ошибки, что подтверждают сами авторы доказательства.

Схемы на основе уравнения Эль-Гамаля. В литературе известно большое количество схем подписи вслепую, в основе которых лежит уравнение Эль-Гамаля [6, 29, 44, 52, 55, 62, 77, 79, 80, 85, 86]. Однако ни для одной из этих схем авторами не было представлено математически строгое обоснование свойства неподделываемости. Единственным исключением является схема, предложенная в работе [85]. Для этой схемы в работе [71] было математически строго обосновано свойство неподделываемости в модели с параллельными сеансами относительно ограниченного множества нарушителей. Однако настоящая схема представляет меньший интерес с точки зрения задачи синтеза схемы подписи на основе группы точек эллиптической кривой, поскольку для обеспечения неотслеживаемости она использует механизм неинтерактивного доказательства с нулевым разглашением, стойкость которого основана на сложности решения задачи факторизации.

Вместе с тем для схем, построенных на основе только группы точек эллиптической кривой, неизвестно каких-либо атак, позволяющих нарушить свойство неподделываемости. Таким образом, актуальной задачей является получение верхних и/или нижних оценок стойкости данных схем.

Цель диссертационной работы — построение новых математических методов получения обоснованных оценок стойкости схем подписи вслепую на основе уравнений Шнорра и Эль-Гамаля.

Для достижения поставленной цели были решены следующие задачи.

1) Разработка методов получения оценок стойкости схемы подписи вслепую Шаума-Педерсена в расширенных моделях безопасности.

2) Разработка методов получения оценок стойкости схем подписи вслепую на основе уравнения Эль-Гамаля в расширенных моделях безопасности.

3) Разработка методов получения оценок стойкости схем подписи и схем подписи вслепую на основе уравнения Эль-Гамаля в специализированных моделях безопасности, актуальных для систем формирования подписи в условиях использования функциональных ключевых носителей.

Научная новизна. В диссертации получены следующие новые результаты.

1) Для схемы подписи вслепую Шаума-Педерсена разработан метод нарушения свойства сильной неподделываемости в модели с параллельными сеансами и доказана содержательная верхняя оценка преимущества нарушителя, реализующего угрозу нарушения

свойства слабой неподделываемости в модели с параллельными сеансами. Полученные результаты демонстрируют, что схема Шаума-Педерсена не обеспечивает стойкость в наиболее сильной расширенной модели безопасности, при этом в основе стойкости схемы в более слабой расширенной модели безопасности лежит новая нестандартная задача в группе точек эллиптической кривой.

2) Синтезирован класс схем подписи вслепую на основе уравнения Эль-Гамаля, не использующих дополнительные криптографические механизмы, покрывающий все существующие схемы такого типа. Для существенной части схем из этого класса разработан метод нарушения свойства неподделываемости в модели с параллельными сеансами. Среди оставшихся схем выявлен подкласс схем, для которых разработан метод нарушения одного из свойств: свойства неподделываемости в модели с последовательными сеансами или свойства неотслеживаемости. Построенные методы демонстрируют, что все существующие схемы подписи вслепую на основе уравнения Эль-Гамаля не обеспечивают стойкость в расширенных моделях безопасности.

3) Разработан метод модификации схемы подписи Эль-Гамаля, позволяющий уменьшить размер подписи на четверть и обеспечить безопасность в условиях использования недоверенного датчика случайных чисел при формировании подписи. Для модифицированной схемы доказана содержательная верхняя оценка величины преимущества нарушителя в специализированной модели безопасности, предоставляющей нарушителю возможность выбирать случайные значения, используемые в процессе формирования подписи.

4) Для схем подписи вслепую на основе уравнения Эль-Гамаля доказаны содержательные верхние оценки преимущества нарушителя в специализированных моделях безопасности, актуальных в системах формирования подписи в условиях использования функциональных ключевых носителей.

Положения, выносимые на защиту.

1) Метод нарушения свойства сильной неподделываемости схемы Шаума-Педерсена в модели с параллельными сеансами.

2) Нижняя оценка стойкости схемы Шаума-Педерсена в модели, учитывающей свойство слабой неподделываемости и атаку с параллельными сеансами.

3) Методы нарушения свойств неподделываемости и неотслеживаемости схем подписи вслепую на основе уравнения Эль-Гамаля.

4) Нижняя оценка стойкости модифицированной схемы подписи Эль-Гамаля в специализированной модели безопасности, актуальной в системах формирования подписи в условиях использования функциональных ключевых носителей.

5) Нижние оценки стойкости схем подписи вслепую на основе уравнения Эль-Гамаля в специализированных моделях безопасности, актуальных в системах формирования подписи в условиях использования функциональных ключевых носителей.

Публикации по теме исследования. Результаты работы изложены в 5 публикациях в изданиях, индексируемых в базе ядра Российского индекса научного цитирования «eLibrary Science Index» и из списка ВАК Минобрнауки России; из них 4 — в изданиях, индексируемых в базе ядра Российского индекса научного цитирования «eLibrary Science Index», рекомендованных для защиты в диссертационном совете МГУ имени М. В. Ломоносова по специальности 2.3.6 «Методы и системы защиты информации, информационная безопасность».

Публикации в рецензируемых научных изданиях, индексируемых в базе ядра Российского индекса научного цитирования «eLibrary Science Index»

1) Akhmetzyanova L. R., Alekseev E. K., Babueva A. A., Smyshlyaev S. V. «On the (im)possibility of secure ElGamal blind signatures» //Математические вопросы криптографии. – 2023. – Т. 14. – №. 2. – С. 25–42 (RSCI WoS, ИФ РИНЦ: 0,071, 1,1 п.л.). EDN: MTAYSS.

   / Соавторам принадлежит постановка задачи и обзор существующих схем подписи вслепую на основе уравнения Эль-Гамаля. Остальные результаты статьи получены Бабуевой А.А. (1 п.л., 90%) /

2) Akhmetzyanova L. R., Alekseev E. K., Babueva A. A., Taraskin O. G. «On blindness of several ElGamal-type blind signatures» //Прикладная дискретная математика. – 2023. – №. 62. – С. 13–20 (Scopus, RSCI WoS, ИФ SJR: 0,135, 0,5 п.л.). EDN: IIXNSY.

   / Бабуевой А.А. принадлежат три метода нарушения свойства неотслеживаемости (0,4 п.л., 80%). Остальные результаты статьи получены соавторами. /

3) Akhmetzyanova L. R., Babueva A. A., Bozhko A. A. «Blind signature as a shield against backdoors in smart-cards» //Прикладная дискретная математика. – 2024. – №. 63. – С. 49–64 (Scopus, RSCI WoS, ИФ SJR: 0,135, 1 п.л.). EDN: KLXDGE.

   / Соавторам принадлежит сравнение разработанного метода обеспечения безопасности систем формирования подписи на основе использования схем подписи вслепую с методом на основе использования доказательства с нулевым разглашением Шнорра. Остальные результаты статьи получены Бабуевой А.А. (0,9 п.л., 90%) /

4) Ахметзянова Л. Р., Бабуева А. А. «О свойстве неподделываемости схемы подписи вслепую Шаума-Педерсена» //Прикладная дискретная математика. – 2024. – №. 65. – С. 41–65 (Scopus, RSCI WoS, ИФ SJR: 0,135, 1,6 п.л.). EDN: VEOFUM.

/ Соавторам принадлежит постановка задачи и обзор сложных задач в группе точек эллиптической кривой. Остальные результаты статьи получены Бабуевой А.А. (1,4 п.л., 88%) /

Публикации в рецензируемых научных изданиях, входящих в перечень ВАК Минобрнауки России

5) Бабуева А. А. «О модификации схемы подписи Эль-Гамаля для применения в одном классе систем голосования, использующих механизм подписи вслепую» //International Journal of Open Information Technologies. – 2023. – Т. 11. – №. 5. – С. 15–21 (ИФ РИНЦ: 0,458, 0,4 п.л., 100%). EDN: GLDDTU.

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

- XVI международной научно-практической конференции «Современные информационные технологии и ИТ-образование», Москва, 25–27 ноября 2021 года;
- семинаре «Математические методы криптографического анализа» кафедры информационной безопасности факультета Вычислительной математики и кибернетики Московского государственного университета им. М.В. Ломоносова под руководством Логачева О.А., Смышляева С.В., Алексеева Е.К., 2022 год;
- XI международной научной конференции «Современные тенденции в криптографии» (CTCrypt 2022), Новосибирск, 6–9 июня 2022 года;
- XII международной научной конференции «Современные тенденции в криптографии» (CTCrypt 2023), Волгоград, 6–9 июня 2023 года;
- XIII международной научной конференции «Современные тенденции в криптографии» (CTCrypt 2024), Петрозаводск, 3–6 июня 2024 года.

Теоретическая значимость. В ходе исследования получены результаты, существенно развивающие математические методы, применяемые при синтезе и анализе схем подписи вслепую.

При анализе схемы подписи вслепую Шаума-Педерсена были выявлены особенности способа маскирования сообщений для защиты от атак с параллельными сеансами и развиты математические методы обоснования оценок стойкости на основе матричного анализа. Также была выявлена новая задача в группе точек эллиптической кривой, сложность которой является достаточным условием обеспечения схемой свойства слабой неподделываемости, и установлена ее связь с другими существующими задачами.

Для схем подписи вслепую на основе уравнения Эль-Гамаля были изучены свойства базовых уравнений, а также способ выработки первой компоненты подписи и их влияние на обеспечение схемой подписи вслепую целевых свойств безопасности. Были выделены условия, которым должна удовлетворять схема подписи вслепую такого типа (в частности, конкретный вид уравнения подписи), чтобы потенциально обеспечивать стойкость в расширенных моделях безопасности. Данные условия могут быть использованы при синтезе новых схем подписи вслепую. Кроме того, доказана связь между специализированными моделями безопасности, релевантными при анализе систем формирования подписи в условиях использования функциональных ключевых носителей, с известными в литературе моделями безопасности для схем подписи вслепую.

Практическая значимость. Результаты диссертации использовались при выборе стандартизируемой в Российской Федерации схемы подписи вслепую. Так, схемы подписи вслепую на основе уравнения Эль-Гамаля и схема Шаума-Педерсена были исключены из рассмотрения в силу разработанных в диссертационной работе методов нарушения свойств неподделываемости и/или неотслеживаемости для этих схем.

Внедрение схем подписи вслепую на основе уравнения Эль-Гамаля в прикладные системы формирования подписи, предполагающие хранение ключа подписи на смарт-карте, позволяет повысить защищенность данных систем относительно внешнего нарушителя и нарушителя с агентом. При этом полученная нижняя оценка стойкости в специализированных моделях безопасности позволяет выбрать безопасные значения параметров эксплуатации схемы подписи вслепую без проведения дополнительных исследований.

Соответствие диссертации паспорту специальности.

Тема, объект и предмет исследований диссертации соответствуют паспорту специальности 2.3.6 (физико-математические науки) по следующим областям исследования:

1. теория и методология обеспечения информационной безопасности и защиты информации;

9. модели противодействия угрозам нарушения информационной безопасности для любо-

го вида информационных систем, позволяющие получать оценки показателей информационной безопасности;

10. модели и методы оценки защищенности информации и информационной безопасности объекта;

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

19. исследования в области безопасности криптографических алгоритмов, криптографических примитивов, криптографических протоколов. Защита инфраструктуры обеспечения применения криптографических методов.

Методология исследования

В рамках исследования применяется математический аппарат и подходы различных разделов математики, таких как теория вероятностей, линейная алгебра, теория сложности вычислений.

Структура и объем работы

Диссертационная работа состоит из введения, двух вспомогательных разделов, трех глав, заключения и списка литературы из 91 наименования. Работа изложена на 120 страницах.

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 разделе **Обозначения, определения и общие сведения** вводятся используемые в работе общие обозначения и определения, а также описываются базовые понятия алгоритмического подхода на основе экспериментатора к формализации моделей безопасности, предложенного в работе [23]. В рамках этого подхода формально вводятся объекты «нарушитель $\mathcal{A}$» и «экспериментатор **Exp**» — пара вероятностных интерактивных алгоритмов, взаимодействующих друг с другом определенным образом и моделирующих функционирование схемы в условиях присутствия нарушителя. Вводится понятие «преимущество нарушителя Adv» как мера успешности нарушителя по реализации угрозы.

Раздел **Модели безопасности для схем подписи вслепую** посвящен описанию известных в литературе моделей безопасности для схем подписи вслепую и их формализации. Для каждого из целевых свойств безопасности подробно рассматриваются особенности фор-

мирования модели нарушителя и модели угроз.

Для свойства неотслеживаемости определяются различные типы атаки (модели нарушителя) в зависимости от возможности нарушителя контролировать процесс генерации ключей, навязывать клиенту для подписи, а также получать информацию об ошибках в процессе формирования подписи. Математически строго определяется расширенная модель безопасности Blind (Blindness) для свойства неотслеживаемости, которая предоставляет нарушителю возможность генерировать ключи произвольным образом, навязывать клиенту сообщения для подписи и узнавать информацию о номере взаимодействия, завершившегося с ошибкой. Кроме того, формально определяется более слабая модель HS-Blind (Honest-Signer Blindness), которая отличается от модели Blind следующим образом: нарушитель не имеет возможности влиять на процесс генерации ключей, а также не получает информацию о номере взаимодействия, завершившегося с ошибкой. В качестве угрозы в обеих моделях рассматривается факт получения нарушителем нетривиальной информации о паре (сообщение, подпись), сформированной в результате взаимодействия с честным пользователем.

Для свойства неподделываемости модель угрозы определяется с помощью понятия «еще одной подделки». Задача нарушителя состоит в создании $(\ell+1)$ корректной пары (сообщение, подпись) в результате ℓ успешных взаимодействий с подписывающим. Под успешным взаимодействием понимается конкретный сеанс протокола формирования подписи, завершившийся с выходным результатом 1 на стороне подписывающего. В зависимости от ограничений, накладываемых на сформированные нарушителем пары, вводятся две различные угрозы: сильная неподделываемость (все пары должны быть различными) и слабая неподделываемость (все сообщения должны быть различными). Для данного свойства определяются также различные модели нарушителя в зависимости от возможности нарушителя осуществлять атаку с последовательными или с параллельными сеансами. Поскольку нарушитель выступает в роли клиента, он в том числе имеет возможность не завершать выполнение сеансов протокола формирования подписи. Математически строго определяется расширенная модель безопасности UF (UnForgeability) для свойства неподделываемости, рассматривающая угрозу нарушения сильной неподделываемости при атаке с параллельными сеансами. Экспериментатор в этой модели не контролирует порядок осуществления запросов на подпись нарушителем, т.е. позволяет начинать новый сеанс протокола формирования подписи в произвольный момент времени, в том числе до завершения предыдущих сеансов. Кроме того, формально определяются более слабые модели, модель SEQ-UF (SEQuential UnForgeability) и модель wUF (weak UnForgeability), отличающиеся от модели UF следующим образом. В модели SEQ-UF нарушителю предоставляется возможность осуществлять атаку только с последовательными

сеансами, в модели wUF рассматривается угроза нарушения слабой неподделываемости.

Глава 1 посвящена анализу безопасности схемы подписи вслепую Шаума-Педерсена. В разделе 1.1 приводится описание схемы для группы точек эллиптической кривой.

В разделе 1.2 приведены результаты анализа схемы Шаума-Педерсена с точки зрения обеспечения схемой свойства сильной неподделываемости в модели с параллельными сеансами (модель UF). Разработан конкретный метод нарушения свойства сильной неподделываемости с вероятностью близкой к 1 в результате открытия $\ell \geqslant \lceil \log q \rceil$ параллельных сеансов с подписывающим (теорема 1.2.1). В результате применения этого метода нарушитель предъявляет подделку для сообщения, ранее подписываемого в результате легитимного взаимодействия с подписывающим. Вместе с тем конструкция схемы Шаума-Педерсена не позволяет обобщить данный метод на случай построения подделки для нового, ранее не подписываемого сообщения.

В разделе 1.3 приведены результаты анализа схемы Шаума-Педерсена с точки зрения обеспечения схемой свойства слабой неподделываемости в модели с параллельными сеансами (модель wUF). С помощью техники сведений удалось доказать, что достаточным условием стойкости схемы в модели wUF с алгебраической группой и случайным оракулом является сложность решения двух задач: задачи REPR и SOMDL (теорема 1.3.1). Задача REPR является модификацией задачи «Representation», определенной в работе [26], ее сложность также является необходимым условием стойкости схемы Шаума-Педерсена. Задача SOMDL является новой задачей, определенной для группы точек элиптической кривой. Доказано, что настоящая задача не сложнее задачи OMDL (One-More Discrete Logarithm, введена в [21]) и задачи SDL (определена в [19] как задача q-dlog). Для задачи SDL доказано также, что ее сложность является необходимым условием стойкости схемы Шаума-Педерсена.

Таким образом, безопасное применение схемы Шаума-Педерсена потенциально возможно только в прикладных информационных системах, в которых обеспечивается уникальность подписываемых сообщений. При этом нижняя оценка стойкости схемы в таких условиях (в модели безопасности wUF) существенно зависит от нижней оценки трудоемкости решения новой нестандартной задачи SOMDL, а потому может быть существенно понижена в будущем. В силу того, что схема Шаума-Педерсена не обеспечивает стойкость в наиболее сильной расширенной модели безопасности, настоящая схема была исключена из рассмотрения в процессе выбора перспективной схемы подписи вслепую для стандартизации в Российской Федерации.

В **Главе 2** представлены результаты, обосновывающие невозможность построения стойкой в расширенных моделях безопасности схемы подписи вслепую на основе уравнения Эль-

Гамаля. В разделе 2.1 вводится общий вид классической схемы подписи семейства Эль-Гамаля, а также определяются допустимые виды уравнений подписи.

В разделе 2.2 для трех схем подписи вслепую на основе уравнения Эль-Гамаля (GYP16 [44], R00 [6], TNHV18 [79]) разработаны конкретные методы нарушения свойства неотслеживаемости с вероятностью близкой к 1 в результате только одного выполнения протокола формирования подписи. Данные методы могут быть применены сторонним пассивным нарушителем, не выступающим в роли подписывающего, т.е. даже в самых слабых моделях безопасности для свойства неотслеживаемости.

В разделе 2.3 определяется новый класс схем подписи вслепую — схемы подписи вслепую Эль-Гамаля GenEG-BS. Серверная сторона протокола формирования подписи в таких схемах зафиксирована и представляет собой классический алгоритм подписи Эль-Гамаля для маскированного хэш-значения сообщения e, сформированного пользователем произвольным образом. Введенная конструкция покрывает все существующие схемы на основе уравнения Эль-Гамаля за исключением схемы из работы [85], в которой алгоритм работы сервера включает в том числе проверку доказательства с нулевым разглашением.

Для введенного класса схем GenEG-BS в разделе 2.4 исследуется вопрос их стойкости в расширенных моделях безопасности. Все схемы класса GenEG-BS разделяются на два типа в зависимости от вида уравнения подписи. Для схем первого типа разработан метод нарушения свойства неподделываемости в модели UF с вероятностью близкой к 1 в результате открытия $\ell \geqslant \lceil \log q \rceil$ параллельных сеансов с подписывающим (теорема 2.4.1). Среди схем второго типа выделен подкласс схем, в которых зафиксирован способ выработки пользователем первой компоненты подписи r'. Все известные в литературе схемы GenEG-BS второго типа лежат в этом подклассе. Схемы из этого подкласса также разделены на два вида в зависимости от способа маскирования первой компоненты подписи и хэш-значения. Для первого вида схем разработан метод нарушения свойства неотслеживаемости в модели Blind с вероятностью близкой к 1 в результате выполнения одного протокола формирования подписи с честным пользователем (теорема 2.4.2). Для второго вида схем разработан метод нарушения свойства неподделываемости в модели SEQ-UF с вероятностью близкой к 1 в результате выполнения одного протокола формирования подписи с честным подписывающим (теорема 2.4.3). Как следствие, удалось доказать, что все известные в литературе схемы GenEG-BS [6, 29, 44, 52, 55, 62, 77, 79, 80, 86] не обеспечивают стойкость в расширенных моделях безопасности. Данные схемы были исключены из рассмотрения в процессе выбора перспективной схемы подписи вслепую для стандартизации в Российской Федерации.

Глава 3 посвящена вопросам безопасности, актуальным в системах формирования клас-

сической подписи в условиях, когда ключ подписи хранится на смарт-карте. В работе [4] введены два типа нарушителей для систем такого типа: внешний нарушитель и нарушитель с агентом. Внешний нарушитель моделирует «честного, но любопытного» нарушителя, действующего на стороне приложения формирования подписи; его цель — сформировать подделку подписи. Нарушитель с агентом моделирует ситуацию использования уязвимой смарт-карты. Этот нарушитель является составным. Первая часть представляет собой активного нарушителя на стороне смарт-карты, который может взаимодействовать только с доверенным приложением, т.е. отсутствуют другие каналы передачи данных от смарт-карты. Вторая часть представляет собой агента, который накапливает пары (сообщение, подпись), вычисленные приложением и недоверенной смарт-картой. Цель агента — сформировать подделку подписи. В диссертации предлагается рассматривать два возможных вида нарушителя с агентом: сильный (нарушитель может формировать значение подписи произвольным образом) и слабый (нарушитель на стороне смарт-карты формирует значение подписи согласно заданному протоколу, но использует недоверенный датчик случайных чисел, т.е. выбирает произвольным образом случайные значения, используемые в алгоритме формирования подписи).

Раздел 3.1 содержит результаты анализа метода обеспечения защиты от внешнего нарушителя и слабого нарушителя с агентом. Для классической схемы подписи определяется специализированная модель безопасности SUF-CMRA [72], которая рассматривает угрозу построения подделки и предоставляет нарушителю возможность адаптивно выбирать сообщения для подписи, а также случайные значения и метки времени, используемые в процессе формирования подписи. Предлагается метод модификации классической схемы подписи Эль-Гамаля, обеспечивающий защиту от использования низкоэнтропийных случайных значений за счет дополнительного замешивания ключа подписи в процесс генерации одноразового секрета с помощью функции HMAC [57]. Кроме того, данный метод позволяет на четверть сократить длину подписи по сравнению с оригинальной схемой, что позволяет повысить производительность рассматриваемого типа систем. Для модифицированной схемы подписи Эль-Гамаля получена содержательная верхняя оценка преимущества нарушителя в модели SUF-CMRA со случайным оракулом (теорема 3.1.1). Данная оценка демонстрирует, что разработанный метод является безопасным в модели SUF-CMRA в предположении псевдослучайности функции HMAC.

Раздел 3.2 содержит результаты анализа метода обеспечения защиты целевых прикладных систем от внешнего нарушителя и сильного нарушителя с агентом, основанный на использовании схем подписи вслепую. В диссертации вводятся новые специализированные мо-

дели безопасности для схем подписи вслепую — SA-UF и HBC-UF, стойкость в которых требуется от схемы подписи вслепую для защиты от сильного нарушителя с агентом и внешнего нарушителя соответственно. Модель SA-UF описывает нарушителя, состоящего из двух алгоритмов. Первый алгоритм обладает всеми возможностями сервера, второй алгоритм (агент) имеет возможность накапливать для адаптивно выбираемых сообщений значения подписей, формируемых честным клиентом в результате взаимодействия с нарушителем-сервером. В качестве угрозы рассматривается построение подделки агентом. Модель HBC-UF описывает так называемого «честного, но любопытного» нарушителя, она предоставляет нарушителю возможность навязывать клиенту сообщения для подписи, получать стенограммы протокола формирования подписи и все случайные значения, выбранные клиентом в процессе выполнения протокола. В качестве угрозы также рассматривается построение поддлелки. Изучается связь между специализированными моделями безопасности и известными в литературе моделями безопасности для схем подписи вслепую. Доказано, что любая схема подписи вслепую обеспечивает стойкость в модели SA-UF, если она обеспечивает свойство неподделываемости относительно внешнего нарушителя и свойство неотслеживаемости в модели HS-Blind (теорема 3.2.1). Для частного случая схем подписи вслепую на основе уравнения Эль-Гамаля доказано, что они обеспечивают стойкость в модели HBC-UF, если базовая схема подписи Эль-Гамаля обеспечивает свойство неподделываемости (теорема 3.2.2).

Для использования в прикладных системах, реализующих схему подписи ГОСТ Р 34.10-2012 [1] на основе эллиптических кривых, определенных документом Р 1323565.1.024-2019 [2][1], предложено использовать схему подписи вслепую Камениша [29], в основе которой лежит такое же уравнение подписи. Удалось доказать, что эта схема обеспечивает защиту целевых прикладных систем при единственном предположении, что схема подписи ГОСТ Р 34.10-2012 обеспечивает свойство неподделываемости (утверждение 3.2.1). Таким образом, схемы подписи вслепую на основе уравнения подписи Эль-Гамаля могут использоваться для защиты целевых прикладных систем несмотря на то, что они не обеспечивают стойкость в расширенных моделях безопасности.

В **Заключении** перечислены основные результаты диссертации.

Благодарности. Автор диссертации выражает благодарность своему научному руково-

[1] Эллиптические кривые, определенные в настоящем документе, имеют достаточно большую степень расширения (англ. embedding degree), что делает неприменимыми для них атаки, описанные в работе: Черепнёв М. А., Грачева С. С. Решение задачи Диффи-Хеллмэна на некоторых эллиптических кривых, удовлетворяющих ГОСТ 34.10-2018 //Информационные технологии. – 2020. – Т. 26. – №. 3. – С. 159-168.

дителю доктору физико-математических наук Смышляеву Станиславу Витальевичу за постановку задачи, постоянное внимание к работе и поддержку, а также доктору физико-математических наук Логачеву Олегу Алексеевичу, кандидату физико-математических наук Ахметзяновой Лилии Руслановне, кандидату физико-математических наук Алексееву Евгению Константиновичу, кандидату физико-математических наук Кяжину Сергею Николаевичу за полезные обсуждения и рекомендации. Автор также признателен заведующему кафедры Информационной безопасности ВМК МГУ имени М.В. Ломоносова академику Соколову Игорю Анатольевичу и всем ее сотрудникам за поддержку и внимание к диссертационной работе.

Обозначения, определения и общие сведения

Обозначим через $\{0,1\}^u$ множество всех u-битовых строк, через $\{0,1\}^*$ — множество всех битовых строк конечной длины, в том числе пустую строку. Битовую строку, состоящую из u нулей, будем обозначать через 0^u.

Для целых чисел $\ell > 0$ и $0 \leqslant i < 2^\ell$ через $\mathrm{str}_\ell(i)$ будем обозначать ℓ-битовое представление числа i, в котором наименее значащий бит находится справа. Для целого числа $\ell > 0$ и битовой строки $U \in \{0,1\}^\ell$ через $\mathrm{int}(U)$ будем обозначать целое число $i < 2^\ell$, такое что $\mathrm{str}_\ell(i) = U$.

Если p простое число, то через $\mathbb{Z}_p$ обозначается поле вычетов по модулю p. Каждый ненулевой элемент x поля $\mathbb{Z}_p$ имеет обратный элемент по умножению $1/x$. Операции сложения и умножения в поле $\mathbb{Z}_p$ обозначаются символами «+» и «·» соответственно. Через $\mathbb{Z}_p^*$ обозначается множество $\mathbb{Z}_p$ без нулевого элемента, т.е. мультипликативная группа поля $\mathbb{Z}_p$.

Группа точек эллиптической кривой, определенной над полем $\mathbb{Z}_p$, обозначается через $\mathbb{G}$, порядок простой подгруппы $\mathbb{G}$ через q и точка эллиптической кривой порядка q через P. Нулевая точка кривой обозначается через $\mathcal{O}$. Через $\mathbb{G}^*$ обозначается множество точек кривой без нулевой точки. Через H обозначается хэш-функция, отображающая двоичные строки в элементы $\mathbb{Z}_q^*$, через $\mathcal{H} : \{0,1\}^* \to \mathbb{G}^*$ — хэш-функция, отображающая двоичные строки в точки кривой. Через $\mathrm{DLog}_B(A)$, $A, B \in \mathbb{G}$, обозначается число $\alpha \in \mathbb{Z}_q$, такое что $A = \alpha B$.

Запись $x \xleftarrow{\mathcal{U}} X$ означает, что элемент x выбирается из множества X случайно в соответствии с равномерным распределением. Далее будем называть такой элемент x случайным. Событие, что алгоритм A вернул значение val в качестве результата работы, обозначается через $A \to val$ ($val \leftarrow A$). Будем обозначать множество всех отображений из A в B через $Func(A, B)$.

Модели безопасности. Модель безопасности определяется двумя компонентами: моделью нарушителя и моделью угрозы. Модель нарушителя определяет качественные и количественные возможности нарушителя по взаимодействию с исследуемой криптографической системой. Под качественными возможностями понимается то, каким образом нарушитель может вмешиваться в процесс работы системы и какую именно информацию о системе он может получать. Количественные возможности определяют ограничения объема вычислительных ресурсов нарушителя и объема информации, которую он может получить в результате взаимодействия с системой. Модель угрозы определяет задачу по нарушению свойств безопасности, которую стремится решить нарушитель.

Для формального задания модели безопасности будем использовать *алгоритмический подход на основе экспериментатора*. В рамках настоящего подхода исследуемая система описывается как набор интерактивных алгоритмов, каждый из которых соответствует конкретному участнику системы, нарушитель $\mathcal{A}$ также определяется как некоторый интерактивный алгоритм. Взаимодействие нарушителя с честными участниками описывается с помощью так называемого «экспериментатора» **Exp** — вероятностного интерактивного алгоритма, моделирующего для нарушителя функционирование исследуемой криптосистемы. Совокупность двух взаимодействующих друг с другом вероятностных алгоритмов, экспериментатора **Exp** и нарушителя $\mathcal{A}$, называется экспериментом.

При задании эксперимента нарушитель $\mathcal{A}$ явно не описывается. Единственными предположениями о внутреннем устройстве алгоритма $\mathcal{A}$ являются его согласованность, как интерактивного алгоритма, с интерфейсом экспериментатора и ограниченность его ресурсов.

Процесс взаимодействия нарушителя и экспериментатора в рамках эксперимента состоит из следующих этапов. В начале эксперимента производится инициализация параметров системы, например, экспериментатор может выбирать секретный ключ подписи и возвращать нарушителю соответствующий открытый ключ проверки подписи. Дальнейшее взаимодействие между нарушителем и экспериментатором осуществляется с помощью определенного набора подпрограмм экспериментатора, которые называются оракулами и являются частью его интерфейса. Нарушитель может делать определенные запросы к данным оракулам (например, запросы на подпись сообщений) и получать от них соответствующие ответы. Данный интерфейс формализует первую компоненту модели безопасности — возможности нарушителя. Для обозначения факта того, что нарушитель $\mathcal{A}$ имеет доступ к некоторому оракулу $\mathcal{O}$ экспериментатора, используется запись $\mathcal{A}^{\mathcal{O}}$. Завершение эксперимента осуществляется с помощью процедуры финализации — экспериментатор обрабатывает данные, полученные от нарушителя (например, проверяет корректность пары (сообщение, подпись), сформированной нарушителем), и возвращает значение $res \in \{0, 1\}$ в качестве результата взаимодействия. Способ определения значения res формализует вторую компоненту модели безопасности, модель угрозы, на качественном уровне. Например, результат эксперимента полагается равным 1, если нарушитель успешно сформировал подделку подписи. Количественная характеристика угрозы (например, вероятность построения подделки подписи) формализуется путем задания «меры успешности» нарушителя $\mathcal{A}$. Такая характеристика называется преимуществом нарушителя и обозначается через $\mathsf{Adv}(\mathcal{A})$.

Формализация ресурсов нарушителя осуществляется следующим образом. Объем доступной нарушителю информации задается с помощью количественных ограничений его взаимо-

действий с экспериментатором. Способ определения вычислительных ресурсов нарушителя зависит от выбранной модели вычислений. Например, в случае использования машины Тьюринга вычислительные ресурсы определяются количеством тактов работы нарушителя и размером его программы. Нарушитель называется полиномиальным, если размер его вычислительных ресурсов определяется полиномом от параметра безопасности (для рассматриваемых в диссертационной работе схем параметр безопасности равен $\lceil \log q \rceil$).

Таким образом, формальное определение модели нарушителя заключается в описании экспериментатора, задании преимущества нарушителя и ограничений доступных ему объемов информации и вычислительных ресурсов. Подробное описание алгоритмического подхода на основе экспериментатора, а также используемые в рамках него термины и обозначения приведены в работах [7, 23].

Схема подписи. Определим формально классическую схему подписи SS и целевые свойства безопасности.

Определение 0.1. Схема подписи SS задается следующими алгоритмами:

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(\ )$: алгоритм генерации ключей, возвращающий пару ключей $(\mathsf{sk}, \mathsf{pk})$, где sk — секретный ключ подписи, pk — открытый ключ проверки подписи;
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$: алгоритм формирования подписи, принимающий на вход секретный ключ подписи sk и сообщение m и возвращающий значение подписи σ;
- $b \leftarrow \mathsf{Verify}(\mathsf{pk}, m, \sigma)$: детерминированный алгоритм проверки подписи, принимающий на вход ключ проверки подписи pk, сообщение m и подпись σ и возвращающий единицу, если значение подписи верное, и ноль в противном случае.

При этом для любой пары ключей $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(\ )$ и для любого сообщения m требуется, чтобы

$$\mathsf{Verify}(\mathsf{pk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = 1.$$

Схема подписи должна обеспечивать свойство неподделываемости при атаке с выбором сообщений, которое традиционно формализуется с помощью моделей UF-CMA и SUF-CMA ([Strong] UnForgeability under Chosen Message Attack). Отличие этих моделей заключается в следующем. В модели UF-CMA нарушитель должен вернуть подделку для нового сообщения, для которого ранее он не запрашивал значение подписи. В модели SUF-CMA подделкой считается в том числе создание новой подписи для старого сообщения. Определим формально модель SUF-CMA.

Определение 0.2. Для нарушителя $\mathcal{A}$ и схемы подписи SS:

$$\mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{SS}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{SUF-CMA}}_{\mathsf{SS}}(\mathcal{A}) \to 1\right],$$

где эксперимент $\mathbf{Exp}^{\text{SUF-CMA}}_{\mathsf{SS}}(\mathcal{A})$ определяется следующим образом:

$\mathbf{Exp}^{\text{SUF-CMA}}_{\mathsf{SS}}(\mathcal{A})$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{SS.KGen}(\ )$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $(m^*, \sigma^*) \leftarrow \mathcal{A}^{Sign}(\mathsf{pk})$
4 : **if** $(m^*, \sigma^*) \in \mathcal{L}$: **return** 0
5 : **return** $\mathsf{SS.Verify}(\mathsf{pk}, m^*, \sigma^*)$

Oracle $Sign(m)$

1 : $\sigma \leftarrow \mathsf{SS.Sign}(\mathsf{sk}, m)$
2 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
3 : **return** σ

Если в настоящем определении экспериментатор сохраняет в множество $\mathcal{L}$ только значения сообщений (см. строку 2 в оракуле подписи) и проверяет вхождение только сообщения m^* в множество $\mathcal{L}$ при финализации эксперимента (см. строку 4 эксперимента), то получим определение модели UF-CMA.

Схема подписи вслепую. Определим формально схему подписи вслепую BS.

Определение 0.3. Схема подписи вслепую BS задается следующими алгоритмами:

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(\ )$: алгоритм генерации ключей, возвращающий пару ключей $(\mathsf{sk}, \mathsf{pk})$, где sk — секретный ключ подписи, pk — открытый ключ проверки подписи;
- $(b, \sigma) \leftarrow \langle \mathsf{Sign}(\mathsf{sk}, \mathsf{pk}), \mathsf{User}(\mathsf{pk}, m) \rangle$: интерактивный протокол, выполняемый между подписывающим, обладающим секретным ключом подписи sk, и клиентом, обладающим сообщением m; подписывающий выдает $b = 1$, если взаимодействие успешно завершилось, и $b = 0$ в противном случае; клиент выдает значение подписи σ в случае успешного завершения протокола и $\perp$ в противном случае.
- $b \leftarrow \mathsf{Verify}(\mathsf{pk}, m, \sigma)$: детерминированный алгоритм проверки подписи, принимающий на вход ключ проверки подписи pk, сообщение m и подпись σ и возвращающий единицу, если значение подписи верное, и ноль в противном случае.

При этом для любой пары ключей $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(\ )$ и для любого сообщения m требуется, чтобы в результате выполнения

$$(b, \sigma) \leftarrow \langle \mathsf{Sign}(\mathsf{sk}, \mathsf{pk}), \mathsf{User}(\mathsf{pk}, m) \rangle$$

$$b' \leftarrow \mathsf{Verify}(\mathsf{pk}, m, \sigma)$$

было выполнено $b = b' = 1$.

Интерактивный протокол формирования подписи может состоять из произвольного количества обменов сообщениями (раундов) между подписывающим и клиентом, каждый обмен представляет собой два сообщения, запрос клиента и ответ подписывающей стороны. Для двухраундовых схем интерактивный протокол формирования подписи $\langle \mathsf{Sign}, \mathsf{User} \rangle$ можно задать следующим образом:

$$(msg_{U,0}, state_{U,0}) \leftarrow \mathsf{BS.User}_0(\mathsf{pk}, m)$$
$$(msg_{S,1}, state_{S,1}) \leftarrow \mathsf{BS.Sign}_1(\mathsf{sk}, \mathsf{pk}, msg_{U,0})$$
$$(msg_{U,1}, state_{U,1}) \leftarrow \mathsf{BS.User}_1(state_{U,0}, msg_{S,1})$$
$$(msg_{S,2}, b) \leftarrow \mathsf{BS.Sign}_2(state_{S,1}, msg_{U,1})$$
$$\sigma \leftarrow \mathsf{BS.User}_2(state_{U,1}, msg_{S,2})$$

где $msg_{role,i}$ означает сообщение с порядковым номером i, отправляемое стороной $role, role \in \{U, S\}$, в рамках протокола, а переменная $state_{role,i}$ позволяет стороне взаимодействия $role$ сохранить некоторое внутреннее состояние на i-м раунде и использовать его на последующем раунде.

Если первая пересылка от пользователя к подписывающему является пустой ($msg_{U,0} = \theta$), то настоящее описание можно упростить следующим образом:

$$(msg_{S,1}, state_{S,1}) \leftarrow \mathsf{BS.Sign}_1(\mathsf{sk}, \mathsf{pk})$$
$$(msg_{U,1}, state_{U,1}) \leftarrow \mathsf{BS.User}_1((\mathsf{pk}, m), msg_{S,1})$$
$$(msg_{S,2}, b) \leftarrow \mathsf{BS.Sign}_2(state_{S,1}, msg_{U,1})$$
$$\sigma \leftarrow \mathsf{BS.User}_2(state_{U,1}, msg_{S,2})$$

Для схемы подписи вслепую можно аналогичным Определению 0.2 образом определить модели UF-CMA/SUF-CMA. Отличие будет заключаться только в строке 1 оракула подписи, где будет вызываться интерактивный протокол формирования подписи вслепую:

$$(1, \sigma) \leftarrow \langle \mathsf{Sign}(\mathsf{sk}, \mathsf{pk}), \mathsf{User}(\mathsf{pk}, m) \rangle.$$

Такая модель будет характеризовать стойкость схемы подписи вслепую относительно пассивного нарушителя, имеющего возможность навязывать честному клиенту подписание определенных сообщений и получать сформированные значения подписей. Однако для схем подписи вслепую на практике требуются более сильные свойства безопасности, настоящие свойства подробно обсуждаются и математически строго определяются в следующем разделе.

Модели безопасности для схем подписи вслепую

Для схем подписи вслепую традиционно рассматривают [31, 53, 68] следующие два свойства безопасности:

- неотслеживаемость (blindness): в результате выполнения протокола формирования подписи подписывающий не получает никакой нетривиальной информации о паре (сообщение, подпись), сформированной клиентом;
- неподделываемость (unforgeability): клиент может сформировать корректную подпись только в результате успешного взаимодействия с подписывающим.

Базовые математически строгие определения настоящих свойств были впервые представлены в работах [53, 68], после чего эти определения были уточнены, расширены, доработаны в работах [12, 17, 28, 40, 47, 58, 76]. В настоящем разделе проводится обзор существующих моделей безопасности для схем подписи вслепую. Математически строгие определения моделей безопасности приведены для двухраундовых схем подписи вслепую, поскольку рассматриваемые в диссертационной работе схемы относятся к схемам именно такого типа.

Свойство неотслеживаемости

Неформально свойство неотслеживаемости означает, что подписывающая сторона (сервер) не получает никакой нетривиальной информации о паре (сообщение, подпись), сформированной клиентом в результате работы протокола. Таким образом, нарушителем в данном случае является подписывающая сторона.

Модель нарушителя

Нарушителю предоставляется возможность взаимодействовать с клиентом, посылая ему произвольные сообщения, при этом предполагается, что клиент функционирует корректным образом.

Генерация ключей. Генерация ключей подписи в схеме подписи вслепую происходит на стороне подписывающего, который является нарушителем с точки зрения свойства неотслеживаемости. Исходное определение данного свойства, предложенное в [53], подразумевает, что генерация ключей происходит честным образом и нарушитель не влияет на этот процесс.

Однако позже в работе [12] было предложено рассматривать более сильную модель, позволяющую нарушителю генерировать ключи произвольным образом. Таким образом, различают следующие два типа атаки:

- слабая атака (модель с честным подписывающим): подписывающий генерирует ключи корректным образом;
- сильная атака (модель с нечестным подписывающим): подписывающий может выбирать ключи как угодно.

Далее для обозначения слабой атаки, т.е. честного процесса генерации ключей, в названии модели используется префикс HS (Honest Signer).

Выбор сообщений. В общем случае сообщения, подписываемые клиентом в рамках протокола формирования подписи вслепую, выбираются клиентом в соответствии с некоторым распределением, которое может иметь сложную структуру (например, это могут быть токены, используемые в рамках некоторых финансовых операций, или бюллетени). Моделирование распределений такого типа зачастую является нетривиальной задачей, поэтому традиционно при формализации свойства неотслеживаемости нарушителю предоставляют возможность самостоятельно выбирать сообщения, подпись для которых будет формировать клиент.

Однако в работе [58] было показано, что для широко используемой на практике схемы подписи вслепую на основе RSA, схемы RSA-BSSA, такое предположение является слишком сильным (существует атака на свойство неотслеживаемости). В связи с этим в этой работе было предложено рассматривать более слабую модель, в которой сообщения для подписи выбираются клиентом случайно в соответствии с некоторым распределением $\mathcal{M}$, являющимся параметром модели. Такая модель получила название «blind token».

Таким образом, с точки зрения выбора сообщений выделяют два возможных типа атаки:

- слабая атака: сообщения для подписи выбираются клиентом;
- сильная атака: нарушитель сам выбирает сообщения, подпись для которых будет формировать клиент.

Обработка ошибок. Поскольку нарушитель выступает в роли подписывающего, ему предоставляется возможность не завершать сеансы и адаптивно провоцировать сбои, т.е. посылать некорректные сообщения, обработка которых на стороне клиента завершится с

ошибкой. Оригинальное определение свойства неотслеживаемости [53] подразумевает, что в случае, если хотя бы одно из взаимодействий завершилось с ошибкой на стороне клиента, нарушителю в результате всех взаимодействий возвращается символ ошибки. Однако в работе [28] отмечено, что в этом случае не учитываются атаки, позволяющие подписывающему вынудить клиента прервать протокол из-за некоторого свойства подписываемого сообщения. В этой работе была введена так называемая модель «selective failure blindness», в которой нарушитель получает возможность узнавать номера взаимодействий, завершившихся с ошибкой. В работе [40] было показано, что если схема обеспечивает свойство неотслеживаемости с такой возможностью нарушителя, то она обеспечивает ее и в отсутствие у нарушителя этой возможности.

Таким образом, в зависимости от информации, которую получает нарушитель в случае возникновения сбоя хотя бы в одном сеансе, выделяют следующие два типа атаки:

- слабая атака: нарушитель не получает никакой информации;
- сильная атака: нарушитель получает информацию о номерах сеансов, взаимодействие в которых завершилось с ошибкой на стороне клиента.

Модель угрозы

Традиционным подходом к формализации «отсутствия какой-либо информации» у нарушителя является формализация с помощью задачи различения (см., например, [43]). Например, свойство конфиденциальности для схем шифрования формализуется с помощью модели LOR-CPA: нарушитель не может определить, какому из двух выбранных им сообщений соответствует шифртекст. Интуитивным аналогом этого свойства для схем подписи вслепую является следующее свойство: нарушитель не может определить, какое из двух выбранных им сообщений подписывалось в ходе взаимодействия.

Однако для схем подписи вслепую такое определение угрозы напрямую неприменимо. Действительно, по значению подписи, в отличие от значения шифртекста, нарушитель может однозначно определить, для какого из двух сообщений сформированная подпись является корректной. В связи с этим при формализации предполагается, что нарушитель два раза взаимодействует с честным клиентом, при этом сообщения, выбранные нарушителем, подписываются в случайном порядке. После этого нарушителю одновременно выдаются оба значения подписи. Задачей нарушителя является сопоставление номера взаимодействия (сеанса) и сформированной в результате него пары (сообщение, подпись). Заметим, что нарушитель всегда может угадать это соответствие с вероятностью $1/2$, даже не используя никакой

имеющейся у него информации. Отклонение вероятности успешного решения задачи от $1/2$ характеризует то, насколько эффективно нарушитель смог использовать информацию, полученную из взаимодействий, а значит, и стойкость схемы подписи вслепую.

Отдельно отметим, что факт успешного решения задачи с вероятностью отличной от $1/2$ означает, что в результате взаимодействий нарушитель получил некоторую дополнительную информацию о сформированной паре (сообщение, подпись), которая позволила ему правильно сопоставить эту пару и сеанс протокола.

Кратное число взаимодействий и обработка ошибок. Оригинальное определение свойства неотслеживаемости подразумевает, что нарушителю доступно только два взаимодействия с клиентом. На практике количество взаимодействий с клиентом может существенно превышать это число. В работах [40, 47] математически строгие определения свойства неотслеживаемости обобщаются на случай наличия у нарушителя $n \geqslant 2$ взаимодействий.

В модели, определенной в работе [40], нарушитель выбирает n сообщений, которые будут подписываться клиентом, а порядок подписания этих сообщений определяется выбором случайной подстановки π на множестве $\{1, \ldots, n\}$. Нарушителю предоставляется возможность компрометировать часть переходов в выбранной перестановке за счет доступа к оракулу $Corrupt$, возвращающему на запрос i значение $\pi(i)$. В случае возникновения хотя бы одной ошибки нарушителю возвращаются номера взаимодействий, завершившихся с ошибкой (аналогично модели selective failure blindness для двух взаимодействий). Угрозу авторы работы [40] предлагают формулировать одним из двух возможных способов:

- через задачу отличения: нарушитель возвращает пару (i, i'), такую что $\pi(i) < \pi(i')$. В этом случае преимущество нарушителя определяется через разность вероятности успешного решения задачи и $1/2$, дополнительно требуется, чтобы количество нескомпрометированных переходов было не менее двух;

- через задачу поиска: нарушитель возвращает пару (i, j), такую что $\pi(i) = j$. В этом случае преимущество нарушителя определяется через разность вероятности успешного решения задачи и $1/r$, где r — количество нескомпрометированных переходов.

Авторы [40] показывают, что эти способы определения угрозы являются эквивалентными, и что частный случай настоящей модели при $n = 2$ совпадает с сильной моделью для свойства неотслеживаемости для двух взаимодействий (ключи и сообщения выбираются нарушителем, нарушитель получает информацию о номерах сеансов, завершившихся с ошибкой).

В модели, определенной в работе [47], сообщения для подписи выбираются клиентом в

соответствии с некоторым распределением, задаваемым нарушителем, они подписываются в прямом порядке, после чего нарушителю возвращаются только успешно сформированные пары (сообщение, подпись) в порядке, определяемом случайно выбранной подстановкой π на множестве $\{1, \ldots, r\}$, где $r \leqslant n$ — количестве успешно завершившихся сеансов. Угроза определяется через задачу поиска: нарушитель возвращает пару (i, j), такую что $\pi(i) = j$ и сеанс с номером i был успешно завершена. В этом случае преимущество нарушителя определяется через разность вероятности успешного решения задачи и $1/r \cdot p$, где p — вероятность того, что ровно r сеансов завершилось успешно. Таким образом, стойкость в модели из работы [47] свидетельствует об обеспечении свойства неотслеживаемости только для успешно завершенных сеансов и ничего не говорит о сеансах, завершившихся с ошибкой. Настоящая модель получила название «a posteriori blindness». Из стойкости в ней не следует стойкость в классической модели для свойства неотслеживаемости и наоборот [47], поэтому она не является широко используемой.

Кроме того, в работе [47] упоминается возможность формализации свойства неотслеживаемости для $n \geqslant 2$ взаимодействий через задачу отличения двух перестановок на множестве сообщений, адаптивно выбираемых нарушителем. Однако никаких математически строгих определений, а также рассуждений об эквивалентности такой формулировки угрозы формулировке через задачу поиска не приводится.

Совершенная и вычислительная неотслеживаемость. Для обеспечения свойства неотслеживаемости необходимо, чтобы стенограммы протокола (т.е. наборы сообщений, пересылаемых от подписывающего клиенту и обратно), созданные в результате формирования подписи для различных сообщений, были неотличимы друг от друга. Если под неотличимостью понимается статистическая неотличимость (статистическое расстояние между стенограммами равно нулю), то говорят, что протокол обеспечивает совершенную неотслеживаемость. Если же имеется в виду вычислительная неотличимость, т.е. «близость» стенограмм измеряется в терминах вычислительного расстояния между ними, то говорят о вычислительной неотслеживаемости.

Математические модели безопасности

Модель Blind (Blindness). Зададим формально наиболее сильную модель Blind, которая предоставляет нарушителю возможность генерировать ключи произвольным образом, выбирать сообщения для подписи и узнавать информацию о номере взаимодействия, завершившегося с ошибкой.

Определение 0.4. Для двухраундовой схемы подписи вслепую BS

$$\mathsf{Adv}^{\mathrm{Blind}}_{\mathsf{BS}}(\mathcal{A}) = \left|\Pr\left[\mathbf{Exp}^{\mathrm{Blind},1}_{\mathsf{BS}}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^{\mathrm{Blind},0}_{\mathsf{BS}}(\mathcal{A}) \to 1\right]\right|,$$

где эксперименты $\mathbf{Exp}^{\mathrm{Blind},b}_{\mathsf{BS}}(\mathcal{A}), b \in \{0,1\}$, определяются следующим образом:

$\underline{\mathbf{Exp}^{\mathrm{Blind},b}_{\mathsf{BS}}(\mathcal{A})}$

1 : $b_0 \leftarrow b$

2 : $b_1 \leftarrow 1 - b$

3 : $st \leftarrow \mathcal{A}^{Init}(\,)$

4 : $b' \leftarrow \mathcal{A}^{User_0, User_1, User_2}(st)$

5 : **return** $(b = b')$

Oracle $Init(\mathsf{pk}, m_0, m_1)$

1 : $sess_0 \leftarrow \mathsf{init}$

2 : $sess_1 \leftarrow \mathsf{init}$

3 : store (pk, m_0, m_1)

4 : **return** θ

Oracle $User_0(i)$

1 : **if** $i \notin \{0,1\} \vee sess_i \neq \mathsf{init}$: **return** $\perp$

2 : $sess_i \leftarrow \mathsf{open}_1$

3 : $(msg_i, state_i) \leftarrow \mathsf{BS.User}_0(\mathsf{pk}, m_{b_i})$

4 : **return** msg_i

Oracle $User_1(i, msg)$

1 : **if** $sess_i \neq \mathsf{open}_1$: **return** $\perp$

2 : $sess_i \leftarrow \mathsf{open}_2$

3 : $(msg_i, state_i) \leftarrow \mathsf{BS.User}_1(state_i, msg)$

4 : **return** msg_i

Oracle $User_2(i, msg)$

1 : **if** $sess_i \neq \mathsf{open}_2$: **return** $\perp$

2 : $sess_i \leftarrow \mathsf{closed}$

3 : $\sigma_{b_i} \leftarrow \mathsf{BS.User}_2(state_i, msg)$

4 : **if** $sess_0 = sess_1 = \mathsf{closed}$:

5 : **if** $\sigma_{b_0} = \perp \wedge \sigma_{b_1} = \perp$: **return** $(\perp, \perp)$

6 : **if** $\sigma_{b_0} = \perp$: **return** $(\perp, \theta)$

7 : **if** $\sigma_{b_1} = \perp$: **return** $(\theta, \perp)$

8 : **return** (σ_0, σ_1)

9 : **return** θ

Первым запросом нарушителя является запрос к оракулу $Init$, определяющий сообщения, подпись для которых будет формироваться клиентом, и ключ проверки подписи. При обработке этого запроса происходит инициализация обоих сеансов формирования подписи, экспериментатор сохраняет значение открытого ключа и сообщений (здесь и далее запись «store x» означает факт сохранения экспериментатором значения x в качестве глобальной переменной). Соответствие идентификаторов сеансов и поданных нарушителем сообщений определяется значением бита b.

Взаимодействие с клиентом моделируется с помощью доступа к оракулам $User_0$, $User_1$ и $User_2$, которые формируют сообщения клиента в протоколе и непосредственно подпись. На вход оракулу $User_0$ подается номер сеанса $i \in \{0, 1\}$, после чего оракул помечает сеанс открытым и формирует первую пересылку клиента для сообщения m_{b_i}, равного m_i при $b = 0$ и m_{1-i} в противном случае. При этом в качестве внутренней переменной сохраняется некоторое состояние $state_i$. На вход оракулу $User_1$ подается номер сеанса $i \in \{0, 1\}$ и первое сообщение подписывающего msg, оракул проверяет, что настоящий сеанс является открытым, после чего формирует вторую пересылку клиента, используя значение $state_i$. Оракул $User_2$ принимает на вход номер сеанса и второе сообщение подписывающего, помечает сеанс как завершенный и формирует подпись. Для контроля правильного порядка запросов в i-й сеансе определяются возможные состояния сеанса: $\mathsf{init}, \mathsf{open}_1, \mathsf{open}_2, \mathsf{closed}$.

Если оба сеанса завершены, оракул $User_2$ проверяет отсутствие ошибок в значениях подписи и выдает нарушителю пару подписей (σ_0, σ_1). Как уже отмечалось выше, тот факт, что нарушитель получает набор сформированных пар (сообщение, подпись) именно после завершения обоих взаимодействий, является принципиальным для формализации свойства неотслеживаемости. В случае, если указанное ограничение не выполняется, свойство неотслеживаемости не является осмысленным: одно завершенное взаимодействие соответствует одной сформированной паре (сообщение, подпись).

Если хотя бы один из сеансов завершился с ошибкой, то нарушителю выдается информация о том, какой это сеанс: $(\perp, \perp)$ – если оба, $(\perp, \theta)$ – если только сеанс с $i = 0$, $(\theta, \perp)$ – если только сеанс с $i = 1$. Подобная обработка ошибок позволяет с помощью введенной модели учитывать атаки, когда нарушитель пытается определить какую-либо информацию о сообщении по факту возникновения ошибки на стороне клиента. Другими словами, обеспечение стойкости в этом случае означает, что нарушитель не сможет вынудить клиента прервать протокол подписи из-за определенного свойства сообщения, которое раскрыло бы некоторую информацию о сообщении нарушителю. Например, пусть схема подписи вслепую предполагает, что клиент проверяет равенство первого бита сообщения, полученного от сервера, и первого бита подписываемого сообщения и завершает работу с ошибкой в случае его невыполнения. В этом случае, узнав, что именно первый сеанс завершился с ошибкой, нарушитель получает дополнительную информацию о сообщении, подписываемом в первом сеансе, — первый бит этого сообщения.

Однако корректная формализация свойства неотслеживаемости в случае ошибок влечет за собой модельное ограничение: если только один из сеансов завершился с ошибкой, нарушитель не получает пару (сообщение, подпись), сформированную в рамках другого успешно

завершенного сеанса (получает только значение θ). В отсутствие подобного ограничения нарушитель может построить тривиальную атаку на любую схему подписи вслепую, подав на вход оракулу $User_2$ в одном из сеансов сообщение msg некорректного формата и установив соответствие между другим сеансом и единственной корректно сформированной парой (сообщение, подпись). В этом случае возникает аналогичная описанной в предыдущем абзаце ситуация — фактически нарушитель получает одну пару (сообщение, подпись) в результате одного успешного взаимодействия с подписывающим. Как уже отмечалось, свойство неотслеживаемости в данном случае не является осмысленным.

В случае, если в протоколе формирования подписи вслепую первая пересылка от клиента к подписывающему пустая, оракул $User_0$ будет всегда возвращать пустые сообщения и не будет давать нарушителю никакой информации. Тогда его можно объединить с оракулом $User_1$, получив в итоге полностью эквивалентную модель.

Модель HS-Blind (Honest-Signer Blindness). Для удобства дальнейшего изложения определим формально более слабую модель HS-Blind, которая отличается от модели Blind в следующем:

- генерация ключей происходит честным образом при инициализации эксперимента;
- нарушитель не получает информацию о номере взаимодействия, завершившегося с ошибкой (если хотя бы одно из взаимодействий завершилось с ошибкой, то нарушителю возвращается значение $(\perp, \perp)$).

Далее в псевдокоде настоящие отличия выделены серым цветом. Для упрощения описания сразу определим настоящую модель для схем, в которых первая пересылка от клиента к серверу является пустой.

Определение 0.5. Для двухраундовой схемы подписи вслепую BS с пустой первой пересылкой от клиента к серверу

$$\mathsf{Adv}_{\mathsf{BS}}^{\text{HS-Blind}}(\mathcal{A}) = \Pr\Big[\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},1}(\mathcal{A}) \to 1\Big] - \Pr\Big[\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},0}(\mathcal{A}) \to 1\Big],$$

где эксперименты $\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},b}(\mathcal{A}),\ b \in \{0,1\}$, определяются следующим образом:

$\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},b}(\mathcal{A})$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $b_0 \leftarrow b$
3 : $b_1 \leftarrow 1 - b$
4 : $st \leftarrow \mathcal{A}^{Init}(\mathsf{sk}, \mathsf{pk})$
5 : $b' \leftarrow \mathcal{A}^{User_1, User_2}(st)$
6 : **return** b'

Oracle $Init(m_0, m_1)$

1 : $sess_0 \leftarrow \mathsf{init}$
2 : $sess_1 \leftarrow \mathsf{init}$
3 : store (m_0, m_1)
4 : **return** θ

Oracle $User_1(i, msg)$

1 : **if** $i \notin \{0, 1\} \vee sess_i \neq \mathsf{init}$: **return** $\perp$
2 : $sess_i \leftarrow \mathsf{open}$
3 : $(msg_i, state_i) \leftarrow \mathsf{BS.User}_1((\mathsf{pk}, m_{b_i}), msg)$
4 : **return** msg_i

Oracle $User_2(i, msg)$

1 : **if** $sess_i \neq \mathsf{open}$: **return** $\perp$
2 : $sess_i \leftarrow \mathsf{closed}$
3 : $\sigma_{b_i} \leftarrow \mathsf{BS.User}_2(state_i, msg)$
4 : **if** $sess_0 = sess_1 = \mathsf{closed}$:
5 : **if** $\sigma_{b_0} = \perp \vee \sigma_{b_1} = \perp$: **return** $(\perp, \perp)$
6 : **return** (σ_0, σ_1)
7 : **return** θ

Свойство неподделываемости

Неформально свойство неподделываемости означает, что клиент может сформировать корректную подпись только в результате успешного взаимодействия с подписывающим. Таким образом, нарушителем в данном случае, в отличие от свойства неотслеживаемости, является клиент.

Модель нарушителя

Нарушителю, выступающему в роли пользователя, предоставляется возможность получать от подписывающего корректные подписи для адаптивно выбираемых им сообщений, при этом предполагается, что подписывающий функционирует корректным образом. При этом для схем подписи вслепую в отличие от классических схем подписи рассматривают атаки двух типов:

- атака с последовательными сеансами: нарушитель может начинать выполнение нового сеанса протокола формирования подписи только после завершения предыдущего;
- атака с параллельными сеансами: нарушитель может начинать выполнение новых сеансов протокола формирования подписи до завершения предыдущих (имеет смысл для протоколов с двумя и более раундами).

В работе [40] подчеркивается важность предоставления нарушителю возможности провоцировать сбои: нарушителю предоставляется возможность не завершать сеансы и адаптивно провоцировать сбои, т.е. посылать некорректные сообщения, обработка которых на стороне подписывающего завершится с ошибкой. Поэтому при рассмотрении свойства неподделываемости нужно требовать, чтобы количество выданных нарушителем пар (сообщение, подпись) превышало именно количество успешно завершенных взаимодействий, при этом среди инициированных нарушителем взаимодействий часть могли остаться незавершенными.

Различные ключевые пары. В работе [17] рассматривается возможность нарушителя получать подписи для адаптивно выбираемых сообщений, формируемые с использованием различных ключевых пар (так называемая multi-signer модель). В качестве подделки нарушитель должен вернуть $(\ell+1)$ пару (сообщение, подпись) корректную относительно хотя бы одного произвольного открытого ключа в результате ℓ успешно завершенных взаимодействий для этого же ключа. При этом для других открытых ключей на количество взаимодействий может не накладываться ограничений (сильная атака) или может требоваться, чтобы количество выданных пар строго совпадало с числом успешных взаимодействий (слабая атака).

Кроме того, в работе [17] математически строго определена модель, позволяющая анализировать схему подписи вслепую относительно атак подмены ключа (key substitution attacks). Задачей нарушителя в настоящей атаке является подбор двух различных открытых ключей и пары (сообщение, подпись), которая является корректной относительно обоих ключей.

Модель угрозы

Свойство неподделываемости для схем подписи вслепую формализуется с помощью понятия «еще одной подделки» (one-more forgery). Задача нарушителя состоит в создании $(\ell+1)$ корректной пары (сообщение, подпись) в результате ℓ успешных взаимодействий с подписывающим.

Тривиальной атакой, доступной нарушителю, является дублирование пары (сообщение, подпись), сформированной в результате успешного взаимодействия с подписывающим. Поэтому, как и в стандартных моделях безопасности для схем подписи, на формируемые пары накладываются дополнительные ограничения:

- слабая угроза: все пары (сообщение, подпись) должны быть различными;
- сильная угроза: все сообщения должны быть различными.

Соответствующие данным угрозам свойства будем называть свойствами сильной неподделываемости для слабой угрозы и слабой неподделываемости для сильной угрозы. Если нарушитель успешно реализует любую из данных угроз, то среди выданных им пар (сообщение, подпись) заведомо есть хотя бы одна пара, полученная не в результате успешного взаимодействия с подписывающим.

Такие ограничения представляются более жесткими, чем ограничения для классических схем подписи. В стандартной для классической схемы подписи модели безопасности нарушитель имеет возможность накапливать произвольное количество подписей для одного и того же сообщения, после чего формировать подделку для другого (сильная угроза) или того же самого (слабая угроза) сообщения. Модели безопасности для схем подписи вслепую не всегда учитывают построение подделки таким образом как реализацию угрозы. Пусть рассматривается слабая угроза. Действительно, если нарушитель для схемы подписи вслепую может провести ℓ успешных взаимодействий для одного и того же сообщения и сформировать одну корректную подделку для другого сообщения, он не решит задачу успешно с точки зрения введенной модели, при этом на практике это может привести к ущербу.

Эта проблема подробно обсуждается в работе [76], в этой же работе предлагается расширенная модель безопасности «honest-user unforgeability» для схем подписи вслепую, предполагающая взаимодействие нарушителя не только с подписывающей стороной, но и с честным пользователем. Нарушитель навязывает честному пользователю получение значений подписи для произвольных, не обязательно различных сообщений $m_1, \ldots, m_n$ и самостоятельно выполняет ℓ успешных взаимодействий с подписывающим, после чего должен выдать $(\ell+1)$ корректную пару (сообщение, подпись) для различных сообщений $m_1^*, \ldots, m_{\ell+1}^*$ таких, что

$$\{m_1^*, \ldots, m_{\ell+1}^*\} \cap \{m_1, \ldots, m_n\} = \emptyset.$$

Отметим, что в случае $\ell = 0$ настоящая модель в точности совпадает с классической моделью UF-CMA для схемы подписи вслепую, которая рассматривает свойство неподделываемости относительно пассивного нарушителя.

Замечание 0.1. *Отметим, что для классических схем подписи при формализации свойства неподделываемости задачу нарушителя традиционно определяют как построение одной корректной пары (сообщение, подпись), которая при этом является нетривиальной, т.е. не была получена в результате штатного запроса к подписывающему. Именно из-за требования нетривиальности возникает отличие в формализации свойства неподделываемости для классических схем подписи и для схем подписи вслепую. В обоих случаях нетривиальность означает одно и то же – результирующая пара (сообщение, подпись) не была*

получена в результате честного взаимодействия с подписывающим. При этом в классическом случае подписывающий знает все честно сформированные пары и может определить, входит ли в их число предъявляемая в качестве подделки пара. Однако в случае схемы подписи вслепую подписывающий не знает пар (сообщение, подпись), которые были сформированы в результате взаимодействий с нарушителем, в силу обеспечения свойства неотслеживаемости. При этом подписывающий знает количество успешно проведенных взаимодействий по формированию подписи, поэтому определяет факт подделки исключительно по числу предъявленных различных пар. Отметим, что такой способ определять, сумел ли нарушитель построить нетривиальную подделку, обобщает классический способ. Действительно, если свойство неподделываемости сформулировать как необходимость вернуть $(\ell+1)$ *различных значений подписи, где* ℓ *— количество запросов к оракулу подписи, то получится эквивалентная классической модель безопасности.*

Математические модели безопасности

Определим формально модели SEQ-UF и UF, которые отличаются типом атаки, доступным нарушителю: с последовательными и параллельными сеансами соответственно. В обеих моделях будем рассматривать слабую угрозу — построение $(\ell+1)$ различных корректных пар (сообщение, подпись) в результате ℓ успешно завершенных взаимодействий, кроме того будем предполагать использование только одной ключевой пары.

Модель SEQ-UF (SEQuential UnForgeability). В настоящей модели предполагается, что нарушителю доступна атака с последовательными сеансами.

Определение 0.6. Для двухраундовой схемы подписи вслепую BS

$$\mathsf{Adv}_{\mathsf{BS}}^{\text{SEQ-UF}}(\mathcal{A}) = \Pr\Big[\mathbf{Exp}_{\mathsf{BS}}^{\text{SEQ-UF}}(\mathcal{A}) \to 1\Big],$$

где эксперимент $\mathbf{Exp}_{\mathsf{BS}}^{\text{SEQ-UF}}(\mathcal{A})$ определен на рисунке 1.

В настоящем определении за взаимодействие с подписывающим отвечают оракулы $Sign_1$ и $Sign_2$, которые реализуют работу подписывающего согласно протоколу на первом и втором раундах соответственно. Счетчик sid характеризует номер (идентификатор) взаимодействия (сеанса), который присваивается при запросе на первом раунде, счетчик ℓ — количество успешно завершенных взаимодействий (согласно определению схемы, взаимодействие закончилось успешно, если в результате работы $\mathsf{BS.Sign}$ вернулась единица). В множестве $\mathcal{I}_{\mathsf{fin}}$ хранятся номера завершенных сеансов.

$\mathbf{Exp}_{\mathsf{BS}}^{\text{SEQ-UF}}(\mathcal{A})$

1 : $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{\mathcal{U}} \mathsf{BS.KGen}(\,)$

2 : $\mathsf{sid}, \ell \leftarrow 0, \mathcal{I}_{\mathsf{fin}} \leftarrow \emptyset$

3 : $\{(m_k^*, \sigma_k^*)\}_{k=1}^{\ell+1} \leftarrow \mathcal{A}^{Sign_1, Sign_2}(\mathsf{pk})$

4 : **return** $(\forall k_1 \neq k_2 \in \{1, \ldots, \ell+1\} : (m_{k_1}^*, \sigma_{k_1}^*) \neq (m_{k_2}^*, \sigma_{k_2}^*) \wedge$

5 : $\quad \wedge \forall k \in \{1, \ldots, \ell+1\} : \mathsf{BS.Verify}(\mathsf{pk}, m_k^*, \sigma_k^*) = 1)$

Oracle $Sign_1(msg)$

1 : **if** $(\mathsf{sid} \notin \mathcal{I}_{\mathsf{fin}}) \wedge (\mathsf{sid} \neq 0)$: **return** $\perp$

2 : $\mathsf{sid} \leftarrow \mathsf{sid} + 1$

3 : $(msg', state_{\mathsf{sid}}) \leftarrow \mathsf{BS.Sign}_1(\mathsf{sk}, \mathsf{pk}, msg)$

4 : **return** (sid, msg')

Oracle $Sign_2(msg)$

1 : **if** $\mathsf{sid} \in \mathcal{I}_{\mathsf{fin}}$: **return** $\perp$

2 : $(msg', b) \leftarrow \mathsf{BS.Sign}_2(state_{\mathsf{sid}}, msg)$

3 : **if** $b = 1 : \ell \leftarrow \ell + 1$

4 : $\mathcal{I}_{\mathsf{fin}} \leftarrow \mathcal{I}_{\mathsf{fin}} \cup \{\mathsf{sid}\}$

5 : **return** msg'

Рис. 1. Эксперимент $\mathbf{Exp}_{\mathsf{BS}}^{\text{SEQ-UF}}(\mathcal{A})$.

Оракул $Sign_1$ принимает на вход сообщение в первом раунде, проверяет, что текущий сеанс с номером sid был завершен (за исключением первого запроса к оракулу, когда $\mathsf{sid} = 0$), увеличивает на единицу счетчик sid и реализует работу подписывающего на первом раунде, сохраняя в качестве внутренней переменной некоторое состояние $state_{\mathsf{sid}}$. Оракул $Sign_2$ принимает на вход сообщение пользователя, реализует работу подписывающего на втором раунде в текущем сеансе с номером sid, используя сохраненное ранее состояние $state_{\mathsf{sid}}$, и добавляет номер sid в множество $\mathcal{I}_{\mathsf{fin}}$. Таким образом, нарушитель не может выполнить второй раунд протокола дважды для одного и того же первого раунда. Если взаимодействие завершилось успешно, оракул $Sign_2$ инкрементирует счетчик успешно завершенных взаимодействий ℓ.

Таким образом, в настоящей модели накладывается следующее ограничение на порядок вызовов оракулов $Sign_1$ и $Sign_2$: эти вызовы должны осуществляться по очереди, причем каждая пара вызовов отвечает за сеанс с текущим номером sid. Это моделирует атаку с последовательными сеансами. Серым цветом в псевдокоде отражено условие, позволяющее контролировать отсутствие параллельных сеансов. Заметим, что возможность не завершать некоторые сеансы моделируется с помощью отправки на вход оракулу $Sign_2$ в этом сеансе заведомо некорректного сообщения msg (например, пустой строки).

Количество корректных пар (сообщение, подпись), возвращаемых нарушителем, на едини-

цу больше количества успешно завершенных сеансов ℓ, при этом все эти пары должны быть различны. Это условие гарантирует, что среди возвращаемых пар хотя бы одна действительно является подделкой, т.е. сформирована нарушителем без успешно завершенного взаимодействия с подписывающим. При этом, на количество инициированных взаимодействий sid не накладывается условий, то есть нарушитель может провоцировать неограниченное число сбоев.

Модель UF (UnForgeability). В настоящей модели предполагается, что нарушителю доступна атака с параллельными сеансами.

Определение 0.7. Для двухраундовой схемы подписи вслепую BS

$$\mathsf{Adv}^{\mathrm{UF}}_{\mathsf{BS}}(\mathcal{A}) = \Pr\big[\mathbf{Exp}^{\mathrm{UF}}_{\mathsf{BS}}(\mathcal{A}) \to 1\big],$$

где эксперимент $\mathbf{Exp}^{\mathrm{UF}}_{\mathsf{BS}}(\mathcal{A})$ определен на рисунке 2.

$\mathbf{Exp}^{\mathrm{UF}}_{\mathsf{BS}}(\mathcal{A})$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}(\,)$
2 : $\mathsf{sid}, \ell \leftarrow 0, \mathcal{I}_{\mathsf{fin}} \leftarrow \emptyset$
3 : $\{(m^*_k, \sigma^*_k)\}_{k=1}^{\ell+1} \leftarrow \mathcal{A}^{Sign_1, Sign_2}(\mathsf{pk})$
4 : **return** $(\forall k_1 \neq k_2 \in \{1, \ldots, \ell+1\} : (m^*_{k_1}, \sigma^*_{k_1}) \neq (m^*_{k_2}, \sigma^*_{k_2})$
5 : $\quad \wedge \forall k \in \{1, \ldots, \ell+1\} : \mathsf{BS.Verify}(\mathsf{pk}, m^*_k, \sigma^*_k) = 1)$

Oracle $Sign_1(msg)$

1 : $\mathsf{sid} \leftarrow \mathsf{sid} + 1$
2 : $(msg', state_{\mathsf{sid}}) \leftarrow \mathsf{BS.Sign}_1(\mathsf{sk}, \mathsf{pk}, msg)$
3 : **return** (sid, msg')

Oracle $Sign_2(j, msg)$

1 : **if** $j \notin [\mathsf{sid}] \setminus \mathcal{I}_{\mathsf{fin}}$: **return** $\perp$
2 : $(msg', b) \leftarrow \mathsf{BS.Sign}_2(state_j, msg)$
3 : **if** $b = 1 : \ell \leftarrow \ell + 1$
4 : $\mathcal{I}_{\mathsf{fin}} \leftarrow \mathcal{I}_{\mathsf{fin}} \cup \{j\}$
5 : **return** msg'

Рис. 2. Эксперимент $\mathbf{Exp}^{\mathrm{UF}}_{\mathsf{BS}}(\mathcal{A})$.

Настоящая модель аналогична модели SEQ-UF за исключением порядка работы оракулов $Sign_1$ и $Sign_2$. Оракул $Sign_1$ не осуществляет никаких проверок и открывает новый сеанс, вне зависимости от того, был ли завершен предыдущий. Оракул $Sign_2$ дополнительно принимает на вход идентификатор сеанса j и проверяет, что этот сеанс был открыт, но не завершен. Таким образом, в настоящей модели на порядок запросов к оракулам $Sign_1$ и $Sign_2$

не накладывается никаких ограничений за исключением того, что делать запрос к $Sign_2$ с идентификатором сеанса j можно только в том случае, когда был выполнен соответствующий запрос к $Sign_1$.

Модель UF, соответствующую слабой угрозе, в которой в строке 4 вместо условия $(m^*_{k_1}, \sigma^*_{k_1}) \neq (m^*_{k_2}, \sigma^*_{k_2})$ проверяется условие $m^*_{k_1} \neq m^*_{k_2}$, будем обозначать через wUF.

Глава 1

Анализ безопасности схемы подписи вслепую Шаума-Педерсена в расширенных моделях безопасности

В настоящей главе проводится анализ схемы Шаума-Педерсена с точки зрения обеспечения свойства неподделываемости при наличии у нарушителя возможности открывать параллельные сеансы протокола формирования подписи.

В разделе 1.1 приводится описание схемы Шаума-Педерсена для группы точек эллиптической кривой. Заметим, что оригинальное описание схемы Шаума-Педерсена [32] представлено для мультипликативной группы конечного поля, и перевод схемы на случай использования группы точек эллиптической кривой, вообще говоря, может быть осуществлен различными способами. Выбор конкретного способа перевода обусловлен стремлением получить нижнюю оценку стойкости схемы, см. подробнее раздел 1.2.

В разделе 1.2 доказывается, что схема Шаума-Педерсена не обеспечивает свойство сильной неподделываемости в модели UF, т.е. позволяет строить подделки для «старого» сообщения, которое было подписано легитимным образом в результате успешного выполнения сеанса протокола формирования подписи. Строится модификация ROS атаки, аналогичная ROS атаке на схему Брандса. При этом конструкция схемы не позволяет расширить данную атаку на случай построения подделки для «нового» сообщения, что позволяет говорить о том, что схема Шаума-Педерсена потенциально обеспечивает свойство слабой неподделываемости в модели wUF (задача нарушителя — построение подделки для нового сообщения).

Далее в разделе 1.3 проводится математически строгий анализ свойства слабой неподделываемости в модели с параллельными сеансами. Заметим, что формальное обоснование свойства неподделываемости для схем подписи вслепую сопряжено с существенными трудностями. В литературе представлен ряд работ, которые показывают невозможность обоснования стойкости схемы подписи вслепую Шнорра в расширенных моделях безопасности [39, 67], а именно, без идеализаций криптографических примитивов (модель со случайным оракулом [22]) и ограничений множества рассматриваемых нарушителей (модели с генерической [63] или алгебраической [41] группой). Более того, в работе [18] показано, что для схем подписи вслепую Шнорра [68], Окамото-Шнорра [69] и Брандса [26] невозможно построить сведение с использованием всех известных техник доказательств для таких схем (например, «random oracle replay» (перезапуск случайного оракула) и «forking lemma» (лемма о разветвлении)) на основе предположений о сложности базовых задач даже в модели со

случайным оракулом. Существующие сведения верны только в моделях с генерической или алгебраической группой со случайным оракулом, которые являются упрощением расширенных моделей. На сегодняшний день не было предложено новых техник построения сведений для схем подписи вслепую, решающих задачу получения оценки стойкости в расширенных моделях безопасности.

Удалось построить сведение в модели wUF с алгебраической группой и случайным оракулом, которое демонстрирует, что достаточным условием стойкости схемы в модели wUF является сложность решения двух задач: задачи REPR и SOMDL. Задача REPR является модификацией задачи «Representation», определенной в работе [26], ее сложность также является необходимым условием стойкости схемы Шаума-Педерсена. Задача SOMDL является новой задачей, определенной для группы точек элиптической кривой. Доказывается, что настоящая задача не сложнее задачи OMDL (One-More Discrete Logarithm, введена в [21]) и задачи SDL (определена в [19] как задача q-dlog). Для задачи SDL доказывается также, что ее сложность является необходимым условием стойкости схемы Шаума-Педерсена. Таким образом, успешно выделены базовые задачи в группе точек эллиптических кривой, сложность которых лежит в основе стойкости схемы Шаума-Педерсена в модели wUF.

1.1. Описание схемы

Приведем описание алгоритмов, задающих работу схемы подписи вслепую Шаума-Педерсена. Далее будем называть эту схему CP-BS.

Оригинальное описание схемы Шаума-Педерсена [32] представлено для мультипликативной группы конечного поля, при этом подписываемое сообщение представляет собой элемент группы. В настоящей главе приведено описание для группы точек эллиптической кривой, при этом для перевода сообщения в элемент группы, т.е. точку кривой, предлагается использовать функцию хэширования $\mathcal{H}$ в группу точек эллиптической кривой. Заметим, что в литературе известны подходы к построению таких функций, см. например [35].

Алгоритм генерации ключей задается следующим образом.

$$\frac{\mathsf{CP\text{-}BS.KGen}(\ )}{d \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*}$$

$$Q \leftarrow dP$$

return (d, Q)

Протокол формирования подписи состоит из двух раундов, инициатором взаимодействия является клиент. Клиент вычисляет элемент группы $M' = \mathcal{H}(m)$, маскирует это значение, вы-

числяя $M = \alpha^{-1}M'$ для случайно выбранного значения $\alpha \in \mathbb{Z}_q^*$, и посылает точку M серверу. Сервер вычисляет значение $Z = dM$, после чего клиент и сервер выполняют интерактивный протокол доказательства равенства дискретных логарифмов Шаума-Педерсена [32] для следующих значений:

$$\mathrm{DLog}_P Q = \mathrm{DLog}_M Z,$$

при этом для вычисления значения c («challenge» в протоколе доказательства Шаума-Педерсена [32]) клиент маскирует все значения, полученные от сервера. Значение $Z' = \alpha Z$ и сформированное доказательство (в маскированном виде) составляют значение подписи.

Алгоритм проверки подписи для сообщения m представляет собой проверку доказательства равенства

$$\mathrm{DLog}_P Q = \mathrm{DLog}_{M'} Z',$$

где $M' = \mathcal{H}(m)$. Заметим, что $\mathrm{DLog}_{M'} Z' = \mathrm{DLog}_{\alpha M}(\alpha Z) = \mathrm{DLog}_M Z$.

Алгоритм проверки подписи и протокол формирования подписи формально определены на рисунках 1.1 и 1.2 соответственно.

В отличие от оригинального описания схемы в представленном описании значение подписи определяется набором (s', c', Z'), а не (s', A', B', Z'). Заметим, что с точки зрения стойкости схемы эти способы задания подписи являются эквивалентными, поскольку по первому набору можно однозначно восстановить второй набор, и наоборот. Вместе с тем, подпись (s', c', Z') является более короткой, а потому представляет больший интерес с практической точки зрения.

$\mathsf{CP\text{-}BS.Verify}(Q, m, (s', c', Z'))$

if $Z' \notin \mathbb{G}$: **return** 0

$M' \leftarrow \mathcal{H}(m)$

if $c' = H(M' \| Z' \| (s'P - c'Q) \| (s'M' - c'Z'))$: **return** 1

else : **return** 0

Рис. 1.1. Алгоритм проверки подписи в схеме CP-BS

1.2. Анализ безопасности относительно свойства сильной неподделываемости

В работе [24] построена атака на свойство неподделываемости для схемы подписи вслепую Брандса [26], применимая в модели с параллельными сеансами и позволяющая сформиро-

Sign (d, Q)		User (Q, m)
		$\alpha \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
		$M' \leftarrow \mathcal{H}(m)$
	$\xleftarrow{\quad M \quad}$	$M \leftarrow \alpha^{-1} M'$
if $M \notin \mathbb{G}$: **return** 0		
$k \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$		
$Z \leftarrow dM$		
$A \leftarrow kP$		
$B \leftarrow kM$	$\xrightarrow{\quad Z, A, B \quad}$	
		if $Z, A, B \notin \mathbb{G}$: **return** $\perp$
		$u, v \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
		$A' \leftarrow uA + vP$
		$B' \leftarrow u\alpha B + vM'$
		$Z' \leftarrow \alpha Z$
		$c' \leftarrow H(M' \| Z' \| A' \| B')$
	$\xleftarrow{\quad c \quad}$	$c \leftarrow c' u^{-1}$
if $c = 0$: **return** 0		
$s \leftarrow k + cd$	$\xrightarrow{\quad s \quad}$	
		if $sP \neq A + cQ$: **return** $\perp$
		if $sM \neq B + cZ$: **return** $\perp$
		$s' \leftarrow us + v$
		$\sigma \leftarrow (s', c', Z')$
return 1		**return** σ

Рис. 1.2. Протокол формирования подписи в схеме CP-BS

вать $(\ell + 1)$ пару (сообщение, подпись) в результате $\ell \geqslant \lceil \log q \rceil$ успешных взаимодействий с подписывающим. При этом данная атака применима только при некоторых ограничениях на множество сообщений, входящих в состав подделок — для их формирования должен использоваться один и тот же «номер аккаунта» пользователя.

Оказалось, что аналогичная атака применима и к схеме Шаума-Педерсена. При этом для данной схемы атака позволяет построить $(\ell + 1)$ подпись для одного и того же сообщения, т.е. реализовать слабую угрозу. Опишем данную атаку, а также условия ее применимости в случае формирования $(\ell+1)$ подписи для различных сообщений. Далее положим $\ell = \lceil \log q \rceil$.

Теорема 1.2.1. *Существует полиномиальный нарушитель $\mathcal{A}$ для схемы* CP-BS *в модели* UF, *открывающий $\ell = \lceil \log q \rceil$ параллельных сеансов протокола формирования подписи, такой что*

$$\mathsf{Adv}^{\mathrm{UF}}_{\mathsf{CP\text{-}BS}}(\mathcal{A}) \geqslant 1 - \frac{\ell}{q}.$$

Доказательство. Построим модифицированную версию ROS атаки, которая позволяет пользователю сформировать $(\ell+1)$ валидную пару (сообщение, подпись) в результате ℓ успешных взаимодействий с подписывающим, где $\ell = \lceil \log q \rceil$. В результате осуществления этой атаки все подписи будут сформированы для одного и того же сообщения.

Пусть нарушитель $\mathcal{A}$ выполняет следующие шаги.

1. Выбирает сообщение $m \in \{0,1\}^*$, для которого будет построена $(\ell + 1)$ подпись, пусть $M' = M = \mathcal{H}(m)$.

2. Открывает ℓ параллельных сеансов, отправляя подписывающему ℓ одинаковых запросов с точкой M.

3. Получает в ответ ℓ наборов $(Z, A_i, B_i), 0 \leqslant i \leqslant \ell - 1$, удовлетворяющих условиям:

$$Z = dM,\ A_i = k_i P,\ B_i = k_i M,$$

   где k_i выбирается подписывающим случайно и равновероятно для каждого открытого сеанса.

4. Выбирает $u_i^0, u_i^1, 0 \leqslant i \leqslant \ell - 1$, таким образом, чтобы $c_{i0} \neq c_{i1}$, где:

$$c'_{i0} = H(M \| Z \| u_i^0 A_i \| u_i^0 B_i),\ c'_{i1} = H(M \| Z \| u_i^1 A_i \| u_i^1 B_i),$$

$$c_{i0} = (u_i^0)^{-1} c'_{i0},\ c_{i1} = (u_i^1)^{-1} c'_{i1}.$$

5. Определяет $\rho_0, \rho_1, \ldots, \rho_\ell$ как коэффициенты перед x_i в выражении $\sum\limits_{i=0}^{\ell-1} 2^i \dfrac{x_i - c_{i0}}{c_{i1} - c_{i0}} = \sum\limits_{i=0}^{\ell-1} \rho_i x_i + \rho_\ell$.

6. Полагает $A_\ell = \sum\limits_{i=0}^{\ell-1} \rho_i A_i - \rho_\ell Q;\ B_\ell = \sum\limits_{i=0}^{\ell-1} \rho_i B_i - \rho_\ell Z$.

7. Вычисляет $c'_\ell = H(M\|Z\|A_\ell\|B_\ell)$.
8. Определяет $b_0, \ldots, b_{\ell-1}$ как $c'_\ell = \sum_{i=0}^{\ell-1} 2^i b_i$. Это возможно, т.к. $\ell = \lceil \log q \rceil$.
9. Полагает $c_i = c_{ib_i}, c'_i = c'_{ib_i}, u_i = u_i^{b_i}, 0 \leqslant i \leqslant \ell - 1$, таким образом, $c'_\ell = \sum_{i=0}^{\ell-1} \rho_i c_i + \rho_\ell$.
10. Отправляет подписывающему значения $c_0, \ldots, c_{\ell-1}$ в соответствующих открытых сеансах.
11. Получает в ответ от подписывающего значения $s_0, \ldots, s_{\ell-1}$ такие, что:

$$s_i P = A_i + c_i Q,$$

$$s_i M = B_i + c_i Z.$$

12. Полагает $s'_i = u_i s_i, , 0 \leqslant i \leqslant \ell - 1$.
13. Полагает $s'_\ell = \sum_{i=0}^{\ell-1} \rho_i s_i$.
14. Выдает $\{m, (s'_i, c'_i, Z)\}_{i=0}^{\ell}$.

Действительно, для $0 \leqslant i \leqslant \ell - 1$ подпись (s'_i, c'_i, Z) будет корректной для сообщения m, так как

$$s'_i P - c'_i Q = u_i s_i P - u_i c_i Q = u_i(s_i P - c_i Q) = u_i A_i;$$

$$s'_i M - c'_i Z = u_i s_i M - u_i c_i Z = u_i(s_i M - c_i Z) = u_i B_i;$$

а по построению $c'_i = H(M\|Z\|u_i A_i\|u_i B_i)$.

Для $i = \ell$ подпись (s'_ℓ, c'_ℓ, Z) будет корректной для сообщения m, так как

$$s'_\ell P - c'_\ell Q = \sum_{i=0}^{\ell-1} \rho_i s_i P - \sum_{i=0}^{\ell-1} \rho_i c_i Q - \rho_\ell Q = \sum_{i=0}^{\ell-1} \rho_i (s_i P - c_i Q) - \rho_\ell Q = \sum_{i=0}^{\ell-1} \rho_i A_i - \rho_\ell Q = A_\ell;$$

$$s'_\ell M - c'_\ell Z = \sum_{i=0}^{\ell-1} \rho_i s_i M - \sum_{i=0}^{\ell-1} \rho_i c_i Z - \rho_\ell Z = \sum_{i=0}^{\ell-1} \rho_i (s_i M - c_i Z) - \rho_\ell Z = \sum_{i=0}^{\ell-1} \rho_i B_i - \rho_\ell Z = B_\ell;$$

а по построению $c'_\ell = H(M\|Z\|A_\ell\|B_\ell)$.

Вероятность успешного применения данного метода равна вероятности успешного осуществления шага 4 (все остальные шаги выполнимы с вероятностью 1). В предположении, что хэш-функция ведет себя как случайная функция, вероятность совпадения значений c_{i0} и c_{i1} для конкретного значения $i \in \{0, \ldots, \ell - 1\}$ равна $\frac{1}{q}$. Тогда вероятность совпадения этих значений хотя бы для одного значения $i \in \{0, \ldots, \ell - 1\}$ равна $\frac{\ell}{q}$. Отсюда получаем, что вероятность успешного применения метода составляет $1 - \frac{\ell}{q}$. □

Таким образом, схема Шаума-Педерсена позволяет построить $(\ell+1)$ корректную подпись для одного и того же сообщения в результате успешного завершения ℓ сеансов протокола формирования подписи.

Модификация атаки на случай разных сообщений. Рассмотрим структурные особенности схемы Шаума-Педерсена, не позволяющие расширить данную атаку на случай формирования подписей для различных сообщений. Ключевым отличием схемы Шаума-Педерсена от схемы Шнорра является наличие первой пересылки $M = \alpha^{-1}M'$ от пользователя к подписывающему, содержимое которой существенно зависит от сообщения, и добавление в подпись элемента $Z' = dM'$. Тогда для построения подделки для некоторого нового сообщения m, которому соответствует точка $M' = \mathcal{H}(m)$, нарушителю необходимо вычислить значение $Z' = dM'$.

Заметим однако, что если дискретный логарифм точки M' по основанию P известен нарушителю (пусть $\mathrm{DLog}_P(M') = \beta$), то соответствующая точка Z' может быть вычислена как $\beta Q = d(\beta P) = dM'$. В этом случае описанная выше атака может быть расширена на случай подписания различных сообщений в каждом сеансе (а значит, различных точек $M'_i = M_i$ и Z_i в каждом сеансе). Все шаги атаки выполняются аналогично за исключением алгоритма вычисления точек Z_ℓ и B_ℓ. Точка Z_ℓ, как уже было сказано, вычисляется как βQ, а точка B_ℓ полагается равной βA_ℓ. Несложно проверить, что подпись $(s'_\ell, c'_\ell, Z_\ell)$ будет корректной подписью для сообщения m, соответствующего точке $M' = \beta P$.

Таким образом, для безопасности схемы критично формирование точки M' таким образом, чтобы ее дискретный логарифм был неизвестен нарушителю. Именно поэтому для формирования M' в схеме Шаума-Педерсена предлагается использовать функцию хэширования в кривую.

Замечание 1.2.1. *В оригинальной схеме подписи вслепую Брандса [26], а также в модификации данной схемы для группы точек эллиптической кривой, используемой в системе U-Prove [66], элемент группы M' формируется как линейная комбинация элементов с взаимно неизвестным дискретным логарифмом, таким образом дискретный логарифм M' неизвестен нарушителю. Однако в вариации данной схемы, определенной в работе [18], элемент M' формируется как αP, поэтому по построению пользователю всегда известен дискретный логарифм M'. Таким образом, для этой вариации схемы свойство неподделываемости не обеспечивается даже в слабом смысле.*

Если дискретный логарифм точки M' по основанию P неизвестен, то описанная в настоящем разделе атака не применима за счет сложности построения точек Z_ℓ, B_ℓ, для которых

будет выполнено условие:

$$B_\ell = \sum_{i=0}^{\ell-1} \rho_i B_i = \sum_{i=0}^{\ell-1} \rho_i (s_i M_i - c_i Z_i) = s_\ell M_\ell - c_\ell Z_\ell.$$

1.3. Анализ безопасности относительно свойства слабой неподделываемости

В настоящем разделе получим оценку стойкости схемы Шаума-Педерсена относительно свойства слабой неподделываемости в модели с параллельными сеансами с помощью метода сведений.

Сведение удалось построить в модели wUF при некоторых дополнительных ограничениях на возможности нарушителя: в модели с алгебраической группой и случайным оракулом. Рассмотрим подробнее, какие ограничения налагают данные модели.

Модель со случайным оракулом. Модель со случайным оракулом была введена в [22] и подразумевает, что на этапе инициализации экспериментатор выбирает случайную функцию, после чего предоставляет нарушителю доступ к ней через так называемый случайный оракул. Нарушитель может получать значения случайной функции на произвольном входе α, подавая запрос вида α к случайному оракулу, при этом он не может вычислять значения случайной функции самостоятельно. При обосновании свойства неподделываемости схемы Шаума-Педерсена будем предполагать, что хэш-функции H и $\mathcal{H}$ моделируются как случайные оракулы. Анализ в данной модели справедливо интерпретировать следующим образом: при получении оценки стойкости не рассматриваются методы криптоанализа, основанные на структурных свойствах конкретных функций H и $\mathcal{H}$, определяющих связь между областью определения и областью значений данных функций.

Модель с алгебраической группой. Настоящая модель была предложена в работе [41]. В модели с алгебраической группой на алгоритм нарушителя накладывается следующее требование: для любого элемента группы, который появляется на выходе алгоритма нарушителя в процессе его работы, нарушитель должен предоставить коэффициенты разложения данного элемента в линейную комбинацию всех элементов, пришедших ему на вход к данному моменту. То есть, если нарушитель возвращает элемент группы Z и на данный момент он получил элементы $X_1, \ldots, X_n$, то вместе с Z он передает набор коэффициентов $z = (z_1, \ldots, z_n)$, такие что $Z = \sum_{i=1}^{n} z_i X_i$. Анализ в данной модели справедливо интерпретировать следующим образом: при получении оценки стойкости не рассматриваются методы криптоанализа,

использующие структурные особенности конкретной группы для формирования новых элементов группы.

1.3.1. Базовые задачи

Сведение стойкости схемы CP-BS построено к следующим базовым задачам: задаче SOMDL и задаче REPR. Определим их формально.

Задача SOMDL (Strong One-More Discrete Logarithm)

Параметрами настоящей задачи являются $k, \ell \in \mathbb{N}$.

Определение 1.3.1. Для нарушителя $\mathcal{A}$ и группы $\mathbb{G}$, параметров ℓ, k:

$$\mathsf{Adv}^{\mathrm{SOMDL}}_{\mathbb{G},k,\ell}(\mathcal{A}) = \Pr\big[\mathbf{Exp}^{\mathrm{SOMDL}}_{\mathbb{G},k,\ell}(\mathcal{A}) \to 1\big],$$

где эксперимент $\mathbf{Exp}^{\mathrm{SOMDL}}_{\mathbb{G},k,\ell}(\mathcal{A})$ определен на рисунке 1.3.

$\underline{\mathbf{Exp}^{\mathrm{SOMDL}}_{\mathbb{G},k,\ell}(\mathcal{A})}$

$x_1, \ldots, x_{\ell+1} \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$

$ctr_1, ctr_2 \leftarrow 0$

$(x'_1, \ldots, x'_{\ell+1}) \leftarrow \mathcal{A}^{O_1, O_2}(x_1 P, \ldots, x_{\ell+1} P)$

return $(x_1 = x'_1) \wedge \ldots \wedge (x_{\ell+1} = x'_{\ell+1})$

$\underline{\text{Oracle } O_1(i, Y)}$

if $ctr_1 > k$: **return** $\perp$

if $i \notin \{1, \ldots, \ell+1\}$: **return** $\perp$

if $Y \notin \mathbb{G}$: **return** $\perp$

$ctr_1 \leftarrow ctr_1 + 1$

return $x_i Y$

$\underline{\text{Oracle } O_2(Y)}$

if $ctr_2 > \ell$: **return** $\perp$

if $Y \notin \mathbb{G}$: **return** $\perp$

$ctr_2 \leftarrow ctr_2 + 1$

return $\mathrm{DLog}_P(Y)$

Рис. 1.3. Эксперимент $\mathbf{Exp}^{\mathrm{SOMDL}}_{\mathbb{G},k,\ell}(\mathcal{A})$.

Нарушитель получает на вход набор точек $x_1 P, \ldots, x_{\ell+1} P$, его задача — найти значения $x_1, \ldots, x_{\ell+1}$. Нарушитель имеет доступ к двум оракулам O_1 и O_2, он может делать не более k запросов к первому оракулу и не более ℓ — ко второму. Настоящие ограничения контролируются с помощью счетчиков ctr_1, ctr_2.

Оракул O_1 в ответ на запрос нарушителя вида $(i, Y), i \in \{1, \ldots, \ell+1\}, Y \in \mathbb{G}$, возвращает значение $x_i Y$. Оракул O_2 в ответ на запрос $Y \in \mathbb{G}$ возвращает дискретный логарифм этой

точки по основанию P. Запросы к оракулам O_1 и O_2 могут быть выполнены в произвольном порядке.

Для оценки трудоемкости решения настоящей задачи рассмотрим ее соотношение с другими известными в литературе задачами.

Соотношение с другими задачами. В литературе, содержащей определения теоретико-сложностных задач для конечных групп, были ранее введены две «близкие» задачи к задаче SOMDL: задача SDL и OMDL. Определим их формально.

Задача SDL**.** Задача SDL (Strong Discrete Logarithm) определена в работе [19] как задача q-dlog и является модификацией задачи SDH (Strong Diffie-Hellman), предложенной в работе [25], ее параметром является значение $s \in \mathbb{N}$.

Определение 1.3.2. Для нарушителя $\mathcal{A}$ и группы $\mathbb{G}$, параметра s:

$$\mathsf{Adv}^{\mathrm{SDL}}_{\mathbb{G},s}(\mathcal{A}) = \Pr\big[\mathbf{Exp}^{\mathrm{SDL}}_{\mathbb{G},s}(\mathcal{A}) \to 1\big],$$

где эксперимент $\mathbf{Exp}^{\mathrm{SDL}}_{\mathbb{G},s}(\mathcal{A})$ определяется следующим образом:

$\mathbf{Exp}^{\mathrm{SDL}}_{\mathbb{G},s}(\mathcal{A})$

$x \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$

$x' \leftarrow \mathcal{A}(xP, \ldots, x^s P)$

return $(x = x')$

Нарушитель получает на вход набор точек $xP, \ldots, x^s P$, его задача — найти значение x.

Покажем, что доступ к оракулу O_1 в задаче SOMDL может предоставлять нарушителю такие же возможности, как в задаче SDL. Действительно, подавая на вход оракулу O_1 сначала запрос $(1, x_1 P)$, а потом запросы вида $(1, Y)$, где Y — ответ оракула O_1 на предыдущий запрос, нарушитель может накопить значения $x_1 P, x_1^2 P, \ldots, x_1^{k+1} P$, что аналогично получению на вход таких значений в задаче SDL с параметром $s = k+1$. Отсюда очевидным образом следует следующее утверждение.

Утверждение 1.3.1. *Для любого нарушителя $\mathcal{A}$, решающего задачу* SDL *с параметром $k+1$, существует нарушитель $\mathcal{B}$ с такими же вычислительными ресурсами, решающий задачу* SOMDL *с параметрами (k, ℓ) для произвольного $\ell \in \mathbb{N}$, такой что*

$$\mathsf{Adv}^{\mathrm{SDL}}_{\mathbb{G},k+1}(\mathcal{A}) \leqslant \mathsf{Adv}^{\mathrm{SOMDL}}_{\mathbb{G},k,\ell}(\mathcal{B}).$$

Таким образом, задача SOMDL с параметрами (k, ℓ) не сложнее задачи SDL с параметром $k+1$.

Задача OMDL. Задача OMDL (One-More Discrete Logarithm) предложена в работе [21], ее параметром является значение $\ell \in \mathbb{N}$.

Определение 1.3.3. Для нарушителя $\mathcal{A}$ и группы $\mathbb{G}$, параметра ℓ:

$$\mathsf{Adv}^{\mathrm{OMDL}}_{\mathbb{G},\ell}(\mathcal{A}) = \Pr\big[\mathbf{Exp}^{\mathrm{OMDL}}_{\mathbb{G},\ell}(\mathcal{A}) \to 1\big],$$

где эксперимент $\mathbf{Exp}^{\mathrm{OMDL}}_{\mathbb{G},\ell}(\mathcal{A})$ определяется следующим образом:

$\underline{\mathbf{Exp}^{\mathrm{OMDL}}_{\mathbb{G},\ell}(\mathcal{A})}$	$\underline{\text{Oracle } DLog(Y)}$
$x_1, \ldots, x_{\ell+1} \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$	**if** $ctr > \ell$: **return** $\perp$
$ctr \leftarrow 0$	**if** $Y \notin \mathbb{G}$: **return** $\perp$
$(x'_1, \ldots, x'_{\ell+1}) \leftarrow \mathcal{A}^{Dlog}(x_1 P, \ldots, x_{\ell+1} P)$	$ctr \leftarrow ctr + 1$
return $(x_1 = x'_1) \wedge \ldots \wedge (x_{\ell+1} = x'_{\ell+1})$	**return** $\mathrm{DLog}_P(Y)$

Нарушитель получает на вход набор точек $x_1 P, \ldots, x_{\ell+1} P$, его задача — найти значения $x_1, \ldots, x_{\ell+1}$. Нарушитель имеет доступ к оракулу $DLog$, который в ответ на запрос $Y \in \mathbb{G}$ возвращает дискретный логарифм этой точки по основанию P. Он может делать не более ℓ запросов к этому оракулу, настоящее ограничение контролируется с помощью счетчика ctr. Заметим, что оракул в задаче OMDL в точности совпадает с оракулом O_2 в задаче SOMDL. Отсюда очевидным образом следует следующее утверждение.

Утверждение 1.3.2. *Для любого нарушителя $\mathcal{A}$, решающего задачу* OMDL *с параметром ℓ, существует нарушитель $\mathcal{B}$ с такими же вычислительными ресурсами, решающий задачу* SOMDL *с параметрами (k, ℓ) для произвольного $k \in \mathbb{N}$, такой что*

$$\mathsf{Adv}^{\mathrm{OMDL}}_{\mathbb{G},\ell}(\mathcal{A}) \leqslant \mathsf{Adv}^{\mathrm{SOMDL}}_{\mathbb{G},k,\ell}(\mathcal{B}).$$

Таким образом, задача SOMDL с параметрами (k, ℓ) не сложнее задачи OMDL с параметром ℓ.

Утверждения 1.3.1 и 1.3.2 показывают, что из сложности задачи SOMDL следует сложность известных задач SDL и OMDL. Однако на данный момент не удалось получить результатов, что сложность этих базовых задач является достаточным условием сложности SOMDL. Таким образом, задача SOMDL — это новая задача, требующая отдельных исследований.

Задача REPR

Настоящая задача является модификацией задачи Representation, определенной в работе [26]. Параметром задачи является значение $s \in \mathbb{N}$.

Определение 1.3.4. Для нарушителя $\mathcal{A}$ и группы $\mathbb{G}$, параметра s:

$$\mathsf{Adv}^{\mathrm{REPR}}_{\mathbb{G},s}(\mathcal{A}) = \Pr\big[\mathbf{Exp}^{\mathrm{REPR}}_{\mathbb{G},s}(\mathcal{A}) \to 1\big],$$

где эксперимент $\mathbf{Exp}^{\mathrm{REPR}}_{\mathbb{G},s}(\mathcal{A})$ определяется следующим образом:

$$\begin{array}{l}\underline{\mathbf{Exp}^{\mathrm{REPR}}_{\mathbb{G},s}(\mathcal{A})}\\ x_1,\ldots,x_s \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*\\ (\alpha_1,\ldots,\alpha_s,\beta) \leftarrow \mathcal{A}(x_1P,\ldots,x_sP)\\ \textbf{return } (\alpha_1x_1+\ldots+\alpha_sx_s+\beta=0)\wedge(\exists i:\alpha_i\neq 0)\end{array}$$

Нарушитель получает на вход набор точек $x_1P,\ldots,x_sP$, его задача — найти такие значения $\alpha_1,\ldots,\alpha_s,\beta$, что линейная комбинация $x_1,\ldots,x_s$ с коэффициентами $\alpha_1,\ldots,\alpha_s$ равна $(-\beta)$.

Таким образом, в настоящем разделе введены две задачи в группе точек эллиптической кривой: задача SOMDL и задача REPR. Нижняя оценка стойкости схемы Шаума-Педерсена в модели wUF, доказанная в следующем разделе, зависит в том числе от вероятности успешного решения данных задач.

1.3.2. Оценка стойкости

В настоящем разделе доказана нижняя оценка стойкости схемы Шаума-Педерсена в модели wUF с алгебраической группой и случайным оракулом.

Теорема 1.3.1. *Для любого нарушителя $\mathcal{A}$ для схемы* CP-BS *в модели* wUF *с алгебраической группой, делающего не более $t,\ell,\ell\leqslant t$, запросов к оракулам $Sign_1, Sign_2$ соответственно и не более q_1,q_2 запросов к случайным оракулам, моделирующим работу хэш-функций $\mathcal{H}$ и H соответственно, существует нарушитель $\mathcal{B}$, решающий задачу* SOMDL *с параметрами $(2t,t)$, и нарушитель $\mathcal{C}$, решающий задачу* REPR *с параметром $(q_1+\ell+1)$, такие что*

$$\mathsf{Adv}^{\mathrm{wUF}}_{\mathsf{CP\text{-}BS}}(\mathcal{A}) \leqslant 2\cdot\mathsf{Adv}^{\mathrm{SOMDL}}_{\mathbb{G},2t,t}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{REPR}}_{\mathbb{G},q_1+\ell+1}(\mathcal{C}) + \frac{2(\ell+1)+q_2}{q}.$$

При этом $T_{\mathcal{B}}$, $T_{\mathcal{C}}$ не превосходят $2T_{\mathcal{A}}$, где $T_{\mathcal{A}},T_{\mathcal{B}},T_{\mathcal{C}}$ — вычислительные ресурсы нарушителей $\mathcal{A},\mathcal{B},\mathcal{C}$ соответственно.

Доказательство. Пусть $\mathcal{A}$ — нарушитель для схемы подписи CP-BS в модели wUF. У него есть доступ к четырем оракулам: $Sign_1, Sign_2, RO_1, RO_2$. Оракул RO_1 моделирует работу хэш-функции $\mathcal{H}$ и возвращает точки эллиптической кривой (за исключением нулевой точки). Оракул RO_2 моделирует работу хэш-функции H и возвращает элементы $\mathbb{Z}_q^*$. Пусть нарушитель $\mathcal{A}$ делает не более t запросов к оракулу $Sign_1$, не более ℓ запросов к оракулу $Sign_2$, $\ell\leqslant t$,

не более q_1 запросов к оракулу RO_1, не более q_2 запросов к оракулу RO_2. Таким образом, нарушитель $\mathcal{A}$ завершит ℓ сеансов протокола формирования подписи и всего откроет t сеансов. В результате своей работы нарушитель $\mathcal{A}$ возвращает $(\ell+1)$ пару (сообщение, подпись).

Для начала сделаем три технических этапа (шага) доказательства.

Шаг 1. Нарушитель $\mathcal{A}$ для любой точки, которую он выдает, обязан предоставить разложение по элементам группы, которые появлялись до этого момента в рамках эксперимента.

За время эксперимента нарушитель $\mathcal{A}$ получает от экспериментатора точки P, Q, $\{M'_i\}_{i=1}^{q_1}$, $\{A_i, B_i, Z_i\}_{i=1}^{t}$.

Нарушитель $\mathcal{A}$ подает точки $M_j, 1 \leqslant j \leqslant t$, на вход оракулу $Sign_1$, точки $\{M'_j,\ Z'_j,\ A'_j,\ B'_j\}_{j=1}^{q_2}$ — на вход оракулу RO_2, а также точки $Z'_i, 1 \leqslant i \leqslant \ell+1,$ — в составе подписей в подделке. Для всех этих точек он должен предоставить разложение.

Зафиксируем наборы коэффициентов

$$(\alpha_i, \beta_i, \{\gamma_{ij}\}_{j=1}^{t}, \{\sigma_{ij}\}_{j=1}^{t}, \{\eta_{ij}\}_{j=1}^{t}, \{\xi_{ij}\}_{j=1}^{q_1}), 1 \leqslant i \leqslant q_2,$$

определяющие разложение точек $B'_i, 1 \leqslant i \leqslant q_2$, подаваемых нарушителем на вход оракулу RO_2. Пусть

$$B'_i = \alpha_i P + \beta_i Q + \sum_{j=1}^{t} \gamma_{ij} A_j + \sum_{j=1}^{t} \sigma_{ij} B_j + \sum_{j=1}^{t} \eta_{ij} Z_j + \sum_{j=1}^{q_1} \xi_{ij} M'_j.$$

Также зафиксируем наборы коэффициентов

$$(\hat{\alpha}_j, \hat{\beta}_j, \{\hat{\gamma}_{ji}\}_{i=1}^{j-1}, \{\hat{\sigma}_{ji}\}_{i=1}^{j-1}, \{\hat{\eta}_{ji}\}_{i=1}^{j-1}, \{\hat{\xi}_{ji}\}_{i=1}^{q_1}), 1 \leqslant j \leqslant t,$$

определяющие разложение точек $M_j, 1 \leqslant j \leqslant t$, подаваемых нарушителем на вход оракулу $Sign_1$. Пусть

$$M_j = \hat{\alpha}_j P + \hat{\beta}_j Q + \sum_{i=1}^{j-1} \hat{\gamma}_{ji} A_i + \sum_{i=1}^{j-1} \hat{\sigma}_{ji} B_i + \sum_{i=1}^{j-1} \hat{\eta}_{ji} Z_i + \sum_{i=1}^{q_1} \hat{\xi}_{ji} M'_i.$$

Шаг 2. Пусть нарушитель $\mathcal{A}$ выдал некоторую корректную пару $(m, (s', c', Z'))$ в качестве подделки. Нарушитель $\mathcal{A}$ мог делать соответствующий запрос $M' \parallel Z' \parallel A' \parallel B'$ к оракулу RO_2, где $M' = \mathcal{H}(m), A' = s'P - c'Q, B' = s'M' - c'Z'$, или не делать его.

Если нарушитель $\mathcal{A}$ не делал запрос, то в процессе проверки подписи будет определено новое значение случайной функции, поэтому вероятность того, что функция Verify вернет 1, не будет превосходить $\dfrac{1}{q-1}$ для конкретного значения подделки. Поскольку количество

подделок равно $\ell+1$, то суммарная вероятность того, что не был сделан хотя бы один запрос, не превышает $\dfrac{\ell+1}{q-1}$.

Далее будем рассматривать только те эксперименты, в которых каждой выданной подделке соответствует запрос нарушителя $\mathcal{A}$ к оракулу RO_2.

Шаг 3. Серверная часть протокола формирования подписи схемы подписи вслепую Шаума-Педерсена в точности повторяет действия доказывающего в протоколе доказательства с нулевым разглашением Шаума–Педерсена [32]. Это протокол доказательства равенства двух дискретных логарифмов:

$$\mathrm{DLog}_P Q = \mathrm{DLog}_M Z.$$

Проверка доказательства осуществляется в точности аналогично проверке подписи в схеме Шаума-Педерсена.

Аналогично обоснованию стойкости протокола доказательства знания Шаума-Педерсена, покажем, что если некоторая подпись (s', c', Z') успешно проходит проверку для сообщения m, то значение Z' в составе этой подписи с большой вероятностью равно dM', где $M' = \mathcal{H}(m)$.

Действительно, пусть для этой подписи $\mathrm{DLog}_{M'} Z' = x \neq d$. Согласно алгоритму проверки подписи, восстановим значения $A' = s'P - c'Q$ и $B' = s'M' - c'Z'$ и рассмотрим соответствующий данной подписи запрос $(M' \parallel Z' \parallel A' \parallel B')$ к оракулу RO_2. Заметим, что в силу шага 2 этот запрос обязательно был сделан. Пусть $k_1 = \mathrm{DLog}_P A'$ и $k_2 = \mathrm{DLog}_{M'} B'$. Тогда для данных значений должны быть выполнены следующие равенства:

$$k_1 = s' - c'd,$$

$$k_2 = s' - c'x.$$

Отсюда получаем $s' = k_1 + c'd = k_2 + c'x$, откуда $c' = \dfrac{k_1 - k_2}{x - d}, d \neq x$. Таким образом, настоящие уравнения выполнены (а значит, подпись успешно проверяется) при единственном значении c', это значение зафиксировано на момент подачи запроса случайному оракулу. Вероятность того, что для конкретного запроса выход случайного оракула примет именно значение $\dfrac{k_1 - k_2}{x - d}$, равна $\dfrac{1}{q-1}$.

Поскольку нарушитель $\mathcal{A}$ делает не более q_2 запросов к случайному оракулу RO_2, вероятность того, что хотя бы для одного запроса будет выполнено условие выше, не превосходит $\dfrac{q_2}{q-1}$. Таким образом, с вероятностью не большей $\dfrac{q_2}{q-1}$ среди точек $Z'_i, 1 \leqslant i \leqslant \ell+1$, в составе всех подделок есть хотя бы одна точка не равная dM'_i.

Далее будем рассматривать только те эксперименты, в которых нарушитель возвращает подделки, для каждой из которых верно, что $Z'_i = dM'_i, 1 \leqslant i \leqslant \ell+1$.

Таким образом, сделав три технических шага доказательства, мы перешли от исходного эксперимента $\mathbf{Exp}(\mathcal{A}) = \mathbf{Exp}^{\text{wUF}}_{\text{CP-BS}}(\mathcal{A})$ к модифицированному эксперименту $\mathbf{Exp}'$, работающему точно так же, как исходный эксперимент, за исключением наступления событий, определяемых шагом 2 и шагом 3. Разницу преимуществ нарушителя в исходном и модифицированном экспериментах можно оценить как:

$$\Pr[\mathbf{Exp}(\mathcal{A}) \Rightarrow 1] - \Pr[\mathbf{Exp}'(\mathcal{A}) \Rightarrow 1] \leqslant \frac{\ell + 1 + q_2}{q - 1}.$$

Далее будем строить алгоритмы работы двух нарушителей — нарушителя $\mathcal{B}$ для задачи SOMDL и нарушителя $\mathcal{C}$ для задачи REPR — использующих нарушителя $\mathcal{A}$ в качестве черного ящика. Покажем, что если нарушитель $\mathcal{A}$ успешно решает свою задачу, т.е. строит $(\ell + 1)$ подделку в результате ℓ успешных взаимодействий с подписывающим, то хотя бы один из нарушителей $\mathcal{B}$ или $\mathcal{C}$ успешно решает свою задачу.

Алгоритм работы нарушителя $\mathcal{B}$. Пусть у нарушителя $\mathcal{B}$ на входе есть точки $(A_1, \ldots, A_t, Q)$. Нарушитель $\mathcal{B}$ заводит два множества Π_1, Π_2, изначально полагая их пустыми, запускает нарушителя $\mathcal{A}$, подавая ему на вход точку Q, и моделирует ответы на запросы к случайным оракулам, используя так называемую технику «lazy sampling», следующим образом:

$SimRO_1(m)$	$SimRO_2(str)$
1 : **if** $m \in \Pi_1$:	1 : **if** $str \in \Pi_2$:
2 : **return** $\Pi_1(m) \cdot P$	2 : **return** $\Pi_2(str)$
3 : $x \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$	3 : $c \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
4 : $\Pi_1 \leftarrow \Pi_1 \cup \{m, x\}$	4 : $\Pi_2 \leftarrow \Pi_2 \cup \{str, c\}$
5 : **return** xP	5 : **return** c

Нарушитель $\mathcal{B}$ фиксирует переходы случайных функций адаптивно по мере запросов нарушителя $\mathcal{A}$ к соответствующим случайным оракулам, сохраняет в множество Π_2 все пары (запрос, ответ), соответствующие работе оракула RO_2, а в множество Π_1 — все запросы к оракулу RO_1 и дискретные логарифмы ответов, соответствующих данным запросам. Если запрос к определенной функции повторяется, то нарушитель $\mathcal{B}$ возвращает то же значение, которое возвращал ранее, используя значения, сохраненные в множествах Π_1, Π_2. Заметим, что по построению нарушитель $\mathcal{B}$ знает дискретные логарифмы точек M' относительно точки P.

Нарушитель $\mathcal{B}$ симулирует работу оракулов $Sign_1$ и $Sign_2$ следующим образом:

$SimSign_1(M)$	$SimSign_2(i, c)$
// i-й запрос нарушителя $\mathcal{A}$	1 : $Y \leftarrow A_i + cQ$
1 : $A \leftarrow A_i$	2 : $s \leftarrow O_2(Y)$
2 : $B \leftarrow O_1(i, M)$	3 : **return** s
3 : $Z \leftarrow O_1(t+1, M)$	
4 : **return** (A, B, Z)	

Пусть нарушитель $\mathcal{A}$ делает i-й запрос вида M к оракулу $Sign_1$. Тогда нарушитель $\mathcal{B}$ полагает точку A равной очередной точке $A_i = k_i P$, полученной на входе. Для получения точки B нарушитель делает запрос (i, M) к оракулу O_1 и получает в ответ точку $k_i M$. Для получения точки Z нарушитель делает запрос $(t+1, M)$ к оракулу O_1 и получает в ответ точку dM. Тройку (A, B, Z) нарушитель возвращает в качестве ответа на запрос к оракулу $Sign_1$. В результате нарушитель $\mathcal{B}$ делает не более $2t$ запросов к собственному оракулу O_1.

Работу оракула $Sign_2$ нарушитель $\mathcal{B}$ симулирует следующим образом: получив запрос (i, c), формирует точку $Y = A_i + cQ$, делает запрос Y своему оракулу O_2 и возвращает полученный ответ $\mathrm{DLog}_P(A_i + cQ) = k_i + cd$ нарушителю $\mathcal{A}$.

В результате своей работы нарушитель $\mathcal{A}$ возвращает $(\ell + 1)$ пару (сообщение, подпись): $\{m_i, (s'_i, c'_i, Z'_i)\}_{i=1}^{\ell+1}$. В силу шага 2, для каждого значения подделки можно найти соответствующий запрос (M'_i, Z'_i, A'_i, B'_i) к оракулу RO_2. В рамках этого запроса нарушитель $\mathcal{A}$ был обязан подать разложение всех точек (см. шаг 1), в частности, точки B'_i.

Тогда для каждой подделки нарушитель $\mathcal{B}$ может выписать следующее соотношение:

$$s'_i M'_i - c'_i Z'_i = B'_i = \alpha_i P + \beta_i Q + \sum_{j=1}^{t} \gamma_{ij} A_j + \sum_{j=1}^{t} \sigma_{ij} B_j + \sum_{j=1}^{t} \eta_{ij} Z_j + \sum_{j=1}^{q_1} \xi_{ij} M'_j, \tag{1.1}$$

где $i \in \{1, \ldots, \ell + 1\}$. Таким образом, нарушитель $\mathcal{B}$ получит систему из $(\ell + 1)$ уравнения.

Левое представление точки B'_i справедливо в силу того, что подпись является корректной. Правое представление — это представление, поданное нарушителем $\mathcal{A}$ при запросе к RO_2.

Заметим, что для всех завершенных сеансов нарушитель $\mathcal{B}$ может представить точки A_j и B_j как $(s_j P - c_j Q)$ и $(s_j M_j - c_j Z_j)$ соответственно. Для всех незавершенных сеансов нарушитель $\mathcal{B}$ не делал запрос к оракулу O_2 (так как нарушитель $\mathcal{A}$ не делал запрос к $Sign_2$), а потому не знает представление k_j через линейную комбинацию $s_j - c_j d$. Пусть для всех незавершенных сеансов с некоторым номером j нарушитель $\mathcal{B}$ после окончания работы нарушителя $\mathcal{A}$ подает запросы вида A_j к своему оракулу O_2. Тогда он получает в ответ соответствующие этим сеансам значения k_j и может в явном виде представить все точки A_j и B_j из незавершенных сеансов как $k_j P$ и $k_j M_j$ соответственно. В результате количество запросов нарушителя $\mathcal{B}$ к оракулу O_2 в точности равно t.

Заметим также, что все значения Z_j в разложении (1.1) можно представить как dM_j, а точка Z'_i, согласно шагу 3 и порядку симулирования оракула RO_1, равна $dM'_i = dx_i P$.

Будем обозначать через $\text{Ç} \subseteq \{1, \ldots, t\}$ множество номеров завершенных сеансов, через $\text{ÍÇ} \subseteq \{1, \ldots, t\}$ — незавершенных. Рассмотрим уравнение (1.1) относительно неизвестного d:

$$s'_i x_i P - c'_i d x_i P = B'_i = \\ = \alpha_i P + \beta_i dP + \sum_{j \in \text{Ç}} \gamma_{ij}(s_j P - c_j dP) + \sum_{j \in \text{ÍÇ}} \gamma_{ij} k_j P + \\ + \sum_{j \in \text{Ç}} \sigma_{ij}(s_j M_j - c_j dM_j) + \sum_{j \in \text{ÍÇ}} \sigma_{ij} k_j M_j + \sum_{j=1}^{t} \eta_{ij} dM_j + \sum_{j=1}^{q_1} \xi_{ij} x_j P. \quad (1.2)$$

Покажем, что в настоящем разложении можно переформировать коэффициенты таким образом, чтобы избавиться от сумм вида $\sum_{j \in \text{ÍÇ}}$.

Действительно, прибавим к α_i значение $\sum_{j \in \text{ÍÇ}} \gamma_{ij} k_j$, известное нарушителю $\mathcal{B}$. Обозначим результирующий коэффициент через α_i^*. Тем самым избавимся от суммы $\sum_{j \in \text{ÍÇ}} \gamma_{ij} k_j P$. Заметим, что коэффициент α_i^* фиксируется в момент подачи запроса оракулу RO_2 нарушителем $\mathcal{A}$, т.к. коэффициенты α_i, γ_{ij} подаются в составе этого запроса, а значения k_j, соответствующие $\gamma_{ij} \neq 0$, уже были выбраны экспериментатором нарушителя $\mathcal{B}$. Действительно, $\mathcal{A}$ подает разложение только по тем точкам, которые возвращались ему в результате эксперимента, а значит, он уже делал запросы $Sign_1$ в соответствующих сеансах.

Рассмотрим точки M_j, соответствующие незавершенным сеансам. Точка M_j из первого незавершенного сеанса очевидно не зависит от других точек из незавершенных сеансов, а потому может быть представлена как линейная комбинация точек только из завершенных сеансов. Второй незавершенный сеанс (пусть его номер равен j') может зависеть от точек из завершенных сеансов, а также от значений A_j, B_j, Z_j первого незавершенного сеанса с номером j. В этом случае можно представить точки A_j, B_j, Z_j как $k_j P, k_j M_j$ и dM_j, где M_j зависит только от точек из завершенных сеансов. Таким образом, перегруппировав коэффициенты, мы получим представление точки $M_{j'}$ через точки, соответствующие завершенным сеансам. Далее аналогично можно представить все точки A_j, B_j из незавершенных сеансов как линейные комбинации точек из завершенных сеансов.

Таким образом, можно переписать систему уравнений (1.2) как

$$s'_i x_i P - c'_i d x_i P = \alpha'_i P + \beta'_i dP + \sum_{j \in \text{Ç}} \gamma'_{ij}(s_j P - c_j dP) + \\ + \sum_{j \in \text{Ç}} \sigma'_{ij}(s_j M_j - c_j dM_j) + \sum_{j=1}^{t} \eta'_{ij} dM_j + \sum_{j=1}^{q_1} \xi'_{ij} x_j P. \quad (1.3)$$

Заметим, что аналогично рассуждениям выше значения всех коэффициентов α'_i, β'_i, γ'_{ij}, σ'_{ij}, η'_{ij}, ξ'_{ij} фиксируются в момент подачи запроса оракулу RO_2 нарушителем $\mathcal{A}$.

Каждая точка M_j также имеет некоторое представление, нарушитель $\mathcal{A}$ подает его при запросе к оракулу $Sign_1$. При этом, точки M_j можно представить в виде

$$M_j = \sum_{t=0}^{j} \tilde{l}_{j,t} d^t P,$$

где $\tilde{l}_{j,t}$ — аффинная функция от значений $x_1, \ldots, x_{q_1}$. Действительно, точка M_1 является разложением только по точкам $P, Q = dP, M'_i = x_i P, 1 \leqslant i \leqslant q_1$, т.е. разложение содержит только первую степень d. Следующая точка M_2 может содержать в разложении точку $Z_1 = dM_1$, а потому степень d в разложении может увеличиться на единицу. Далее аналогично точка M_j содержит в разложении не более j-й степени d. В силу того, что в результате запроса к $Sign_1$ ни одна из точек не умножается на значения x_i, эти значения никогда не умножаются друг на друга и могут входить в разложения только в первой степени через замешивание точек M'_i. Таким образом, каждый коэффициент перед $d^t P$ можно представить как аффинную функцию от значений $x_1, \ldots, x_{q_1}$.

Заметим, что получить такое представление точки M_j можно за полиномиальное время. Действительно, в ходе эксперимента нарушитель сам подает разложения точек M_j, а в результате запроса к оракулу $Sign_1$ через это разложение однозначно определяется разложение точек B_j, Z_j за счет домножения всех коэффициентов на k_j и d соответственно и представления точек $M_{j'}$ из предыдущих запросов. Таким образом, нарушитель на каждом шаге контролирует разложения всех точек.

Тогда для дискретных логарифмов (по основанию P) точек, представленных в системе уравнений (1.3), получаем следующую систему уравнений:

$$s'_i x_i - c'_i d x_i = \alpha'_i + \beta'_i d + \sum_{j \in \text{Ç}} \gamma'_{ij}(s_j - c_j d) + \\ + \sum_{j \in \text{Ç}} \sigma'_{ij}(s_j - c_j d) \sum_{t=0}^{j} \tilde{l}_{j,t} d^t + \sum_{j=1}^{t} \eta'_{ij} d \sum_{t=0}^{j} \tilde{l}_{j,t} d^t + \sum_{j=1}^{q_1} \xi'_{ij} x_j. \quad (1.4)$$

Для нарушителя $\mathcal{B}$ единственным неизвестным значением в этой системе является значение d. Каждое уравнение представляет собой полином от d степени не большей $\ell+1$. При этом,

в силу того, что подписи корректные и нарушитель предоставляет корректное разложение точек, корень d обязательно существует. Тогда если хотя бы одно из уравнений этой системы существенным образом зависит от d, то нарушитель $\mathcal{B}$ может найти значение d с помощью вероятностного алгоритма факторизации полиномов, описанного в [60] (см. алгоритм 4). Положим количество итераций данного алгоритма равным $2(\log \ell + 1)$. Тогда трудоемкость поиска всех корней полинома составляет $O(\ell)$ операций. Событие, что алгоритм успешно решает задачу, обозначим через factor, вероятность успешного запуска алгоритма составляет $\Pr[\mathsf{factor}] \geqslant \frac{1}{2}$. Нарушитель $\mathcal{B}$, найдя все корни системы (1.4), может найти правильное значение ключа d перебором по всем корням $d_i, 1 \leqslant i \leqslant \ell$, и сравнением $d_i P$ с открытым ключом Q, трудоемкость этого шага составляет $O(\ell)$ операций. Трудоемкость поиска ключа d, таким образом, не превосходит значения $T_{\mathcal{A}}$. Если нарушитель $\mathcal{B}$ успешно восстанавливает значение d, то он успешно решает задачу SOMDL, т.к. может восстановить все остальные значения k_j из линейных комбинаций, полученных им от оракула O_2.

Единственным случаем, при котором нарушитель $\mathcal{B}$ не может найти корень d, является случай, когда система (1.4) является тривиальной относительно d, т.е. если во всех уравнениях коэффициент перед всеми степенями d равен 0. В частности, в этом случае свободный член (коэффициент перед нулевой степенью d) во всех уравнениях равен 0. Выпишем это условие:

$$s'_i x_i = \alpha'_i + \sum_{j \in \mathsf{Ç}} \gamma'_{ij} s_j + \sum_{j \in \mathsf{Ç}} \sigma'_{ij} s_j \tilde{l}_{j,0} + \sum_{j=1}^{q_1} \xi'_{ij} x_j \tag{1.5}$$

для всех $i = 1, \ldots, \ell + 1$.

Обозначим через event событие, когда не выполнено условие (1.5). Тогда

$$\mathsf{Adv}^{\mathrm{SOMDL}}_{\mathbb{G},2t,t}(\mathcal{B}) = \Pr\left[\mathbf{Exp}^{\mathrm{SOMDL}}_{\mathbb{G},2t,t}(\mathcal{B}) \to 1\right] = \Pr[(\mathbf{Exp}'(\mathcal{A}) \to 1) \wedge \mathsf{event} \wedge \mathsf{factor}] =$$
$$= \Pr[(\mathbf{Exp}'(\mathcal{A}) \to 1) \wedge \mathsf{event}] \Pr[\mathsf{factor}] \geqslant \frac{1}{2} \cdot \Pr[(\mathbf{Exp}'(\mathcal{A}) \to 1) \wedge \mathsf{event}].$$

Далее покажем, что если событие event не произошло, т.е. условие (1.5) выполнено, то нарушитель $\mathcal{C}$ с большой вероятностью успешно решает задачу REPR.

Алгоритм работы нарушителя $\mathcal{C}$. Построим алгоритм работы нарушителя $\mathcal{C}$, решающего задачу REPR. Пусть нарушитель $\mathcal{C}$ получает на вход точки $x_1 P, \ldots, x_{q_1} P$, где значения x_i выбраны случайно равновероятно из $\mathbb{Z}_q^*$.

Нарушитель $\mathcal{C}$ самостоятельно генерирует ключ подписи d и значения k_i, поэтому моделирует работу оракулов $Sign_1$ и $Sign_2$ в точности так же, как экспериментатор нарушителя $\mathcal{A}$. Работу оракула RO_2 нарушитель $\mathcal{C}$ симулирует так же, как и нарушитель $\mathcal{B}$. Работу оракула

RO_1 нарушитель $\mathcal{C}$ симулирует, отдавая на каждый новый запрос m очередную точку x_iP, полученную нарушителем $\mathcal{C}$ на вход от своего собственного экспериментатора.

Нарушитель $\mathcal{C}$ в точности так же, как и нарушитель $\mathcal{B}$, может составить систему уравнений (1.1), получив $(\ell+1)$ подделку от нарушителя $\mathcal{A}$. Заметим, что для нарушителя $\mathcal{C}$ неизвестными в этом уравнении будут являться только величины x_i. Значения d и k_i ему известны, поскольку он генерирует их самостоятельно. Нарушитель $\mathcal{C}$ может преобразовать это уравнение точно так же, как и нарушитель $\mathcal{B}$, выразив k_j от завершенных сеансов через d и подставив известные ему k_j для незавершенных сеансов.

Покажем, что если выполнено условие (1.5), то нарушитель $\mathcal{C}$ с большой вероятностью успешно решает задачу REPR, т.е. находит нетривиальную линейную комбинацию значений $x_1, \ldots, x_{q_1}$.

Пусть есть $(\ell+1)$ уравнение относительно переменных $x_1, \ldots, x_{q_1}$. Если хотя бы в одном из этих уравнений перед некоторым x_i стоит ненулевой коэффициент, то это уравнение задает нетривиальную линейную комбинацию. Таким образом, «плохим» случаем является следующий: коэффициенты перед всеми x_i во всех уравнениях равны нулю.

Прежде чем выписать это условие, сделаем два технических преобразования.

1. Перенумеруем подделки таким образом, чтобы они были упорядочены по порядку соответствующих им запросов к случайному оракулу RO_2. В силу шага 2 для каждой подделки можно найти запрос к RO_2. Тогда в результате настоящего переупорядочивания получим, что запрос для i-й подделки выполнен раньше, чем запрос для $(i+1)$-й подделки, $1 \leqslant i \leqslant \ell$. Технически это преобразование означает, что мы поменяли местами уравнения в системе (1.4). Очевидно, что подобное изменение не влияет на решение системы и может быть сделано за полиномиальное время.

2. Переобозначим переменные $x_1, \ldots, x_{q_1}$ таким образом, чтобы сообщению m_i в составе i-й подделки (номер подделки в результате преобразования 1) соответствовала переменная x_i, т.е. чтобы было выполнено $\mathcal{H}(m_i) = x_iP$. Если переменная x_i не соответствует ни одной подделке, то ее индекс может быть произвольным. Очевидно, что подобное изменение также не влияет на решение системы и может быть сделано за полиномиальное время.

В результате выполнения этих технических преобразований получаем систему уравнений (1.4), уравнения в которой упорядочены по порядку запросов к RO_2, а переменные x_i — по вхождению в набор подделок. Напомним условие (1.5), при котором нарушитель $\mathcal{B}$ не может

успешно решить свою задачу:

$$\left\{ s'_i x_i = \alpha'_i + \sum_{j=1}^{l} \gamma'_{ij} s_j + \sum_{j=1}^{l} \sigma'_{ij} s_j \tilde{l}_{j,0}(x_1, \ldots, x_{q_1}) + \sum_{j=1}^{q_1} \xi'_{ij} x_j, \quad 1 \leqslant i \leqslant \ell + 1 \right.$$

Теперь выпишем в матричном виде условие, означающее, что во всех уравнениях этой системы коэффициенты перед всеми x_j равны нулю, т.е. условие, при котором нарушитель $\mathcal{C}$ не может успешно решить свою задачу.

$$\ell+1 \left\{ \overbrace{\begin{pmatrix} s'_1 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & s'_2 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 0 & s'_3 & \ldots & 0 & 0 & \ldots & 0 \\ & & & \ddots & & & & \\ 0 & 0 & 0 & \ldots & s'_{\ell+1} & 0 & \ldots & 0 \end{pmatrix}}^{q_1} \right. =$$

$$= \underbrace{\begin{pmatrix} \ddots & & \\ & \sigma'_{ij} s_j & \\ & & \ddots \end{pmatrix}}_{\ell} \cdot \underbrace{\begin{pmatrix} \ddots & & \\ & \tilde{l}_{j,0} & \\ & & \ddots \end{pmatrix}}_{q_1} + \underbrace{\begin{pmatrix} \ddots & & \\ & \xi'_{ij} & \\ & & \ddots \end{pmatrix}}_{q_1} \quad (1.6)$$

Заметим, что коэффициенты ξ'_{ij}, составляющие матрицу справа, фиксируются при подаче запросов к случайному оракулу RO_2. Эти коэффициенты определяются коэффициентами разложения точки B'_i на входе случайного оракула и коэффициентами разложений точек M_j, которые на текущий момент уже были отправлены оракулу $Sign_1$, при этом, значения ξ'_{ij} фиксируются в момент подачи нарушителем $\mathcal{A}$ соответствующего запроса к оракулу RO_2. Тогда при подаче первого запроса к RO_2 фиксируется первая строка матрицы $\left(\xi'_{ij}\right)$, при подаче второго запроса — вторая строка и так далее.

Если сделан запрос $(M' \parallel Z' \parallel A' \parallel B')$ к оракулу RO_2, то зафиксировано в том числе значение $k' = \mathrm{DLog}_{M'} B'$, значение d также является фиксированным. В результате подачи запроса выбирается некоторое случайное c'. Поскольку подпись (s', c', Z) является корректной, должно быть верно следующее условие:

$$B' = s'M' - c'Z',$$

откуда следует

$$s' = k' + c'd.$$

Тогда, поскольку значения k' и d фиксированные, а c' выбирается случайно равновероятно из множества мощности $(q-1)$, можно считать, что значение s' также выбирается случайно

равновероятно из множества мощности $(q-1)$. Значит, можно считать, что элементы матрицы, стоящей слева в уравнении (1.6), выбираются случайно равновероятно после фиксации определенным образом матрицы справа, состоящей из значений ξ'_{ij}.

Перепишем матричное уравнение (1.6) следующим образом:

$$\ell+1\left\{\overbrace{\begin{pmatrix} s'_1-\xi'_{11} & \cdot & \cdot & \dots & \cdot & \dots \\ \cdot & s'_2-\xi'_{22} & \cdot & \dots & \cdot & \dots \\ \cdot & \cdot & s'_3-\xi'_{33} & \dots & \cdot & \dots \\ & & & \ddots & & \\ \cdot & \cdot & \cdot & \dots & s'_{\ell+1}-\xi'_{(\ell+1)(\ell+1)} & \dots \end{pmatrix}}^{q_1}\right. =$$

$$= \underbrace{\begin{pmatrix} \ddots & & \\ & \sigma'_{ij}s_j & \\ & & \ddots \end{pmatrix}}_{\ell} \cdot \underbrace{\begin{pmatrix} \ddots & & \\ & \tilde{l}_{j,0} & \\ & & \ddots \end{pmatrix}}_{q_1}$$

Заметим, что справа в этом уравнении стоит произведение двух матриц, ранг каждой из которых не превосходит ℓ (их размер $(\ell+1)\times\ell$ и $\ell\times q_1$ соответственно). Тогда ранг произведения также не превосходит ℓ.

Оценим, с какой вероятностью ранг матрицы слева будет отличен от $(\ell+1)$. Для этого будем рассматривать квадратную подматрицу размера $(\ell+1)\times(\ell+1)$, взяв левые $(\ell+1)$ столбцов исходной матрицы. Можно считать, что элементы, стоящие на главной диагонали этой подматрицы, выбираются случайно равновероятно из множества мощности $(q-1)$ (т.к. s' выбираются случайно). Подматрица, согласно рассуждениям выше, формируется следующим образом: фиксируется определенным образом произвольным первая строка, кроме элемента, стоящего на главной диагонали, после чего случайно выбирается этот элемент. Далее фиксируется определенным произвольным образом вторая строка и случайно выбирается элемент, стоящий на главной диагонали, и так далее.

Заметим, что ранг квадратной матрицы размера $m\times m$ отличен от m тогда и только тогда, когда ее определитель равен нулю. Будем обозначать квадратную подматрицу размера

t, стоящую в левом верхнем углу, через A_t. Искомую вероятность можно оценить как

$$\begin{aligned}\Pr[\det(A_{\ell+1})=0] &= \Pr[\det(A_{\ell+1})=0 \wedge \det(A_\ell)=0] + \\ &+ \Pr[\det(A_{\ell+1})=0 \wedge \det(A_\ell)\neq 0] \leqslant \\ &\leqslant \Pr[\det(A_\ell)=0] + \Pr[\det(A_{\ell+1})=0 \mid \det(A_\ell)\neq 0] \leqslant \\ &\leqslant \ldots \leqslant \Pr[\det(A_1)=0] + \sum_{i=1}^{\ell} \Pr[\det(A_{i+1})=0 \mid \det(A_i)\neq 0].\end{aligned}$$

Матрица A_1 состоит только из элемента $s'_1 - \xi'_{11}$, этот элемент выбирается случайно. Определитель будет равен нулю, если этот элемент равен нулю, вероятность такого события равна $1/(q-1)$, т.е. $\Pr[\det(A_1)=0] = 1/(q-1)$. Далее по индукции по размеру t матрицы A_t.

Пусть $\det(A_i) = d_i \neq 0$. Оценим $\Pr[\det(A_{i+1})=0 \mid \det(A_i)\neq 0]$. Матрица A_{i+1} получается из матрицы A_i приписыванием справа и снизу еще одного столбца и еще одной строки, при этом сначала произвольным образом фиксируются все элементы кроме элемента с номером $(i+1, i+1)$, после чего он выбирается случайно равновероятно. Разложим $\det(A_{i+1})$ по последней строке, тогда $\det(A_{i+1})$ состоит из значения $(s'_{i+1,i+1} - \xi'_{i+1,i+1}) \cdot \det(A_i)$, к которому прибавляется некоторые фиксированные значения. Таким образом, $\det(A_{i+1}) = 0$ только в том случае, когда $s'_{i+1,i+1}$ принимает некоторое фиксированное значение, т.е. с вероятностью $1/(q-1)$. Откуда получаем, что

$$\Pr[\det(A_{i+1})=0 \mid \det(A_i)\neq 0] = \frac{1}{q-1}.$$

Откуда следует

$$\Pr[\det(A_{\ell+1})=0] \leqslant \frac{\ell+1}{q-1}.$$

Таким образом, слева с вероятностью больше либо равной $1 - \dfrac{\ell+1}{q-1}$ стоит матрица ранга $(\ell+1)$, справа стоит матрица ранга не больше ℓ. Это означает, что условие (1.6) с большой вероятностью не выполнено, а значит, среди уравнений системы (1.5) есть нетривиальная линейная комбинация переменных x_i.

Замечание 1.3.1. *Если нарушитель $\mathcal{A}$ в составе подделок выдает сообщения, для которых он не делал запросы к оракулу RO_1, то в системе (1.5) слева будут одни значения x_i (от подделок), а справа — другие (от разложений). Тогда в этой системе точно есть нетривиальные комбинации x_i от подделок, т.к. s'_i ненулевые. В общем случае нарушитель $\mathcal{C}$ принимает на вход и использует для формирования ответов случайного оракула $(q_1+\ell+1)$ точек, т.е. параметр s в задаче* REPR *равен $q_1+\ell+1$.*

Обозначим через eqrank событие, что $\det(A_{\ell+1}) = 0$. Тогда, если условие (1.5) выполнено, т.е. произошло событие $\overline{\mathsf{event}}$, и событие eqrank не произошло, нарушитель $\mathcal{C}$ успешно решает свою задачу. Тогда

$$\begin{aligned}\Pr[\mathbf{Exp}'(\mathcal{A}) \to 1 \wedge \overline{\mathsf{event}}] = \\ = \underbrace{\Pr\big[\mathbf{Exp}'(\mathcal{A}) \to 1 \wedge \overline{\mathsf{event}} \wedge \overline{\mathsf{eqrank}}\big]}_{=\Pr\left[\mathbf{Exp}^{\mathrm{REPR}}_{\mathbb{G},q_1+\ell+1}(\mathcal{C})\right]} + \underbrace{\Pr[\mathbf{Exp}'(\mathcal{A}) \to 1 \wedge \overline{\mathsf{event}} \wedge \mathsf{eqrank}]}_{\leqslant \Pr[\mathsf{eqrank}]} \leqslant \\ \leqslant \Pr\left[\mathbf{Exp}^{\mathrm{REPR}}_{\mathbb{G},q_1+\ell+1}(\mathcal{C}) \to 1\right] + \Pr[\mathsf{eqrank}] \leqslant \mathsf{Adv}^{\mathrm{REPR}}_{\mathbb{G},q_1+\ell+1}(\mathcal{C}) + \frac{\ell+1}{q-1}.\end{aligned}$$

Итоговая оценка. Таким образом, предъявлены алгоритмы работы двух нарушителей $\mathcal{B}$ и $\mathcal{C}$, хотя бы один из которых с большой вероятностью решает свою задачу, если нарушитель $\mathcal{A}$ успешно строит подделки. Суммируя полученные выше результаты, получаем

$$\begin{aligned}\Pr[\mathbf{Exp}'(\mathcal{A}) \to 1] = \Pr[\mathbf{Exp}'(\mathcal{A}) \to 1 \wedge \mathsf{event}] + \Pr[\mathbf{Exp}'(\mathcal{A}) \to 1 \wedge \overline{\mathsf{event}}] \leqslant \\ \leqslant 2 \cdot \mathsf{Adv}^{\mathrm{SOMDL}}_{\mathbb{G},t,2t}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{REPR}}_{\mathbb{G},q_1+\ell+1}(\mathcal{C}) + \frac{\ell+1}{q-1}.\end{aligned}$$

Откуда следует, что

$$\mathsf{Adv}^{\mathrm{wUF}}_{\mathsf{CP\text{-}BS}}(\mathcal{A}) \leqslant 2 \cdot \mathsf{Adv}^{\mathrm{SOMDL}}_{\mathbb{G},t,2t}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{REPR}}_{\mathbb{G},q_1+\ell+1}(\mathcal{C}) + \frac{2(\ell+1)+q_2}{q-1}.$$

□

1.3.3. Трактовка полученной оценки

Полученная оценка стойкости свидетельствует о том, что сложность задач SOMDL и REPR является достаточным условием стойкости схемы $\mathsf{CP\text{-}BS}$ в модели wUF со случайным оракулом и алгебраической группой при соответствующих значениях параметров схемы. Более того, рассмотрение каждого из слагаемых, входящих в оценку, позволяет сделать вывод о некоторых необходимых условиях стойкости схемы. Далее сопоставим каждое слагаемое конкретным методам взлома схемы $\mathsf{CP\text{-}BS}$.

Первое слагаемое учитывает атаки на схему $\mathsf{CP\text{-}BS}$, направленные на восстановление ключа подписи d или эфемерного значения k хотя бы в одном из сеансов (в частности, за счет совпадения значений k в нескольких сеансах).

Для задачи SOMDL на текущий момент неизвестно методов решения более эффективных, чем решение за счет решения либо задачи SDL, либо задачи OMDL. Для задачи OMDL на

текущий момент неизвестно методов решения более эффективных, чем решение задачи дискретного логарифмирования [56]. Для задачи SDL известен метод решения, вычислительная трудоемкость которого ниже, чем трудоемкость известных методов решения задачи дискретного логарифмирования, — метод, предложенный в работе [33] для случая, когда параметр s является делителем $(q-1)$. Настоящий метод требует $T_{\text{op}} = c_1 \cdot \log q \cdot (\sqrt{q/s} + \sqrt{s})$ вычислений групповых операций и $T_{\text{mem}} = c_2 \cdot \max\{\sqrt{q/s}, \sqrt{s}\}$ памяти, где c_1, c_2 — константы, зависящие только от используемой модели вычислений.

Заметим, что настоящий метод позволяет восстановить ключ подписи схемы CP-BS. Действительно, нарушитель, выступающий в роли клиента, может последовательно открыть t сеансов, посылая в первом сеансе в первой пересылке значение $M_1 = Q = dP$, а в последующих сеансах — значение $M_i = Z_{i-1} = dM_{i-1} = d^i P$, $2 \leqslant i \leqslant t$, где Z_{i-1} — значение Z, полученное в ответ от сервера в $(i-1)$-м сеансе. Таким образом, в качестве значений $Z_i, 1 \leqslant i \leqslant t$, в открытых сеансах нарушитель получит значения $d^2P, \ldots, d^{t+1}P$. Тогда, используя метод решения задачи SDL с параметром $s = t + 1$ из работы [33], нарушитель восстанавливает ключ подписи d и успешно реализует угрозу.

Пусть количество t открытых сеансов протокола формирования подписи вслепую не превышает 2^{64}. Обозначим через s_m максимальный делитель числа $(q-1)$ для заданной кривой $\mathcal{E}$, такой что $s_m \leqslant 2^{64} + 1$. Приведем значения s_m и параметры рассматриваемого метода для стандартизированных в России эллиптических кривых простого порядка, определенных в [2].

Кривая	$\log q$	s_m	op	mem
id-tc26-gost-3410-2012-256-paramSetB	256	$\approx 2^{32}$	2^{120}	2^{112}
id-tc26-gost-3410-2012-256-paramSetC	256	$\approx 2^{62}$	2^{105}	2^{97}
id-tc26-gost-3410-2012-256-paramSetD	256	$\approx 2^{64}$	2^{104}	2^{96}
id-tc26-gost-3410-12-512-paramSetA	512	$\approx 2^{25}$	2^{252}	2^{243}
id-tc26-gost-3410-12-512-paramSetB	512	$\approx 2^{11}$	2^{259}	2^{250}

Значения T_{op} и T_{mem} рассматриваемого метода равны $c_1 \cdot \text{op}$ и $c_2 \cdot \text{mem}$ соответственно.

Таким образом, в схеме CP-BS рекомендуется использовать эллиптические кривые с как можно меньшим значением s_m.

Второе слагаемое учитывает атаки на схему CP-BS, направленные на поиск соотношений между точками $M'_i = \mathcal{H}(m_i)$ для различных сообщений m_i и генерационной точкой P.

Для задачи REPR неизвестно методов решения лучших, чем решение задачи дискретного логарифмирования [26]. Заметим, что решение задачи дискретного логарифмирования для

хотя бы одной точки $M'_i = \mathcal{H}(m_i)$, $1 \leqslant i \leqslant q_1$, позволяет реализовать атаку типа ROS, описанную в разделе 1.2.

Третье слагаемое учитывает атаки на схему CP-BS, направленные на перебор значения c' в подписи. Действительно, для фиксированных значений Z', s' и сообщения m вероятность найти значение c', такое что алгоритм проверки подписи завершится успешно, можно оценить как $\dfrac{q_2}{q-1}$, где q_2 — количество вычислений хэш-функции H.

Таким образом, полученная нижняя оценка стойкости показывает, что достаточным условием стойкости схемы CP-BS в модели wUF со случайным оракулом и алгебраической группой является сложность задач SOMDL и REPR. Исходя из рассуждений, представленных ранее, необходимым условием является сложность другой задачи в группе — задачи SDL. Для задачи SDL, в свою очередь, в работе [19] были получены результаты, косвенно свидетельствующие в пользу того, что эта задача в общем случае может быть легче, чем задача дискретного логарифмирования. Более того, так как решение задачи SOMDL не удалось свести только к решению SDL, могут быть обнаружены другие методы, решающие задачу SOMDL и приводящие к взлому схемы CP-BS.

Выводы

В настоящей главе впервые проведен математически строгий анализ свойства неподделываемости схемы подписи вслепую Шаума-Педерсена. Полученные результаты позволяют сделать вывод о том, что схема Шаума-Педерсена не является стойкой в наиболее сильной расширенной модели безопасности. Так, в разделе 1.2 разработан метод нарушения свойства сильной неподделываемости в модели с параллельными сеансами с вероятностью близкой к 1, а в разделе 1.3 доказано, что свойство слабой неподделываемости в модели с параллельными сеансами обеспечивается лишь в предположении сложности недостаточно изученных базовых задач в группе точек эллиптической кривой.

В результате проведенных исследований схема Шаума-Педерсена была исключена из рассмотрения в процессе выбора перспективной для стандартизации в Российской Федерации схемы подписи вслепую.

Глава 2

Анализ безопасности схем подписи вслепую на основе уравнения Эль-Гамаля в расширенных моделях безопасности

Настоящая глава посвящена исследованию свойств безопасности схем подписи вслепую, в основе которых лежит уравнение подписи Эль-Гамаля.

2.1. Классические схемы подписи на основе уравнения Эль-Гамаля

Понятие обобщенной схемы подписи Эль-Гамаля было впервые введено в работе [45] и далее расширено в работе [37], далее схемы подписи Эль-Гамаля обозначаются через GenEG.

Алгоритм генерации ключей в схемах подписи Эль-Гамаля определяется следующим образом:

GenEG.KGen

1 : $d \xleftarrow{U} \mathbb{Z}_q^*$

2 : $Q \leftarrow dP$

3 : **return** (d, Q)

Алгоритм подписи сообщения m включает в себя вычисление хэш-значения сообщения $e = H(m)$, выбор случайного k из множества $\mathbb{Z}_q^*$ и вычисление значения r как $kP.x \bmod q$. Для поддержания возможности проверки подписи, а также обеспечения ее стойкости, некоторые значения e и r отбрасываются. Значение s вычисляется в соответствии с уравнением подписи Эль-Гамаля. Согласно работе [45], уравнение подписи Эль-Гамаля в общем виде определяется следующим образом:

$$G_d(r, e, s) \cdot d + G_k(r, e, s) \cdot k + G_0(r, e, s) = 0, \tag{2.1}$$

где G_d, G_k, G_0 — функции $\mathbb{Z}_q^3 \to \mathbb{Z}_q^*$, которые являются аффинными по переменной z или z^{-1} для всех $z \in \{r, e, s\}$. Если существует единственное s, которое обращает уравнение (2.1) в верное равенство, то алгоритм подписи возвращает пару (r, s) в качестве результата своей работы, в противном случае он возвращает символ ошибки.

Например, уравнение подписи в схеме ECDSA [11] задается следующим уравнением подписи Эль-Гамаля:

$$s = k^{-1}(e + dr),$$

т.е. $G_d(r,e,s) = -r, G_k(r,e,s) = s, G_0(r,e,s) = -e$.

В работе [45] перечислены все возможные уравнения подписи Эль-Гамаля (здесь опущена разница между $+z$ и $-z$ и разница между z и z^{-1}, где $z \in \{r,e,s,k,d\}$):

1 :	$ed = rk + s$	7 :	$red = k + s$	13 :	$(r+e)d = k + s$
2 :	$ed = sk + r$	8 :	$d = rek + s$	14 :	$d = (r+e)k + s$
3 :	$rd = ek + s$	9 :	$sd = k + re$	15 :	$sd = k + (r+e)$
4 :	$rd = sk + e$	10 :	$d = sk + re$	16 :	$d = sk + (r+e)$
5 :	$sd = rk + e$	11 :	$red = sk + 1$	17 :	$(r+e)d = sk + 1$
6 :	$sd = ek + r$	12 :	$sd = rek + 1$	18 :	$sd = (r+e)k + 1$

Рис. 2.1. Уравнения подписи Эль-Гамаля

Результаты, полученные в настоящей главе, опираются именно на этот список.

Алгоритм проверки подписи (r,s) для сообщения m заключается в проверке равенства

$$r = R.x \bmod q,$$

где $R = -\dfrac{1}{G_k(r,e,s)} \left(G_d(r,e,s) \cdot Q + G_0(r,e,s) \cdot P\right),\ e = H(m).$

2.2. Атаки на свойство неотслеживаемости на некоторые схемы подписи вслепую на основе уравнения Эль-Гамаля

Прежде чем переходить к определению общего класса схем подписи вслепую Эль-Гамаля, рассмотрим атаки на свойство неотслеживаемости на три схемы подписи вслепую [6, 44, 79], в основе которых лежит уравнение Эль-Гамаля. Будем называть эти схемы по инициалам авторов, предложивших их, и году публикации соответствующей работы.

Алгоритм генерации ключа во всех схемах совпадает с алгоритмом генерации ключа классической схемы подписи Эль-Гамаля. Таким образом, секретным ключом подписи является элемент $d \in \mathbb{Z}_q^*$, а ключом проверки подписи является точка $Q = dP$.

Для избежания тривиальных атак предполагается, что во время протокола формирования подписи подписывающий и пользователь проверяют, что все полученные из канала и вычисленные самостоятельно элементы поля являются ненулевыми, точки принадлежат используемой группе и не равны нулевой точке кривой. Более того, пользователь всегда проверяет, что значения, полученные от подписывающего, являются корректными относительно его запроса. Если хотя бы одна из проверок не выполняется, участник прекращает выполнение протокола и возвращает ошибку в качестве результата работы.

Все описанные далее атаки применимы даже в самой слабой модели для свойства неотслеживаемости, предполагающей:

- нарушитель не влияет на процесс генерации ключей;
- честный пользователь самостоятельно выбирает сообщения, которые он будет подписывать;
- нарушитель не знает секретный ключ подписи;
- нарушитель завершает все сеансы и не провоцирует возникновение ошибок на стороне пользователя.

Более того, все эти атаки могут быть выполнены даже сторонним пассивным нарушителем, который не выступает в роли подписывающего.

2.2.1. Схема GYP16

В работе [44] в 2016 году было предложено четыре схемы подписи вслепую, построенные на основе классических схем подписи ECDSA, GDSA, KCDSA and DSTU. В настоящем разделе приведена атака на схему на основе подписи ECDSA, атаки на остальные схемы строятся аналогичным образом.

Описание схемы. Протокол формирования подписи определен на рис. 2.2.

Алгоритм проверки подписи (r, s) для сообщения m заключается в проверке равенства $R.x \bmod q = r$, где $R = s^{-1}(rQ + eP)$ и $e = H(m)$.

Атака. Покажем, что для фиксированных стенограммы протокола и сообщения существует лишь малое число корректных значений подписи, которые могли бы быть получены в результате выполнения этого протокола. Действительно, если стенограмма протокола (R, e, s) и сообщение m зафиксированы, то значения $r = R.x \bmod q$ и $e' = H(m)$ также являются фиксированными. Уравнение (1) позволяет однозначно определить значение r' следующим образом:

$$r' = re^{-1}e',$$

а потому точка R' также является фиксированной с точностью до знака. Для каждого возможного значения R' существует единственное значение α, такое что $R' = \alpha R$. Однако, согласно протоколу, значение α выбирается случайно равновероятно из множества $\mathbb{Z}_q^*$, поэтому вероятность в двух сеансах протокола выбрать значение α, такое что $(\alpha R).x \bmod q = re^{-1}e'$,

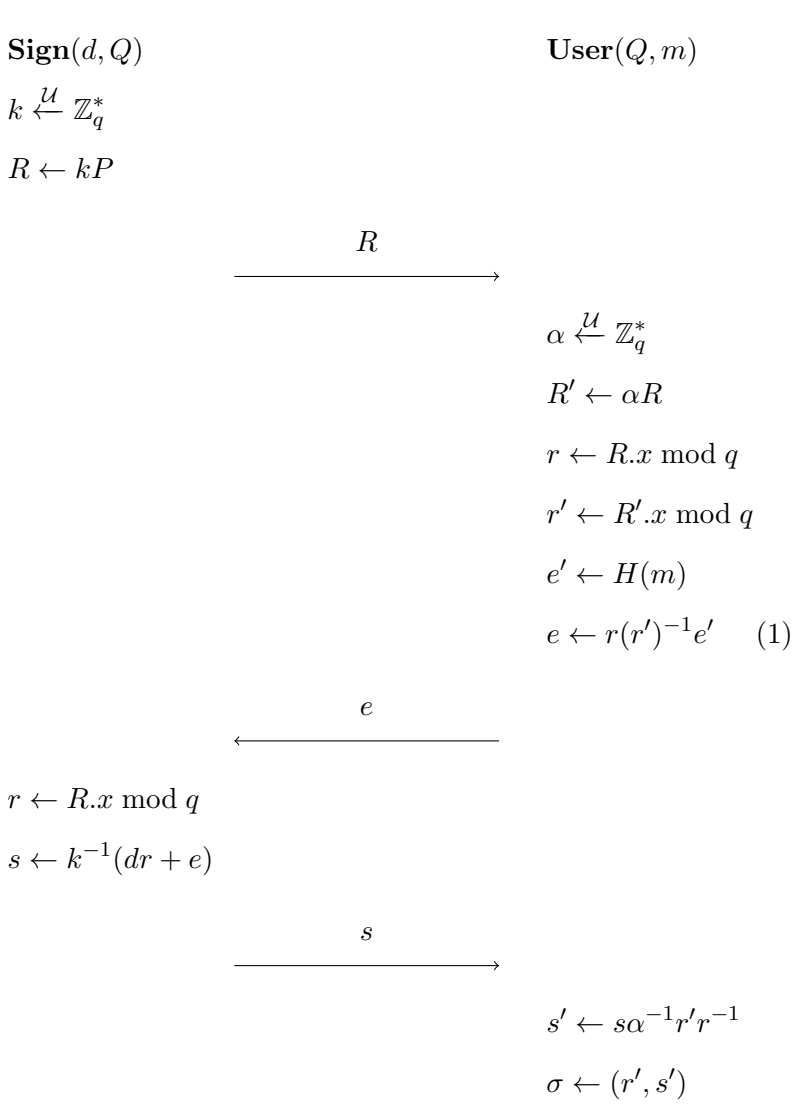

Рис. 2.2. Схема GYP16: протокол формирования подписи.

является пренебрежимо малой. Таким образом, с вероятностью близкой к единице существует только одна подпись с компонентой r', удовлетворяющая условию в строке (1).

Тогда строка (1) позволяет построить критерий для нарушения свойства неотслеживаемости. Транскрипция протокола (R, e, s) соответствует паре из сообщения m и подписи (r', s') тогда и только тогда, когда выполнено следующее условие:

$$e = r(r')^{-1}e',$$

где $e' = H(m)$.

2.2.2. Схема R00

В работе [6] в 2000 году было представлено две схемы подписи вслепую на основе схем подписи Шнорра и Эль-Гамаля. Обе схемы подвержены похожей атаке на свойство неотслеживаемости. Рассмотрим эту атаку на примере схемы подписи Эль-Гамаля.

Далее предполагается, что точки эллиптической кривой могут быть представлены в виде двоичных строк, соответствующих битовому представлению их координат, а потому могут быть поданы в качестве входного аргумента функции хэширования.

Описание схемы. Протокол формирования подписи определен на рис. 2.3.

Sign(d, Q)		**User**(Q, m)
$k \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$		
$R \leftarrow kP$		
$r \leftarrow H(R)$		
	$\xrightarrow{\quad R \quad}$	
		$\alpha \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
		$R' \leftarrow \alpha R \quad (1)$
		$r' \leftarrow H(R') \ (2)$
		$r \leftarrow H(R)$
		$\beta \leftarrow r' r^{-1} \quad (3)$
		$e' \leftarrow H(m)$
		$e \leftarrow \alpha \beta^{-1} e' \ (4)$
	$\xleftarrow{\quad e \quad}$	
$s \leftarrow ke + dr$		
	$\xrightarrow{\quad s \quad}$	
		$s' \leftarrow s\beta$
		$\sigma \leftarrow (R', s')$

Рис. 2.3. Схема R00: протокол формирования подписи.

Алгоритм проверки подписи (R, s) для сообщения m заключается в проверке равенства

$sP = H(R)Q + eR$, где $e = H(m)$.

Атака. Аналогично атаке, описанной в предыдущем разделе, покажем, что для фиксированных стенограммы протокола и сообщения существует лишь малое число корректных значений подписи, которые могли бы быть получены в результате выполнения этого протокола. Действительно, если стенограмма протокола (R, e, s) и сообщение m зафиксированы, то значения $r = R.x \bmod q$ и $e' = H(m)$ также являются фиксированными. Рассмотрим соотношение (4), принимая во внимание соотношения (1)–(3):

$$e = \alpha\beta^{-1}e' = \alpha(r'r^{-1})^{-1}e' = \alpha(r')^{-1}re' = \alpha(H(\alpha R))^{-1}re'.$$

Уравнение $e = \alpha(H(\alpha R))^{-1}re'$ относительно переменной α имеет малое число корней. Однако значение α выбирается случайно равновероятно из множества $\mathbb{Z}_q^*$, поэтому вероятность выбрать в нескольких сеансах протокола значения α, удовлетворяющие этому уравнению, является пренебрежимо малой. Тогда с вероятностью близкой к единице существует только одно значение подписи с компонентой $R' = \alpha R$, для которого α удовлетворяет условию (4).

Таким образом, соотношения (1)–(4) могут быть использованы для построения критерия для нарушения свойства неотслеживаемости. Транскрипция протокола (R, e, s) соответствует паре из сообщения m с хэш-значением e' и подписи (R', s') тогда и только тогда, когда выполнено следующее условие:

$$\alpha R = R',$$

где $\alpha = e(e')^{-1}H(R')H(R)^{-1}$.

Атака на схему подписи вслепую на основе схемы Шнорра, предложенную в [6], строится с использованием аналогичных рассуждений.

Интерпретация свойства неотслеживаемости. Представляется, что причины применимости описанной атаки кроются в некорректном понимании авторами работы [6] свойства неотслеживаемости. В этой работе обеспечение схемой свойства неотслеживаемости рассматривается как устойчивость схемы к атакам, ведущим к раскрытию сообщения m в результате выполнения протокола формирования подписи. Однако, как подробно обсуждалось в разделе, посвященном обзору моделей безопасности для схем подписи вслепую, свойство неотслеживаемости является более широким. Действительно, если для некоторой схемы подписи вслепую стенограмма протокола раскрывает информацию о сформированном значении подписи, то свойство неотслеживаемости также не обеспечивается.

2.2.3. Схема TNHV18

Аналогичная атака применима к агрегированной схеме подписи вслепую, предложенной в 2018 году в работе [79] (если быть точнее, было предложено два протокола формирования подписи, отличие которых заключалось в алгоритме работы пользователя). Без ограничения общности, опустим описание деталей схемы, связанных с агрегированием, и представим частный случай схемы с одним подписывающим. Действительно, описанная далее атака не требует знания нарушителем секретного ключа подписи и может быть осуществлена любым участником, наблюдающим стенограммы протокола формирования подписи и сформированные пары (сообщение, подпись).

Описание схемы. Протокол формирования подписи определен на рис. 2.4.

Sign(d, Q)		**User**(Q, m)		
		Вариант 1	Вариант 2	
$k \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$				
$R \leftarrow kP$				
	$\xrightarrow{R}$			
		$\alpha, \beta \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$	$\alpha, \beta \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$	
		$e' \leftarrow H(m)$	$e' \leftarrow H(m)$	
		$e \leftarrow \alpha e'$	$e \leftarrow \beta e'$	(1)
		$R' \leftarrow \beta R + \alpha P$	$R' \leftarrow \alpha^{-1} R + P + Q$	(2)
		$r' \leftarrow R'.x \bmod q$	$r' \leftarrow R'.x \bmod q$	(3)
		$r \leftarrow r'\beta^{-1}\alpha$	$r \leftarrow \alpha\beta(r' + e')$	(4)
	$\xleftarrow{r, e}$			
$s \leftarrow ke + dr$				
	$\xrightarrow{s}$			
		$s' \leftarrow \beta\alpha^{-1}s + \alpha e'$	$s' \leftarrow \beta^{-1}\alpha^{-1}s + e'$	
		$\sigma \leftarrow (r', s')$	$\sigma \leftarrow (r', s')$	

Рис. 2.4. Схема TNHV18: протокол формирования подписи.

Алгоритм проверки подписи (r, s) для сообщения m в обеих случаях заключается в проверке равенства $R.x \bmod q = r$, где $R = e^{-1}sP - e^{-1}rQ$ и $e = H(m)$.

Атака. Рассмотрим первый вариант схемы. Как и в предыдущих случаях, покажем, что для фиксированных стенограммы протокола и сообщения существует лишь малое число корректных значений подписи, которые могли бы быть получены в результате выполнения этого протокола. Действительно, если стенограмма протокола (R, r, e, s) и сообщение m зафиксированы, то значение $e' = H(m)$ также является фиксированным. Рассмотрим соотношение (4), принимая во внимание соотношения (1)–(3):

$$r = r'\beta^{-1}\alpha = (R'.x \bmod q)\beta^{-1}e(e')^{-1} = ((\beta R + \alpha P).x \bmod q)\beta^{-1}e(e')^{-1} = \\ = ((\beta R + e(e')^{-1}P).x \bmod q)\beta^{-1}e(e')^{-1}.$$

Уравнение

$$r = ((\beta R + e(e')^{-1}P).x \bmod q)\beta^{-1}e(e')^{-1}$$

относительно переменной β имеет малое число корней. Однако значение β выбирается случайно равновероятно из множества $\mathbb{Z}_q^*$, поэтому вероятность выбрать в нескольких сеансах протокола значения β, удовлетворяющие этому уравнению, является пренебрежимо малой. Тогда с вероятностью близкой к единице существует только одно значение подписи с компонентой r' равной $(\beta R + e(e')^{-1}P).x \bmod q$, для которого β удовлетворяет условию (4).

Таким образом, соотношения (1)–(4) могут быть использованы для построения критерия для нарушения свойства неотслеживаемости. Транскрипция протокола $(R,\ r,\ e,\ s)$ соответствует паре из сообщения m с хэш-значением e' и подписи (r', s') тогда и только тогда, когда выполнено следующее условие:

$$R'.x \bmod q = r',$$

где $R' = \beta R + \alpha P,\ \alpha = e(e')^{-1},\ \beta = r'r^{-1}\alpha$.

Атака на второй вариант схемы строится аналогичным образом. Транскрипция протокола (R, r, e, s) соответствует паре из сообщения m с хэш-значением e' и подписи (r', s') тогда и только тогда, когда выполнено следующее условие:

$$R'.x \bmod q = r',$$

где $R' = \alpha^{-1}R + P + Q,\ \alpha = r\beta^{-1}(r' + e')^{-1},\ \beta = e(e')^{-1}$.

2.3. Синтез схем подписи вслепую Эль-Гамаля GenEG-BS

В настоящем разделе вводится новый класс схем подписи вслепую — схемы подписи вслепую Эль-Гамаля. Каждая схема из этого класса далее обозначается через GenEG-BS. Параметрами таких схем являются используемая группа $\mathbb{G}$ и порядок ее простой подгруппы q. Таким образом, далее неявно рассматриваются семейства схем с растущим параметром безопасности $\lceil \log q \rceil$. Определим алгоритм генерации ключей, протокол формирования подписи и алгоритм ее проверки для схемы GenEG-BS.

Алгоритм генерации ключей осуществляется на стороне подписывающего и полностью совпадает с алгоритмом генерации ключей в классической схеме подписи Эль-Гамаля.

Протокол формирования подписи определен на рисунке 2.5. Инициатором начала сеанса является подписывающий, он вычисляет случайную эфемерную точку $R = kP$ и отправляет ее пользователю. Получив это значение, пользователь некоторым образом вычисляет значение $e \in \mathbb{Z}_q$ и отправляет его подписывающему. Последний вычисляет значение r по точке R и находит значение s из уравнения подписи Эль-Гамаля, используя для этого долговременный ключ подписи d, эфемерное значение k и полученное значение e. Далее значение s отправляется пользователю, который некоторым образом вычисляет на основе него значение подписи (r', s'), где $r', s' \in \mathbb{Z}_q$, для выбранного сообщения m. Таким образом, алгоритм работы подписывающего в протоколе формирования подписи схемы GenEG-BS полностью зафиксирован и представляет собой алгоритм подписи Эль-Гамаля для значения e, полученного от пользователя, тогда как алгоритм работы пользователя, т.е. алгоритм вычисления им значений e, r', s', не является фиксированным и вообще говоря может быть произвольным.

Алгоритм проверки подписи (r', s') для сообщения m полностью совпадает с алгоритмом проверки подписи в классической схеме Эль-Гамаля и заключается в проверке равенства

$$r' = R'.x \bmod q,$$

где $R' = -\dfrac{1}{G_k(r', e', s')} \left(G_d(r', e', s') \cdot Q + G_0(r', e', s') \right),\ e' = H(m).$

2.4. Анализ безопасности схем GenEG-BS в расширенных моделях безопасности

В настоящем разделе исследуется возможность построения схемы подписи вслепую GenEG-BS, обеспечивающей стойкость в расширенных моделях безопасности: Blind и UF. Заметим, что все известные в литературе схемы GenEG-BS были предложены без математически строгого обоснования свойства неподделываемости, свойство неотслеживаемости обосновано

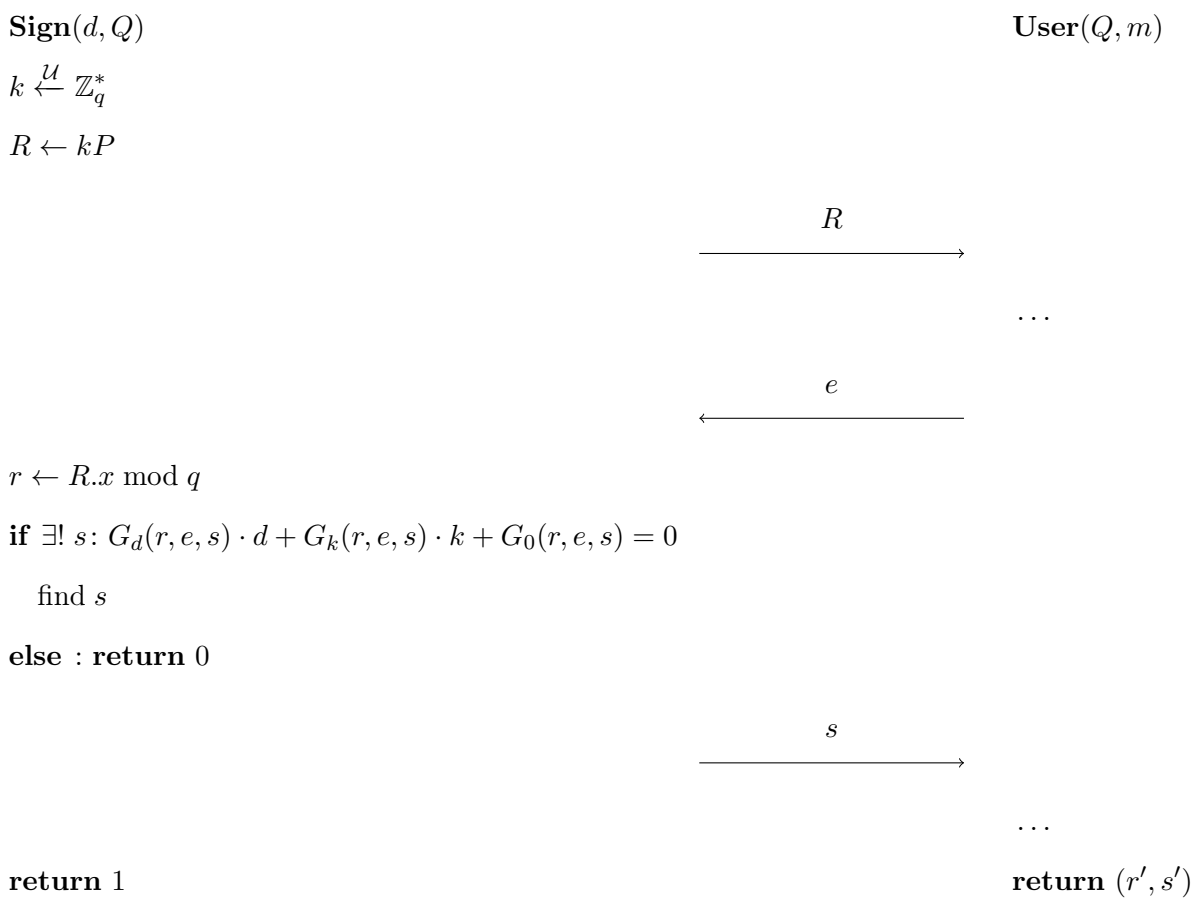

Рис. 2.5. Протокол формирования подписи в схеме GenEG-BS

лишь для части из них. Более того, в разделе 2.2 было доказано, что ряд схем подписи вслепую не обеспечивает свойство неотслеживаемости даже в слабой модели. Таким образом, до настоящего момента стойкость этих схем являлась открытым вопросом.

В ходе исследования выявляется два типа схем GenEG-BS. Они покрывают все известные в литературе схемы такого типа [6, 29, 44, 52, 55, 62, 77, 79, 80, 86]. Доказывается, что схемы обоих типов не обеспечивают либо свойство неподделываемости, либо неотслеживаемости.

Отправной точкой разделения схем GenEG-BS на два типа стало исследование применимости к этим схемам так называемой ROS атаки [24]. Как уже было сказано ранее, большинство схем подписи вслепую на основе уравнений Шнорра оказались уязвимыми к данной атаке, поэтому вопрос ее применимости к схемам GenEG-BS является актуальным.

2.4.1. Схемы типа I

Оказалось, что модифицированная версия ROS атаки применима к значительному количеству существующих схем [6, 29, 52, 55, 62, 77, 79, 86]. Выделим необходимые условия применимости этой атаки в виде условий, налагаемых на структуру уравнения подписи. Бу-

дем называть схемами типа I все схемы GenEG-BS с уравнением подписи, удовлетворяющим условию 1.

Условие 1: уравнение подписи может быть представлено в следующем виде:

$$k + Y_1(r, e) \cdot G_1(d) + Y_2(r, e, s) \cdot G_2(d) = 0, \tag{2.2}$$

где функции G_1 и G_2 аффинны по d, функция Y_1 существенно зависит от значения e и функция Y_2 является дробно-линейной по s.

Следующая теорема доказывает, что схемы типа I не являются стойкими в модели UF.

Теорема 2.4.1. *Пусть схема* GenEG-BS *удовлетворяет условию 1. Тогда для нее существует полиномиальный нарушитель $\mathcal{A}$ в модели* UF*, открывающий $\ell = \lceil \log q \rceil$ параллельных сеансов протокола формирования подписи, такой что*

$$\mathsf{Adv}^{\mathrm{UF}}_{\mathsf{GenEG\text{-}BS}}(\mathcal{A}) \geqslant 1 - \frac{\ell + 1}{q}.$$

Доказательство. Согласно условию 1, уравнение подписи может быть представлено в виде

$$k + Y_1(r, e) \cdot G_1(d) + Y_2(r, e, s) \cdot G_2(d) = 0,$$

где функции G_1 и G_2 аффинны по d, функция Y_1 существенно зависит от значения e и функция Y_2 является дробно-линейной по s.

Алгоритм проверки подписи (r', s') для сообщения m заключается в проверке равенства

$$r' = R'.x \bmod q,$$

где $R' = -Y_1(r', e') \cdot G_1(d)P - Y_2(r', e', s') \cdot G_2(d)P,\ e' = H(m)$. Заметим, что значения $G_1(d)P$ и $G_2(d)P$ всегда могут быть вычислены, поскольку функции G_1, G_2 являются аффинными по d и значение $Q = dP$ известно.

Построим атаку, которая позволяет нарушителю сформировать $(\ell + 1)$ корректную пару (сообщение, подпись) в результате $\ell = \lceil \log q \rceil$ успешных взаимодействий с подписывающим. Нарушитель выступает в роли нечестного пользователя и может вычислять значения e, r', s' произвольным образом. Таким образом, атака применима вне зависимости от способа вычисления этих значений, зафиксированных в некоторой конкретной схеме.

Пусть нарушитель выполняет следующие шаги.

1. Выбирает сообщение $m_\ell \in \{0,1\}^*$, для которого строит подделку, пусть $e'_\ell = H(m_\ell)$.
2. Открывает ℓ параллельных сеансов, отправляя соответствующие запросы подписывающему, получает в ответ $R_0, \ldots, R_{\ell-1}$.

3. Вычисляет $r_i = R_i.x \bmod q, 0 \leqslant i \leqslant \ell - 1$, полагает $r'_i = r_i, 0 \leqslant i \leqslant \ell - 1$.

4. Выбирает различные $m_{i0}, m_{i1} \in \{0,1\}^*, 0 \leqslant i \leqslant \ell - 1$, такие что $z_{i0} = Y_1(r'_i, e'_{i0}) \neq Y_1(r'_i, e'_{i1}) = z_{i1}$, где $e'_{i0} = H(m_{i0}), e'_{i1} = H(m_{i1})$.

   В предположении, что хэш-функция ведет себя как случайная функция, для всех уравнений Эль-Гамаля, перечисленных на рисунке 2.1, вероятность совпадения значений z_{i0} и z_{i1} для конкретного значения $i \in \{0, \ldots, \ell - 1\}$ равна $\frac{1}{q}$. Тогда вероятность совпадения этих значений хотя бы для одного значения $i \in \{0, \ldots, \ell - 1\}$ не превосходит $\frac{\ell}{q}$.

5. Полагает $(\rho_0, \rho_1, \ldots, \rho_\ell)$ равными вектору коэффициентов перед x_i в функции $f : \mathbb{Z}_q^\ell \to \mathbb{Z}_q$; $f(x_0, \ldots, x_{\ell-1}) = \sum\limits_{i=0}^{\ell-1} 2^i \underbrace{\frac{x_i - z_{i0}}{z_{i1} - z_{i0}}}_{b'_i} = \sum\limits_{i=0}^{\ell-1} \rho_i x_i + \rho_\ell$. Заметим, что если $x_i = z_{i0}$, то $b'_i = 0$, если $x_i = z_{i1}$, то $b'_i = 1$.

6. Полагает $R'_\ell = \sum\limits_{i=0}^{\ell-1} \rho_i R_i - \rho_\ell G_1(d)P$.

7. Вычисляет $r'_\ell = R'_\ell.x \mod q$.

8. Определяет $b_0, \ldots, b_{\ell-1}$ из следующего уравнения: $Y_1(r'_\ell, e'_\ell) = \sum\limits_{i=0}^{\ell-1} 2^i b_i$. Это возможно, т.к. $\ell = \lceil \log q \rceil$.

9. Полагает $z_i = z_{ib_i}, e'_i = e'_{ib_i}, m_i = m_{ib_i}, 0 \leqslant i \leqslant \ell - 1$; тогда, согласно шагу 5, $Y_1(r'_\ell, e'_\ell) = \sum\limits_{i=0}^{\ell-1} \rho_i z_i + \rho_\ell = \sum\limits_{i=0}^{\ell-1} \rho_i Y_1(r'_i, e'_i) + \rho_\ell$.

10. Полагает $e_i = e'_i, 0 \leqslant i \leqslant \ell - 1$.

11. Отправляет значения $e_0, \ldots, e_{\ell-1}$ подписывающему в соответствующих сеансах.

12. Получает ответы $s_0, \ldots, s_{\ell-1}$, такие что:

$$R_i + Y_1(r_i, e_i) \cdot G_1(d)P + Y_2(r_i, e_i, s_i) \cdot G_2(d)P = 0,\ 0 \leqslant i \leqslant \ell - 1.$$

13. Полагает $s'_i = s_i, 0 \leqslant i \leqslant \ell - 1$.

14. Определяет s'_ℓ из уравнения:

$$\sum_{i=0}^{\ell-1} \rho_i Y_2(r'_i, e'_i, s'_i) = Y_2(r'_\ell, e'_\ell, s'_\ell).$$

   Согласно условию 1, функция Y_2 является дробно-линейной по s. Тогда настоящее уравнение может быть представлено в виде $a_1 s'_\ell + a_2 = 0$, где значения $a_1, a_2 \in \mathbb{Z}_q$

фиксированные и зависят от значений $d, e'_\ell, R_i, e'_{i0}, e'_{i1}, 0 \leqslant i \leqslant \ell-1$. Если $a_1 \neq 0$, всегда можно найти значение s'_ℓ, которое обратит уравнение в верное равенство. Если $a_1 = 0$, нарушитель возвращается на шаг 1. Для всех уравнений Эль-Гамаля, перечисленных на рисунке 2.1, для любого ключа подписи d и для любых значений $e'_\ell, e'_{i0}, e'_{i1}, 0 \leqslant i \leqslant \ell-1$, выбранных нарушителем, событие $a_1 = 0$ происходит с вероятностью $\dfrac{1}{q}$ по выбору случайных значений R_i подписывающим.

15. Выдает набор $\{m_i, (r'_i, s'_i)\}_{i=0}^{\ell}$.

Действительно, для $0 \leqslant i \leqslant \ell-1$ подпись (r'_i, s'_i) является корректной для сообщения m_i из построения атаки и факта, что $r'_i = r_i, s'_i = s_i, e'_i = e_i$, см. шаг 12. Рассмотрим случай $i = \ell$. Суммируем все уравнения, полученные на шаге 12, с соответствующими коэффициентами:

$$\sum_{i=0}^{\ell-1} \rho_i R_i + \sum_{i=0}^{\ell-1} \rho_i \underbrace{Y_1(r_i, e_i)}_{=Y_1(r'_i, e'_i)} \cdot G_1(d)P + \sum_{i=0}^{\ell-1} \rho_i \underbrace{Y_2(r_i, e_i, s_i)}_{=Y_2(r'_i, e'_i, s'_i)} \cdot G_2(d)P = 0.$$

Вычтем и прибавим слагаемое $\rho_\ell G_1(d)P$ в левой части уравнения:

$$\underbrace{\sum_{i=0}^{\ell-1} \rho_i R_i - \rho_\ell G_1(d)P}_{=R'_\ell} + \underbrace{\left(\sum_{i=0}^{\ell-1} \rho_i Y_1(r'_i, e'_i) + \rho_\ell\right)}_{=Y_1(r'_\ell, e'_\ell)} \cdot G_1(d)P + \underbrace{\sum_{i=0}^{\ell-1} \rho_i Y_2(r'_i, e'_i, s'_i)}_{=Y_2(r'_\ell, e'_\ell, s'_\ell)} \cdot G_2(d)P = 0.$$

Согласно шагам 6, 9, 14, это уравнение эквивалентно следующему уравнению:

$$R'_\ell = -Y_1(r'_\ell, e'_\ell) \cdot G_1(d)P - Y_2(r'_\ell, e'_\ell, s'_\ell) \cdot G_2(d)P,$$

и $R'_\ell.x \bmod q = r'_\ell$ по построению, см. шаг 7. Следовательно, подпись (r'_ℓ, s'_ℓ) будет корректной для сообщения m_ℓ.

Вероятность успешного применения атаки определяется вероятностью успешного осуществления шагов 4 и 14 и, таким образом, больше либо равна $1 - \dfrac{\ell+1}{q}$. □

Причина, по которой возможна приведенная выше атака, состоит в том, что нарушитель имееет возможность варьировать значения $Y_1(r'_i, e'_i)$ на шаге 4 за счет изменения значений сообщения. Это, в свою очередь, возможно в силу наличия в уравнении подписи (2.2) слагаемого, которое не зависит от значения s. Этот факт и определяет вид условия 1.

Заметим, что пары (сообщение, подпись), сформированные нарушителем в результате настоящей атаки, могут быть сопоставлены с соответствующими сеансами протокола формирования подписи, поскольку нарушитель не использует никаких маскирующих значений. Тем не менее представляется, что для любой конкретной схемы, для которой математически строго доказано свойство неотслеживаемости (например, [29, 62]), атака может быть несложным образом модифицирована с целью возвращения $(\ell+1)$ подделки с сохранением свойства неотслеживаемости.

2.4.2. Схемы типа II

Рассмотрим уравнения подписи Эль-Гамаля, для которых не выполнено условие 1. Это уравнения 2, 4, 10, 11, 16 на рисунке 2.1, все они имеют следующий вид:

$$sk = F_1(r, e)d + F_2(r, e) \tag{2.3}$$

или

$$s^{-1}k = F_1(r, e)d + F_2(r, e), \tag{2.4}$$

где функции F_1 и F_2 являются аффинными по переменным z или z^{-1} для всех $z \in \{r, e\}$. Более того, только одна из функций F_1 и F_2 существенно зависит от значения r.

Далее рассматривается частный случай схем GenEG-BS на основе уравнений такого типа. А именно, рассматриваются схемы, в которых значение r' на стороне пользователя вычисляется следующим образом (способ вычисления значений e, s' по-прежнему может быть произвольным):

$$r' \leftarrow R'.x \bmod q, \tag{2.5}$$

где $R' = \alpha R + \beta Q + \gamma P$, каждое из значений α, β, γ (называемых маскирующими значениями) либо выбирается пользователем случайно равновероятно из множества $\mathbb{Z}_q^*$, либо равно нулю. Рассмотрение только равновероятного распределения на множестве маскирующий значений обусловлено тем, что другие распределения вероятно не позволяют обеспечить свойство совершенной неотслеживаемости. Все существующие в литературе схемы, известные автору диссертации, подразумевают именно такой способ вычисления первой компоненты r' подписи (вне зависимости от вида уравнения подписи). Таким образом, сужение рассматриваемого класса схем подписи не выводит нас из множества известных схем, а потому полученные результаты являются важными с практической точки зрения.

Итак, схема GenEG-BS является схемой типа II, если:

- уравнение подписи имеет вид (2.3) или (2.4);
- значение r' вычисляется в соответствии с (2.5).

Единственная представленная в литературе схема типа II — это схема, определенная в работе [44]. Однако для этой схемы в разделе 2.2 была представлена атака, нарушающая свойство неотслеживаемости. Настоящая атака позволяет сформулировать следующее условие, налагаемое на конструкцию схемы.

Пусть (R, e, s) — стенограмма протокола формирования подписи и $r = R.x \bmod q$. Пусть также (r', s') — подпись, вычисленная пользователем для сообщения m с хэш-значением $e' = H(m)$ в результате выполнения протокола с этой стенограммой.

Условие 2: равенство

$$F_1(r, e) \cdot F_2(r', e') = F_1(r', e') \cdot F_2(r, e) \tag{2.6}$$

выполняется с вероятностью $p \geqslant 1 - \dfrac{1}{q}$.

Здесь вероятностное пространство определяется выбором ключей, а также всех случайных значений на стороне подписывающего и пользователя.

Оказалось, что выполнение условия 2 позволяет построить критерий сопоставления стенограммы протокола и пары (сообщение, подпись).

Теорема 2.4.2. *Пусть схема* GenEG-BS *удовлетворяет условию 2. Тогда для нее существует полиномиальный нарушитель $\mathcal{A}$ в модели* Blind*, такой что*

$$\mathsf{Adv}^{\text{Blind}}_{\text{GenEG-BS}}(\mathcal{A}) \geqslant 1 - \frac{2}{q}.$$

Доказательство. Рассмотрим схемы GenEG-BS типа II, которые удовлетворяют условию 2. Покажем, что для фиксированных стенограммы протокола и сообщения существует лишь малое множество допустимых корректных значений подписи, которые могли бы быть получены в результате этого сеанса протокола.

Действительно, если стенограмма (R, e, s) и сообщение m зафиксированы, то значения $r = R.x \bmod q$ и $e' = H(m)$ также зафиксированы. Уравнение (2.6) аффинно по r', т.к. функции $F_1(r', e')$ и $F_2(r', e')$ также аффинны по r' и только одна из них существенно зависит от r'. Тогда значение r' может быть однозначным образом восстановлено из уравнения (2.6). Заметим, что α, β, γ либо равны нулю, либо выбраны случайно равновероятно из $\mathbb{Z}_q^*$. Вероятность выбрать α, β, γ так, что $(\alpha R + \beta Q + \gamma P).x \bmod q = r'$, не превосходит $\dfrac{2}{q}$. Таким образом, с вероятностью $1 - \dfrac{2}{q}$ существует единственная подпись, которая может быть сформирована для сообщения m в результате данного запуска протокола. □

Насколько известно автору диссертации, в литературе не представлено ни одной схемы типа II, для которой не выполнено условие 2. Это наблюдение позволило сформулировать следующую теорему, обосновывающую невозможность построения стойкой схемы подписи вслепую такого типа.

Теорема 2.4.3. *Если существует схема* GenEG-BS *типа II, которая не удовлетворяет условию 2, то для нее существует полиномиальный нарушитель $\mathcal{A}$ в модели* SEQ-UF*, такой что*

$$\mathsf{Adv}^{\text{SEQ-UF}}_{\text{GenEG-BS}}(\mathcal{A}) \geqslant 1 - \frac{1}{q}.$$

Доказательство. В ходе доказательства покажем следующее: факт существования подобной схемы ведет к тому, что либо в данной схеме возможно восстановление секретного ключа подписи из одной стенограммы протокола и полученного в результате нее значения подписи, либо нарушитель имеет возможность формировать корректные подписи без знания ключа и взаимодействия с подписывающим.

Предположим, что существует схема GenEG-BS типа II, для которой не выполнено условие 2. Это означает, что существует алгоритм User, который работает на стороне пользователя и выполняет следующие действия. Для произвольного открытого ключа pk, выработанного в результате работы алгоритма генерации ключей, произвольных сообщения m, точки R и маскирующих значений α, β, γ, выбранных в соответствии с распределением, зафиксированным в конкретной схеме, он возвращает значение e. Далее, после получения значения s, вычисленного согласно (2.3) или (2.4), алгоритм User выдает корректную подпись (r', s') для сообщения m.

Далее будет построен нарушитель $\mathcal{A}$ в модели SEQ-UF, который предъявляет подделку для схемы GenEG-BS и использует алгоритм User в качестве черного ящика. Нарушителю $\mathcal{A}$ известно значение открытого ключа Q, далее он выполняет следующие шаги.

1. Выбирает сообщение m и вычисляет $e' = H(m)$.

2. Выбирает значения α, β, γ случайно равновероятно из $\mathbb{Z}_q^*$ или полагает их равными нулю (в зависимости от порядка работы алгоритма User).

3. Открывает сеанс с подписывающим и получает в ответ точку R, вычисляет $r = R.x \bmod q$.

4. Вычисляет $r' = (\alpha R + \beta Q + \gamma P).x \bmod q$.

5. Запускает алгоритм User, подавая ему на вход точку Q, точку R, сообщение m и значения α, β, γ.

6. Получает значение e от алгоритма User.

7. Если $\gamma F_1(r, e) - \beta F_2(r, e) = 0$, переходит на следующий шаг.

   Если $\gamma F_1(r, e) - \beta F_2(r, e) \neq 0$, вычисляет

   $$s^* = (\gamma F_1(r, e) - \beta F_2(r, e))^{-1}(F_1(r, e)F_2(r', e') - F_2(r, e)F_1(r', e'))$$

   и проверяет корректность подписи, вычисляя

   $$b = \mathsf{GenEG\text{-}BS.Verify}(Q, m, (r', s^*)).$$

Если $b = 1$, нарушитель $\mathcal{A}$ возвращает пару $(m, (r', s^*))$ в качестве подделки и завершает работу.

8. Оправляет значение e подписывающему и транслирует полученное в ответ значение s алгоритму User.

9. Получает подпись (r', s') от алгоритма User. Это подпись должна быть корректной для сообщения m и открытого ключа Q, поэтому $s' \neq s^*$.

10. Если условие (2.6) не выполняется, восстанавливает ключ подписи d, используя алгоритм, описанный далее. После этого он вычисляет корректную подпись (r'_1, s'_1) для произвольного сообщения $m_1 \neq m$, используя знание ключа d, и возвращает две пары $(m, (r', s'))$ и $(m_1, (r'_1, s'_1))$. Если уравнение (2.6) обратилось в верное равенство, нарушитель $\mathcal{A}$ возвращает символ ошибки и завершает работу.

Схематично эта атака отражена на рисунке 2.6.

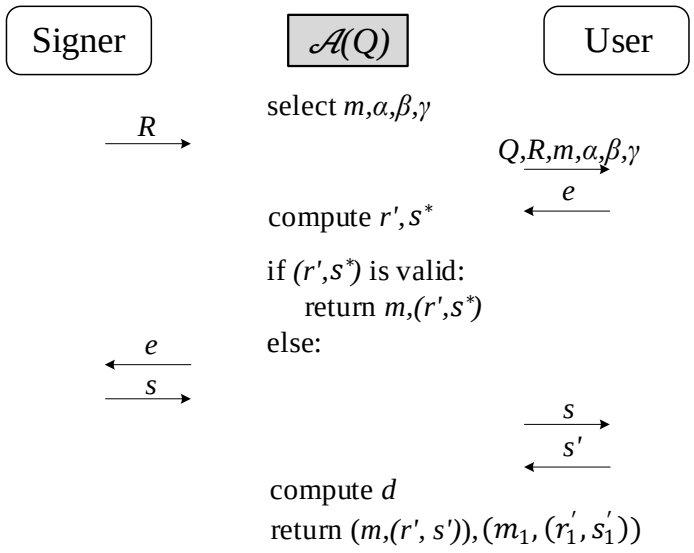

Рис. 2.6. Атака на схему GenEG-BS типа II

Если нарушитель $\mathcal{A}$ завершил работу на шаге 7, он совершил 0 успешных взаимодействий с подписывающим и выдал 1 подделку. В противном случае нарушитель $\mathcal{A}$ завершил 1 успешное взаимодействие с подписывающим и выдал 2 подделки, если условие (2.6) не выполняется. Согласно условию теоремы 2.4.3, вероятность выполнения условия (2.6) не превосходит $\frac{1}{q}$. Таким образом, нарушитель $\mathcal{A}$ предъявляет подделку с вероятностью $1 - \frac{1}{q}$.

Вычисление d**.** Рассмотрим случай, когда схема GenEG-BS типа II основана на уравнении (2.3), случай уравнения (2.4) рассматривается аналогично.

Имея корректную подпись (r', s') для сообщения m с хэш-значением e' и стенограмму протокола (R, e, s), нарушитель $\mathcal{A}$ может составить следующую систему уравнений с неизвестными k и d:

$$\begin{cases} sk = F_1(r, e)d + F_2(r, e), \\ s'(\alpha k + \beta d + \gamma) = F_1(r', e')d + F_2(r', e'), \end{cases} \tag{2.7}$$

где $r = R.x \bmod q$. Первое уравнение следует из алгоритма вычисления значения s согласно уравнению (2.3). Второе уравнение следует из того, что $r' = R'.x \bmod q = (\alpha R + \beta Q + \gamma P).x \bmod q$ и подпись (r', s') является корректной, т.е. $s'R' = F_1(r', e')Q + F_2(r', e')P$.

Согласно построению схемы, система (2.7) должна иметь решение относительное переменных k и d. Согласно теореме Кронекера-Капелли [82], система имеет решение тогда и только тогда когда ранг расширенной матрицы A равен рангу присоединенной матрицы A', т.е. $\operatorname{rank}(A) = \operatorname{rank}(A')$. Выпишем эти матрицы для системы (2.7):

$$A = \begin{pmatrix} s & -F_1(r, e) \\ s'\alpha & s'\beta - F_1(r', e') \end{pmatrix},$$

$$A' = \begin{pmatrix} s & -F_1(r, e) & F_2(r, e) \\ s'\alpha & s'\beta - F_1(r', e') & F_2(r', e') - s'\gamma \end{pmatrix}.$$

Пусть $\operatorname{rank}(A) = \operatorname{rank}(A') = t$. Далее покажем, что t принимает максимально возможное значение, т.е. $t = 2$. Тогда система имеет единственное решение, а значит, нарушитель $\mathcal{A}$ восстанавливает ключ подписи d, решая эту систему.

Предположим обратное. Пусть $t \leqslant 1$. Тогда любые два столбца матрицы A', в частности, второй и третий столбцы, являются линейно зависимыми. Это означает, что определитель квадратной матрицы, составленной из этих столбцов, равен нулю. Выпишем это условие:

$$\begin{aligned} 0 = \begin{vmatrix} -F_1(r, e) & F_2(r, e) \\ s'\beta - F_1(r', e') & F_2(r', e') - s'\gamma \end{vmatrix} = \\ = F_1(r, e)(s'\gamma - F_2(r', e')) - (s'\beta - F_1(r', e'))F_2(r, e) = \\ = s' \left(\gamma F_1(r, e) - \beta F_2(r, e)\right) - \left(F_1(r, e)F_2(r', e') - F_2(r, e)F_1(r', e')\right). \end{aligned}$$

Поскольку условие (2.6) не выполняется, $F_1(r, e)F_2(r', e') - F_2(r, e)F_1(r', e') \neq 0$. Тогда если $\gamma F_1(r, e) - \beta F_2(r, e) = 0$, определитель не может быть равен нулю и приходим к противоречию, откуда $t = 2$. Пусть $\gamma F_1(r, e) - \beta F_2(r, e) \neq 0$, тогда для значения s' должно быть верно

следующее соотношение:

$$s' = (\gamma F_1(r,e) - \beta F_2(r,e))^{-1}(F_1(r,e)F_2(r',e') - F_2(r,e)F_1(r',e')) = s^*.$$

Однако $s' \neq s^*$ согласно построению атаки (см. шаг 9), поэтому приходим к противоречию и $t = 2$. Таким образом, нарушитель $\mathcal{A}$ успешно восстанавливает ключ подписи d, решая систему (2.7). □

Выводы

В настоящей главе впервые проведен системный анализ схем подписи вслепую, построенных на основе уравнения подписи Эль-Гамаля, в расширенных моделях безопасности. Строго определен класс схем GenEG-BS, покрывающий все известные схемы такого типа. В данном классе выделено два типа схем (схемы типа I и схемы типа II) в зависимости от вида базового уравнения подписи и способа выработки первой компоненты подписи. Все известные схемы принадлежат к одному из данных типов.

Удалось доказать, что схемы типа I и схемы типа II не обеспечивают либо свойство неподделываемости, либо свойство неотслеживаемости. Как следствие, все известные существующие схемы не являются стойкими в расширенных моделях безопасности. В результате проведенных исследований данные схемы были исключены из рассмотрения в процессе выбора перспективной для стандартизации в Российской Федерации схемы подписи вслепую.

Кроме того, полученные результаты демонстрируют, что если существует стойкая в расширенных моделях безопасности схема подписи вслепую на основе уравнения Эль-Гамаля, то либо на стороне подписывающего выполняется алгоритм отличный от того, который зафиксирован в схемах типа GenEG-BS, либо на стороне пользователя используется принципиально новый подход к вычислению первой компоненты подписи, при этом уравнение подписи обязательно имеет вид (2.3) или (2.4).

Заметим, однако, что схемы подписи вслепую на основе уравнения Эль-Гамаля потенциально могут обеспечивать стойкость в более слабых моделях безопасности, не подразумевающих, например, возможность нарушителя открывать параллельные сеансы протокола формирования подписи.

Глава 3

Анализ безопасности схем подписи и схем подписи вслепую на основе уравнения Эль-Гамаля в специализированных моделях безопасности

В настоящей главе рассматриваются специализированные модели безопасности для схем подписи и схем подписи вслепую, релевантные в одном классе прикладных информационных систем, предназначенных для формирования подписи. Для начала определим данный класс систем.

Рассмотрим информационную систему, состоящую из двух компонентов: смарт-карты (или токена), используемой в качестве функционального ключевого носителя, и приложения, установленного на пользовательском устройстве (стационарном или мобильном). Прикладной функцией системы является создание подписи для любого документа, передаваемого через приложение, с помощью ключа подписи, хранящегося на смарт-карте. Компоненты обычно взаимодействуют следующим образом.

1. Пользователь открывает приложение, выбирает документ для подписания и нажимает кнопку «Подписать».

2. Приложение соединяется со смарт-картой (обычно путем установления защищенного паролем канала, подробнее см. [5]) и отправляет ей выбранный для подписи документ или его хэш-значение.

3. Смарт-карта самостоятельно вычисляет значение подписи для документа с помощью хранящегося на ней секретного ключа подписи и возвращает вычисленное значение приложению.

4. Приложение проверяет полученное значение подписи и возвращает подписанный документ пользователю.

Использование функциональных ключевых носителей с неизвлекаемыми ключами считается одним из наиболее безопасных подходов к управлению ключами. Он позволяет защититься от нарушителей, которые могут получить физический доступ к смарт-картам [7]. Однако у него есть и свои недостатки. В отличие от программного обеспечения, исходный код которого может быть открытым (а значит, любой наблюдатель может полностью проверить

его, самостоятельно скомпилировать и запустить), разработка смарт-карт представляет собой гораздо более сложный с технической точки зрения процесс, который обычно реализуется конкретными компаниями, специализирующимися в этой области. Действительно, алгоритм подписи часто реализуется непосредственно в микрочипах смарт-карт для повышения производительности и, следовательно, не может быть верифицирован внешними наблюдателями — пользователям предоставляется готовый к использованию «черный ящик». Это дает возможность недобросовестным разработчикам реализовать алгоритм подписи некорректным, в том числе уязвимым образом.

Для систем на основе схем подписи Эль-Гамаля вопросы безопасности, возникающие, когда используемая смарт-карта считается потенциально уязвимой, являются критически важными. Эти схемы используют для генерации подписи одноразовые случайные значения, которые генерируются с помощью смарт-карты, и компрометация которых немедленно приводит к восстановлению ключа подписи. Например, недоверенная смарт-карта может использовать низкоэнтропийные одноразовые значения, позволяющие нарушителю провести атаку методом перебора и восстановить ключ пользователя, используя одно корректное значение подписи.

Предшествующие работы. Этой проблеме посвящена работа [4]. В ней вводятся следующие две модели безопасности.

Внешний нарушитель: моделирует «честного, но любопытного» нарушителя, действующего на стороне приложения; цель нарушителя — создать новую корректную пару (сообщение, подпись) без взаимодействия со смарт-картой или, другими словами, сделать подделку. Заметим, что такая угроза покрывает и более сильную угрозу — восстановление ключа. Рассмотрение таких нарушителей соответствует сценарию, когда только честный пользователь взаимодействует со смарт-картой через доверенное приложение, но это приложение менее защищено от утечки памяти по сравнению со смарт-картой.

Замечание 3.0.1. *Заметим, что нарушитель такого типа не обладает возможностями активного нарушителя, который может напрямую взаимодействовать (например, с помощью собственного приложения) со смарт-картой. На практике это означает, что нарушитель, укравший смарт-карту, не может получить доступ к ее API. Рассмотрение только пассивных нарушителей обусловлено тем, что доступ к API смарт-карты, как правило, дополнительно защищен запоминаемым паролем (см. [84]).*

Нарушитель с агентом: этот нарушитель является составным. Первая часть представляет собой активного нарушителя на стороне смарт-карты, который может взаимодействовать только с доверенным приложением, т.е. отсутствуют другие каналы передачи данных от смарт-карты. Вторая часть представляет собой агента, который накапливает пары (сообщение, подпись), вычисленные приложением и недоверенной смарт-картой. Как и для первого типа нарушителей, задача нарушителя состоит в построении подделки.

Определим следующие два вида нарушителей с агентом:

- *слабый*: нарушитель на стороне смарт-карты корректным образом реализует целевой алгоритм формирования подписи, но использует для этого недоверенный датчик случайных чисел;
- *сильный*: нарушитель на стороне смарт-карты формирует значение подписи произвольным образом.

Для защиты от нарушителей такого типа в работе [4] предлагается решение для схемы подписи Эль-Гамаля, основанное на использовании интерактивного протокола доказательства с нулевым разглашением Шнорра. Этот протокол выполняется одновременно с основным алгоритмом подписи, его цель — доказать приложению, что смарт-карта использует для генерации подписи высокоэнтропийное одноразовое значение (подробнее см. [4]).

Это решение имеет два существенных недостатка. Во-первых, оно позволяет защититься только от частично доверенных смарт-карт: критически важным вопросом для безопасности является корректная реализация низкоуровневых арифметических операций в смарт-картах. Хотя это и реалистичное предположение, на практике отсутствуют удобные способы проверить его. Во-вторых, решение не обеспечивает защиту, если смарт-карта может спровоцировать завершение процесса подписания с ошибкой на стороне приложения. В работе [4] описывается конкретная атака, в которой недоверенная смарт-карта успешно завершает протокол подписи только в том случае, если определенные биты значения подписи равны определенным битам ключа подписи. Одним из подходов к защите от атак такого типа является удаление ключа подписи сразу после возникновения подобных ошибок. Однако на практике ошибки могут возникать не только из-за действий нарушителя, но и в результате технических сбоев, поэтому удаление ключа подписи после каждой ошибки не является практичным решением.

В настоящей главе проводится анализ двух новых методов обеспечения защиты данных систем. Первый метод (см. раздел 3.1) позволяет обеспечить защиту от внешнего нарушите-

ля и слабого нарушителя с агентом и основан на модификации классической схемы подписи Эль-Гамаля. Второй метод (см. раздел 3.2) позволяет обеспечить защиту от внешнего нарушителя и сильного нарушителя с агентом и предполагает подписание сообщений в результате выполнения интерактивного протокола формирования подписи вслепую вместо классической схемы подписи.

3.1. Анализ модифицированной схемы подписи Эль-Гамаля в специализированной модели безопасности

В рассматриваемом классе систем одноразовые секретные значения, используемые при формировании подписи, генерируются на стороне потенциально уязвимой смарт-карты, а потому в общем случае могут быть низкоэнтропийными. Одним из способов защиты от потенциально уязвимых смарт-карт является использование в целевых системах классической схемы подписи, обеспечивающей свойство неподделываемости даже в случае использования недоверенных источников случайности. Кроме того, для повышения производительности данных систем актуальна задача сокращения длины формируемой подписи.

В настоящем разделе разработан метод модификации схем подписи Эль-Гамаля с использованием идей из работ [3, 16] с целью достижения данных свойств. Предлагаемая модификация описана на примере схемы подписи ГОСТ Р 34.10-2012 [1] с использованием эллиптических кривых, определенных в документе Р 1323565.1.024-2019 [2] (далее — схема GOST). Длина подписи в результирующей схеме на четверть меньше, чем в классической схеме GOST. Алгоритм формирования подписи не является детерминированным при повторе случайных значений за счет добавления метки времени в процесс выработки одноразового значения (см. [3, раздел 7]). Для модифицированной схемы получена оценка стойкости в специализированной модели безопасности SUF-CMRA, учитывающей возможности внешнего нарушителя и слабого нарушителя с агентом для целевого класса прикладных систем.

3.1.1. Короткая усиленная подпись на основе схемы GOST

Определим вспомогательный параметр ℓ равным $\lceil \log q \rceil$. Пусть $\mathsf{HMAC} : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^\ell$ — ключевая хэш-функция, определенная в [57], отображающая ключ K длины κ бит и строку T произвольной длины в двоичный вектор длины ℓ бит. Будем считать, что размер ключа κ может принимать любые значения от 256 до 512 бит. Будем считать, что хэш-функция H_1, используемая в схеме подписи, действует из $\{0,1\}^*$ в $\mathbb{Z}_q$.

Схема подписи GOST задается алгоритмами генерации ключей, формирования и провер-

ки подписи. Приведем их описание в виде псевдокода:

$\mathsf{KGen}(\,)$	$\mathsf{Sig}(d, m)$	$\mathsf{Verify}(Q, m, (r, s))$
$d \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$	$k \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$	**if** $(s = 0) \vee (r = 0)$: **return** 0
$Q \leftarrow dP$	$e \leftarrow \mathsf{H}_1(m)$	$e \leftarrow \mathsf{H}_1(m)$
return (d, Q)	**if** $e = 0 : e \leftarrow 1$	**if** $e = 0 : e \leftarrow 1$
	$R \leftarrow kP$	$R \leftarrow e^{-1}sP - e^{-1}rQ$
	$r \leftarrow R.x \bmod q$	**if** $R.x \bmod q \neq r$:
	if $r = 0$: **return** $\perp$	**return** 0
	$s \leftarrow ke + dr$	**return** 1
	if $s = 0$: **return** $\perp$	
	return (r, s)	

Длина подписи в схеме GOST составляет $2\lceil \log_2 q \rceil$ бит. Определим модифицированную схему подписи, значение подписи в которой имеет длину $\left(\frac{3}{2}\lceil \log q \rceil + 1\right)$ бит и стойкость которой обеспечивается даже в случае вырождения случайности в рамках алгоритма формирования подписи.

Для укорочения длины значения подписи применим метод 1 из работы [16], в результате получим схему $\mathsf{GOST\text{-}H}$. Настоящий метод заключается в изменении способа вычисления первой компоненты подписи r из эфемерной точки R. Если в оригинальной схеме GOST значение r полагается равным $R.x \bmod q$, то в модифицированной схеме оно вычисляется следующим образом:

$$r = \phi(\mathsf{H}_2(R.x)),$$

где H_2 отображает $\mathbb{Z}_p$ в $\{0,1\}^{\ell/2}$, и ϕ отображает $\{0,1\}^{\ell/2}$ в $\mathbb{Z}_q^*$. Функция ϕ определяется следующим образом: она ставит в соответствие значению $x \in \{0,1\}^{\ell/2} \setminus \{0\}$ значение $\mathrm{int}(x)$ и переводит 0 в $2^{\ell/2}$. Таким образом, длина компоненты r подписи не превосходит $\ell/2 + 1 = \lceil \log q \rceil/2 + 1$ бит. Отметим, что в силу задания области значений функции ϕ значение r никогда не может быть нулевым, а потому из алгоритмов формирования и проверки подписи можно исключить проверку на равенство r нулю.

Заметим, что метод, предложенный в [16], является более общим и позволяет генерировать подписи, первая компонента r которых имеет длину b бит, $b < \lceil \log q \rceil$, где b — настраиваемый параметр. Однако в настоящем разделе фиксируется значение $b = \ell/2$ из следующих соображений. Оценка стойкости схемы $\mathsf{GOST\text{-}H}$, полученная в работе [16], свидетельствует о том, что при уменьшении значения b с ℓ до $\ell/2$ оценка стойкости схемы в модели SUF-CMA не изменяется, в то время как при дальнейшем уменьшении параметра оценка стойкости в мо-

дели SUF-CMA начинает ухудшаться. Таким образом, значение $b = \ell/2$ позволяет сохранить стойкость схемы в модели SUF-CMA, улучшив ее эксплуатационные характеристики.

Согласно работе [16] схема подписи GOST-H также является схемой подписи Эль-Гамаля, а потому для этой схемы критично использование доверенного источника случайности (предположение о том, что эфемерный ключ подписи k выбирается в соответствии с равновероятным распределением). Действительно, если нарушитель вместе со значением подписи (r, s) получает также значение k, он может восстановить ключ подписи d из уравнения подписи. Применим к схеме GOST-H метод из работы [3], обеспечивающий защиту от вырождения случайности. Настоящий метод заключается в изменении способа выработки эфемерного ключа подписи k. В оригинальной схеме GOST и схеме GOST-H предполагается, что значение k выбирается из множества $\mathbb{Z}_q^*$ в соответствии с равновероятным распределением. В модифицированной схеме значение k вырабатывается из ключа подписи d и хэш-значения сообщения e в результате следующей последовательности действий:

$$K \leftarrow \mathsf{HMAC}(0^{256}, \mathrm{str}_\ell(d))$$
$$k' \xleftarrow{\mathcal{U}} \{0,1\}^\ell$$
$$k'' \leftarrow \mathsf{HMAC}(K, \mathrm{str}_\ell(e) \parallel k' \parallel time)$$
$$k \leftarrow \mathrm{int}(k'') \bmod q$$

где $time$ — текущее значение времени (в мс), прошедшее с полуночи 01.01.1970 UTC+0, представленное в виде битовой строки длины ℓ бит. Заметим, что в [3] настоящий метод определен для случая, когда ключ подписи хранится в маскированном виде, однако наличие масок существенно зависит от конкретной реализации и, вообще говоря, не является обязательным. Поэтому в настоящем разделе определяется модифицированная схема в случае отсутствия масок. Кроме того, используется идея из [3, раздел 7]: в эфемерный ключ подписи замешивается текущее значение времени $time$.

Результирующую схему будем называть $\widetilde{\mathsf{GOST\text{-}H}}$. На рисунке 3.1 представлено ее описание в виде псевдокода.

Заметим, что методы из работ [3, 16] являются общими и применимы ко всем схемам подписи Эль-Гамаля.

3.1.2. Свойства безопасности

В данном разделе для схемы подписи определим специализированную модель безопасности SUF-CMRA, учитывающую возможности внешнего нарушителя и слабого нарушителя с агентом. Кроме того, определим вспомогательные модели безопасности для функции хэширования и функции HMAC и задачу дискретного логарифмирования.

$\mathsf{KGen}(\,)$	$\mathsf{Sig}(d, m)$	$\mathsf{Verify}(Q, m, (r, s))$
$d \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$	$k' \xleftarrow{\mathcal{U}} \{0,1\}^\ell$	**if** $s = 0$: **return** 0
$Q \leftarrow dP$	$e \leftarrow \mathsf{H}_1(m)$	$e \leftarrow \mathsf{H}_1(m)$
return (d, Q)	$K \leftarrow \mathsf{HMAC}(0^{256}, \mathrm{str}_\ell(d))$	**if** $e = 0 : e \leftarrow 1$
	$k'' \leftarrow \mathsf{HMAC}(K, \mathrm{str}_\ell(e) \parallel k' \parallel time)$	$R \leftarrow e^{-1}sP - e^{-1}rQ$
	$k \leftarrow \mathrm{int}(k'') \bmod q$	**if** $\phi(\mathsf{H}_2(R.x)) \neq r$:
	if $k = 0$: **return** $\perp$	**return** 0
	if $e = 0 : e \leftarrow 1$	**return** 1
	$R \leftarrow kP$	
	$r \leftarrow \phi(\mathsf{H}_2(R.x))$	
	$s \leftarrow ke + dr$	
	if $s = 0$: **return** $\perp$	
	return (r, s)	

Рис. 3.1. Схема $\widetilde{\mathsf{GOST}}$-$\mathsf{H}$.

Модель SUF-CMRA**.** Модель SUF-CMRA, предложенная в работе [72] для классической схемы подписи, рассматривает стойкость к угрозе нахождения подделки при атаке не только с выбором сообщений, но и с выбором случайных значений (Chosen Message and Randomness Attack).

Определим модель SUF-CMRA со случайным оракулом для схемы подписи $\widetilde{\mathsf{GOST}}$-$\mathsf{H}$. Единственным случайным значением, использующимся в процессе формирования подписи в схеме $\widetilde{\mathsf{GOST}}$-$\mathsf{H}$, является значение k'. При формализации предполагается, что это значение подается нарушителем на вход оракулу подписи вместе с значением сообщения. Кроме того, настоящая модель учитывает, что источник времени, вообще говоря, может быть недоверенным, а потому предоставляет нарушителю контролировать значение времени $time$, также подавая его на вход оракулу подписи.

В рамках настоящей модели нарушителю также предоставляется доступ к двум случайным оракулам: оракул RO_1 моделирует вычисление значения функции $\mathsf{HMAC}(0^{256}, \cdot)$, оракул RO_2 моделирует вычисление значения хэш-функции H_2. Заметим, что в работах [3, 16] при обосновании стойкости также предполагалось, что соответствующие функции моделируются как случайные оракулы. Релевантность подобных предположений и интерпретация результатов, полученных в модели со случайным оракулом, подробно обсуждается в оригинальных работах.

Определение 3.1.1. Преимущество нарушителя $\mathcal{A}$ для схемы подписи $\widetilde{\mathsf{GOST\text{-}H}}$ в модели SUF-CMRA со случайным оракулом определяется следующим образом:

$$\mathsf{Adv}^{\text{SUF-CMRA}}_{\widetilde{\mathsf{GOST\text{-}H}}}(\mathcal{A}) = \Pr\Big[\mathbf{Exp}^{\text{SUF-CMRA}}_{\widetilde{\mathsf{GOST\text{-}H}}}(\mathcal{A}) \to 1\Big],$$

где эксперимент $\mathbf{Exp}^{\text{SUF-CMRA}}_{\widetilde{\mathsf{GOST\text{-}H}}}(\mathcal{A})$ определяется следующим образом:

$\underline{\mathbf{Exp}^{\text{SUF-CMRA}}_{\widetilde{\mathsf{GOST\text{-}H}}}(\mathcal{A})}$

$\mathcal{F}_1 \xleftarrow{\mathcal{U}} Func(\{0,1\}^{\ell}, \{0,1\}^{\ell})$

$\mathcal{F}_2 \xleftarrow{\mathcal{U}} Func(\mathbb{Z}_p, \{0,1\}^{\ell/2})$

$d \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$

$Q \leftarrow dP$

$\mathcal{L} \leftarrow \emptyset$

$(m, sgn) \leftarrow \mathcal{A}^{Sign, RO_1, RO_2}(Q)$

if $(m, sgn) \in \mathcal{L}$: **return** 0

return $\widetilde{\mathsf{GOST\text{-}H}}.\mathsf{Verify}(Q, m, sgn)$

$\underline{\text{Oracle } RO_1(x)}$

$y \leftarrow \mathcal{F}_1(x)$

$\underline{\text{Oracle } RO_2(x)}$

$y \leftarrow \mathcal{F}_2(x)$

$\underline{\text{Oracle } Sign(m, k', time)}$

$sgn \leftarrow \widetilde{\mathsf{GOST\text{-}H}}.\mathsf{SigDet}(d, m, k', time)$

$\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, sgn)\}$

return sgn

Через $\widetilde{\mathsf{GOST\text{-}H}}.\mathsf{SigDet}$ обозначен алгоритм подписи $\widetilde{\mathsf{GOST\text{-}H}}.\mathsf{Sig}$ без генерации значения k' и с определенным значением $time$ (все строки алгоритма $\widetilde{\mathsf{GOST\text{-}H}}.\mathsf{Sig}$, кроме первой).

Покажем, что данная модель учитывает возможности внешнего нарушителя и слабого нарушителя с агентом. Прежде всего заметим, что угроза (построение подделки) для обоих нарушителей определяется так же, как и в модели SUF-CMRA. Возможности внешнего нарушителя являются более слабыми, чем возможности, предоставляемые нарушителю в модели SUF-CMRA. Действительно, внешний нарушитель имеет возможность лишь накапливать значения подписей для адаптивно выбираемых сообщений, при этом нарушитель в модели SUF-CMRA дополнительно имеет возможность выбирать случайные значения, используемые в процессе формирования подписи. Возможности слабого нарушителя с агентом в худшем случае совпадают с возможностями нарушителя в модели SUF-CMRA. Действительно, в случае, если агент имеет полный контроль над датчиком случайных чисел, используемом на стороне смарт-карты, он может навязывать значение одноразового секрета для каждой формируемой подписи.

Псевдослучайность функции HMAC **(модель** PRF**).** В модели PRF нарушитель имеет возможность для адаптивно выбираемых сообщений получать соответствующие им значения. При этом, в зависимости от бита, выбранного экспериментатором при инициализации эксперимента, эти значения будут либо результатом «честного» вычисления функции HMAC на секретном ключе, либо результатом работы случайной функции. Задачей нарушителя является различение этих двух ситуаций, т.е. определение значения бита с вероятностью, существенно отличной от $\frac{1}{2}$.

Определение 3.1.2. Преимущество нарушителя $\mathcal{A}$ для схемы HMAC в модели PRF определяется следующим образом:

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{HMAC}}(\mathcal{A}) = \left|2\Pr\left[\mathbf{Exp}^{\mathrm{PRF}}_{\mathsf{HMAC}}(\mathcal{A}) \to 1\right] - 1\right|,$$

где эксперимент $\mathbf{Exp}^{\mathrm{PRF}}_{\mathsf{HMAC}}(\mathcal{A})$ определяется следующим образом:

$\underline{\mathbf{Exp}^{\mathrm{PRF}}_{\mathsf{HMAC}}(\mathcal{A})}$	$\underline{\text{Oracle } HMAC(m)}$
$b \xleftarrow{\mathcal{U}} \{0,1\}$	**if** $b = 1$:
if $b = 1$:	**return** $\mathsf{HMAC}(K, m)$
$K \xleftarrow{\mathcal{U}} \{0,1\}^{\kappa}$	**else** :
else :	**return** $\mathcal{F}(m)$
$\mathcal{F} \xleftarrow{\mathcal{U}} Func(\{0,1\}^*, \{0,1\}^{\ell})$	
$b' \leftarrow \mathcal{A}^{HMAC}()$	
return $b = b'$	

Задача дискретного логарифмирования ECDLP**.** Определим математически строго модель ECDLP, задачей нарушителя в которой является нахождение для некоторой случайно выбранной точки Q ее дискретного логарифма по основанию образующей точки P.

Определение 3.1.3. Преимущество нарушителя $\mathcal{A}$ для группы $\mathbb{G}$ в модели ECDLP определяется следующим образом:

$$\mathsf{Adv}^{\mathrm{ECDLP}}_{\mathbb{G}}(\mathcal{A}) = \Pr\left[Q \xleftarrow{\mathcal{U}} \langle P\rangle;\ d \leftarrow \mathcal{A}(Q, P) : dP = Q\right]$$

Свойства хэш-функции. Будем рассматривать ключевые хэш-функции H_1: $\{0,1\}^* \to \mathbb{Z}_q$, неявно инициализированные соответствующим вектором; предполагается, что эксперименты в определениях моделей начинаются со случайного выбора вектора $IV \in \mathcal{IV}$ и передачи его нарушителю.

Определим для семейства хэш-функций H_1 свойство устойчивости к поиску коллизий с точностью до знака (свойство SCR) и свойство устойчивости к поиску делителей с точностью до знака (свойство SDR).

Определение 3.1.4 (свойство SCR)**.** Преимущество нарушителя $\mathcal{A}$ для семейства хэш-функций H_1 в модели SCR определяется следующим образом:

$$\mathsf{Adv}^{\mathrm{SCR}}_{\mathsf{H}_1}(\mathcal{A}) = \Pr[(m_1, m_2) \leftarrow \mathcal{A} : \mathsf{H}_1(m_1) = \pm\mathsf{H}_1(m_2) \wedge m_1 \neq m_2].$$

Определение 3.1.5 (свойство SDR)**.** Преимущество нарушителя $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ для семейства хэш-функций H_1 в модели SDR определяется следующим образом:

$$\mathsf{Adv}^{\mathrm{SDR}}_{\mathsf{H}_1}(\mathcal{A}) = \Pr\left[\beta_1, \beta_2 \xleftarrow{\mathcal{U}} \{0,1\}^b; (m_1, \Gamma) \leftarrow \mathcal{A}_1(\beta_1), m_2 \leftarrow \mathcal{A}_2(\Gamma, \beta_2) : \frac{H_1(m_1)}{\phi(\beta_1)} = \pm\frac{H_1(m_2)}{\phi(\beta_2)}\right].$$

Напомним, что функция ϕ ставит в соответствие значению $x \in \{0,1\}^{\ell/2} \setminus \{0\}$ значение $\mathrm{int}(x)$ и переводит 0 в $2^{\ell/2}$.

3.1.3. Анализ безопасности в модели SUF-CMRA

Получим оценку стойкости схемы $\widetilde{\mathsf{GOST}}\text{-}\mathsf{H}$ в модели SUF-CMRA, опираясь на результаты работ [3, 16].

Теорема 3.1.1. *Пусть $\mathcal{A}$ — нарушитель с вычислительными ресурсами $T_{\mathcal{A}}$ в модели* SUF-CMRA *для схемы $\widetilde{\mathsf{GOST}}\text{-}\mathsf{H}$, делающий не более Q_S запросов к оракулу Sign, $Q_{O,1}$ и $Q_{O,2}$ запросов к случайным оракулам RO_1 и RO_2 соответственно. Тогда существуют*

- *нарушитель $\mathcal{D}$, решающий задачу* ECDLP *в используемой группе точек $\mathbb{G}$,*
- *нарушитель $\mathcal{C}$, решающий задачу* SCR *(поиска коллизий с точностью до знака) для хэш-функции H_1,*
- *нарушитель $\mathcal{M}$, решающий задачу* SDR *(поиска делителей с точностью до знака) для хэш-функции H_1,*
- *нарушитель $\mathcal{P}$ для схемы HMAC в модели* PRF*, делающий не более Q_S запросов к оракулу, длина каждого из которых не превосходит 3ℓ битов,*

такие, что:

$$\mathsf{Adv}^{\text{SUF-CMRA}}_{\widetilde{\mathsf{GOST}}\text{-}\mathsf{H}}(\mathcal{A}) \leqslant \frac{Q_{O,2}+3}{\sqrt{q}} + \mathsf{Adv}^{\mathrm{SCR}}_{\mathsf{H}_1}(\mathcal{C}) + \sqrt{(Q_{O,2}+2)\left(\mathsf{Adv}^{\mathrm{SDR}}_{\mathsf{H}_1}(\mathcal{M})\cdot(Q_{O,2}+2) + \mathsf{Adv}^{\mathrm{ECDLP}}_{\mathbb{G}}(\mathcal{D})\right)} + \frac{(2Q_{O,2}+Q_S+1)Q_S}{q-1} + Q_S\cdot\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{HMAC}}(\mathcal{P}).$$

Для вычислительных ресурсов нарушителей $\mathcal{C}, \mathcal{D}, \mathcal{M}$ и $\mathcal{P}$ верно следующее:

$$T_{\mathcal{C}} \leqslant T_{\mathcal{A}} + c(Q_{O,1} + Q_{O,2} + 2Q_S + T_{\mathsf{Sig}} + (Q_S + 3)T_{\mathsf{Verify}}),$$
$$T_{\mathcal{D}}, T_{\mathcal{M}} \leqslant 2T_{\mathcal{A}} + 2c(Q_{O,1} + 2Q_S + T_{\mathsf{Sig}} + (Q_S + 4)T_{\mathsf{Verify}} + 2Q_{O,2} + 4),$$
$$T_{\mathcal{P}} \leqslant T_{\mathcal{A}} + c(Q_{O,1} + T_{\mathsf{Sig}} + Q_S(\ell + 1 + T_{\mathsf{Sig}} + 3\ell \log Q_S + T_{\mathsf{HMAC}})),$$

где $T_{\mathsf{Sig}}, T_{\mathsf{Verify}}$ — вычислительные ресурсы, необходимые для подписи одного сообщения и проверки подписи для схемы $\widetilde{\mathsf{GOST\text{-}H}}$, T_{HMAC} — вычислительные ресурсы, необходимые для вычисления функции HMAC на входе длины 3ℓ, c — константа, зависящая только от модели вычислений и способа представления данных.

Доказательство. Настоящая оценка следует из следующих двух оценок, применяемых последовательно.

Оценка в модели SUF-CMRA для метода усиления случайности (Теорема 1 из работы [3] для схемы подписи $\mathsf{SS} = \mathsf{GOST\text{-}H}$):

$$\mathsf{Adv}^{\text{SUF-CMRA}}_{\widetilde{\mathsf{GOST\text{-}H}}}(\mathcal{A}) \leqslant \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{GOST\text{-}H}}(\mathcal{B}) + Q_S \cdot \mathsf{Adv}^{\text{PRF}}_{\mathsf{HMAC}}(\mathcal{P}).$$

Заметим, что настоящая оценка остается верной и в случае, когда маскирование ключа подписи не производится. Действительно, в этом случае можно считать, что рассматривается частный случай нарушителя, а именно, нарушитель, который всегда подает на вход оракулу подписи значение $mask = 1$. Таким образом, оценка остается верной.

Также заметим, что ограничение на длину запросов нарушителя $\mathcal{P}$ поменялось по сравнению с оценкой из работы [3]. Это связано с добавлением времени $time$ в аргументы функции HMAC при выработке значения k''. Поскольку на вход функции HMAC приходят значения $\text{str}_\ell(e), k', time$, каждое из которых имеет длину ℓ бит, длина запросов нарушителя $\mathcal{P}$ составляет 3ℓ бит. При этом, все остальное доказательство остается неизменным с добавлением $time$ и в точности повторяет доказательство [3].

Оценка в модели SUF-CMA для метода уменьшения длины подписи (Теорема 1 из работы [16]):

$$\mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{GOST\text{-}H}}(\mathcal{B}) \leqslant \frac{Q_O + 3}{2^b} + \mathsf{Adv}^{\text{SCR}}_{\mathsf{H}_1}(\mathcal{C}) +$$
$$\sqrt{(Q_O + 2)\left(\mathsf{Adv}^{\text{SDR}}_{\mathsf{H}_1}(\mathcal{M}) \cdot (Q_O + 2) + \mathsf{Adv}^{\text{ECDLP}}_{\mathbb{G}}(\mathcal{D})\right)} + \frac{(2Q_O + Q_S + 1)Q_S}{q - 1}.$$

Для схемы $\widetilde{\mathsf{GOST\text{-}H}}$, как было указано ранее, в эту оценку подставляется значение $b = \ell/2$.

Заметим, что наличие двух случайных оракулов не влияет на возможность «совмещения» приведенных оценок, поскольку при построении сведений нарушители $\mathcal{C}, \mathcal{D}$ и $\mathcal{M}$ имеют возможность честно моделировать оракул RO_1, адаптивно выбирая значения выходов случайной функции $\mathcal{F}_1$ на соответствующих входах, а $\mathcal{P}$ — честно моделировать оракул RO_2 аналогичным образом.

Вычислительные ресурсы нарушителей также определяются путем совмещения соответствующих оценок из работ [3, 16].

□

Аналогичная теорема может быть сформулирована для модифицированных схем подписи, основанных на других схемах подписи Эль-Гамаля. Этап а) доказательства в таком случае проводится идентично, поскольку оценка из работы [3] применима к произвольной схеме подписи Эль-Гамаля. При этом этап б) доказательства требует оценки стойкости укороченной схемы подписи в модели SUF-CMA, которая, в свою очередь, может быть получена с применением идей из работы [37] (в работе [16] настоящая оценка явно приводится только для схемы подписи GOST-H).

3.2. Анализ схем подписи вслепую на основе уравнения Эль-Гамаля в специализированных моделях безопасности

В настоящем разделе предлагается новый подход к обеспечению защиты целевых прикладных систем от внешнего нарушителя и сильного нарушителя с агентом, основная идея которого заключается в использовании схем подписи вслепую, построенных на основе классических схем подписи. Предлагается в качестве подписывающей стороны использовать смарт-карту, в качестве запрашивающей — приложение. В силу свойства неотслеживаемости смарт-карта в результате выполнения протокола формирования подписи не получает информацию о сформированном значении подписи и, следовательно, не может контролировать эти значения, например, передавая через них биты секретного ключа подписи.

Вводятся два специализированных свойства безопасности для схем подписи вслепую: свойство неподделываемости относительно «честного, но любопытного» нарушителя и свойство неподделываемости относительно сервера с агентом, которые характеризуют стойкость предложенного решения относительно внешнего нарушителя и сильного нарушителя с агентом. Анализируется связь данных свойств безопасности с известными в литературе моделями безопасности, учитывающими угроза нарушения свойств неподделываемости и неотслеживаемости.

Доказывается, что для любой схемы подписи вслепую из обеспечения схемой свойства неотслеживаемости при условии честной генерации ключей и свойства неподделываемости относительно внешнего нарушителя следует обеспечение неподделываемости относительно сервера с агентом. Для частного случая схем подписи вслепую на основе уравнения Эль-Гамаля доказывается, что данные схемы обеспечивают защиту от «честного, но любопытного» нарушителя, если соответствующая классическая схема подписи Эль-Гамаля обеспечивает свойство неподделываемости. Более того, что для систем на основе схемы подписи GOST предлагается использовать конкретную схему подписи вслепую — схему Камениша [29], которая обеспечивает совершенную неотслеживаемость (а потому неотслеживаемость при условии честной генерации ключей) и неподделываемость относительно «честного, но любопытного» нарушителя (а потому неподделываемость относительно внешнего нарушителя), которая, в свою очередь, обеспечивается только за счет неподделываемости схемы GOST. Это означает, что схема подписи вслепую Камениша обеспечивает защиту как от внешнего нарушителя, так и от сильного нарушителя с агентом при единственном предположении, что схема подписи GOST является стойкой, т.е. для нее невозможно построить подделку при атаке с выбором сообщений.

Слепая модификация схемы подписи. В настоящем разделе рассматриваются схемы подписи вслепую, построенные на основе классических схем подписи. Будем говорить, что схема подписи вслепую BS является слепой модификацией классической схемы подписи SS, если их алгоритмы генерации ключей KGen и проверки подписи Verify совпадают, и для любых $(\mathsf{sk}, \mathsf{pk})$, сообщения m и подписи σ

$$\Pr[(1, \sigma) \leftarrow \langle \mathsf{BS.Sign}(\mathsf{sk}, \mathsf{pk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle] = \Pr[\sigma \leftarrow \mathsf{SS.Sig}(\mathsf{sk}, m)],$$

где вероятность определяется выбором случайных значений, используемых в протоколе формирования подписи схемы BS и алгоритме подписи схемы SS.

Несложно заметить, что из SUF-CMA-стойкости классической схемы SS следует SUF-CMA-стойкость ее слепой модификации BS, и наоборот.

3.2.1. Специализированные модели безопасности для схем подписи вслепую

В настоящем разделе вводятся математически строгие определения двух моделей безопасности: неподделываемости относительно сервера с агентом и неподделываемости относительно «честного, но любопытного» нарушителя.

Неподделываемость относительно сервера с агентом/Стойкость относительно сильного нарушителя с агентом. Рассмотрим нарушителя $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, состоящего из двух алгоритмов. Алгоритм $\mathcal{A}_1$ обозначает активного нарушителя, действующего на стороне недоверенной смарт-карты. Алгоритм $\mathcal{A}_2$ обозначает агента, который накапливает значения подписей для адаптивно выбираемых сообщений.

Формальное определение модели SA-UF (Server with Agent UnForgeability) для схем подписи вслепую приведено далее (см. Определение 3.2.1). Настоящая модель параметризована значением k, определяющим количество попыток, доступных экспериментатору для формирования корректной подписи для одного сообщения (см. детали ниже).

Определение 3.2.1. Для нарушителя $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ и схемы подписи вслепую BS:

$$\mathsf{Adv}^{\text{SA-UF}_k}_{\mathsf{BS}}(\mathcal{A}) = \Pr\Big[\mathbf{Exp}^{\text{SA-UF}_k}_{\mathsf{BS}}(\mathcal{A}) \to 1\Big],$$

где эксперимент $\mathbf{Exp}^{\text{SA-UF}_k}_{\mathsf{BS}}(\mathcal{A}),\ k \in \mathbb{N}$, определяется следующим образом:

$\mathbf{Exp}^{\text{SA-UF}_k}_{\mathsf{BS}}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $\mathsf{lost} \leftarrow \mathsf{false}$
4 : $st \leftarrow \mathcal{A}_1(\mathsf{sk}, \mathsf{pk})$
5 : $(m, \sigma) \leftarrow \mathcal{A}_2^{Sign}(\mathsf{pk})$
6 : **if** $((m, \sigma) \in \mathcal{L}) \vee (\mathsf{lost} = \mathsf{true})$:
7 : **return** 0
8 : **return** $\mathsf{BS.Verify}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

1 : $i \leftarrow 0$
2 : **do**
3 : $(st, \sigma) \leftarrow \langle \mathcal{A}_1(st), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
4 : $i \leftarrow i + 1$
5 : **until** $(i \geqslant k) \vee (\sigma \neq \perp)$
6 : **if** $\sigma = \perp$:
7 : $\mathsf{lost} \leftarrow \mathsf{true}$
8 : **return** $\perp$
9 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
10 : **return** σ

На этапе инициализации эксперимента (строка 1) экспериментатор, моделирующий работу честного приложения, генерирует ключевую пару $(\mathsf{sk}, \mathsf{pk})$ с помощью алгоритма генерации ключей. Далее он подает оба ключа на вход нарушителю $\mathcal{A}_1$ (строка 4) и только ключ проверки подписи pk нарушителю $\mathcal{A}_2$ (строка 5). Этот этап моделирует доверенный процесс генерации ключей, выдачи сертификата и загрузки ключевого материала на смарт-карту.

Нарушитель $\mathcal{A}_2$ может делать запросы к оракулу подписи $Sign$, возвращающему в ответ значения подписи σ для сообщений m, адаптивно выбираемых нарушителем. Каждое значение подписи вычисляется в результате выполнения протокола формирования подписи между оракулом, моделирующим работу честного пользователя, и нарушителем $\mathcal{A}_1$, моделирующим

работу нечестной смарт-карты (строка 3 в оракуле подписи). Здесь переменная st обозначает внутреннее состояние нарушителя $\mathcal{A}_1$, сохраняемое от вызова к вызову.

Нарушитель $\mathcal{A}_1$ может провоцировать завершение протокола формирования подписи с ошибкой $\perp$ на стороне пользователя (строка 5 в оракуле подписи). В связи с этим для каждого сообщения m оракул делает k попыток создать корректную подпись. Если все k попыток не успешны, экспериментатор возвращает 0 в качестве результата эксперимента (это означает, что нарушитель не решил задачу успешно, см. строку 7 в оракуле подписи). Это моделирует сценарий, когда в работе смарт-карты произошел сбой и она больше не используется.

Замечание 3.2.1. *Заметим, что если агент $\mathcal{A}_2$ может получать символ ошибки от оракула подписи, то всегда существует тривиальная атака на схему. Рассмотрим алгоритм $\mathcal{A}_1$, который успешно завершает протокол формирования подписи тогда и только тогда, когда i-й бит ключа sk равен 1, где i — порядковый номер запроса к оракулу подписи. Имея такого нарушителя на стороне смарт-карты, агент $\mathcal{A}_2$ может восстановить все биты ключа подписи и сформировать подделку тривиальным образом.*

Для успешного решения задачи в модели SA-UF$_k$ нарушитель $\mathcal{A}_2$ должен сформировать подделку (m, σ), подпись σ в которой ранее не возвращалась в качестве ответа оракула $Sign$ на запрос m.

Неподделываемость относительно «честного, но любопытного» нарушителя/ Стойкость относительно внешнего нарушителя. Эта модель рассматривает «честного, но любопытного» нарушителя, действующего на стороне пользователя. Этот нарушитель может адаптивно выбирать сообщения для подписи, делая запросы m к оракулу подписи, и получать для них значение подписи σ и специальное значение $view$. Это значение содержит все входящие в рамках протокола формирования подписи сообщения, а также все случайные значения, выбранные пользователем в процессе выполнения протокола. Это моделирует сценарий, когда нарушитель получает доступ к памяти доверенного приложения.

Далее приведем формальное определение модели HBC-UF.

Определение 3.2.2. Для нарушителя $\mathcal{A}$ и схемы подписи вслепую BS:

$$\mathsf{Adv}_{\mathsf{BS}}^{\text{HBC-UF}}(\mathcal{A}) = \Pr\big[\mathbf{Exp}_{\mathsf{BS}}^{\text{HBC-UF}}(\mathcal{A}) \to 1\big],$$

где эксперимент $\mathbf{Exp}_{\mathsf{BS}}^{\text{HBC-UF}}(\mathcal{A})$ определяется следующим образом:

$\mathbf{Exp}^{\text{HBC-UF}}_{\mathsf{BS}}(\mathcal{A})$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $(m, \sigma) \leftarrow \mathcal{A}^{Sign}(\mathsf{pk})$
4 : **if** $(m, \sigma) \in \mathcal{L}$: **return** 0
5 : **return** $\mathsf{BS.Verify}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

1 : $(1, (\sigma; view)) \leftarrow \langle \mathsf{BS.Sign}(\mathsf{sk}, \mathsf{pk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
2 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
3 : **return** $\sigma, view$

Легко видеть, что для любой схемы подписи вслепую из стойкости в модели HBC-UF следует стойкость в модели SUF-CMA.

3.2.2. Анализ безопасности относительно сильного нарушителя с агентом

В настоящем разделе докажем, что из обеспечения схемой свойства неотслеживаемости в условиях честной генерации ключей и свойства неподделываемости относительно внешнего нарушителя (SUF-CMA) следует обеспечение неподделываемости относительно сервера с агентом.

Теорема 3.2.1. *Пусть $k \in \mathbb{N}$. Для любого нарушителя $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ в модели* SA-UF$_k$ *с вычислительными ресурсами t, делающего не более q_S запросов к оракулу подписи, существует нарушитель $\mathcal{B}$ в модели* SUF-CMA*, делающий не более q_S запросов к оракулу подписи, и нарушитель $\mathcal{C}$ в модели* HS-Blind*, такие что*

$$\mathsf{Adv}^{\text{SA-UF}_k}_{\mathsf{BS}}(\mathcal{A}) \leqslant \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{BS}}(\mathcal{B}) + q_S \cdot k \cdot \mathsf{Adv}^{\text{HS-Blind}}_{\mathsf{BS}}(\mathcal{C}).$$

Вычислительные ресурсы $\mathcal{B}$ и $\mathcal{C}$ не превосходят t и tkq_S соответственно.

Замечание 3.2.2. *Если схема подписи вслепую обеспечивает совершенную неотслеживаемость (т.е. $\mathsf{Adv}^{\text{HS-Blind}}_{\mathsf{BS}}(\mathcal{C}) = 0$ для любого нарушителя $\mathcal{C}$ с любыми вычислительными ресурсами), оценка преобразуется следующим образом*

$$\mathsf{Adv}^{\text{SA-UF}_k}_{\mathsf{BS}}(\mathcal{A}) \leqslant \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{BS}}(\mathcal{B}).$$

С точки зрения использования классической схемы подписи SS*, настоящее неравенство означает, что для обеспечения неподделываемости относительно сервера с агентом достаточно использования «слепой модификации» этой схемы* BS *(для которой $\mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{BS}}(\mathcal{B}) = \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{SS}}(\mathcal{B})$) и того факта, что классическая схема обеспечивает свойство неподделываемости. Заметим, что оценка не зависит от значения k, поэтому оно может быть выбрано произвольным образом разработчиками приложения.*

Замечание 3.2.3. *Для упрощения изложения доказательство приводится для двухраундовых схем подписи вслепую с пустой первой пересылкой от пользователя серверу. Однако доказательство не использует существенным образом никаких особенностей схем такого типа и может быть несложным образом расширено на случай произвольной схемы подписи вслепую.*

Доказательство. Доказательство состоит из двух этапов.

Этап 1. Рассмотрим последовательность экспериментов, в которой каждый следующий эксперимент получен в результате некоторой модификации предыдущего.

$\mathbf{Exp}^0$. Пусть $\mathbf{Exp}^0_{\mathsf{BS}}(\mathcal{A}) = \mathbf{Exp}^{\text{SA-UF}_k}_{\mathsf{BS}}(\mathcal{A})$.

$\mathbf{Exp}^1$. Рассмотрим следующий модифицированный эксперимент $\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A})$:

$\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $\mathsf{lost} \leftarrow \mathsf{false}$
4 : $st \leftarrow \mathcal{A}_1(\mathsf{sk}, \mathsf{pk})$
5 : $(m, \sigma) \leftarrow \mathcal{A}_2^{Sign}(\mathsf{pk})$
6 : **if** $((m, \sigma) \in \mathcal{L}) \vee (\mathsf{lost} = \mathsf{true})$:
7 : **return** 0
8 : **return** $\mathsf{BS.Verify}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

1 : $i \leftarrow 0$
2 : **do**
3 : $(st, \sigma) \leftarrow \langle \mathcal{A}_1(st), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
4 : **if** $\sigma \neq \perp$:
5 : $(1, \sigma) \leftarrow \langle \mathsf{BS.Sign}(\mathsf{sk}, \mathsf{pk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
6 : $i \leftarrow i + 1$
7 : **until** $(i \geqslant k) \vee (\sigma \neq \perp)$
8 : **if** $\sigma = \perp$:
9 : $\mathsf{lost} \leftarrow \mathsf{true}$
10 : **return** $\perp$
11 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
12 : **return** σ

$\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A})$ отличается от $\mathbf{Exp}^0_{\mathsf{BS}}(\mathcal{A})$ наличием дополнительных строк 4 и 5 в оракуле подписи $Sign$. Если оракул, взаимодействуя с алгоритмом $\mathcal{A}_1$ в качестве честного пользователя, завершает выполнение протокола формирования подписи успешно, то он заново вычисляет новое значение подписи, самостоятельно выполняя честный протокол формирования подписи (без взаимодействия с $\mathcal{A}_1$). Второй этап доказательства посвящен оценке разницы вероятностей успешного завершения эксперимента в экспериментах $\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A})$ и $\mathbf{Exp}^0_{\mathsf{BS}}(\mathcal{A})$.

$\mathbf{Exp}^2$. Рассмотрим следующую модификацию: эксперимент $\mathbf{Exp}^2_{\mathsf{BS}}(\mathcal{A})$. В нем оракул подписи всегда возвращает в ответ на запросы $\mathcal{A}_2$ честно сгенерированное значение подписи, даже если нарушитель $\mathcal{A}_1$ спровоцировал возникновение ошибки k раз подряд, что приводит к выставлению флага lost в $\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A})$.

$\mathbf{Exp}^{2}_{\mathsf{BS}}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $st \leftarrow \mathcal{A}_1(\mathsf{sk}, \mathsf{pk})$
4 : $(m, \sigma) \leftarrow \mathcal{A}_2^{Sign}(\mathsf{pk})$
5 : **if** $((m, \sigma) \in \mathcal{L})$:
6 : **return** 0
7 : **return** $\mathsf{BS.Verify}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

1 : $i \leftarrow 0$
2 : **do**
3 : $(st, \sigma) \leftarrow \langle \mathcal{A}_1(st), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
4 : $i \leftarrow i + 1$
5 : **until** $(i \geqslant k) \vee (\sigma \neq \perp)$
6 : $(1, \sigma) \leftarrow \langle \mathsf{BS.Sign}(\mathsf{sk}, \mathsf{pk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
7 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
8 : **return** σ

Для этого эксперимента:

$$\Pr\left[\mathbf{Exp}^{1}_{\mathsf{BS}}(\mathcal{A}) \to 1\right] = \Pr\left[\mathbf{Exp}^{1}_{\mathsf{BS}}(\mathcal{A}) \to 1 \wedge (\mathsf{lost} = \mathsf{false})\right] + \\ + \underbrace{\Pr\left[\mathbf{Exp}^{1}_{\mathsf{BS}}(\mathcal{A}) \to 1 \wedge (\mathsf{lost} = \mathsf{true})\right]}_{= 0 \text{ в силу строки 6 } \mathbf{Exp}^{1}_{\mathsf{BS}}(\mathcal{A})} \leqslant \Pr\left[\mathbf{Exp}^{2}_{\mathsf{BS}}(\mathcal{A}) \to 1\right].$$

$\mathbf{Exp}^{3}$. В эксперименте $\mathbf{Exp}^{2}_{\mathsf{BS}}(\mathcal{A})$ алгоритм $\mathcal{A}_1$ может быть исключен, поскольку его работа никак не влияет на значения подписи, возвращаемые оракулом подписи (см. эксперимент $\mathbf{Exp}^{3}_{\mathsf{BS}}(\mathcal{A}_2)$ ниже). Заметим, что

$$\Pr\left[\mathbf{Exp}^{2}_{\mathsf{BS}}(\mathcal{A}_1, \mathcal{A}_2) \to 1\right] = \Pr\left[\mathbf{Exp}^{3}_{\mathsf{BS}}(\mathcal{A}_2) \to 1\right].$$

$\mathbf{Exp}^{3}_{\mathsf{BS}}(\mathcal{A}_2)$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $(m, \sigma) \leftarrow \mathcal{A}_2^{Sign}(\mathsf{pk})$
4 : **if** $(m, \sigma) \in \mathcal{L}$:
5 : **return** 0
6 : **return** $\mathsf{BS.Verify}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

1 : $(1, \sigma) \leftarrow \langle \mathsf{BS.Sign}(\mathsf{sk}, \mathsf{pk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
2 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
3 : **return** σ

Заметим, что эксперимент $\mathbf{Exp}^{3}_{\mathsf{BS}}$ в точности совпадает с экспериментом $\mathbf{Exp}^{\text{SUF-CMA}}_{\mathsf{BS}}$, поэтому $\Pr\left[\mathbf{Exp}^{2}_{\mathsf{BS}}(\mathcal{A}) \to 1\right] \leqslant \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{BS}}(\mathcal{B})$ для $\mathcal{B} = \mathcal{A}_2$.

Этап 2. Для завершения доказательства построим алгоритм работы нарушителя $\mathcal{C}$ для свойства неотслеживаемости. Определим следующий вспомогательный эксперимент:

$\mathbf{Exp}^{4,b}_{\mathsf{BS}}(\mathcal{C})$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $st \leftarrow \mathcal{C}^{Init}(\mathsf{sk}, \mathsf{pk})$
3 : $b' \leftarrow \mathcal{C}^{User_1, User_2}(st)$
4 : **return** b'

Oracle $Init(m)$

1 : $sess \leftarrow \mathsf{init}$
2 : store m
3 : **return** θ

Oracle $User_1(msg)$

1 : **if** $sess \neq \mathsf{init}$: **return** $\perp$
2 : $sess \leftarrow \mathsf{open}$
3 : $(msg, state) \leftarrow \mathsf{BS.User}_1((\mathsf{pk}, m), msg)$
4 : **return** msg

Oracle $User_2(msg)$

1 : **if** $sess \neq \mathsf{open}$: **return** $\perp$
2 : $\sigma \leftarrow \mathsf{BS.User}_2(state, msg)$
3 : **if** $(\sigma \neq \perp) \wedge (b = 0)$:
4 : $\quad (1, \sigma) \leftarrow \langle \mathsf{BS.Sign}(\mathsf{sk}, \mathsf{pk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
5 : **return** σ

В этом эксперименте нарушитель может сделать только один запрос к каждому из оракулов (выполнить только один сеанс). Нарушитель получает значение подписи, сформированное оракулами в результате взаимодействия с нарушителем, если $b = 1$, и подпись, сформированную в результате честного запуска протокола, в противном случае. Заметим, что если нарушитель провоцирует возникновение ошибки в сеансе, то он всегда получает символ $\perp$ от оракула $User_2$ вне зависимости от значения бита b.

Используя стандартную технику, называемую «гибридным аргументом» (см., например, [38]), можно показать, что существует нарушитель $\mathcal{C}'$, такой что

$$\Pr\left[\mathbf{Exp}^{0}_{\mathsf{BS}}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^{1}_{\mathsf{BS}}(\mathcal{A}) \to 1\right] = \\ = q_S \cdot k \cdot \left(\Pr\left[\mathbf{Exp}^{4,1}_{\mathsf{BS}}(\mathcal{C}') \to 1\right] - \Pr\left[\mathbf{Exp}^{4,0}_{\mathsf{BS}}(\mathcal{C}') \to 1\right]\right).$$

Наконец построим алгоритм работы нарушителя $\mathcal{C}$, который использует нарушителя $\mathcal{C}'$ в качестве черного ящика. Нарушитель $\mathcal{C}$ действует следующим образом.

1. Нарушитель $\mathcal{C}$ получает ключевую пару $(\mathsf{sk}, \mathsf{pk})$ и запускает нарушителя $\mathcal{C}'$, подавая ему на вход эти значения.

2. Когда $\mathcal{C}'$ делает запрос m к оракулу $Init$, нарушитель $\mathcal{C}$ делает запрос (m, m) к своему собственному оракулу $Init$.

3. После запуска сеансов нарушитель $\mathcal{C}$ сначала моделирует $sess_0$ согласно протоколу:

   a. он вычисляет $(msg^0_{S,1}, state_{S,1}) \leftarrow \mathsf{BS.Sign}_1(\mathsf{sk}, \mathsf{pk})$ и делает запрос $(0, msg^0_{S,1})$ к своему собственному оракулу $User_1$;

б. после получения $msg^0_{U,1}$, нарушитель $\mathcal{C}$ вычисляет

$$(msg^0_{S,2}, 1) \leftarrow \mathsf{BS.Sign}_2(state_{S,1}, msg_{U,1}),$$

делает запрос $(0, msg^0_{S,2})$ к своему собственному оракулу $User_2$ и получает в ответ значение θ.

Заметим, что $\sigma_{b_0} \neq \perp$ в силу свойства корректности схемы подписи вслепую.

4. Далее нарушитель $\mathcal{C}$ перехватывает все запросы $\mathcal{C}'$ и транслирует их своим оракулам в сеансе $sess_1$.

   а. Перехватив запрос msg_1 нарушителя $\mathcal{C}'$ к оракулу $User_1$, нарушитель $\mathcal{C}$ делает запрос $(1, msg^1_{S,1})$, где $msg^1_{S,1} = msg_1$, к своему собственному оракулу $User_1$ and и транслирует ответ $msg^1_{U,1}$ нарушителю $\mathcal{C}'$.

   б. Перехватив запрос msg_2 нарушителя $\mathcal{C}'$ к оракулу $User_2$, нарушитель $\mathcal{C}$ делает запрос $(1, msg^1_{S,2})$, где $msg^1_{S,2} = msg_2$, к своему собственному оракулу $User_2$. В ответ $\mathcal{C}$ получает пару (σ_0, σ_1) и возвращает нарушителю $\mathcal{C}'$ первое значение σ_0. Заметим, что пара (σ_0, σ_1) может быть равна $(\perp, \perp)$.

5. Нарушитель $\mathcal{C}$ возвращает тот же самый бит, что и нарушитель $\mathcal{C}'$.

Если $\mathcal{C}$ взаимодействует с экспериментатором $\mathbf{Exp}^{\text{HS-Blind},1}_{\mathsf{BS}}$ ($\mathbf{Exp}^{\text{HS-Blind},0}_{\mathsf{BS}}$), то $\sigma_0 = \sigma_{b_1}$ ($\sigma_0 = \sigma_{b_0}$). Более того, $\mathcal{C}$ возвращает $\perp$ на шаге 4 тогда и только тогда, когда $\mathcal{C}'$ провоцирует возникновение ошибки в $sess_1$, что в точности совпадает с поведением эксперимента $\mathbf{Exp}^{4,b}_{\mathsf{BS}}$. Таким образом,

$$\Pr\big[\mathbf{Exp}^{4,1}_{\mathsf{BS}}(\mathcal{C}') \to 1\big] = \Pr\Big[\mathbf{Exp}^{\text{HS-Blind},1}_{\mathsf{BS}}(\mathcal{C}) \to 1\Big],$$

$$\Pr\big[\mathbf{Exp}^{4,0}_{\mathsf{BS}}(\mathcal{C}') \to 1\big] = \Pr\Big[\mathbf{Exp}^{\text{HS-Blind},0}_{\mathsf{BS}}(\mathcal{C}) \to 1\Big].$$

Суммируя полученные оценки, получаем

$$\begin{aligned}
\Pr\big[\mathbf{Exp}^0_{\mathsf{BS}}(\mathcal{A}) \to 1\big] - \Pr\big[\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A}) \to 1\big] = \\
= q_S \cdot k \cdot \big(\Pr\big[\mathbf{Exp}^{4,1}_{\mathsf{BS}}(\mathcal{C}') \to 1\big] - \Pr\big[\mathbf{Exp}^{4,0}_{\mathsf{BS}}(\mathcal{C}') \to 1\big]\big) = \\
= q_S \cdot k \cdot \Big(\Pr\Big[\mathbf{Exp}^{\text{HS-Blind},1}_{\mathsf{BS}}(\mathcal{C}) \to 1\Big] - \Pr\Big[\mathbf{Exp}^{\text{HS-Blind},0}_{\mathsf{BS}}(\mathcal{C}) \to 1\Big]\Big) = \\
= q_S \cdot k \cdot \mathsf{Adv}^{\text{HS-Blind}}_{\mathsf{BS}}(\mathcal{C}).
\end{aligned}$$

□

3.2.3. Анализ безопасности относительно внешнего нарушителя

В настоящем разделе определяется класс схем, частный случай схем подписи вслепую Эль-Гамаля, которые обеспечивают свойство неподделываемости относительно «честного, но любопытного» нарушителя. А именно, для этих схем строится сведение к свойству неподделываемости классической схемы подписи Эль-Гамаля. Напомним, что все существующие схемы подписи вслепую Эль-Гамаля не обеспечивают свойство неподделываемости в расширенной модели безопасности (см. раздел 2).

Класс схем подписи вслепую Эль-Гамаля GenEG-BS определен в разделе 2. Параметрами протокола формирования подписи в настоящих схемах являются генерационная точка P, открытый ключ Q и сообщение m, обозначим их через par.

В настоящем разделе на алгоритм работы пользователя в протоколе формирования подписи налагаются дополнительные ограничения:

- все маскирующие факторы (обозначим их через rnd), используемые пользователем, выбираются в соответствии с некоторым распределением $\mathcal{D}$, независимым от значений, полученных от подписывающего;
- первая компонента подписи r' представляет собой x-координату точки R', которая вычисляется в результате применения функции, параметризованной значением par (обозначим эту функцию через $\mathcal{L}_1^{par}$), которая принимает на вход точку R, полученную от подписывающего, и значения rnd. Эта функция является линейной по R для всех значений rnd, выбранных в соответствии с протоколом;
- вторая компонента подписи s' вычисляется в результате применения функции, параметризованной значением par (обозначим эту функцию через $\mathcal{L}_2^{par}$), которая принимает на вход значение s, полученное от подписывающего, значения rnd и точку R. Эта функция является линейной по s для всех значений rnd и R, сгенерированных в соответствии с протоколом.

Все схемы такого типа далее обозначаются через GenEG-BS$_{\mathcal{L}}$. Соответствующий протокол формирования подписи изображен на рис. 3.2.

Покажем, что схемы GenEG-BS$_{\mathcal{L}}$ действительно являются «слепыми модификациями» классических схем подписи GenEG, т.е. генерируют такое же распределение на множестве значений подписи. Распределение подписей в схемах подписи GenEG определяется равновероятным распределением на множестве значений k. Распределение в схемах GenEG-BS$_{\mathcal{L}}$ определяется распределением на множестве значений k', где k' определяется из условия

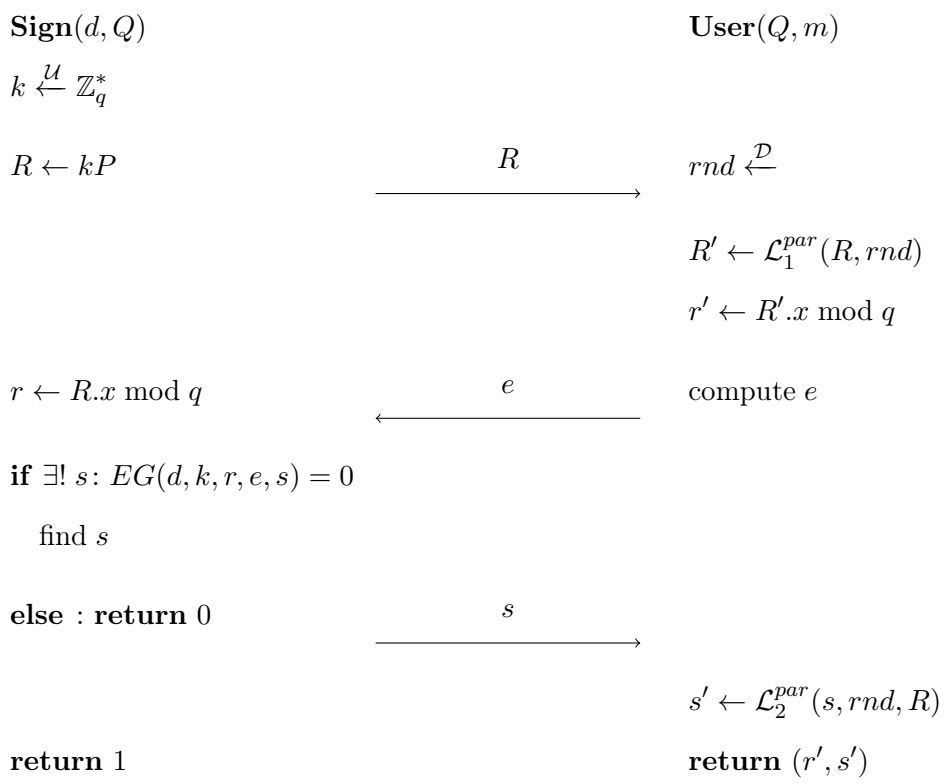

Рис. 3.2. Протокол формирования подписи в схеме $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$

$(k'P).x \bmod q = r'$. Значение k' линейно зависит от k, так как значение R' линейно по R, а значения rnd выбираются независимо от R. Таким образом, распределение на значениях k' также является равновероятным.

Заметим, что значение $view$, определяющее вспомогательные значения, используемые пользователем в протоколе формирования подписи схемы $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$, включает в себя входящие сообщения R, s и маскирующие факторы rnd.

Теорема 3.2.2. *Для любого нарушителя $\mathcal{A}$ для схемы $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$ в модели* HBC-UF *с вычислительными ресурсами t, делающего не более q запросов к оракулу подписи, существует нарушитель $\mathcal{B}$ для классической схемы подписи GenEG в модели* SUF-CMA *с такими же вычислительными ресурсами, делающий не более q запросов к оракулу подписи, такой что*

$$\mathsf{Adv}^{\text{HBC-UF}}_{\mathsf{GenEG\text{-}BS}_{\mathcal{L}}}(\mathcal{A}) \leqslant \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{GenEG}}(\mathcal{B}).$$

Доказательство. Построим алгоритм работы нарушителя $\mathcal{B}$ для схемы GenEG, использующий нарушителя $\mathcal{A}$ в качестве черного ящика. Он перехватывает запросы $\mathcal{A}$ к оракулу подписи и обрабатывает их самостоятельно, используя свой собственный оракул подписи.

Получив запрос m, нарушитель $\mathcal{B}$ перенаправляет m своему собственному оракулу и получает в ответ значение (r', s'). Далее он восстанавливает точку R' из уравнения проверки подписи и выбирает значение rnd в соответствии с распределением $\mathcal{D}$. После этого он вычис-

ляет значение R, применяя функцию $\mathcal{L}_1^{-1}$, и значение s, применяя функцию $\mathcal{L}_2^{-1}$. В качестве ответа он возвращает подпись (r', s') и значение $view = (R, s, rnd)$.

Заметим, что $\mathcal{B}$ генерирует в точности такое же распределение на множестве ответов оракула подписи, поскольку схема $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$ является «слепой модификацией» схемы GenEG. Значение rnd выбирается так же, как и при честном выполнении протокола, значения R и s также вычисляются как и при честном запуске протокола, т.к. функции $\mathcal{L}_1$ и $\mathcal{L}_2$ обратимы однозначным образом.

Когда нарушитель $\mathcal{A}$ возвращает подделку, $\mathcal{B}$ транслирует ее своему собственному экспериментатору и завершает работу. Очевидно, что если $\mathcal{A}$ успешно решает задачу, то $\mathcal{B}$ также успешно решает свою задачу, откуда следует утверждение теоремы. □

Замечание 3.2.4. *Аналогичный результат может быть сформулирован для схемы подписи Шнорра и ее «слепой модификации», определенной в [31]. Доказательство проводится аналогичным образом.*

3.2.4. Схема подписи вслепую на основе GOST

Для защиты систем на основе схемы подписи GOST предлагается использовать конкретную схему подписи вслепую — схему, предложенную в работе [29] в 1994 году и обычно называемую схемой Камениша. Определим эту схему для группы точек эллиптической кривой. Будем обозначать ее через Cam-BS.

Схема Камениша принадлежит классу схем GenEG-BS, алгоритм генерации ключей в этой схеме совпадает с алгоритмом генерации ключей в схеме подписи Эль-Гамаля. Протокол формирования подписи определен на рис. 3.3. Алгоритм проверки подписи (r', s') для сообщения m заключается в проверке условия $r' \neq 0$ и равенства $r' = R'.x \bmod q$, где $R' = (e')^{-1}(s'P - r'Q)$, e' равно $H(m)$, если $H(m) \neq 0$, и 1 в противном случае. Заметим, что протокол формирования подписи на рис. 3.3 определен для случая использования эллиптических кривых простого порядка. Тем не менее, он может быть расширен за счет включения дополнительных проверок для случая использования кривых непростого порядка, например, кривых Эдвардса.

Докажем, что использование схемы Камениша в системах на основе подписи GOST позволяет обеспечить защиту как от сильного нарушителя с агентом, так и от внешнего нарушителя. Заметим, что в разделе 2 было показано, что схема Камениша уязвима к модифицированной ROS атаке, если нарушитель, выступающий в роли пользователя, имеет возможность

Sign(d, Q)		**User**(Q, m)
rand1: $k \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$		
$R \leftarrow kP$		
$r \leftarrow R.x \bmod q$		
if $r = 0$: goto rand1	$\xrightarrow{\quad R \quad}$	**if** $R = \mathcal{O}$: **return** $\perp$
		$r \leftarrow R.x \bmod q$
		if $r = 0$: **return** $\perp$
		rand2 : $\alpha, \beta \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
		$R' \leftarrow \alpha R + \beta P$
		if $R' = \mathcal{O}$: goto rand2
		$r' \leftarrow R'.x \bmod q$
		if $r' = 0$: goto rand2
		$e' \leftarrow H(m)$
		if $e' = 0$: $e' \leftarrow 1$
if $e = 0$: **return** 0	$\xleftarrow{\quad e \quad}$	$e \leftarrow \alpha e' r (r')^{-1}$
$s \leftarrow ke + dr$		
return 1	$\xrightarrow{\quad s \quad}$	**if** $sP \neq eR + rQ$: **return** $\perp$
		$s' \leftarrow sr'r^{-1} + \beta e'$
		$\sigma \leftarrow (r', s')$
		return σ

Рис. 3.3. Протокол формирования подписи в схеме Камениша

открыть $\ell \geqslant \lceil \log q \rceil$ параллельных сеансов протокола формирования подписи, т.е. данная схема не обеспечивает свойство неподделываемости в сильном смысле. Однако рассматриваемая прикладная система не требует от схемы подписи вслепую столь сильных свойств безопасности, достаточно обеспечения свойства неподделываемости относительно «честного, но любопытного» нарушителя и сервера с агентом.

Утверждение 3.2.1. *Пусть $k \in \mathbb{N}$. Для любого нарушителя $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ для схемы* Cam-BS *в модели* SA-UF$_k$ *с вычислительными ресурсами t, делающего не более q запросов к оракулу подписи, существует нарушитель $\mathcal{B}$ для схемы* GOST *в модели* SUF-CMA *с та-*

кими же вычислительными ресурсами, делающий не более q запросов к оракулу подписи, такой что

$$\mathsf{Adv}^{\text{SA-UF}_k}_{\mathsf{Cam\text{-}BS}}(\mathcal{A}) \leqslant \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{GOST}}(\mathcal{B}).$$

Для любого нарушителя $\mathcal{D}$ для схемы Cam-BS *в модели* HBC-UF *с вычислительными ресурсами t, делающего не более q запросов к оракулу подписи, существует нарушитель $\mathcal{C}$ для схемы* GOST *в модели* SUF-CMA *с такими же вычислительными ресурсами, делающий не более q запросов к оракулу подписи, такой что*

$$\mathsf{Adv}^{\text{HBC-UF}}_{\mathsf{Cam\text{-}BS}}(\mathcal{D}) \leqslant \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{GOST}}(\mathcal{C}).$$

Доказательство. Схема Камениша обеспечивает совершенную неотслеживаемость (см. теорему 2 [29]), поэтому для любого нарушителя $\mathcal{M}$ для схемы Cam-BS в модели HS-Blind

$$\mathsf{Adv}^{\text{HS-Blind}}_{\mathsf{Cam\text{-}BS}}(\mathcal{M}) = 0.$$

Покажем, что схема Камениша является представителем класса схем $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$, определенного в подразделе 3.2.3. Действительно, распределение $\mathcal{D}$ в этой схеме представляет собой равновероятное распределение на множестве $\mathbb{Z}_q^* \times \mathbb{Z}_q^*$, которое независимо от значения R, функции $\mathcal{L}_1^{(P,Q,m)}$ и $\mathcal{L}_2^{(P,Q,m)}$ определяются следующим образом:

$$\mathcal{L}_1^{(P,Q,m)}(R,(\alpha,\beta)) = \alpha R + \beta P; \quad \mathcal{L}_2^{(P,Q,m)}(s,(\alpha,\beta),R) = sr'r^{-1} + \beta e',$$

где $e' = H(m),\ r = R.x \bmod q,\ r' = (\alpha R + \beta P).x \bmod q$. Эти функции являются линейными по значениям R и s соответственно для всех возможных значений rnd. Более того, нулевые значения r и e исключаются за счет соответствующих проверок на стороне подписывающего так же, как и в схеме подписи GOST. Таким образом, схема Камениша является «слепой модификацией» схемы подписи GOST.

Тогда для любого нарушителя $\mathcal{B}$ в модели SUF-CMA справедливо, что

$$\mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{Cam\text{-}BS}}(\mathcal{B}) = \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{GOST}}(\mathcal{B}).$$

Суммируя рассуждения выше и применяя теорему 3.2.1, получаем первое утверждение теоремы. А именно, для любого нарушителя $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ для схемы Cam-BS в модели SA-UF_k существует нарушитель $\mathcal{B}$ для схемы GOST в модели SUF-CMA, такой что

$$\mathsf{Adv}^{\text{SA-UF}_k}_{\mathsf{Cam\text{-}BS}}(\mathcal{A}) \leqslant \mathsf{Adv}^{\text{SUF-CMA}}_{\mathsf{GOST}}(\mathcal{B}).$$

Поскольку схема Камениша является представителем класса $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$, к ней применима также теорема 3.2.2. Это означает справедливость второго утверждения теоремы. А

именно, для любого нарушителя $\mathcal{D}$ в модели HBC-UF, существует нарушитель $\mathcal{C}$ для схемы GOST в модели SUF-CMA, такой что

$$\mathsf{Adv}^{\text{HBC-UF}}_{\text{Cam-BS}}(\mathcal{D}) \leqslant \mathsf{Adv}^{\text{SUF-CMA}}_{\text{GOST}}(\mathcal{C}).$$

□

Таким образом, схема Камениша является «слепой модификацией» схемы подписи GOST и может применяться в прикладных системах на основе подписи GOST в условиях использования потенциально уязвимых ключевых носителей. Она обеспечивает стойкость относительно сильного нарушителя с агентом и внешнего нарушителя при единственном предположении, что схема подписи GOST является стойкой. Заметим, что настоящее решение, в отличие от решения, предложенного в работе [4], не требует никаких дополнительных предположений о корректной реализации низкоуровневых арифметических операций и отсутствии сбоев. Более того, оно подразумевает меньшее количество вычислений на стороне смарт-карты.

Выводы

В настоящем разделе исследуются специализированные модели безопасности для схем подписи и схем подписи вслепую, актуальные в прикладных системах формирования подписи, в которых ключ подписи хранится на потенциально уязвимой смарт-карте.

Специализированная модель безопасности для схем подписи является более сильной, чем классическая модель безопасности SUF-CMA. Доказывается, что схемы подписи Эль-Гамаля могут быть модифицированы с целью обеспечения стойкости в данной модели. Как следствие, данные схемы позволяют обеспечить защиту рассматриваемого класса прикладных систем от внешнего нарушителя и слабого нарушителя с агентом.

Специализированные модели безопасности для схем подписи вслепую являются более слабыми, чем расширенные модели безопасности для этих схем. Доказывается, что схемы подписи вслепую на основе уравнения Эль-Гамаля обеспечивают (при некоторых ограничениях на значения параметров) стойкость в специализированных моделях безопасности, если они обеспечивают свойство неотслеживаемости при условии честной генерации ключей, свойство неподделываемости относительно внешнего нарушителя, а также базовая схема подписи Эль-Гамаля обеспечивает свойство неподделываемости в модели SUF-CMA. Таким образом, данные схемы потенциально позволяют обеспечить защиту рассматриваемого класса прикладных систем от внешнего нарушителя и сильного нарушителя с агентом несмотря на то, что в общем случае они не являются стойкими в расширенных моделях безопасности.

Заключение

Основные результаты диссертационной работы состоят в следующем.

1) Для схемы подписи вслепую Шаума-Педерсена разработан метод нарушения свойства сильной неподделываемости в модели с параллельными сеансами и доказана содержательная верхняя оценка преимущества нарушителя, реализующего угрозу нарушения свойства слабой неподделываемости в модели с параллельными сеансами. Полученные результаты демонстрируют, что схема Шаума-Педерсена не обеспечивает стойкость в наиболее сильной расширенной модели безопасности, при этом в основе стойкости схемы в более слабой расширенной модели безопасности лежит новая нестандартная задача в группе точек эллиптической кривой.

2) Синтезирован класс схем подписи вслепую на основе уравнения Эль-Гамаля, не использующих дополнительные криптографические механизмы, покрывающий все существующие схемы такого типа. Для существенной части схем из этого класса разработан метод нарушения свойства неподделываемости в модели с параллельными сеансами. Среди оставшихся схем выявлен подкласс схем, для которых разработан метод нарушения одного из свойств: свойства неподделываемости в модели с последовательными сеансами или свойства неотслеживаемости. Построенные методы демонстрируют, что все существующие схемы подписи вслепую на основе уравнения Эль-Гамаля не обеспечивают стойкость в расширенных моделях безопасности.

3) Разработан метод модификации схемы подписи Эль-Гамаля, позволяющий уменьшить размер подписи на четверть и обеспечить безопасность в условиях использования недоверенного датчика случайных чисел при формировании подписи. Для модифицированной схемы доказана содержательная верхняя оценка величины преимущества нарушителя в специализированной модели безопасности, предоставляющей нарушителю возможность выбирать случайные значения, используемые в процессе формирования подписи.

4) Для схем подписи вслепую на основе уравнения Эль-Гамаля доказаны содержательные верхние оценки преимущества нарушителя в специализированных моделях безопасности, актуальных в системах формирования подписи в условиях использования функциональных ключевых носителей.

Разработанные в диссертации методы могут применяться при разработке и исследова-

ниях средств защиты информации, а также использоваться в учебном процессе студентов, проходящих обучение в рамках специализации «Математические и программные методы обеспечения информационной безопасности».

Список литературы

1. ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи». Стандартинформ, 2012.
2. Рекомендации по стандартизации Р 1323565.1.024-2019 «Информационная технология. Криптографическая защита информации. Параметры эллиптических кривых для криптографических алгоритмов и протоколов». М.: Стандартинформ, 2019.
3. Алексеев Е. К. , Ахметзянова Л. Р., Бабуева А. А., Смышляев С. В. «О повышении безопасности схем подписи Эль-Гамаля», Матем. вопр. криптогр., 12:3 (2021), 5–30.
4. Алексеев Е. К. , Ахметзянова Л. Р., Божко А. А., Смышляев С. В. «Безопасная реализация электронной подписи с использованием слабодоверенного вычислителя», Матем. вопр. криптогр., 12:4 (2021), 5–23.
5. Алексеев Е. К. , Ахметзянова Л. Р., Ошкин И. Б., Смышляев С. В. «Обзор уязвимостей некоторых протоколов выработки общего ключа с аутентификацией на основе пароля и принципы построения протокола SESPAKE», Матем. вопр. криптогр., 7:4 (2016), 7–28.
6. Ростовцев А. Г. «Подпись «вслепую» на эллиптической кривой для электронных денег» //Проблемы информационной безопасности. Компьютерные системы. – 2000. – №. 1. – С. 40-45.
7. Смышляев С. В. «Математические методы обоснования оценок уровня информационной безопасности программных средств защиты информации, функционирующих в слабодоверенном окружении» Дис. ... док. физ.-мат. наук, МГУ имени М.В. Ломоносова, Москва, 2022, 593 с.
8. Information technology – Security techniques – Blind digital signatures – Part 2: Discrete logarithm based mechanisms. ISO/IEC DIS 18370-2:2016(E) – 70 p.
9. Information technology – Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms : ISO/IEC 14888-3 (Edition 2) : 2014. – 130 p.
10. DSTU 4145:2002. Information technology. Cryptographic information security. Digital signature, which based on elliptic curve. Generation and verification, Kyiv, 2004.
11. FIPS 186-5. Digital Signature Standard (DSS). Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8900. 2023.
12. Abdalla M., Namprempre C., Neven G. On the (im) possibility of blind message authentication codes //Topics in Cryptology–CT-RSA 2006: The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2005. Proceedings. Springer Berlin

Heidelberg, 2006. pp. 262–279.

13. Abe M. «A secure three-move blind signature scheme for polynomially many signatures» //International Conference on the Theory and Applications of Cryptographic Techniques. – Springer, Berlin, Heidelberg, 2001. – C. 136-151.
14. Abe M., Ohkubo M. «Security of Some Three-move Blind Signature Schemes Reconsidered». The 2003 Symposium on Cryptography and Information Security. Hamamatsu, Japan, 2003.
15. Abe M., Okamoto T. Provably secure partially blind signatures // Advances in Cryptology — CRYPTO 2000, Springer, Berlin, Heidelberg, pp. 271–286, 2000.
16. Akhmetzyanova L., Alekseev E., Babueva A., Smyshlyaev S. «On methods of shortening ElGamal-type signatures». Matematicheskie Voprosy Kriptografii [Mathematical Aspects of Cryptography], 12(2), pp. 75–91, 2021.
17. Alemu S. B., Kastner J. On the Security of Blind Signatures in the Multi-Signer Setting //Cryptology ePrint Archive. 2023.
18. Baldimtsi F., Lysyanskaya A. «On the security of one-witness blind signature schemes» //International Conference on the Theory and Application of Cryptology and Information Security. – Springer, Berlin, Heidelberg, 2013. – C. 82-99.
19. Bauer B., Fuchsbauer G., Loss J. «A classification of computational assumptions in the algebraic group model» //Annual International Cryptology Conference. – Cham: Springer International Publishing, 2020. – C. 121-151.
20. Bauer B., Fuchsbauer G., Plouviez A. «The one-more discrete logarithm assumption in the generic group model» //Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV 27. – Springer International Publishing, 2021. – C. 587-617.
21. Bellare M., Namprempre C., Pointcheval D., Semanko M. «The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme» //Journal of Cryptology. – 2003. – Т. 16. – №. 3.
22. Bellare M., Rogaway P. «Random oracles are practical: A paradigm for designing efficient protocols»// In Proceedings of the 1st ACM conference on Computer and communications security, pp. 62–73. 1993.
23. Bellare M., Rogaway P. «The security of triple encryption and a framework for code-based game-playing proofs» // Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 409–426. Springer, Berlin, Heidelberg, 2006.
24. Benhamouda F, Lepoint T., Loss J., Orru M., Raykova M. «On the (in)security of

ROS»// Annual International Conference on the Theory and Applications of Cryptographic Techniques. – Cham : Springer International Publishing, 2021. – С. 33-53.

25. Boneh D., Boyen X. «Short signatures without random oracles» //International conference on the theory and applications of cryptographic techniques. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2004. – С. 56-73.
26. Brands S. Untraceable off-line cash in wallet with observers //Advances in Cryptology—CRYPTO'93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22–26, 1993 Proceedings 13. – Springer Berlin Heidelberg, 1994. – С. 302-318.
27. Bresson E., Monnerat J., Vergnaud D. «Separation results on the "one-more" computational problems». //Topics in Cryptology–CT-RSA 2008: The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings. – Springer Berlin Heidelberg, 2008. – С. 71-87.
28. Camenisch J., Neven G., Shelat A. «Simulatable adaptive oblivious transfer» //Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg : Springer Berlin Heidelberg, 2007. pp. 573–590.
29. Camenisch J. L., Piveteau J. M., Stadler M. A. «Blind signatures based on the discrete logarithm problem» //Workshop on the Theory and Application of of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1994. pp. 428–432.
30. Chairattana-Apirom R., Tessaro S., Zhu C. «Pairing-Free Blind Signatures from CDH Assumptions» //Cryptology ePrint Archive, Paper 2023/1780, 2023.
31. Chaum D. «Blind signatures for untraceable payments» //Advances in cryptology. – Springer, Boston, MA, 1983. – С. 199–203.
32. Chaum D., Pedersen T. P. «Wallet databases with observers». //Annual international cryptology conference. – Springer, Berlin, Heidelberg, 1992. – С. 89–105.
33. Cheon J. H. «Security analysis of the strong Diffie-Hellman problem». //Annual International Conference on the Theory and Applications of Cryptographic Techniques. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2006. – С. 1–11.
34. Crites, E., Komlo, C., Maller, M., Tessaro, S., Zhu, C. «Snowblind: A Threshold Blind Signature in Pairing-Free Groups» //Annual International Cryptology Conference (pp. 710-742). Cham: Springer Nature Switzerland. 2023.
35. Faz-Hernandez, A., Scott, S., Sullivan, N., Wahby, R. S., Wood, C. A. «Hashing to elliptic curves» //Internet Research Task Force, Informational, RFC 9380, 2021.
36. Fersch M., Kiltz E., Poettering B. «On the provable security of (EC) DSA signatures»

//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. – 2016. – С. 1651-1662.

37. Fersch M. «The provable security of Elgamal-type signature schemes» //Diss. Bochum, Ruhr-Universitet Bochum, 2018.
38. Fischlin M., Mittelbach A. «An Overview of the Hybrid Argument» //Cryptology ePrint Archive, Paper 2021/088, 2021.
39. Fischlin M., Schroder D. «On the impossibility of three-move blind signature schemes» //Annual International Conference on the Theory and Applications of Cryptographic Techniques. – Springer, Berlin, Heidelberg, 2010. – С. 197-215.
40. Fischlin M., Schroder D. «Security of blind signatures under aborts» //International Workshop on Public Key Cryptography. – Springer, Berlin, Heidelberg, 2009. – С. 297-316.
41. Fuchsbauer G., Kiltz E., Loss J. «The Algebraic Group Model and its Applications». In: Shacham H., Boldyreva A. (eds) Advances in Cryptology – CRYPTO 2018. CRYPTO 2018. Lecture Notes in Computer Science, vol 10992. Springer, Cham. 2018.
42. Fuchsbauer G., Plouviez A., Seurin Y. «Blind Schnorr signatures and signed ElGamal encryption in the algebraic group model» //Annual International Conference on the Theory and Applications of Cryptographic Techniques. – Springer, Cham, 2020. – С. 63-95.
43. Goldreich O. «Foundations of cryptography» // Cambridge university press, 2009.
44. Gorbenko I., Yesina M., Ponomar V. «Anonymous electronic signature method» //2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T). – IEEE, 2016. – С. 47-50.
45. Harn L., Xu Y. «Design of generalised ElGamal type digital signature schemes based on discrete logarithm» // Electronics Letters, 30(24), pp. 2025–2026, 1994.
46. Hauck E., Kiltz E., Loss J., Nguyen N. K. «Lattice-Based Blind Signatures, Revisited» //Annual International Cryptology Conference. – Springer, Cham, 2020. – С. 500-529.
47. Hazay, C., Katz, J., Koo, C. Y., Lindell, Y. «Concurrently-Secure Blind Signatures Without Random Oracles or Setup Assumptions» //Theory of Cryptography. TCC 2007. Lecture Notes in Computer Science, vol 4392. Springer, Berlin, Heidelberg, pp. 323-341 (2007).
48. Hosseini H., Bahrak B., Hessar F. «A GOST-like Blind Signature Scheme Based on Elliptic Curve Discrete Logarithm Problem» //arXiv preprint arXiv:1304.2094. – 2013.
49. Huang Z., Wang Y. Blind Signature Schemes Based on Gost Signature //Progress on Cryptography. – Springer, Boston, MA, 2004. – С. 123-128.
50. Hufschmitt E., Traor J. «Fair blind signatures revisited» //International Conference on Pairing-Based Cryptography. – Springer, Berlin, Heidelberg, 2007. – С. 268-292.

51. Jena D., Jena S. K., Majhi B. «A novel blind signature scheme based on nyberg-rueppel signature scheme and applying in off-line digital cash» //10th International Conference on Information Technology (ICIT 2007). – IEEE, 2007. – C. 19-22.
52. Jena D., Panigrahy S. K., Acharya B., Jena S. K. «A Novel ECDLP-Based Blind Signature Scheme» //National Conference on Information Security – Issues & Challenges, NCISIC 08, pp. 37–40, 2008.
53. Juels A., Luby M., Ostrovsky R. «Security of blind digital signatures» //Annual International Cryptology Conference. – Springer, Berlin, Heidelberg, 1997. – C. 150-164.
54. Kastner J., Loss J., Xu J. «On Pairing-Free Blind Signature Schemes in the Algebraic Group Model» //Cryptology ePrint Archive. – 2020. – T. 2020. – №. 1071.
55. Khater M. M., Al-Ahwal A., Selim M. M., Zayed H. H. «New Blind Signature Scheme Based on Modified ElGamal Signature for Secure Electronic Voting» //International Journal of Scientific & Engineering Research, 9(3), pp. 917–921, 2018.
56. Koblitz N., Menezes A. «Another look at non-standard discrete log and Diffie-Hellman problems» //Journal of Mathematical Cryptology. – 2008. – T. 2. – №. 4. – C. 311-326.
57. Krawczyk H., Bellare M., Canetti R. «HMAC: Keyed-Hashing for Message Authentication» //RFC 2104, RFC Editor, 1997.
58. Lysyanskaya A. «Security analysis of RSA-BSSA» //IACR International Conference on Public-Key Cryptography. Cham : Springer Nature Switzerland, 2023. pp. 251–280.
59. Mala H., Nezhadansari N. «New blind signature schemes based on the (elliptic curve) discrete logarithm problem» //ICCKE 2013. – IEEE, 2013. – C. 196-201.
60. van der Meer N. «Root Finding over Finite Fields for Secure Multiparty Computation» //Bachelor Thesis, Eindhoven University of Technology, 2021.
61. Mohammed E., Emarah A.E., El-Shennawy K. «A blind signature scheme based on ElGamal signature». //IEEE/AFCEA EUROCOMM 2000 Information Systems for Enhanced Public Safety and Security, pp. 51-53, 2000.
62. Moldovyan N.A. «Blind Signature Protocols from Digital Signature Standards» //International Journal of Network Security, Vol.13, No.1, PP.22–30, July 2011.
63. Nechaev V. I.«Complexity of a determinate algorithm for the discrete logarithm» //Mathematical Notes, 55(2):165–172, 1994.
64. Ohkubo M., Abe M. «Security of Some Three-move Blind Signature Schemes Reconsidered» //The 2003 Symposium on Cryptography and Information Security. Hamamatsu,Japan, 2003.
65. Paillier P. «Public-key cryptosystems based on composite degree residuosity classes»

//EUROCRYPT 1999, pp. 223–238.

66. Paquin C., Zaverucha G. «U-Prove Cryptographic Specification. V. 1.1» //Microsoft Corporation. 2013. https://www.microsoft.com/en-us/research/publication/u-prove-cryptographic-specification-v1-1-revision-3/
67. Pass R. «Limits of provable security from standard assumptions» //Proceedings of the forty-third annual ACM symposium on Theory of computing. – 2011. – C. 109-118.
68. Pointcheval D., Stern J. «Provably secure blind signature schemes» //International Conference on the Theory and Application of Cryptology and Information Security. – Berlin, Heidelberg : Springer Berlin Heidelberg, 1996. – C. 252–265.
69. Pointcheval D., Stern J. «Security arguments for digital signatures and blind signatures» //Journal of cryptology. – 2000. – T. 13. – C. 361–396.
70. Pointcheval D. «Strengthened security for blind signatures» //Lecture Notes in Computer Science, Volume 1403, Advances in Cryptology — EUROCRYPT'98, pp. 391–405, 1998.
71. Qin X., Cai C., Yuen T. H. One-more unforgeability of blind ecdsa //Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part II 26. – Springer International Publishing, 2021. – C. 313-331.
72. Ristenpart T., Yilek S. «When Good Randomness Goes Bad: Virtual Machine Reset Vulnerabilities and Hedging Deployed Cryptography» //NDSS, 2010.
73. Sahu R. A., Gini A., Pal A. «Supersingular Isogeny-Based Designated Verifier Blind Signature» //IACR Cryptol. ePrint Arch. – 2019. – T. 2019. – C. 1498.
74. Schnorr C. P. «Efficient identification and signatures for smart cards» //Conference on the Theory and Application of Cryptology. – Springer, New York, NY, 1989. – C. 239-252.
75. Schnorr C. P. «Security of blind discrete log signatures against interactive attacks» //ICICS 01, volume 2229 of LNCS, pages 1–12. Springer, Heidelberg, November 2001.
76. Schroder D., Unruh D. «Security of blind signatures revisited» //International Workshop on Public Key Cryptography. – Springer, Berlin, Heidelberg, 2012. – C. 662-679.
77. Shen, V. R., Chung, Y. F., Chen, T. S., Lin, Y. A. «A blind signature based on discrete logarithm problem» // International Journal of Innovative Computing, Information and Control, 7(9), pp. 5403–5416, 2011.
78. Srinath M., Chandrasekaran V. «Isogeny-based Quantum-resistant Undeniable Blind Signature Scheme» //IACR Cryptol. ePrint Arch. – 2016. – T. 2016. – C. 148.
79. Tan D. N., Nam H. N., Hieu M. N., Van H. N «New blind muti-signature schemes based on ECDLP» //International Journal of Electrical and Computer Engineering. – 2018. – T. 8. –

№. 2. – С. 1074.

80. Tan D. N., Nam H. N., Van H. N., Thi, L. T., Hieu M. N. «New blind mutisignature schemes based on signature standards» // 2017 International Conference on Advanced Computing and Applications (ACOMP), IEEE, pp. 23–27, 2017.
81. Tessaro S., Zhu C. «Short pairing-free blind signatures with exponential security» //Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cham, Springer International Publishing, pp. 782–811, 2022.
82. Vinberg E. B. «A course in algebra» //American Mathematical Soc, 56, 2003.
83. Wagner D. «A generalized birthday problem» //CRYPTO 2002, volume 2442 of LNCS, pp. 288–303. Springer, Heidelberg, August 2002.
84. Wang Y. «Password Protected Smart Card and Memory Stick Authentication against Off-Line Dictionary Attacks». IFIP international information security conference. Springer Berlin Heidelberg, pp. 489–500, 2012.
85. Yi X., Lam K. Y. «A new blind ECDSA scheme for bitcoin transaction anonymity» //Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. – 2019. – С. 613-620.
86. Zhang Y., He D., Zhang F., Huang X., Li D. «An efficient blind signature scheme based on SM2 signature algorithm» //LNCS, 12612. International Conference on Information Security and Cryptology, Springer, Cham, pp. 368–384, 2020.

Публикации автора по теме диссертации

Публикации в рецензируемых научных изданиях, индексируемых в базе ядра Российского индекса научного цитирования «eLibrary Science Index»

87. Akhmetzyanova L. R., Alekseev E. K., Babueva A. A., Smyshlyaev S. V. «On the (im)possibility of secure ElGamal blind signatures» //Математические вопросы криптографии. – 2023. – Т. 14. – №. 2. – С. 25–42.
88. Akhmetzyanova L. R., Alekseev E. K., Babueva A. A., Taraskin O. G. «On blindness of several ElGamal-type blind signatures» //Прикладная дискретная математика. – 2023. – №. 62. – С. 13–20.
89. Akhmetzyanova L. R., Babueva A. A., Bozhko A. A. «Blind signature as a shield against backdoors in smart-cards» //Прикладная дискретная математика. – 2024. – №. 63. – С. 49–64.
90. Ахметзянова Л. Р., Бабуева А. А. «О свойстве неподделываемости схемы подписи всле-

пую Шаума-Педерсена» //Прикладная дискретная математика. – 2024. – №. 65. – С. 41–65.

Публикации в рецензируемых научных изданиях, входящих в перечень ВАК Минобрнауки России:

91. Бабуева А. А. «О модификации схемы подписи Эль-Гамаля для применения в одном классе систем голосования, использующих механизм подписи вслепую» //International Journal of Open Information Technologies. – 2023. – Т. 11. – №. 5. – С. 15–21.