УДК 519.719.2 DOI 10.17223/20710410/59/3

ТРОИЧНАЯ ЛЕММА О РАЗВЛЕТВЛЕНИИ И ЕЁ ПРИЛОЖЕНИЕ К АНАЛИЗУ СТОЙКОСТИ ОДНОЙ КОДОВОЙ СХЕМЫ ПОДПИСИ

К. Д. Царегородцев

АО «НПК "Криптонит"», г. Москва, Россия

E-mail: k.tsaregorodtsev@kryptonite.ru

Работа посвящена обобщению леммы о развлетвлении на случай, когда хеш-функция возвращает набор тритов, и приложению леммы к альтернативному доказательству стойкости в модели SUF-CMA одной кодовой схемы подписи, основанной на протоколе идентификации Штерна.

Ключевые слова: *лемма о развлетвлении, электронная подпись, доказуемая стойкость.*

TERNARY FORKING LEMMA AND ITS APPLICATION TO THE ANALYSIS OF ONE CODE-BASED SIGNATURE

K. D. Tsaregorodtsev

JSC "NPK "Kryptonite", Moscow, Russia

The paper is devoted to the generalization of the so-called "forking lemma" to the case when the hash function returns a tuple of trits (trit is a variable that can take one out of the three values 0, 1, 2). It can be stated as follows. Let $\mathcal{A}(par, b)$ be an algorithm that, when run on randomly chosen $par \leftarrow^{\mathrm{R}}$ Params and $b \leftarrow^{\mathrm{R}} \{0,1,2\}^{\delta}$, successfully stops and returns the correct answer x with probability ϵ. Then there exists an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine and returns a triple (x_1, x_2, x_3), where $x_i \leftarrow \mathcal{A}(par, b^i)$, $i = 1, 2, 3$, with the additional condition that there exists a position j such that $\{b_j^1, b_j^2, b_j^3\} = \{0, 1, 2\}$ (i.e., all trits in this position are different); the success probability of $\mathcal{B}$ can be bounded from above as follows: $p_{\mathcal{B}} \geqslant \varepsilon^3 - \varepsilon\,(17/27)^{\delta/2}$, and the running time of $\mathcal{B}$ does not exceed $4\,t_{\mathcal{A}}$, where $t_{\mathcal{A}}$ is the time complexity of $\mathcal{A}$. The lemma is then applied to the analysis of code-based signature based on Stern identification protocol in the (standard) SUF-CMA model (with the outer hash function modelled as a programmable random oracle). First, we show that the SUF-CMA model can be reduced to the NMA model (in which the adversary makes no sign queries, only random oracle queries), thanks to the zero-knowledge property of the original Stern identification scheme and the programmability of an oracle. We then show that in the NMA model we can restrict attention only to the case, where the adversary makes a single random oracle query. The b value from the forking lemma acts as the random oracle's answer to the adversarial query. Using the lemma, we are able to fork the process of forging a signature. Having three valid signatures for the same message, we can to extract a secret key (or to find a collision of an inner hash function). Hence, we can bound from above the probability of successful forgery in terms of the probability of successful execution of collision finding and syndrome decoding algorithms.

Keywords: *forking lemma, digital signature, provable security.*

Введение

Лемма о развлетвлении (forking lemma) [1] часто используется для доказательства стойкости схем подписей, полученных из схем идентификации (сигма-протоколов) [2] на основе преобразования Фиата — Шамира [3]. Основой для её применения является следующее соображение. Базовая схема идентификации, как правило, обладает свойством корректности (soundness) [4]. Неформально, данное свойство гласит, что если противник может с большой вероятностью корректно пройти аутентификацию на особым образом связанных запросах, то фактически он должен знать секретное значение (более формально: можно запустить противника дважды, получить два ответа и извлечь из них секретное значение). При этом в стандартной версии леммы [1, 5] противнику, по сути, нужно успешно пройти аутентификацию на любых двух неравных запросах. Этого оказывается достаточно для протоколов, в которых вероятность пройти раунд протокола без знания секрета составляет $1/2$ (см. также свойство специальной корректности (special soundness), например, в [2]).

Для рассматриваемой в работе [6] кодовой схемы подписи основой является протокол идентификации Штерна [7], вероятность пройти раунд протокола без знания секрета в котором равна $2/3$. В этих условиях для извлечения секрета необходимо, чтобы противник трижды успешно прошел аутентификацию, причем требование попарного неравенства запросов необходимо усилить следующим образом: должно найтись такое число j, что все триты в j-й координате запросов попарно различны (трит — величина, которая может принимать одно из трёх значений 0, 1, 2). В описанных условиях стандартная лемма о развлетвлении неприменима. В настоящей работе рассматривается обобщение леммы о развлетвлении на троичный случай.

В п. 1 приводятся краткое описание рассматриваемой схемы подписи, модели SUF-CMA и NMA для изучения стойкости схем подписи, а также используемые обозначения. Пункт 2 посвящён формулировке и доказательству троичной леммы о развлетвлении. В п. 3 доказанная лемма применяется к схеме подписи для доказательства стойкости в модели SUF-CMA; показано, что оценка из этой работы асимптотически (при стремлении числа раундов $\delta \to \infty$) превосходит оценку, полученную в работе [6], но для конкретных значимых на практике параметров оказывается более слабой.

1. Определения и используемые обозначения

1.1. Используемые обозначения

Определение 1. Схемой подписи будем называть тройку вероятностных алгоритмов [8, разд. 13]:

1) Алгоритм генерации пары ключей $(\mathrm{pk}, \mathrm{sk}) \leftarrow^{\mathrm{R}} \mathrm{KGen}()$.
2) Алгоритм генерации подписи для сообщения m:

$$\tau \leftarrow^{\mathrm{R}} \mathrm{Sign}_{\mathrm{sk}}(m).$$

3) Алгоритм проверки подписи τ для сообщения m:

$$res \leftarrow \mathrm{Verify}_{\mathrm{pk}}(m, \tau) \in \{0, 1\}.$$

К схеме подписи предъявляется (синтаксическое) требование корректности: для любого сообщения m и любой пары ключей $(\mathrm{pk}, \mathrm{sk}) \leftarrow^{\mathrm{R}} \mathrm{KGen}()$ выполняется равенство

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

Определение 2. Код (линейный двоичный код) — линейное k-мерное подпространство $\mathcal{C}$ векторного пространства $\mathbb{F}_2^n$.

Определение 3. Проверочная матрица H кода $\mathcal{C}$ — матрица полного ранга размера $(n-k)\times n$, такая, что $Hx=0$ для любого $x\in\mathcal{C}$.

Определение 4. Задача синдромного декодирования [9] формулируется следующим образом. Противнику даются параметры (H, y, ω):

— $H\in \mathrm{Mat}_{n-k,n}(\mathbb{F}_2)$: проверочная матрица двоичного кода;
— $y\in\{0,1\}^{n-k}$: ненулевой синдром;
— $\omega>0$: число, вес вектора ошибок.

Его задачей является нахождение вектора $s\in\{0,1\}^n$, такого, что $\mathrm{wt}(s)=\omega$ и $Hs=y$.

Введём величину $\mathrm{InSec}^{\mathrm{SD}}(t)$ — максимальную вероятность успешного решения противником задачи SD (при заданных n, k и ω), где максимум берется по всем противникам, временные ресурсы которых ограничены t тактами вычислений.

Определение 5. Задача поиска коллизии формулируется следующим образом [8, разд. 6.1]. В начале эксперимента выбирается случайная функция $h\leftarrow^{\mathrm{R}}\mathrm{Hash}$ из некоторого семейства хеш-функций $\mathrm{Hash}=\{h\}$, $h\colon\{0,1\}^*\to\{0,1\}^\ell$, её описание предоставляется противнику (можно предполагать, что семейство хеш-функций параметризовано некоторым ключом-строкой s и противнику предоставляется конкретный индекс функции). Задачей противника является нахождение векторов $x', x''\in\{0,1\}^*$, $x'\neq x''$, со свойством $h(x')=h(x'')$.

Введём величину $\mathrm{InSec}^{\mathrm{Coll}}(t)$ — максимальную вероятность успешного решения противником задачи Coll (для заданного семейства Hash), где максимум берется по всем противникам, временные ресурсы которых ограничены t тактами вычислений. Во временные ресурсы также часто включают размер программы противника.

Будем использовать следующие обозначения:

δ — число раундов в схеме подписи;
H — проверочная матрица кода, используемого в схеме подписи;
h — «внутренняя» хеш-функция в схеме подписи;
f — «внешняя» хеш-функция в схеме подписи;
$\mathbb{B}$ — множество $\{0,1,2\}^\delta$; хеш-функция f принимает значения из $\mathbb{B}$;
n — длина используемого в схеме подписи кода;
S_n — группа подстановок на множестве $\{1,\ldots,n\}$;
$\sigma(x)$ — вектор $(x_{\sigma(1)},\ldots,x_{\sigma(n)})$, $\sigma\in S_n$;
$\mathbb{I}[A]$ — индикатор события A;
$\langle U\mid V\rangle$ — скалярное произведение векторов U и V над полем вещественных чисел $\mathbb{R}$;
$x\leftarrow^{\mathrm{R}}X$ — выбор случайного равновероятного элемента x из конечного множества X; если X — вероятностный алгоритм, то присвоить переменной x случайный выход алгоритма X;
$x\leftarrow y$ — присвоить переменной x значение переменной y;
Λ — неинициализированное значение массива (пустая строка).

1.2. Схема подписи на основе схемы идентификации Штерна

В работе [6] описана схемы подписи на основе схемы идентификации Штерна. Приведём её краткое описание и некоторые свойства.

Алгоритм генерации ключей. Дана проверочная матрица H некоторого кода. Алгоритм KGen() задаётся следующим образом: необходимо выбрать случайный вектор $s\in\{0,1\}^n$, такой, что $\mathrm{wt}(s)=\omega$, затем вычислить вектор $y\in\{0,1\}^{n-k}$ как $y=Hs$.

Вектор y является открытым ключом проверки подписи, вектор s — секретным ключом подписи.

Алгоритм подписи. Даны: ключ подписи $\mathrm{sk} = s$, подписываемое сообщение m. Для вычисления подписи необходимо сделать следующие шаги:

1) Для всех раундов $i = 1, \ldots, \delta$:
   - выбрать $u_i \leftarrow^{\mathrm{R}} \{0,1\}^n$, $\sigma_i \leftarrow^{\mathrm{R}} S_n$;
   - вычислить $c_{i0} \leftarrow h(\sigma_i \| Hu_i)$, $c_{i1} \leftarrow h(\sigma_i(u_i))$, $c_{i2} \leftarrow h(\sigma_i(u_i \oplus s))$, $C_i = (c_{i0} \| c_{i1} \| c_{i2})$.
2) Вычислить *challenge*: $b = f(m \| C_1 \| \ldots \| C_\delta) \in \mathbb{B}$.
3) Вычислить *response*: для всех раундов $i = 1, \ldots, \delta$:
   - если $b_i = 0$, положить $R_i \leftarrow \sigma_i \| u_i$;
   - если $b_i = 1$, положить $R_i \leftarrow \sigma_i \| (u_i \oplus s)$;
   - если $b_i = 2$, положить $R_i \leftarrow \sigma_i(u_i) \| \sigma_i(s)$.
4) Сформировать результат-подпись:

$$\zeta = (C, R) = (C_1 \| \ldots \| C_\delta \| R_1 \| \ldots \| R_\delta).$$

Алгоритм проверки подписи. Даны: открытый ключ y, сообщение m, подпись $\zeta = (C, R) = (C_1 \| \ldots \| C_\delta \| R_1 \| \ldots \| R_\delta)$. Для проверки подписи необходимо сделать следующие шаги:

1) Вычислить $b = f(m \| C_1 \| \ldots \| C_\delta)$.
2) Для всех раундов $i = 1, \ldots, \delta$ проверить:
   - если $b_i = 0$, то $c_{i0} = h\left(R_{i0} \| HR_{i1}\right)$, $c_{i1} = h(R_{i0}(R_{i1}))$;
   - если $b_i = 1$, то $c_{i0} = h\left(R_{i0} \| HR_{i1} \oplus y\right)$, $c_{i2} = h(R_{i0}(R_{i1}))$;
   - если $b_i = 2$, то $c_{i1} = h(R_{i0})$, $c_{i2} = h\left(R_{i0} \oplus R_{i1}\right)$, $\mathrm{wt}(R_{i1}) = \omega$.
3) Если все проверки успешны, то вернуть 1 (иначе 0).

Свойства схемы подписи. Обозначим некоторые свойства представленной схемы подписи:

— Симуляция раунда подписи без знания секрета [6, 7]: существует алгоритм Sim, который по заданному $b \in \{0,1,2\}$ моделирует такие случайные величины (C'_i, R'_i), что их распределение совпадает с величинами (C_i, R_i), построенными в соответствии с «честным» алгоритмом подписи. Это свойство следует из свойства нулевого разглашения протокола идентификации Штерна.

— Подделка подписи без знания секрета: противник может подделать подпись с вероятностью не менее чем $(2/3)^\delta$.

— Извлечение секрета (см. работу [6], а также доказательство теоремы 3): если противник может построить три подделки подписи (C^i, R^i), $i = 1, 2, 3$, такие, что соответствующие им векторы тритов b^i имеют позицию j, в которой все три вектора попарно различны (т. е. $\{b^1_j, b^2_j, b^3_j\} = \{0,1,2\}$), то существует алгоритм Extract, который с вероятностью, стремящейся к 1 при стремлении числа раундов δ к бесконечности, извлекает секретный ключ s схемы подписи либо находит коллизию внутренней хеш-функции h.

Замечание 1. Опишем подробнее свойство симуляции. Для симуляции подписи генерируется общий для всех шагов i, $1 \leqslant i \leqslant \delta$, случайный равновероятный вектор s' из множества $\mathbb{F}_2^n$ с дополнительным ограничением на вес. На каждом из шагов $i = 1, \ldots, \delta$ необходимо:

1) сгенерировать $u_i \leftarrow^{\mathrm{R}} \mathbb{F}_2^n$, $\sigma_i \leftarrow^{\mathrm{R}} S_n$;

2) в зависимости от b_i сгенерировать следующие величины:
   — если $b_i = 0$, положить

$$c_{i0} \leftarrow h(\sigma_i \| Hu_i), \quad c_{i1} \leftarrow h(\sigma_i(u_i)), \quad c_{i2} \leftarrow h(\sigma_i(u_i \oplus s')), \quad R_i \leftarrow \sigma_i \| u_i;$$

   — если $b_i = 1$, положить

$$c_{i0} = h(\sigma_i \| Hu_i \oplus y), c_{i1} = h(\sigma_i(u_i) \oplus s'), c_{i2} = h(\sigma_i(u_i)), \quad R_i \leftarrow \sigma_i \| u_i;$$

   — если $b_i = 2$, положить

$$c_{i0} = h(\sigma_i \| H(u_i \oplus s')), c_{i1} = h(\sigma_i(u_i)), c_{i2} = h(\sigma_i(u_i \oplus s')), \quad R_i \leftarrow \sigma_i(u_i) \| \sigma_i(s').$$

Заметим также, что при генерации c_{ij} в каждом из C_i есть подстрока вида $h(X)$, где X выбирается случайно равновероятно из множества $\{0,1\}^n$.

1.3. Модели SUF-CMA и NMA

Рассмотрим следующие две стандартные модели, используемые для изучения стойкости схем подписи. Противнику предоставляется доступ к оракулу подписи Sign, его задачей является подделка подписи для нового (ранее не запрашиваемого) сообщения. В случае модели со случайным оракулом (ROM, Random Oracle Model, см. [10, 11] и [8, разд. 6.5]) противнику также даётся доступ к случайному оракулу f (случайной функции, моделирующей поведение реальной хеш-функции). Приведём псевдокод эксперимента, используемого в моделях SUF-CMA и NMA:

$\mathrm{Exp}^{\text{SUF-CMA}}(\mathcal{A})$	$\mathrm{Sign}(m)$
$(\mathrm{sk}, \mathrm{pk}) \leftarrow^{\mathrm{R}} \mathrm{KGen}()$	$\tau \leftarrow^{\mathrm{R}} \mathrm{Sign}_{\mathrm{sk}}(m)$
$sent = \varnothing$	$sent \leftarrow sent \cup \{(m, \tau)\}$
$(m, \tau) \leftarrow^{\mathrm{R}} \mathcal{A}^{\mathrm{Sign},f}(\mathrm{pk})$	**return** τ
if $((m, \tau) \notin sent)$	
return $\mathrm{Verify}_{\mathrm{pk}}(m, \tau)$	
else	
return 0	
fi	

Определение 6. Уровнем нестойкости схемы подписи в модели SUF-CMA (Strong UnForgeability under Chosen Message Attack) со случайным оракулом будем называть число

$$\mathrm{InSec}^{\text{SUF-CMA}}(t, q_f, q_s) = \max_{\mathcal{A} \in A(t, q_f, q_s)} \mathsf{P}[\mathrm{Exp}^{\text{SUF-CMA}}(\mathcal{A}) \to 1],$$

где за $A(t, q_f, q_s)$ обозначено множество алгоритмов, работающих не более t тактов и делающих не более q_f запросов к случайному оракулу и q_s запросов к оракулу подписи.

Определение 7. Нестойкостью схемы подписи в модели NMA (No Message Attack) со случайным оракулом будем называть следующее число:

$$\mathrm{InSec}^{\mathrm{NMA}}(t, q_f) = \mathrm{InSec}^{\text{SUF-CMA}}(t, q_f, 0).$$

Другими словами, противник в модели NMA не делает ни одного запроса к оракулу подписи, и его задачей является подделка подписи для любого сообщения.

2. Троичная лемма о развлетвлении

Сформулируем и докажем основной результат работы.

Лемма 1. Пусть $\mathcal{A}(par, b)$ — алгоритм, который на случайно выбранных входах $par \leftarrow^{\mathrm{R}} \mathrm{Params}$ и $b \leftarrow^{\mathrm{R}} \mathbb{B}$ успешно завершает работу с некоторым выходом x с вероятностью ε. Тогда мы можем построить алгоритм $\mathcal{B}$, использующий алгоритм $\mathcal{A}$ как подпроцедуру, который возвращает тройку (x_1, x_2, x_3), где x_i есть результат работы $\mathcal{A}(par, b^i)$, с дополнительным условием, что в (b^1, b^2, b^3) найдётся такой номер i, для которого все триты в этой позиции различны ($\{b_i^1, b_i^2, b_i^3\} = \{0, 1, 2\}$), причём вероятность успешной работы алгоритма $\mathcal{B}$ оценивается снизу как

$$p_{\mathcal{B}} \geqslant \varepsilon^3 - \varepsilon\,(17/27)^{\delta/2}\,,$$

время работы алгоритма $\mathcal{B}$ оценивается сверху как $4\,t_{\mathcal{A}}$, где $t_{\mathcal{A}}$ — время работы алгоритма $\mathcal{A}$.

Доказательство. Построим алгоритм $\mathcal{B}$ следующим образом:

— В начале эксперимента алгоритм $\mathcal{B}$ генерирует параметры $par \leftarrow^{\mathrm{R}} \mathrm{Params}$ и три случайных независимых равновероятных элемента $b^i \leftarrow^{\mathrm{R}} \mathbb{B}$, $i = 1, 2, 3$.

— Алгоритм $\mathcal{B}$ трижды запускает алгоритм $\mathcal{A}$ на входах (par, b^i) и получает результаты работы x_i, $i = 1, 2, 3$.

— Если все три запуска завершились успешно, а также найдётся такой номер i, для которого все триты b_i^1, b_i^2, b_i^3 различны, то алгоритм $\mathcal{B}$ возвращает тройку (x_1, x_2, x_3) в качестве ответа.

Алгоритм $\mathcal{B}$ трижды запускает алгоритм $\mathcal{A}$ и генерирует параметры par, длина которых не может превышать $t_{\mathcal{A}}$, что даёт оценку сверху на время работы $t_{\mathcal{B}} \leqslant 4\,t_{\mathcal{A}}$.

Найдём вероятность успешного завершения работы алгоритма $\mathcal{B}$. Введём события A — все три запуска алгоритма $\mathcal{A}$ успешны; B — в векторах тритов (b^1, b^2, b^3) найдётся позиция i, в которой все триты различны (в таком случае событие $\overline{B}$ заключается в том, что для каждой позиции i хотя бы два из трёх тритов совпадают). В таком случае вероятность $p_{\mathcal{B}}$ можно оценить следующим образом:

$$p_{\mathcal{B}} = \mathsf{P}[A \cap B] = \mathsf{P}[A] - \mathsf{P}[A \cap \overline{B}].$$

Далее мы введём случайную величину X, выразим вероятность каждого из событий A и $A \cap \overline{B}$ через математические ожидания функций от X и применим к полученной оценке неравенство Йенсена. Определим индикатор $\mathbb{I}(par, b)$ и случайную величину X на пространстве Params следующим образом:

$$\mathbb{I}(par, b) = \mathbb{I}[\mathcal{A}(par, b) \text{ завершается успешно}],$$
$$X(par) = \mathsf{P}[b \in \mathbb{B}\colon \mathcal{A}(par, b) \text{ завершается успешно}].$$

В таком случае имеем (по определению математического ожидания и величины ε как вероятности успешного завершения алгоритма $\mathcal{A}$):

$$\mathbb{E}[X] = \sum_{par} \mathsf{P}[par] \cdot X(par) = \sum_{par,b} \mathsf{P}[par] \cdot \mathsf{P}[b] \cdot \mathbb{I}(par, b) = \varepsilon.$$

Заметим, что вероятности событий A и $A \cap \overline{B}$ выражаются через введённые величины следующим образом:

$$\mathsf{P}[A] = \sum_{par} \sum_{(b^1,b^2,b^3)\in\mathbb{B}^3} \mathsf{P}[par] \cdot \mathsf{P}[b^1,b^2,b^3] \cdot \mathbb{I}(par,b^1)\,\mathbb{I}(par,b^2)\,\mathbb{I}(par,b^3) =$$

$$= \sum_{par} \mathsf{P}[par] \left(\sum_{b\in\mathbb{B}} \mathsf{P}[b] \cdot \mathbb{I}(par,b) \right)^3 = \mathbb{E}[X^3];$$

$$\mathsf{P}[A \cap \overline{B}] = \sum_{par} \sum_{(b^1,b^2,b^3)\in\mathbb{B}'} \mathsf{P}[par] \cdot \mathsf{P}[b^1,b^2,b^3] \cdot \mathbb{I}(par,b^1)\,\mathbb{I}(par,b^2)\,\mathbb{I}(par,b^3) =$$

$$= \sum_{par} \mathsf{P}[par] \frac{1}{|\mathbb{B}|^3} \sum_{b^1,b^2\in\mathbb{B}^2} \mathbb{I}(par,b^1)\,\mathbb{I}(par,b^2)\,S(b^1,b^2).$$

Здесь $\mathbb{B}'$ — множество таких троек $(b^1,b^2,b^3) \in \mathbb{B}^3$, что для каждой позиции принимается не более двух значений трита из трёх возможных (в частности, $|\mathbb{B}'| = 21^\delta$), а $S(b^1,b^2)$ — величина, равная количеству таких векторов b^3, что для тройки (b^1,b^2,b^3) в каждой позиции $i = 1,\ldots,\delta$ встречаются не более двух тритов из трёх возможных (группировка суммы по (b^1,b^2)).

Оценим сверху вероятность $\mathsf{P}[A\cap\overline{B}]$ с помощью неравенства Коши — Буняковского. Для этого введём векторы $U(b^1,b^2) = \mathbb{I}(par,b^1)\,\mathbb{I}(par,b^2) \in \{0,1\}$ и $V(b^1,b^2) = S(b^1,b^2)$, а также скалярное произведение векторов $\langle U \mid V\rangle = \sum\limits_{b^1,b^2\in\mathbb{B}^2} \mathbb{I}(par,b^1)\,\mathbb{I}(par,b^2)\,S(b^1,b^2)$. Тогда

$$\sum_{b^1,b^2\in\mathbb{B}^2} \mathbb{I}(par,b^1)\,\mathbb{I}(par,b^2)\,S(b^1,b^2) = \langle U \mid V\rangle \leqslant \sqrt{\langle U \mid U\rangle} \cdot \sqrt{\langle V \mid V\rangle}.$$

Рассмотрим отдельно каждое из скалярных произведений. Для выражения $\langle U \mid U\rangle$ выполнено следующее (используем то, что $U(b^1,b^2) \in \{0,1\}$):

$$\langle U \mid U\rangle = \sum_{b^1,b^2} \left(U(b^1,b^2)\right)^2 = \sum_{b^1,b^2} U(b^1,b^2) = \sum_{b^1,b^2} \mathbb{I}(a,b^1)\,\mathbb{I}(a,b^2) = \left(\sum_{b} \mathbb{I}(a,b) \right)^2.$$

Для величины $\langle V \mid V\rangle$ найдём точное значение:

$$\langle V \mid V\rangle = \sum_{b^1,b^2} S(b^1,b^2)^2 = \sum_{t=0}^{\delta} \binom{\delta}{t} 3^\delta \cdot 2^t (3^{\delta-t}\cdot 2^t)^2.$$

Дадим пояснения:

— если в (b^1,b^2) различны t позиций, то $S(b^1,b^2) = 3^{\delta-t}\cdot 2^t$ (есть два варианта зафиксировать трит в b^3, в котором b^1 и b^2 не совпали, и три варианта зафиксировать трит, в котором они совпали);

— всего имеется $\binom{\delta}{t} 3^\delta \cdot 2^t$ способов выбрать два вектора из тритов так, чтобы у них было n различных позиций:

1) первый вектор выбирается любым из возможных 3^δ способов,
2) второй вектор выбирается отличным от первого в t фиксированных позициях, что даёт сомножитель 2^t.

Упрощая последнюю сумму, получим

$$\langle V \mid V\rangle = 3^{3\delta} \sum_{t=0}^{\delta} \binom{\delta}{t} \left(\frac{2^3}{3^2}\right)^t = 3^{3\delta} \left(1+\frac{8}{9}\right)^\delta = 51^\delta.$$

Таким образом, имеем

$$\mathsf{P}[A \cap \overline{B}] = \sum_{par} \mathsf{P}[par] \frac{1}{|\mathbb{B}|^3} \sum_{b^1, b^2 \in \mathbb{B}^2} \mathbb{I}(par, b^1)\, \mathbb{I}(par, b^2)\, S(b^1, b^2) = \sum_{par} \mathsf{P}[par] \frac{1}{|\mathbb{B}|^3} \langle U \mid V \rangle \leqslant$$

$$\leqslant \sum_{par} \mathsf{P}[par] \frac{1}{|\mathbb{B}|^3} \sqrt{\langle U \mid U \rangle}\, \sqrt{\langle V \mid V \rangle} \leqslant \sum_{par} \mathsf{P}[par] \frac{1}{|\mathbb{B}|^3} \sqrt{\left(\sum_b \mathbb{I}(a, b) \right)^2}\, \sqrt{51^\delta} =$$

$$= \frac{(51)^{\delta/2}}{3^{2\delta}} \sum_{par} \mathsf{P}[par] \frac{1}{|\mathbb{B}|} \sum_b \mathbb{I}(a, b) = \mathbb{E}[X] \left(\frac{17}{27} \right)^{\delta/2}.$$

Следовательно, вероятность успешного завершения алгоритма $\mathcal{B}$ ограничена снизу следующим образом:

$$p_{\mathcal{B}} = \mathsf{P}[A \cap B] = \mathsf{P}[A] - \mathsf{P}[A \cap \overline{B}] \geqslant \mathbb{E}[X^3] - \mathbb{E}[X]\, (17/27)^{\delta/2}.$$

Функция $\phi(x) = x^3 - \alpha x$, $\alpha = (17/27)^{\delta/2}$, является выпуклой при $x \geqslant 0$, и для неё выполняется неравенство Йенсена $\phi(\mathbb{E}[X]) \leqslant \mathbb{E}[\phi(X)]$, а значит,

$$p_{\mathcal{B}} \geqslant (\mathbb{E}[X])^3 - \mathbb{E}[X]\, (17/27)^{\delta/2} = \varepsilon^3 - \varepsilon\, (17/27)^{\delta/2}.$$

Лемма 1 доказана. ■

3. Приложение леммы к анализу схемы подписи

Применим лемму 1 к анализу стойкости схемы подписи на основе схемы идентификации Штерна. Доказательство состоит из нескольких шагов:

1) В модели программируемого случайного оракула (подробнее см. [11]) запросы противника на получение подписи сообщения можно моделировать (это свойство по сути следует из свойства нулевого разглашения протокола идентификации и возможности задать ответы случайного оракула на выбранных входах). Таким образом, возможно получить сведение модели SUF-CMA к модели NMA.
2) Модель NMA с q_f запросами к случайному оракулу может быть сведена к случаю одного запроса. Такая модель проще поддаётся анализу.
3) Если противник в случае одного запроса к случайному оракулу с высокой вероятностью успешно завершает атаку, то его можно перезапустить несколько раз (см. лемму 1) и с высокой вероятностью получить несколько «существенно различных» подделок подписи. В нашем случае это означает, что, имея три различные подделки подписи, можно либо найти коллизию внутренней хеш-функции h, либо решить задачу синдромного декодирования. Поскольку обе задачи на данный момент считаются сложными, отсюда можно заключить, что схема подписи является стойкой в исходной модели SUF-CMA.

Первые два этапа уже рассматривались ранее в [6], здесь мы приведём альтернативное доказательство этих фактов.

3.1. Сведение модели SUF-CMA к модели NMA

Основная идея доказательства заключается в следующем: если бы противник знал заранее, каким будет бит b_i для каждого раунда схемы подписи (т. е. знал бы выход внешней хеш-функции $f(\cdot)$), то он мог бы самостоятельно смоделировать подписание сообщения, не зная секретного ключа (свойство нулевого разглашения схемы идентификации Штерна). В модели случайного оракула мы делаем именно это: сначала

заранее выбираем значение выхода b, генерируем корректную подпись (C, R) для выбранного b, а затем перепрограммируем случайный оракул так, чтобы на сформированном входе $(m\|C)$ оракул выдавал бы значение b: $f(m\|C) \leftarrow b$.

Теорема 1. Выполняется неравенство

$$\mathrm{InSec}^{\text{SUF-CMA}}(t, q_f, q_s) \leqslant \mathrm{InSec}^{\text{SUF-CMA}}(t + T_{\text{SIG}} \cdot q_s, q_f) + \frac{(2q_f + q_s)q_s}{2^{\delta\lambda+1}},$$

где используются следующие обозначения:

— q_f: число запросов к случайному оракулу F;
— q_s: число запросов к Sign;
— δ: число раундов в схеме подписи;
— λ: min-энтропия величины $h(X)$, X выбирается случайно равновероятно из множества $\{0,1\}^n$;
— T_{SIG}: время вычисления подписи (количество элементарных операций, необходимых для вычисления).

Доказательство. Пусть $\mathcal{A}$ — противник для схемы подписи в модели SUF-CMA. По нему мы построим противника $\mathcal{B}$ в модели NMA для схемы подписи. Поскольку в модели NMA отсутствует оракул подписи Sign, противник $\mathcal{B}$ должен моделировать оракул Sign самостоятельно. Для этого противник $\mathcal{B}$ будет вести таблицу запросов к случайному оракулу $F[\cdot]$.

При запросе противника $\mathcal{A}$ к оракулу подписи на входе m:

1) $\mathcal{B}$ генерирует элемент $b \leftarrow^{\text{R}} \mathbb{B}$;
2) $\mathcal{B}$ генерирует векторы u_i и перестановки σ_i в соответствии со значением b_i и вычисляет C_i и R_i. Это можно сделать в силу свойства нулевого разглашения схемы идентификации Штерна (см. [7] и разд. 1.2), причём распределение сгенерированных (без знания секрета) элементов (C_i, R_i) статистически неотличимо от распределения элементов (C_i, R_i), сгенерированных по секретному значению;
3) если значение $m\|C$ уже запрашивалось ранее (т. е. $F[m\|C] \neq \Lambda$), то $\mathcal{B}$ прерывает эксперимент (выдаёт ошибку). Если $F[m\|C] = \Lambda$, то задать значение $F[m\|C] \leftarrow b$ (программирование случайного оракула f).

При запросе x противника $\mathcal{A}$ к случайному оракулу F:

— если $F[x] \neq \Lambda$, то возвращается $F[x]$;
— в противном случае $\mathcal{B}$ запрашивает значение на входе x у своего случайного оракула, его ответ b записывает в таблицу $F[x] \leftarrow b$ и возвращает вектор b противнику $\mathcal{A}$.

Выпишем псевдокод противника $\mathcal{B}$ (генерация (R_i, C_i) описана в замечании 1):

$\mathcal{B}^{\mathcal{F}}(\mathcal{A})$

$F \leftarrow [\,]$
$(C, R) \leftarrow^{\mathrm{R}} \mathcal{A}^{\mathrm{SimSign},F}$
return (C, R)

$F(x)$

if $F[x] = \Lambda$
  $F[x] \leftarrow^{\mathrm{R}} \mathcal{F}(x)$
fi
return $F[x]$

SimSign(m)

$b \leftarrow^{\mathrm{R}} \{0,1,2\}^{\delta}$
for $(0 \leqslant i < \delta)$
  generate R_i, C_i
endfor
$x \leftarrow m \| C_0 \| \ldots \| C_{\delta-1}$
if $(F[x] \neq \Lambda)$
  $flag \leftarrow \texttt{true}$
  return $\perp$
else
  $F[x] \leftarrow b$
  return (C, R)
fi

Как подчёркивалось ранее (см. замечание 1), распределение величин C_i и R_i такое же, как и в случае знания секрета. До тех пор, пока не случилась коллизия (строка $flag \leftarrow \texttt{true}$), распределение ответов в «настоящей» модели SUF-CMA не отличается от распределения ответов в модели NMA, где оракул Sign заменяется на оракул SimSign. При этом для корректной симуляции необходимо затратить порядка $T_{\mathrm{SIG}} \cdot q_s$ элементарных операций (симуляция подписи q_s раз).

Найдём теперь вероятность коллизии. Ключи массива F формируются следующим образом: либо они были запрошены через оракул $F(x)$ у случайного оракула $\mathcal{F}$, либо сформированы при симуляции подписи в SimSign. Ключи при формировании подписи в интерфейсе SimSign имеют вид $m\|C_1\| \ldots \|C_\delta$. В каждый из C_i входит подстрока вида $h(X)$, где X выбирается случайно независимо равновероятно из множества $\{0,1\}^n$ (см. замечание 1). Для того чтобы случилась коллизия, необходимо, чтобы все δ раундовых значений C_i совпали как подстроки с подстроками запрашиваемых ранее у оракула $\mathcal{F}$ запросов.

Обозначим $y \leftarrow C_1\| \ldots \|C_\delta$. Пусть среди ключей массива F на очередном шаге уже содержатся q некоторых значений $x_1, \ldots, x_q$, тогда вероятность коллизии не превышает $q \cdot \mathsf{P}[y = x]$ для некоторого $x \in \{x_1, \ldots, x_q\}$. Эту вероятность (в силу независимости подстрок вида $h(X)$ в строках C_i) можно оценить сверху следующим образом:

$$\mathsf{P}[y = x] \leqslant \prod_{i=1}^{\delta} \left(\mathsf{P}[h(X) = x']\right) = \left(\sum_c \mathsf{P}[h(X) = c]\,\mathsf{P}[x' = c]\right)^{\delta} \leqslant \left(\sum_c 2^{-\lambda}\,\mathsf{P}[x' = c]\right)^{\delta} = 2^{-\delta\lambda},$$

где λ — min-энтропия величины $h(X)$, $X \leftarrow^{\mathrm{R}} \{0,1\}^n$.

В таком случае вероятность коллизии ключей F можно оценить как

$$\mathsf{P}[F\text{-coll}] \leqslant (q_f + (q_f + 1) + \ldots + (q_f + q_s - 1))\, 2^{-\delta\lambda} \leqslant \frac{(2q_f + q_s)q_s}{2^{\delta\lambda+1}},$$

поскольку при первом запросе к оракулу SimSign в массиве $F[\cdot]$ содержится не более q_f ключей, при втором — не более $q_f + 1$ ключей и так далее.

Таким образом, по лемме 1 из работы [12] имеем

$$\mathrm{InSec}^{\text{SUF-CMA}}(t, q_f, q_s) \leqslant \mathrm{InSec}^{\text{SUF-CMA}}(t + T_{\mathrm{SIG}} \cdot q_s, q_f) + \frac{(2q_f + q_s)q_s}{2^{\delta\lambda+1}}.$$

Теорема 1 доказана. ■

Замечание 2. Заметим, что в доказательстве теоремы 1 мы не использовали условия (не)равенства различных значений m. Это соответствует тому факту, что в модели SUF-CMA противнику для успеха необходимо подделать новую (не запрошенную ранее) подпись под некоторым сообщением m (при этом, вообще говоря, не требуется, чтобы сообщение никогда ранее не запрашивалось у оракула подписи).

3.2. Сведение случая q запросов к одному запросу

Теорема 2. Выполняется следующее неравенство:

$$\mathrm{InSec}^{\mathrm{NMA}}(t, q_f) \leqslant q_f \,\mathrm{InSec}^{\mathrm{NMA}}(t + \delta \cdot q_f, 1).$$

Доказательство. Без ограничения общности можем считать, что противник $\mathcal{A}$ не повторяет запросы к оракулу. Перед началом эксперимента противник $\mathcal{B}$ выбирает $l \leftarrow^{\mathrm{R}} \{1, \ldots, q_f\}$ — номер запроса, который он перенаправит «настоящему» случайному оракулу. На i-й запрос m_i к оракулу хеширования от противника $\mathcal{A}$ противник $\mathcal{B}$:

— при $i \neq l$ возвращает значение $b \leftarrow^{\mathrm{R}} \mathbb{B}$;
— при $i = l$ возвращает ответ оракула $\mathcal{F}(m_i)$.

В конце эксперимента $\mathcal{B}$ выдаёт подделку (m, τ), полученную от $\mathcal{A}$. Найдём вероятность успешной подделки подписи противником $\mathcal{B}$. Если $m \in \{m_1, \ldots, m_{q_f}\}$, то $m = m_l$ с вероятностью q_f^{-1}, поскольку распределения всех $\mathcal{F}(m_i)$ одинаковы и индекс l выбран случайно независимо. Если $m \notin \{m_1, \ldots, m_{q_f}\}$, то вероятность успеха противника $\mathcal{A}$ равна вероятности успеха противника $\mathcal{B}$, поскольку для корректности подписи необходимо, чтобы хеш-значение b, задействованное в подписи, совпало с $f(m)$, не запрашиваемым ранее. Отсюда следует утверждение теоремы. ■

3.3. Применение троичной леммы о разветвлении для схемы подписи

Теорема 3. Выполнены следующие неравенства:

$$\mathrm{InSec}^{\mathrm{SUF\text{-}CMA}}(t, q_f, q_s) \leqslant q_f \,\mathrm{InSec}^{\mathrm{NMA}}(t + T_{\mathrm{SIG}} \cdot q_s + \delta \cdot q_f, 1) + \frac{(2q_f + q_s)\, q_s}{2^{\delta\lambda+1}}; \tag{1}$$

$$\varepsilon_t^3 - \varepsilon_t \,(17/27)^{\delta/2} \leqslant \mathrm{InSec}^{\mathrm{SD}}(4t) + \mathrm{InSec}^{\mathrm{Coll}}(4t) + 3^{1-\delta}, \tag{2}$$

где $\varepsilon_t = \mathrm{InSec}^{\mathrm{NMA}}(t, 1)$; остальные обозначения те же, что в теореме 1.

Доказательство. Из теоремы 1 следует

$$\mathrm{InSec}^{\mathrm{SUF\text{-}CMA}}(t, q_f, q_s) \leqslant \mathrm{InSec}^{\mathrm{NMA}}(t + T_{\mathrm{SIG}} \cdot q_s, q_f),$$

из теоремы 2 следует оценка (1).

Найдём ограничение на величину ε_t. Рассмотрим алгоритм $\mathcal{A}(par, b)$, где $b \in \mathbb{B}$ — ответ случайного оракула $\mathcal{F}$ на единственный запрос от $\mathcal{A}$; par — открытый ключ $par = \mathrm{pk} = Hs$ рассматриваемой криптосистемы (зафиксирован в эксперименте). Пусть ρ — случайная лента противника $\mathcal{A}$. Выходом противника x является пара (m, τ) — подделка подписи сообщения. Согласно теореме 1, найдётся противник $\mathcal{B}$, который выдаёт три подделки (m_0, τ_0), (m_1, τ_1), (m_2, τ_2), $\tau_i = (C^i, R^i)$, полученные на некоторых значениях b^0, b^1, b^2, в которых найдётся позиция j, такая, что (без ограничения общности)

$$b_j^0 = 0, \quad b_j^1 = 1, \quad b_j^2 = 2.$$

При этом для сообщений m_i выполняется равенство $\mathcal{F}(m_i\|C^i) = b^i$ (в противном случае подпись τ_i неверна). Следовательно, сообщение $m_i\|C^i$ с большой вероятностью запрашивалось в качестве единственного запроса к оракулу (в противном случае каждая подпись верна с вероятностью не более чем $3^{-\delta}$). Но при этом запрос к оракулу $\mathcal{F}$ (значение $m\|C_1\|\ldots\|C_\delta$) зависит только от ленты противника ρ и открытого ключа pk (т. е. от параметров par), а следовательно, не меняется от запуска к запуску. Таким образом, с вероятностью не менее чем $1 - 3^{1-\delta}$ выполнено равенство $m_0\|C^0 = m_1\|C^1 = m_2\|C^2$. Если равенство не выполнено, то эксперимент прерывается.

Рассмотрим соответствующие позиции j значения подписей (C_j^0, R_j^0), (C_j^1, R_j^1), (C_j^2, R_j^2). По определению схемы подписи имеем

— $R_j^0 = (\sigma\|u)\colon h(\sigma\|Hu) = C_j^0, h(\sigma(u)) = C_j^1$;
— $R_j^1 = (\pi\|v)\colon h(\pi\|Hv \oplus y) = C_j^0, h(\pi(v)) = C_j^2$;
— $R_j^2 = (w_1\|w_2)\colon h(w_1) = C_j^1, h(w_1 \oplus w_2) = C_j^2, \mathrm{wt}(w_2) = \omega$.

Если при этом была построена коллизия для внутренней хеш-функции h, то эксперимент прерывается. Вероятность этого события может быть оценена сверху величиной $\mathrm{InSec}^{\mathrm{Coll}}(t_{\mathcal{B}})$.

В случае отсутствия коллизий $\mathcal{B}$ восстанавливает вектор $s = \pi^{-1}(w_2)$. Из условий отсутствия коллизий получим

$$H\pi^{-1}(w_2) = H(v \oplus \pi^{-1}(w_1)) = H(v \oplus u) = Hv \oplus Hv \oplus y = y,$$

причём $\mathrm{wt}(\pi^{-1}(w_2)) = \mathrm{wt}(w_2) = \omega$.

Вероятность восстановления вектора s (секретного ключа) может быть оценена сверху величиной $\mathrm{InSec}^{\mathrm{SD}}(t_{\mathcal{B}})$. Отсюда следует, что выполнено неравенство

$$\varepsilon_t^3 - \varepsilon_t\,(17/27)^{\delta/2} \leqslant \mathrm{InSec}^{\mathrm{SD}}(4t) + \mathrm{InSec}^{\mathrm{Coll}}(4t) + 3^{1-\delta},$$

что и требовалось доказать. ■

3.4. Практическая значимость оценки

Асимптотическое поведение оценки

При большом числе раундов $\delta \geqslant \Delta$ слагаемое вида $\varepsilon_t\,(17/27)^{\delta/2}$ вносит всё меньший вклад, и в пределе неравенство (2) переходит в асимптотическое неравенство вида

$$\mathrm{InSec}^{\mathrm{NMA}}(t, 1) \leqslant \sqrt[3]{\mathrm{InSec}^{\mathrm{SD}}(4t) + \mathrm{InSec}^{\mathrm{Coll}}(4t) + 3^{1-\delta}}.$$

Совместно с оценкой (1) получаем следующую оценку сверху на величину $\mathrm{InSec}^{\text{SUF-CMA}}$:

$$\mathrm{InSec}^{\text{SUF-CMA}}(t, q_f, q_s) \leqslant q_f \sqrt[3]{\mathrm{InSec}^{\mathrm{SD}}(4T) + \mathrm{InSec}^{\mathrm{Coll}}(4T) + 3^{1-\delta}} + \frac{(2q_f + q_s)\,q_s}{2^{\delta\lambda+1}},$$

где $T = t + T_{\mathrm{SIG}}\cdot q_s + \delta\cdot q_f$, что меньше оценки, полученной в [6], примерно в $14\cdot\sqrt[3]{\delta^2/4}$ раз (время работы противника, рассматриваемого в [6], при $q_f, q_s \ll t$, примерно равно $\delta^2 t$, также в оценке [6] присутствует сомножитель $\sqrt[3]{1920/(1-e^{-1}} \approx 14$).

Сравнение оценок при конкретных значениях параметров

К сожалению, для интересных на практике значений числа раундов δ оценка для $\mathrm{InSec}^{\mathrm{NMA}}(t, 1)$, полученная в настоящей работе, намного хуже полученной в [6].

Примем следующие значения параметров задач SD и Coll:

$$n = 2896,\ k = 1448,\ \omega = 318,\ \ell = 512.$$

Согласно работе [13] и программному коду [14], оптимальный алгоритм для заданных (n, k, ω) решает задачу SD с вероятностью 1 за $T_{\mathrm{SD}} = 2^{323}$ битовых операций.

Пусть

$$\mathrm{InSec}^{\mathrm{SD}}(t) = t/T_{\mathrm{SD}}, \quad \mathrm{InSec}^{\mathrm{Coll}}(t) = t^2/2^{\ell}.$$

Примем (следуя работе [6]) следующие значения параметров: $t = 2^{70}$ элементарных операций, $\ell = 512$. В таблице в первом столбце указано число раундов в схеме подписи δ, во втором и третьем — двоичный логарифм от оценки сверху для величины $\mathrm{InSec}^{\mathrm{NMA}}(t, 1)$ (для второго столбца использована оценка [6, теорема 1], для третьего — оценка (2)). Заметим, что оценка (2) становится лучше при $\delta = 414$ раундах схемы подписи.

δ	Прошлая оценка	Текущая оценка
100	−58,5	−16,69
137	−71,16	−22,86
200	−70,43	−33,37
300	−69,65	−50,06
500	−68,67	−83,21

Заключение

В работе доказана троичная версия леммы о развлетвлении. Полученная лемма применена для анализа стойкости схемы подписи, основанной на схеме идентификации Штерна, в модели SUF-CMA. Направлением дальнейших исследований может быть получение более сильных оценок для практически значимых результатов при малом числе раундов, а также обобщение доказательства на модель квантового доступа к случайному оракулу QROM.

Автор благодарит Александру Бабуеву и Викторию Высоцкую за полезные обсуждения в ходе работы, а также рецензента, замечания которого помогли улучшить статью.

ЛИТЕРАТУРА

1. *Pointcheval D. and Stern J.* Security proofs for signature schemes // EUROCRYPT'96. LNCS. 1996. V. 1070. P. 387–398.
2. *Damgård I.* On Σ-protocols. Lecture Notes. University of Aarhus, Department of Computer Science, 2002.
3. *Fiat A. and Shamir A.* How to prove yourself: Practical solutions to identification and signature problems // LNCS. 1987. V. 263. P. 186–194.
4. *Черемушкин А. В.* Криптографические протоколы: основные свойства и уязвимости // Прикладная дискретная математика. Приложение. 2009. № 2. С. 115–150.
5. *Bellare M. and Neven G.* Multi-signatures in the plain public-key model and a general forking lemma // Proc. 13th ACM Conf. CCM'06. 2006. P. 390–399.
6. *Vysotskaya V. V. and Chizhov I. V.* The security of the code-based signature scheme based on the Stern identification protocol // Прикладная дискретная математика. 2022. Т. 57. С. 67–90.
7. *Stern J.* A new identification scheme based on syndrome decoding // LNCS. 1993. V. 773. P. 13–21.
8. *Katz J. and Lindell Y.* Introduction to Modern Cryptography: 3d Ed. Chapman & Hall/CRC Cryptography and Network Security Series, 2021. 628 pp.

9. *Berlekamp E., McEliece R., and Van Tilborg H.* On the inherent intractability of certain coding problems // IEEE Trans. Inform. Theory. 1978. V. 24. No. 3. P. 384–386.
10. *Bellare M. and Rogaway P.* Random oracles are practical: A paradigm for designing efficient protocols // Proc. 1st ACM Conf. CCS'93. 1993. P. 62–73.
11. *Fischlin M., Lehmann A., Ristenpart T., et al.* Random oracles with (out) programmability // LNCS. 2010. V. 6477. P. 303–320.
12. *Shoup V.* Sequences of Games: a Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive. 2004. Paper 2004/332.
13. *Esser A. and Bellini E.* Syndrome Decoding Estimator. Cryptology ePrint Archive. 2021. Paper 2021/1243.
14. `https://github.com/Crypto-TII/syndrome_decoding_estimator` — Syndrome Decoding Estimator. 2021.

REFERENCES

1. *Pointcheval D. and Stern J.* Security proofs for signature schemes. EUROCRYPT'96, LNCS, 1996, vol. 1070, pp. 387–398.
2. *Damgård I.* On Σ-protocols. Lecture Notes, University of Aarhus, Department of Computer Science, 2002.
3. *Fiat A. and Shamir A.* How to prove yourself: Practical solutions to identification and signature problems. LNCS, 1987, vol. 263, pp. 186–194.
4. *Cheremushkin A. V.* Kriptograficheskie protokoly: osnovnye svoystva i uyazvimosti [Cryptographic protocols: main properties and vulnerabilities]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2009, no. 2, pp. 115–150. (in Russian)
5. *Bellare M. and Neven G.* Multi-signatures in the plain public-key model and a general forking lemma. Proc. 13th ACM Conf. CCM'06, 2006, pp. 390–399.
6. *Vysotskaya V. V. and Chizhov I. V.* The security of the code-based signature scheme based on the Stern identification protocol. Prikladnaya Diskretnaya Matematika, 2022, vol. 57, pp. 67–90.
7. *Stern, J.* A new identification scheme based on syndrome decoding. LNCS, 1993, vol. 773, pp. 13–21.
8. *Katz J. and Lindell Y.* Introduction to Modern Cryptography. 3d Ed. Chapman & Hall/CRC Cryptography and Network Security Series, 2021. 628 p.
9. *Berlekamp E., McEliece R., and Van Tilborg H.* On the inherent intractability of certain coding problems. IEEE Trans. Inform. Theory, 1978, vol. 24, no. 3, pp. 384–386.
10. *Bellare M. and Rogaway P.* Random oracles are practical: A paradigm for designing efficient protocols. Proc. 1st ACM Conf. CCS'93, 1993, pp. 62–73.
11. *Fischlin M., Lehmann A., Ristenpart T., et al.* Random oracles with (out) programmability. LNCS, 2010, vol. 6477, pp. 303–320.
12. *Shoup V.* Sequences of Games: a Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive, 2004, Paper 2004/332.
13. *Esser A. and Bellini E.* Syndrome Decoding Estimator. Cryptology ePrint Archive, 2021, Paper 2021/1243.
14. `https://github.com/Crypto-TII/syndrome_decoding_estimator` — Syndrome Decoding Estimator, 2021.