МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7 DOI 10.17223/20710410/65/3

О СВОЙСТВЕ НЕПОДДЕЛЫВАЕМОСТИ СХЕМЫ ПОДПИСИ ВСЛЕПУЮ ШАУМА — ПЕДЕРСЕНА

Л. Р. Ахметзянова, А. А. Бабуева

КриптоПро, г. Москва, Россия

E-mail: {lah, babueva}@cryptopro.ru

Анализируется свойство неподделываемости схемы подписи вслепую Шаума — Педерсена в условиях, когда нарушитель имеет возможность открывать параллельные сеансы протокола формирования подписи. Показано, что схема не обеспечивает свойство неподделываемости в сильном смысле, т. е. позволяет строить подделки для «старого» сообщения, которое было подписано легитимным образом в результате взаимодействия с подписывающим. Проведён анализ свойства неподделываемости в слабом смысле (задача нарушителя — построение подделки для нового сообщения). С помощью метода сведений получена оценка стойкости схемы относительно свойства слабой неподделываемости в модели с алгебраической группой и случайным оракулом. Полученная оценка позволяет выделить базовые задачи, сложность которых лежит в основе стойкости схемы.

Ключевые слова: *схема подписи вслепую, схема Шаума — Педерсена, ROS-атака.*

ON THE UNFORGEABILITY OF THE CHAUM — PEDERSEN BLIND SIGNATURE SCHEME

L. R. Akhmetzyanova, A. A. Babueva

CryptoPro, Moscow, Russia

The paper is devoted to the analysis of the unforgeability property of the Chaum — Pedersen blind signature scheme in case an adversary is able to initiate parallel sessions of the signature generation protocol. It is shown that the scheme does not ensure strong unforgeability, i.e., it allows to create the forgeries for “old” messages that were legitimately signed. An analysis of the weak unforgeability property (the adversary’s task is to create a forgery for a new message) is also conducted. Using the reduction method, we obtain a security bound on the weak unforgeability property in the algebraic group model and random oracle model. This estimation identifies the base problems whose complexity underpins the scheme security.

Keywords: *blind signature scheme, Chaum — Pedersen blind signature, ROS attack.*

Введение

Механизм подписи вслепую был впервые предложен Д. Шаумом в 1982 г. [1]. Формирование подписи вслепую представляет собой интерактивный протокол, выполняемый между подписывающей стороной (сервером) и клиентом. В результате клиент получает подпись для некоторого сообщения, при этом подписывающий не получает информации ни о сообщении, ни о сформированном значении подписи (свойство неотслеживаемости), а клиент не может сформировать корректное значение подписи без взаимодействия с подписывающим (свойство неподделываемости). Как показывает история развития схем подписи вслепую, построение схем на основе стандартных эллиптических кривых, обеспечивающих свойство неподделываемости в условиях, когда нарушитель имеет возможность открывать параллельные сеансы протокола формирования подписи, — нетривиальная задача. Заметим, что подобные условия актуально рассматривать для приложений, где множество клиентов единовременно подключаются к серверу выдачи подписи вслепую и при этом важна высокая скорость получения подписи. Ярким примером таких систем являются системы дистанционного электронного голосования [2].

Существующие работы. Классической схемой подписи вслепую на основе стандартных эллиптических кривых является схема Шнорра, предложенная в 1996 г. в [3]. Анализ стойкости данной схемы проведен в 2001 г. в [4]. Свойство неподделываемости было доказано в предположении сложности задачи ROS (Random inhomogeneities in a Overdetermined Solvable system of linear equations) в генерической модели со случайным оракулом. Задача ROS на протяжении 20 лет считалась сложной. Однако в 2020 г. был предложен полиномиальный алгоритм решения этой задачи [5], 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 группы точек эллиптической кривой.

Оказалось [5, 6], что аналогичная атака применима не только к схеме подписи вслепую Шнорра, но и к ряду других схем на основе стандартных эллиптических кривых: схеме Окамото— Шнорра [7], схеме Абе [8], обеспечивающей частичную неотслеживаемость, схемам на основе уравнения Эль-Гамаля [6], а также схеме Брандса [9]. При этом для схемы Брандса атака позволяет строить подделки только для одного и того же «номера аккаунта». Схема Брандса, в свою очередь, построена на основе схемы подписи вслепую Шаума — Педерсена [10]. Для схемы Шаума— Педерсена в литературе не представлено ни атак, ни формального обоснования свойства неподделываемости, поэтому вопрос её стойкости является открытым.

Формальное обоснование свойства неподделываемости для схем подписи вслепую сопряжено с некоторыми трудностями. В литературе представлен ряд работ, которые показывают невозможность обоснования стойкости схемы подписи вслепую Шнорра в стандартных моделях безопасности [11, 12], а именно: без идеализаций криптографических примитивов (модель со случайным оракулом [13]) и ограничений множества рассматриваемых нарушителей (модели с генерической [14] или алгебраической [15] группой). Более того, в работе [16] показано, что для схемы подписи вслепую Шнорра [3], Окамото — Шнорра [7] и схемы Брандса [9] невозможно построить сведение с использованием всех известных техник доказательств для таких схем (например, Random oracle replay и forking lemma) на основе предположений о сложности базовых задач даже в модели со случайным оракулом. Существующие сведения верны толь-

ко в моделях с генерической или алгебраической группой со случайным оракулом, которые являются упрощением стандартных моделей. Насколько известно авторам настоящей работы, на сегодняшний день не предложено каких-либо принципиально новых техник построения сведений для схем подписи вслепую.

С момента публикации ROS-атаки в литературе было предложено несколько схем подписи вслепую [17–19] на основе стандартных эллиптических кривых, стойких (при некоторых предположениях) даже в том случае, когда нарушитель имеет возможность открывать параллельные сеансы протокола формирования подписи. Оценка стойкости всех этих схем получена в модели со случайным оракулом, а при обосновании схем [18, 19] используется также модель с алгебраической группой.

Наши результаты. В настоящей работе проводится анализ схемы Шаума — Педерсена с точки зрения обеспечения свойства неподделываемости при наличии у нарушителя возможности открывать несколько параллельных сеансов протокола формирования подписи.

Показано, что схема Шаума — Педерсена не обеспечивает свойство неподделываемости в сильном смысле, т. е. позволяет строить подделки для «старого» сообщения, которое было подписано легитимным образом в результате взаимодействия с подписывающим. Построена 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, это означает, что схема Шаума — Педерсена потенциально обеспечивает свойство неподделываемости в слабом смысле (задача нарушителя — построение подделки для нового сообщения).

Проведён формальный анализ свойства неподделываемости в слабом смысле. Построено сведение в модели с алгебраической группой и случайным оракулом, которое демонстрирует, что достаточным условием стойкости схемы является сложность решения двух задач: задачи REPR и SOMDL. Задача REPR является модификацией задачи Representation, определённой в работе [9], её сложность также является необходимым условием стойкости схемы Шаума — Педерсена. Задача SOMDL является новой задачей, определённой для группы точек эллиптической кривой. Показано, что настоящая задача не сложнее задачи OMDL (One-More Discrete Logarithm, введена в [20]) и задачи SDL (определена в [21] как задача q-dlog). Для задачи SDL показано также, что её сложность является необходимым условием стойкости схемы Шаума — Педерсена. Таким образом, выделены базовые задачи, дальнейшее изучение которых необходимо для получения больших гарантий безопасности схемы Шаума — Педерсена.

Структура работы. В п. 1 даются основные определения и обозначения, п. 2 посвящён формальному определению схемы подписи вслепую Шаума — Педерсена. В п. 3 вводится свойство неподделываемости для схем подписи вслепую. Пункты 4 и 5 посвящены анализу свойства неподделываемости в сильном и слабом смысле соответственно. В приложении A формально определяется модель wUF, а в приложении B приводится доказательство теоремы 1 о стойкости этой модели.

1. Определения и обозначения

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 $1/x$; $\mathbb{Z}_p^*$ — множество $\mathbb{Z}_p$ без нулевого элемента, т. е. мультипликативная группа поля $\mathbb{Z}_p$.

Группа точек эллиптической кривой, определённой над полем $\mathbb{Z}_p$, простого порядка q обозначается через $\mathbb{G}$, точка эллиптической кривой порядка q — через P, нулевая точка кривой — через $\mathcal{O}$; $\mathbb{G}^*$ — множество точек кривой без нулевой точки; H — хеш-

функция, отображающая двоичные строки в элементы $\mathbb{Z}_q^*$; $\mathcal{H} : \{0,1\}^* \to \mathbb{G}^*$ — хеш-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 $x \xleftarrow{U} X$ означает, что элемент x выбирается из множества X случайно в соответствии с равновероятным распределением, будем называть такой элемент x случайным. Событие, что алгоритм A вернул значение val в качестве результата работы, обозначается через $A \to val$ ($val \leftarrow A$).

Определение 1. Схема подписи вслепую BS задается следующими алгоритмами:

— $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}()$: алгоритм выработки ключей, возвращающий пару ключей $(\mathsf{sk}, \mathsf{pk})$, где sk — ключ подписи; pk — 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 ключом подписи sk и ключом проверки подписи pk, и клиентом, обладающим сообщением m и ключом проверки подписи pk; подписывающий выдаёт $b = 1$, если взаимодействие успешно завершилось, и $b = 0$ в противном случае; клиент выдаёт 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 нуль в противном случае.

При этом для любой пары ключей $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}()$ и 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 $b = b' = 1$.

2. Схема Шаума — Педерсена

Приведём описание алгоритмов, задающих работу схемы подписи вслепую Шаума — Педерсена. Далее будем называть эту схему $\mathsf{CP\text{-}BS}$.

Оригинальное описание схемы Шаума — Педерсена [10] представлено для мультипликативной группы конечного поля, при этом подписываемое сообщение представляет собой элемент группы. В настоящей работе мы приводим описание для группы точек эллиптической кривой, при этом для перевода сообщения в элемент группы, т. е. точку кривой, мы предлагаем использовать функцию хеширования $\mathcal{H}$ в группу точек эллиптической кривой. Заметим, что в литературе известны подходы к построению таких функций (см., например, [22]).

Алгоритм выработки ключей задаётся следующим образом:

$\mathsf{CP\text{-}BS.KGen}()$

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

$Q \leftarrow dP$

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, вычисляя $M = \alpha^{-1} M'$ 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 [10] для значений $\mathrm{DLog}_P Q = \mathrm{DLog}_M Z$, при этом для вычисления значения c (challenge в протоколе доказательства Шаума — Педерсена) клиент маскирует все значения, полученные от сервера. Значение $Z' = \alpha Z$ и сформированное доказательство (в маскированном виде) составляют подпись.

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 $\mathrm{DLog}_{M'} Z' = \mathrm{DLog}_{\alpha M}(\alpha Z) = \mathrm{DLog}_M Z$.

Протокол формирования и алгоритм проверки подписи формально определены на рис. 1 и 2 соответственно.

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** $\perp$
		$u, v \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
		$A' \leftarrow uA + vP$
		$B' \leftarrow u\alpha B + vM'$
		$Z' \leftarrow \alpha Z$
		$c' \leftarrow H(M' \Vert Z' \Vert A' \Vert B')$
	$\xleftarrow{\quad c \quad}$	$c \leftarrow c' u^{-1}$
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. Протокол формирования подписи в схеме CP-BS

CP-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. 2. Алгоритм проверки подписи в схеме CP-BS

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.

3. Свойства безопасности схем подписи вслепую

Для схем подписи вслепую традиционно рассматривают [1, 3] следующие два свойства безопасности:

— неотслеживаемость (blindness): нарушитель, выступающий в роли подписывающего, в результате выполнения протокола формирования подписи не получает никакой информации о паре (сообщение, подпись), сформированной клиентом;
— неподделываемость (unforgeability): нарушитель, выступающий в роли клиента, может сформировать корректную подпись только в результате успешного взаимодействия с подписывающим.

Поскольку настоящая работа посвящена анализу свойства неподделываемости, рассмотрим подробнее именно данное свойство.

Возможности нарушителя. Нарушителю, выступающему в роли клиента, предоставляется возможность получать от подписывающего корректные подписи для адаптивно выбираемых им сообщений, при этом предполагается, что подписывающий функционирует корректным образом. Наиболее сильной возможностью нарушителя, которую целесообразно рассматривать, является проведение атаки с параллельными сеансами: нарушитель может начинать выполнение новых сеансов протокола формирования подписи до завершения предыдущих.

Угроза. Угроза формализуется с помощью понятия «ещё 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.

4. Анализ безопасности относительно свойства сильной неподделываемости

В работе [5] построена атака на свойство неподделываемости для схемы подписи вслепую Брандса [9], применимая в модели с параллельными сеансами и позволяющая сформировать $(\ell+1)$ подпись для одного и того же «номера аккаунта» в результате $\ell \geqslant \lceil \log q \rceil$ успешных взаимодействий с подписывающим.

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 её применимости в случае формирования $(\ell+1)$ подписи для различных сообщений.

Атака типа ROS. Пусть нарушитель $\mathcal{A}$:

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

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

5) Определяет $\rho_0, \rho_1, \ldots, \rho_\ell$ как коэффициенты перед x_i в выражении

$$\sum_{i=0}^{\ell-1} 2^i \frac{x_i - c_{i0}}{c_{i1} - c_{i0}} = \sum_{i=0}^{\ell-1} \rho_i x_i + \rho_\ell.$$

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 $c'_\ell = \sum\limits_{i=0}^{\ell-1} 2^i b_i$.
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, $c'_\ell = \sum\limits_{i=0}^{\ell-1} \rho_i c_i + \rho_\ell$.
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

$$s_i P = A_i + c_i Q, \quad s_i M = B_i + c_i Z.$$

12) Полагает $s'_i = u_i s_i, , 0 \leqslant i \leqslant \ell-1$.
13) Полагает $s'_\ell = \sum\limits_{i=0}^{\ell-1} \rho_i s_i$.
14) Выдаёт $\{\big(m, (s'_i, c'_i, Z)\big) : i = 0, \ldots, \ell\}$.

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

Условие $\ell \geqslant \lceil \log q \rceil$ необходимо для того, чтобы было возможно осуществить шаг 8.

В настоящей атаке нарушитель не использует значения маскирующих факторов α и v (полагает их равными единице и нулю соответственно), в связи с этим пары (сообщение, подпись), построенные на шаге 14, можно соотнести с соответствующими стенограммами протокола, т. е. для них не выполняется свойство неотслеживаемости. Однако представляется, что атака может быть расширена с целью обеспечения неотслеживаемости для сформированных пар.

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 $Z' = dM'$. Для построения подделки для 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. Нетрудно 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 важно формировать точку M' таким образом, чтобы её дискретный логарифм был неизвестен нарушителю. Именно поэтому для формирования M' в схеме Шаума — Педерсена предлагается использовать функцию хеширования в кривую.

Замечание 1. В оригинальной схеме подписи вслепую Брандса [9], а также в модификации данной схемы для группы точек эллиптической кривой, используемой в системе U-Prove [23], 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, определённой

в работе [16], 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 атака неприменима из-за сложности построения точек Z_ℓ, B_ℓ, для которых выполнено условие

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

5. Анализ безопасности относительно свойства слабой неподделываемости

Получим оценку стойкости схемы Шаума — Педерсена относительно свойства слабой неподделываемости с помощью метода сведений.

Сведение удалось построить в модели wUF (weak UnForgeability), см. формальное определение в Приложении А, 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 введена в [13] 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 хеш-функции H и $\mathcal{H}$ моделируются как случайные оракулы. Анализ в данной модели можно 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 предложена в работе [15]. 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 передаёт набор коэффициентов $z = (z_1, \ldots, z_n)$, таких, что $Z = \sum_{i=1}^{n} z_i X_i$. Анализ в данной модели можно интерпретировать следующим образом: при получении оценки стойкости не рассматриваются методы криптоанализа, использующие структурные особенности конкретной группы для формирования новых элементов группы.

5.1. 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 2. Для нарушителя $\mathcal{A}$, группы $\mathbb{G}$, параметров ℓ и k положим

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

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

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

$x_1,\ldots,x_{\ell+1} \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
$ctr_1, ctr_2 \leftarrow 0$
$(x'_1,\ldots,x'_{\ell+1}) \leftarrow \mathcal{A}^{O_1,O_2}(x_1P,\ldots,x_{\ell+1}P)$
return $(x_1 = x'_1) \wedge \ldots \wedge (x_{\ell+1} = x'_{\ell+1})$

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_iY

Oracle $O_2(Y)$

if $ctr_2 > \ell$: **return** $\perp$
if $Y \notin \mathbb{G}$: **return** $\perp$
$ctr_2 \leftarrow ctr_2 + 1$
return $\mathrm{DLog}_P(Y)$

Нарушитель получает на вход набор точек $x_1P,\ldots,x_{\ell+1}P$, 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. Эти ограничения контролируются с помощью счётчиков 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_iY. 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.

О соотношении с другими задачами. В результате обзора существующих в литературе задач для конечных групп найдены две наиболее «близкие» к задаче SOMDL: задачи SDL и OMDL. Определим их формально.

Задача SDL (*Strong Discrete Logarithm*) определена в работе [21] как задача q-dlog и является модификацией задачи SDH (*Strong Diffie — Hellman*), предложенной в [24], её параметром является значение $s \in \mathbb{N}$.

Определение 3. Для нарушителя $\mathcal{A}$, группы $\mathbb{G}$ и параметра s положим

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

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

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

$x \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
$x' \leftarrow \mathcal{A}(xP,\ldots,x^sP)$
return $(x = x')$

Нарушитель получает на вход набор точек $xP,\ldots,x^sP$, 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 $(1, x_1P)$, а потом запросы вида $(1, Y)$, где Y — ответ оракула O_1 на предыдущий запрос, нарушитель может накопить значения $x_1P, x_1^2P,\ldots,x_1^{k+1}P$, что аналогично получению на вход таких значений в задаче SDL с параметром $s = k+1$. Отсюда следует

Утверждение 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 (*One-More Discrete Logarithm*) предложена в работе [20], её параметром является значение $\ell \in \mathbb{N}$.

Определение 4. Для нарушителя $\mathcal{A}$, группы $\mathbb{G}$ и параметра ℓ положим

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

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

$\mathbf{Exp}^{\mathrm{OMDL}}_{\mathbb{G},\ell}(\mathcal{A})$	Oracle $\mathrm{DLog}(Y)$
$x_1,\ldots,x_{\ell+1} \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$	**if** $ctr > \ell$: **return** $\perp$
$ctr \leftarrow 0$	**if** $Y \notin \mathbb{G}$: **return** $\perp$
$(x'_1,\ldots,x'_{\ell+1}) \leftarrow \mathcal{A}^{\mathrm{Dlog}}(x_1P,\ldots,x_{\ell+1}P)$	$ctr \leftarrow ctr + 1$
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 $x_1P,\ldots,x_{\ell+1}P$, 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, это ограничение контролируется с помощью счётчика ctr. Заметим, что оракул в задаче OMDL в точности совпадает с оракулом O_2 в задаче SOMDL. Отсюда следует

Утверждение 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 и 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

Эта задача — модификация задачи Representation [9], её параметром является значение $s \in \mathbb{N}$.

Определение 5. Для нарушителя $\mathcal{A}$, группы $\mathbb{G}$ и параметра s положим

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

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

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

$x_1, \ldots, x_s \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$

$(\alpha_1, \ldots, \alpha_s, \beta) \leftarrow \mathcal{A}(x_1 P, \ldots, x_s P)$

return $(\alpha_1 x_1 + \ldots + \alpha_s x_s + \beta = 0) \wedge (\exists i : \alpha_i \neq 0)$

Нарушитель получает на вход набор точек $x_1 P, \ldots, x_s P$, 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 $(-\beta)$.

5.2. Оценка стойкости в модели wUF

Через $\mathsf{Adv}^{\mathrm{wUF}}_{\mathsf{CP\text{-}BS}}(\mathcal{A})$ будем обозначать преимущество нарушителя $\mathcal{A}$ для схемы CP-BS в модели wUF со случайным оракулом и алгебраической группой (вероятность построения подделки).

Теорема 1. Для любого нарушителя $\mathcal{A}$ для схемы CP-BS в модели wUF с алгебраической группой, делающего не более t и ℓ запросов к оракулам Sign_1 и Sign_2 соответственно и не более q_1 и q_2 запросов к случайным оракулам, моделирующим работу хеш-функций $\mathcal{H}$ и H соответственно, существуют 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

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

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

Доказательство теоремы 1 приведено в Приложении В.

Интерпретация оценки. Полученная оценка стойкости свидетельствует о том, что сложность задач SOMDL и REPR является достаточным условием стойкости схемы CP-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 CP-BS.

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

Для задачи SOMDL на текущий момент неизвестно более эффективных методов решения, чем за счёт решения либо задачи SDL, либо задачи OMDL. Для задачи OMDL неизвестно более эффективных методов решения, чем решение задачи дискретного логарифмирования [25]. Для задачи SDL известен метод решения, вычислительная трудоёмкость которого ниже, чем трудоёмкость известных методов решения задачи дискретного логарифмирования, — метод, предложенный в [26] для случая, когда параметр s является делителем $(q-1)$. Он требует $T_{\mathrm{op}} = c_1 \log q\,(\sqrt{q/s} + \sqrt{s})$ вычислений групповых операций и $T_{\mathrm{mem}} = c_2 \max(\sqrt{q/s}, \sqrt{s})$ памяти, где c_1, c_2 — константы, зависящие только от используемой модели вычислений.

Заметим, что метод [26] 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 — значения $M_i = Z_{i-1} = dM_{i-1} = d^iP$, $2 \leqslant i \leqslant t$, 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 [26], 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 $s_m \leqslant 2^{64}+1$. В таблице приведены значения s_m и параметры метода для стандартизированных в России эллиптических кривых простого порядка, определённых в [27].

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 лучших методов решения, чем решение задачи дискретного логарифмирования [9]. Заметим, что решение задачи дискретного логарифмирования хотя бы для одной точки $M'_i = \mathcal{H}(m_i)$, $1 \leqslant i \leqslant q_1$, позволяет реализовать атаку типа ROS, описанную в п. 4.

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 хеш-функции H.

Вывод. Полученная оценка указывает, что достаточным условием стойкости схемы CP-BS в модели wUF со случайным оракулом и алгебраической группой является сложность задач SOMDL и REPR. На данный момент не удалось доказать, что сложность задачи SOMDL является необходимым условием стойкости схемы в указанной модели. Исходя из рассуждений выше, необходимым условием является сложность другой задачи в группе — задачи SDL. Для задачи SDL, в свою очередь, в [21] 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 пока свести только к решению SDL, могут быть обнаружены другие методы, решающие задачу SOMDL и приводящие к взлому схемы CP-BS. Таким образом, из-за слабой изученности как необходимых, так и достаточных для стойкости схемы базовых задач требуется проведение дополнительных исследований их сложности.

ЛИТЕРАТУРА

1. *Chaum D.* Blind signatures for untraceable payments // D. Chaum, R. L. Rivest, and A. T. Sherman (eds.). Advances in Cryptology. Boston, MA: Springer, 1983. P. 199–203.
2. *Fujioka A., Okamoto T., and Ohta K.* A practical secret voting scheme for large scale elections // LNCS. 1993. V. 718. P. 244–251.
3. *Pointcheval D. and Stern J.* Provably secure blind signature schemes // LNCS. 1996. V. 1163. P. 252–265.
4. *Schnorr C. P.* Security of blind discrete log signatures against interactive attacks // LNCS. 2001. V. 2229. P. 1–12.
5. *Benhamouda F., Lepoint T., Loss J., et al.* On the (in) security of ROS // J. Cryptology. 2022. V. 35. No. 4. Article 25.
6. *Akhmetzyanova L., Alekseev E., Babueva A., and Smyshlyaev S.* On the (im)possibility of secure ElGamal blind signatures // Матем. вопр. криптогр. 2023. Т. 14. № 2. С. 25–42.
7. *Pointcheval D. and Stern J.* Security arguments for digital signatures and blind signatures // J. Cryptology. 2000. V. 13. P. 361–396.
8. *Abe M. and Okamoto T.* Provably secure partially blind signatures // LNCS. 2000. V. 1880. P. 271–286.
9. *Brands S.* Untraceable off-line cash in wallets with observers // LNCS. 1994. V. 773. P. 302–318.
10. *Chaum D. and Pedersen T. P.* Wallet databases with observers // LNCS. 1993. V. 740. P. 89–105.
11. *Fischlin M. and Schroder D.* On the impossibility of three-move blind signature schemes // LNCS. 2010. V. 6110. P. 197–215.
12. *Pass R.* Limits of provable security from standard assumptions // Proc. 43rd Ann. ACM Symp. Theory Computing. San Jose, California, USA, 2011. P. 109–118.
13. *Bellare M. and Rogaway P.* Random oracles are practical: A paradigm for designing efficient protocols // Proc. CCS'93. Fairfax, Virginia, USA, 1993. P. 62–73.
14. *Nechaev V. I.* Complexity of a determinate algorithm for the discrete logarithm // Math. Notes. 1994. V. 55. No. 2. P. 165–172.
15. *Fuchsbauer G., Kiltz E., and Loss J.* The algebraic group model and its applications // LNCS. 2018. V. 10992. P. 33–62.
16. *Baldimtsi F. and Lysyanskaya A.* On the security of one-witness blind signature schemes // LNCS. 2013. V. 8270. P. 82–99.
17. *Chairattana-Apirom R., Tessaro S., and Zhu C.* Pairing-Free Blind Signatures from CDH Assumptions. Cryptology ePrint Archive. 2023. Paper 2023/1780. `https://eprint.iacr.org/2023/1780`.
18. *Crites E., Komlo C., Maller M., et al.* Snowblind: A threshold blind signature in pairing-free groups // LNCS. 2023. V. 14081. P. 710–742.
19. *Tessaro S. and Zhu C.* Short pairing-free blind signatures with exponential security // LNCS. 2022. V. 13276. P. 782–811.
20. *Bellare M., Namprempre C., Pointcheval D., and Semanko M.* The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme // J. Cryptology. 2003. V. 16. No. 3. P. 185–215.
21. *Bauer B., Fuchsbauer G., and Loss J.* A classification of computational assumptions in the algebraic group model // LNCS. 2020. V. 12171. P. 121–151.
22. *Faz-Hernandez A., Scott S., Sullivan N., et al.* Hashing to Elliptic Curves. `https://datatracker.ietf.org/doc/rfc9380/`.

23. *Paquin C. and Zaverucha G.* U-Prove Cryptographic Specification V1.1 (Revision 3). https://www.microsoft.com/en-us/research/publication/u-prove-cryptographic-specification-v1-1-revision-3/. 2013.
24. *Boneh D. and Boyen X.* Short signatures without random oracles // LNCS. 2004. V. 3027. P. 56–73.
25. *Koblitz N. and Menezes A.* Another look at non-standard discrete log and Diffie — Hellman problems // J. Math. Cryptology. 2008. V. 2. No. 4. P. 311–326.
26. *Cheon J. H.* Security analysis of the strong Diffie — Hellman problem // LNCS. 2006. V. 4004. P. 1–11.
27. Р 1323565.1.024-2019 «Информационная технология. Криптографическая защита информации. Параметры эллиптических кривых для криптографических алгоритмов и протоколов». М.: Стандартинформ, 2019.
28. *Van der Meer N.* Root Finding over Finite Fields for Secure Multiparty Computation. Bachelor Thesis. Eindhoven University of Technology, 2021.

REFERENCES

1. *Chaum D.* Blind signatures for untraceable payments. D. Chaum, R. L. Rivest, and A. T. Sherman (eds.). Advances in Cryptology, Boston, MA, Springer, 1983, pp. 199–203.
2. *Fujioka A., Okamoto T., and Ohta K.* A practical secret voting scheme for large scale elections. LNCS, 1993, vol. 718, pp. 244–251.
3. *Pointcheval D. and Stern J.* Provably secure blind signature schemes. LNCS, 1996, vol. 1163, pp. 252–265.
4. *Schnorr C. P.* Security of blind discrete log signatures against interactive attacks. LNCS, 2001, vol. 2229, pp. 1–12.
5. *Benhamouda F., Lepoint T., Loss J., et al.* On the (in) security of ROS. J. Cryptology, 2022, vol. 35, no. 4, Article 25.
6. *Akhmetzyanova L., Alekseev E., Babueva A., and Smyshlyaev S.* On the (im)possibility of secure ElGamal blind signatures. Matem. Vopr. Kriptogr., 2023, vol. 14, no. 2, pp. 25–42.
7. *Pointcheval D. and Stern J.* Security arguments for digital signatures and blind signatures. J. Cryptology, 2000, vol. 13, pp. 361–396.
8. *Abe M. and Okamoto T.* Provably secure partially blind signatures. LNCS, 2000, vol. 1880, pp. 271–286.
9. *Brands S.* Untraceable off-line cash in wallets with observers. LNCS, 1994, vol. 773, pp. 302–318.
10. *Chaum D. and Pedersen T. P.* Wallet databases with observers. LNCS, 1993, vol. 740, pp. 89–105.
11. *Fischlin M. and Schroder D.* On the impossibility of three-move blind signature schemes. LNCS, 2010, vol. 6110, pp. 197–215.
12. *Pass R.* Limits of provable security from standard assumptions. Proc. 43rd Ann. ACM Symp. Theory Computing, San Jose, California, USA, 2011, pp. 109–118.
13. *Bellare M. and Rogaway P.* Random oracles are practical: A paradigm for designing efficient protocols. Proc. CCS'93, Fairfax, Virginia, USA, 1993, pp. 62–73.
14. *Nechaev V. I.* Complexity of a determinate algorithm for the discrete logarithm. Math. Notes, 1994, vol. 55, no. 2, pp. 165–172.
15. *Fuchsbauer G., Kiltz E., and Loss J.* The algebraic group model and its applications. LNCS, 2018, vol. 10992, pp. 33–62.
16. *Baldimtsi F. and Lysyanskaya A.* On the security of one-witness blind signature schemes. LNCS, 2013, vol. 8270, pp. 82–99.

17. *Chairattana-Apirom R., Tessaro S., and Zhu C.* Pairing-Free Blind Signatures from CDH Assumptions. Cryptology ePrint Archive, 2023, Paper 2023/1780, https://eprint.iacr.org/2023/1780.
18. *Crites E., Komlo C., Maller M., et al.* Snowblind: A threshold blind signature in pairing-free groups. LNCS, 2023, vol. 14081, pp. 710–742.
19. *Tessaro S. and Zhu C.* Short pairing-free blind signatures with exponential security. LNCS, 2022, vol. 13276, pp. 782–811.
20. *Bellare M., Namprempre C., Pointcheval D., and Semanko M.* The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. J. Cryptology, 2003, vol. 16, no. 3, pp. 185–215.
21. *Bauer B., Fuchsbauer G., and Loss J.* A classification of computational assumptions in the algebraic group model. LNCS, 2020, vol. 12171, pp. 121–151.
22. *Faz-Hernandez A., Scott S., Sullivan N., et al.* Hashing to Elliptic Curves. https://datatracker.ietf.org/doc/rfc9380/.
23. *Paquin C. and Zaverucha G.* U-Prove Cryptographic Specification V1.1 (Revision 3). https://www.microsoft.com/en-us/research/publication/u-prove-cryptographic-specification-v1-1-revision-3/, 2013.
24. *Boneh D. and Boyen X.* Short signatures without random oracles. LNCS, 2004, vol. 3027, pp. 56–73.
25. *Koblitz N. and Menezes A.* Another look at non-standard discrete log and Diffie — Hellman problems. J. Math. Cryptology, 2008, vol. 2, no. 4, pp. 311–326.
26. *Cheon J. H.* Security analysis of the strong Diffie — Hellman problem. LNCS, 2006, vol. 4004, pp. 1–11.
27. R 1323565.1.024-2019 «Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Parametry ellipticheskih krivyh dlya kriptograficheskih algoritmov i protokolov» [R 1323565.1.024-2019 "Information Technology. Cryptographic Data Security. Elliptic Curve Parameters for the Cryptographic Algorithms and Protocols"]. Moscow, Standartinform Publ., 2019. (in Russian)
28. *Van der Meer N.* Root Finding over Finite Fields for Secure Multiparty Computation. Bachelor Thesis, Eindhoven University of Technology, 2021.

Приложение А. Модель wUF (weak UnForgeability)

Определим формально модель wUF, в которой предполагается, что нарушитель имеет возможность проводить атаку с параллельными сеансами, а его задача — сформировать $(\ell + 1)$ корректную пару (сообщение, подпись) для различных сообщений в результате ℓ успешных взаимодействий с подписывающим.

Данную модель определим для двухраундовых схем подписи вслепую. Протокол формирования подписи $\langle \mathsf{Sign}, \mathsf{User} \rangle$ в таких схемах можно формально задать следующим образом:

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

Здесь $msg_{role,i}$ означает сообщение с порядковым номером i, отправляемое стороной $role \in \{U, S\}$ в рамках протокола, а переменная $state_{role,i}$ позволяет стороне взаимо-

действия *role* сохранить некоторое внутреннее состояние на i-м раунде и использовать его на последующем раунде.

Определение 6. Для двухраундовой схемы подписи вслепую BS положим

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

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

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

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}(\,)$
2 : $\mathsf{sid}, \ell \leftarrow 0, \mathcal{I}_{\mathsf{fin}} \leftarrow \varnothing$
3 : $\{(m^*_k, \sigma^*_k)\}_{k=1}^{\ell+1} \leftarrow \mathcal{A}^{\mathrm{Sign}_1, \mathrm{Sign}_2}(\mathsf{pk})$
4 : **return** $(\forall k_1 \neq k_2 \in \{1, \ldots, \ell+1\} : m^*_{k_1} \neq m^*_{k_2}$
5 : $\quad \wedge \forall k \in \{1, \ldots, \ell+1\} : \mathsf{BS.Verify}(\mathsf{pk}, m^*_k, \sigma^*_k) = 1)$

Oracle $\mathrm{Sign}_1(msg)$

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 $\mathrm{Sign}_2(j, msg)$

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'

Приложение B. Доказательство теоремы 1

Пусть $\mathcal{A}$ — нарушитель для схемы подписи CP-BS в модели wUF. У него есть доступ к четырём оракулам: $\mathrm{Sign}_1, \mathrm{Sign}_2, \mathrm{RO}_1, \mathrm{RO}_2$. Оракул RO_1 моделирует работу хеш-функции $\mathcal{H}$ и возвращает точки эллиптической кривой (за исключением нулевой точки). Оракул RO_2 моделирует работу хеш-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 откроет t сеансов. В результате своей работы нарушитель $\mathcal{A}$ возвращает $(\ell+1)$ пару (сообщение, подпись).

Ш а г 1. Нарушитель $\mathcal{A}$ для любой точки, которую он выдаёт, обязан предоставить разложение по элементам группы, которые появлялись до этого момента в рамках эксперимента (в силу того, что рассматривается модель с алгебраической группой).

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

Нарушитель $\mathcal{A}$ подаёт точки M_j, $1 \leqslant j \leqslant t$, на вход оракулу Sign_1, точки M'_j, Z'_j, A'_j, B'_j, $j = 1 \ \ldots, q_2$, — 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

$$\bigl(\alpha_i, \beta_i, \{\gamma_{ij} : j = 1, \ldots, t\}, \{\sigma_{ij} : j = 1, \ldots, t\}, \{\eta_{ij} : j = 1, \ldots, t\}, \{\xi_{ij} : j = 1, \ldots, q_1\}\bigr),$$

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

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

Зафиксируем также наборы коэффициентов

$$\big(\hat\alpha_j, \hat\beta_j, \{\hat\gamma_{ji} : i{=}1,\dots,j-1\}, \{\hat\sigma_{ji} : i{=}1,\dots,j-1\}, \{\hat\eta_{ji} : i{=}1,\dots,j-1\}, \{\hat\xi_{ji} : i{=}1,\dots,q_1\}\big),$$

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

$$M_j = \hat\alpha_j P + \hat\beta_j Q + \sum_{i=1}^{j-1} \hat\gamma_{ji} A_i + \sum_{i=1}^{j-1} \hat\sigma_{ji} B_i + \sum_{i=1}^{j-1} \hat\eta_{ji} Z_i + \sum_{i=1}^{q_1} \hat\xi_{ji} M'_i.$$

Ш а г 2. Пусть нарушитель $\mathcal{A}$ выдал некоторую корректную пару $(m, (s', c', Z'))$ в качестве подделки. Нарушитель $\mathcal{A}$ мог делать 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 вернёт 1, не будет превосходить $(q-1)^{-1}$ для конкретного значения подделки. Поскольку количество подделок равно $\ell + 1$, то суммарная вероятность того, что не был сделан хотя бы один запрос, не превышает $(\ell+1)/(q-1)$.

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

Ш а г 3. Серверная часть протокола формирования подписи схемы подписи вслепую Шаума — Педерсена в точности повторяет действия доказывающего в протоколе доказательства с нулевым разглашением Шаума — Педерсена [10]. Это протокол доказательства равенства двух дискретных логарифмов:

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

Проверка доказательства осуществляется аналогично проверке подписи в схеме Шаума — Педерсена.

Аналогично протоколу доказательства Шаума — Педерсена, можно показать, 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 равенства

$$k_1 = s' - c'd, \quad k_2 = s' - c'x.$$

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

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 $q_2/(q-1)$. Таким образом, с вероятностью не больше $q_2/(q-1)$ среди точек Z'_i, $1 \leqslant i \leqslant \ell+1$, в составе всех подделок есть хотя бы одна точка, не равная dM'_i.

Далее рассмотрим 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, мы перешли от исходного эксперимента $\mathbf{Exp}(\mathcal{A}) = \mathbf{Exp}^{\mathrm{wUF}}_{\mathsf{CP\text{-}BS}}(\mathcal{A})$ к модифицированному эксперименту $\mathbf{Exp}'$, работающему так же, как исходный эксперимент, за исключением наступления определённых событий (см. шаги 2–3). Разницу преимуществ нарушителя в исходном и модифицированном экспериментах можно оценить как

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

Будем строить двух нарушителей: нарушителя $\mathcal{B}$ для задачи SOMDL и нарушителя $\mathcal{C}$ для задачи REPR, использующих нарушителя $\mathcal{A}$ в качестве чёрного 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.

Построение 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:

$\mathrm{SimRO}_1(m)$	$\mathrm{SimRO}_2(str)$
1 : **if** $m \in \Pi_1$:	1 : **if** $str \in \Pi_2$:
2 : **return** $\Pi_1(m)P$	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 определённой функции повторяется, то нарушитель $\mathcal{B}$ возвращает то же значение, что и в прошлый раз, 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:

$\mathrm{SimSign}_1(M)$	$\mathrm{SimSign}_2(i, c)$
1 : // i-й запрос нарушителя $\mathcal{A}$	1 : $Y \leftarrow A_i + cQ$
2 : $A \leftarrow A_i$	2 : $s \leftarrow O_2(Y)$
3 : $B \leftarrow O_1(i, M)$	3 : **return** s
4 : $Z \leftarrow O_1(t + 1, M)$	
5 : **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) $\big(m_i, (s'_i, c'_i, Z'_i)\big)$, $i = 1, \ldots, \ell+1$. 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 соотношение

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

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 — это представление, поданное нарушителем $\mathcal{A}$ при запросе к RO_2.

Заметим, что для всех завершённых 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 незавершённых 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 незавершённых сеансов с некоторым номером j нарушитель $\mathcal{B}$ после окончания работы нарушителя $\mathcal{A}$ подаёт запросы вида A_j к своему оракулу O_2. Тогда он получает в ответ соответствующие значения k_j и может в явном виде представить все точки A_j и B_j из незавершённых сеансов как $k_j P$ и $k_j M_j$ соответственно. В результате количество запросов нарушителя $\mathcal{B}$ к оракулу O_2 равно t.

Заметим также, что все значения Z_j в разложении (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 $\mathsf{З} \subseteq \{1, \ldots, t\}$ множество номеров завершённых сеансов, $\mathsf{НЗ} \subseteq \{1, \ldots, t\}$ — незавершённых. Рассмотрим уравнение (1) относительно неизвестного d:

$$\begin{gathered} s'_i x_i P - c'_i d x_i P = B'_i = \\ = \alpha_i P + \beta_i dP + \sum_{j \in \mathsf{З}} \gamma_{ij} (s_j P - c_j dP) + \sum_{j \in \mathsf{НЗ}} \gamma_{ij} k_j P + \\ + \sum_{j \in \mathsf{З}} \sigma_{ij} (s_j M_j - c_j dM_j) + \sum_{j \in \mathsf{НЗ}} \sigma_{ij} k_j M_j + \sum_{j=1}^{t} \eta_{ij} dM_j + \sum_{j=1}^{q_1} \xi_{ij} x_j P. \end{gathered} \qquad (2)$$

Покажем, что в этом разложении можно переформировать коэффициенты таким образом, чтобы избавиться от сумм вида $\sum\limits_{j \in \mathsf{НЗ}}$.

Прибавим к α_i значение $\sum\limits_{j \in \mathsf{НЗ}} \gamma_{ij} k_j$, известное нарушителю $\mathcal{B}$. Обозначим результирующий коэффициент через α_i^*, избавившись тем самым от суммы $\sum\limits_{j \in \mathsf{НЗ}} \gamma_{ij} k_j P$. Заметим, что коэффициент α_i^* фиксируется в момент подачи запроса оракулу RO_2 нарушителем $\mathcal{A}$, так как 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, $\mathcal{A}$ подаёт разложение только по тем точкам, которые возвращались ему в результате эксперимента, а значит, он уже делал запросы Sign_1 в j-х сеансах.

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

Можно переписать систему уравнений (2) как

$$\begin{gathered} s'_i x_i P - c'_i d x_i P = \alpha'_i P + \beta'_i dP + \sum_{j \in \mathsf{З}} \gamma'_{ij}(s_j P - c_j dP) + \\ + \sum_{j \in \mathsf{З}} \sigma'_{ij}(s_j M_j - c_j dM_j) + \sum_{j=1}^{t} \eta'_{ij} dM_j + \sum_{j=1}^{q_1} \xi'_{ij} x_j P. \end{gathered} \tag{3}$$

Аналогично значения всех коэффициентов $\alpha'_i, \beta'_i, \gamma'_{ij}, \sigma'_{ij}, \eta'_{ij}, \xi'_{ij}$ фиксируются в момент подачи запроса оракулу RO_2 нарушителем $\mathcal{A}$.

Каждая точка M_j также имеет некоторое представление, нарушитель $\mathcal{A}$ подаёт 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 $x_1, \ldots, x_{q_1}$.

Получить такое представление точки M_j можно за полиномиальное время. В ходе эксперимента нарушитель сам подаёт разложения точек M_j, а в результате запроса к оракулу Sign_1 через это разложение однозначно определяется разложение точек B_j, Z_j за счёт 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, прологарифмировав по основанию P, можно записать систему уравнений (3) следующим образом:

$$\begin{gathered} s'_i x_i - c'_i d x_i = \alpha'_i + \beta'_i d + \sum_{j \in \mathsf{З}} \gamma'_{ij}(s_j - c_j d) + \\ + \sum_{j \in \mathsf{З}} \sigma'_{ij}(s_j - c_j d) \sum_{t=0}^{j} \tilde{l}_{j,t} d^t + \sum_{j=1}^{t} \eta'_{ij} d \sum_{t=0}^{j} \tilde{l}_{j,t} d^t + \sum_{j=1}^{q_1} \xi'_{ij} x_j. \end{gathered} \tag{4}$$

Для нарушителя $\mathcal{B}$ единственным неизвестным в этой системе является значение d. Каждое уравнение представляет собой полином от d степени не больше $\ell+1$. При этом в силу того, что подписи корректные и нарушитель предоставляет корректное разложение точек, корень d обязательно существует. Если хотя бы одно из уравнений системы существенно зависит от d, то нарушитель $\mathcal{B}$ может найти значение d с помощью вероятностного алгоритма факторизации полиномов, описанного в [28, алгоритм 4]. Положим количество итераций данного алгоритма равным $2(\log \ell+1)$. Тогда трудоёмкость поиска всех корней полинома составляет $O(\ell)$ операций. Событие, что алгоритм успешно решает задачу, обозначим через factor, вероятность успешного запуска алгоритма составляет $\Pr[\mathsf{factor}] \geqslant 1/2$. Нарушитель $\mathcal{B}$, найдя все корни системы (4), может найти правильное значение ключа d перебором по всем корням d_i, $1 \leqslant i \leqslant \ell$, и сравнением d_iP с открытым ключом Q, трудоёмкость этого шага составляет $O(\ell)$ операций. Трудоёмкость поиска ключа d, таким образом, не превосходит $T_{\mathcal{A}}$. Если нарушитель $\mathcal{B}$ успешно восстанавливает значение d, то он успешно решает задачу SOMDL, так как 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 (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:

$$s'_i x_i = \alpha'_i + \sum_{j\in\mathfrak{Z}} \gamma'_{ij} s_j + \sum_{j\in\mathfrak{Z}} \sigma'_{ij} s_j \tilde{l}_{j,0} + \sum_{j=1}^{q_1} \xi'_{ij} x_j, \quad i = 1, \ldots, \ell+1. \tag{5}$$

Обозначим через event событие, когда не выполнено условие (5). Тогда

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

Далее покажем, что если событие event не произошло, т. е. условие (5) выполнено, то нарушитель $\mathcal{C}$ с большой вероятностью успешно решает задачу OMDL.

Построение нарушителя $\mathcal{C}$. Сначала построим нарушителя $\mathcal{C}$, решающего задачу REPR. Пусть нарушитель $\mathcal{C}$ получает на вход точки $x_1P, \ldots, x_{q_1}P$, 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 точно 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 противник $\mathcal{C}$ симулирует, отдавая на каждый новый запрос m очередную точку x_iP, полученную нарушителем $\mathcal{C}$ на вход от своего собственного экспериментатора.

Нарушитель $\mathcal{C}$ так же, как и нарушитель $\mathcal{B}$, может составить систему уравнений (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 завершённых сеансов через d и подставив известные ему k_j для незавершённых сеансов.

Покажем, что если выполнено условие (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 задаёт 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 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 (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 $\mathcal{H}(m_i) = x_i P$. Если переменная x_i не соответствует ни одной подделке, то её индекс может быть произвольным. Очевидно, что подобное изменение также не влияет на решение системы и может быть сделано за полиномиальное время.

В результате этих преобразований получаем систему уравнений (4), уравнения в которой упорядочены по порядку запросов к RO_2, а переменные x_i — по вхождению в набор подделок. Напомним условие (5), при котором нарушитель $\mathcal{B}$ не может успешно решить свою задачу:

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

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:

$$\ell + 1 \left\{ \overbrace{\begin{pmatrix} s'_1 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & s'_2 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 0 & s'_3 & \ldots & 0 & 0 & \ldots & 0 \\ & & & \ddots & & & & \\ 0 & 0 & 0 & \ldots & s'_{\ell+1} & 0 & \ldots & 0 \end{pmatrix}}^{q_1} \right. = \tag{6}$$

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

Заметим, что коэффициенты ξ'_{ij}, составляющие матрицу справа, фиксируются при подаче запросов к случайному оракулу RO_2. Они определяются коэффициентами раз-

ложения 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 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 фиксировано. В результате подачи запроса выбирается некоторое случайное c'. Поскольку подпись (s', c', Z) является корректной, должно быть верно 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 (6), выбираются случайно равновероятно после фиксации определённым образом матрицы справа, состоящей из значений ξ'_{ij}.

Перепишем матричное уравнение (6) следующим образом:

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

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

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 $(q-1)$ (так как s' выбираются случайно). Подматрица, согласно рассуждениям выше, формируется следующим образом: фиксируется определённым произвольным образом первая строка, кроме элемента, стоящего на главной диагонали, после чего случайно выбирается этот элемент. Далее фиксируется вторая строка и случайно выбирается элемент, стоящий на главной диагонали, и так далее.

Ранг квадратной матрицы размера $m\times m$ отличен от m тогда и только тогда, когда её определитель равен нулю. Будем обозначать квадратную подматрицу размера t, стоящую в левом верхнем углу, через A_t. Искомую вероятность можно оценить как

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

Матрица A_1 состоит только из элемента $s'_1 - \xi'_{11}$, который выбирается случайно. Определитель будет равен нулю, если этот элемент равен нулю, вероятность такого события равна $(q-1)^{-1}$, т. е. $\Pr[\det(A_1) = 0] = (q-1)^{-1}$. Далее — индукция 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 ещё одного столбца и ещё 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 $\det(A_{i+1})$ равен сумме $(s'_{i+1,i+1} - \xi'_{i+1,i+1}) \det(A_i)$ и некоторых фиксированных значений. Таким образом, $\det(A_{i+1}) = 0$ только в том случае, когда $s'_{i+1,i+1}$ принимает фиксированное значение, т. е. с вероятностью $(q-1)^{-1}$. Получаем, что

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

откуда следует

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

Итак, в (6) слева с вероятностью больше либо равной $\big(1 - (\ell+1)/(q-1)\big)$ стоит матрица ранга $(\ell+1)$, справа — матрица ранга не больше ℓ. Это означает, что условие (6) с большой вероятностью не выполнено, а значит, среди уравнений системы (5) есть нетривиальная линейная комбинация переменных x_i.

Замечание 2. Если нарушитель $\mathcal{A}$ в составе подделок выдаёт сообщения, для которых он не делал запросы к оракулу RO_1, то в системе (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, так как 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 $\det(A_{\ell+1}) = 0$. Если условие (5) выполнено (т. е. произошло событие $\overline{\mathsf{event}}$) и событие eqrank не произошло, то нарушитель $\mathcal{C}$ успешно решает свою задачу. Тогда

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

Итоговая оценка

Таким образом, мы построили 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. В итоге получаем

$$\Pr[\mathbf{Exp}'(\mathcal{A}) \to 1] = \Pr[\mathbf{Exp}'(\mathcal{A}) \to 1 \wedge \mathsf{event}] + \Pr[\mathbf{Exp}'(\mathcal{A}) \to 1 \wedge \overline{\mathsf{event}}] \leqslant$$
$$\leqslant 2\,\mathsf{Adv}^{\mathrm{SOMDL}}_{\mathbb{G},t,2t}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{REPR}}_{\mathbb{G},q_1+\ell+1}(\mathcal{C}) + \frac{\ell+1}{q-1},$$
$$\mathsf{Adv}^{\mathrm{wUF}}_{\mathsf{CP\text{-}BS}}(\mathcal{A}) \leqslant 2\,\mathsf{Adv}^{\mathrm{SOMDL}}_{\mathbb{G},t,2t}(\mathcal{B}) + \mathsf{Adv}^{\mathrm{REPR}}_{\mathbb{G},q_1+\ell+1}(\mathcal{C}) + \frac{2(\ell+1)+q_2}{q-1}.$$