# О модификации схемы подписи Эль-Гамаля для применения в одном классе систем голосования, использующих механизм подписи вслепую

А. А. Бабуева

***Аннотация*—В настоящей работе рассматривается сценарий применения классической схемы подписи в одном классе систем дистанционного электронного голосования, использующих механизм подписи вслепую. Особенностями такого класса систем являются формирование подписи на мобильном устройстве избирателя и включение значения подписи в состав бюллетеня. Подчеркивается, что в таком случае от классической схемы подписи может требоваться малая длина подписи, а также обеспечение стойкости в случае использования вырожденных источников случайности. Для стандартных схем подписи Эль-Гамаля настоящие свойства не выполняются. При повторе случайного значения, используемого при формировании подписи, такие схемы позволяют восстановить ключ подписи. В настоящей работе показано, что схемы подписи Эль-Гамаля могут быть модифицированы с целью достижения указанных выше свойств. Предлагаемая модификация описана на примере стандартизированной в РФ схемы подписи, определенной в документе ГОСТ Р 34.10-2012. Она позволяет сократить длину подписи на четверть и дополнительно использует функцию HMAC. Получена оценка стойкости предложенной схемы в модели SUF-CMRA, предоставляющей нарушителю возможность контролировать случайные значения, а также метки времени, используемые в процессе формирования подписи.**

***Ключевые слова*—схема подписи Эль-Гамаля, схема подписи ГОСТ, короткая подпись, схема подписи вслепую**

## I. Введение

Различные области применения одного и того же криптографического механизма могут накладывать различные криптографические и эксплуатационные требования к нему. Рассмотрим такую область применения схем подписи, как системы дистанционного электронного голосования (ДЭГ). Как правило, при проведении голосования избиратель аутентифицируется перед Организатором голосования и получает некоторый «тикет», подтверждающий его право на голосование, после чего «бросает в урну» свой голос с использованием настоящего «тикета». Так, в случае традиционной (очной) системы голосования аналогом такого «тикета» является незаполненный бюллетень установленного образца. В случае систем ДЭГ «тикет» может представлять собой подписанное Организатором вслепую сообщение. В качестве подписываемого сообщения может выступать заполненный бюллетень избирателя (см. [1—4]) или одноразовый открытый ключ избирателя (см. [5—8]).

Рассмотрим сценарий применения схем подписи в системах ДЭГ, когда в качестве подписываемого вслепую сообщения выступает одноразовый открытый ключ избирателя. В этом случае предполагается, что после получения «тикета» избиратель заполнит бюллетень, подпишет его с помощью классической схемы подписи, после чего отправит Организатору по анонимному каналу связи транзакцию (подписанный бюллетень; открытый ключ проверки подписи; подпись открытого ключа проверки подписи, полученная вслепую). Таким образом, классическая схема подписи в данном случае позволяет обеспечить целостность бюллетеня, в то время как схема подписи вслепую позволяет обеспечить аутентичность этого бюллетеня, т.е. подтверждает, что настоящий открытый ключ проверки подписи действительно принадлежит легитимному избирателю.

Такой сценарий обладает следующими особенностями, требующими от классической схемы подписи наличия дополнительных криптографических и эксплуатационных свойств.

Формирование подписи происходит полностью на стороне избирателя. На практике это осуществляется на некотором пользовательском мобильном устройстве, на котором в общем случае сложно обеспечить выполнение требований по корректной генерации случайных значений [9]. Это связано, в первую очередь, с возможным отсутствием на пользовательских устройствах хороших, равновероятно распределенных начальных значений для программных датчиков случайных чисел. Поэтому в рассматриваемом случае целесообразно использовать схему подписи, обеспечивающую стойкость даже в случае, если одноразовые случайные значения, используемые для формирования подписи, выбираются не в соответствии с равновероятным распределением, в частности, могут быть зависимыми и даже повторяться.

Заметим, что в ряде систем может требоваться, чтобы схема подписи оставалась вероятностной даже при повторе случайных значений. Например, это является актуальным в системах, предоставляющих избирателю возможность повторной отправки бюллетеня и определяющих уникальность полученного бюллетеня по его хэш-значению. Тогда, в случае отсутствия данного свойства, возможна следующая ситуация, нарушающая коррект-

Статья получена 1 марта 2023

Александра Алексеевна Бабуева, МГУ им. М.В. Ломоносова, ООО ”КРИПТО-ПРО”, (email: babueva@cryptopro.ru).

ность работы системы. Легитимный избиратель отправляет бюллетень за кандидата А, далее отправляет бюллетень за кандидата Б, после чего — снова за кандидата А. Поскольку содержимое первого и третьего бюллетеней одинаковое, избиратель сформирует то же самое значение подписи и отправит идентичную транзакцию. С точки зрения системы, такие транзакции будут неразличимы, а потому третий бюллетень будет отброшен и будет учтен голос за кандидата Б. Заметим, что проверка уникальности полученных бюллетеней необходима, поскольку она защищает от повторной отправки нарушителем бюллетеней, сформированных легитимными избирателями.

Другой особенностью рассматриваемого класса систем ДЭГ является высокая нагрузка на Организатора голосования, для сокращения которой желательно сократить размер отправляемого бюллетеня, в частности, длину подписи.

Схемы подписи Эль-Гамаля [10] не являются стойкими в случае вырождения случайности [11], более того, повтор одноразовых случайных значений приводит к восстановлению ключа подписи. Длина подписи в этих схемах составляет $2\lceil \log q \rceil$ бит, где $q$ — порядок подгруппы группы точек используемой эллиптической кривой.

В настоящей работе показано, что схемы подписи Эль-Гамаля могут быть модифицированы с использованием идей из работ [11, 12] с целью достижения обсуждаемых выше свойств, представляющих интерес для рассматриваемого класса систем ДЭГ, использующих подпись вслепую. Предлагаемая модификация описана на примере схемы подписи, определенной в ГОСТ Р 34.10-2012 [13] (далее — схема GOST). Длина подписи в результирующей схеме составляет $\left(\frac{3}{2}\lceil \log q \rceil + 1\right)$ бит. Алгоритм формирования подписи остается вероятностным при повторе случайных значений за счет добавления метки времени в процесс выработки одноразового значения (см. раздел 7 [11]). Для модифицированной схемы получена оценка стойкости в модели SUF-CMRA, учитывающей возможность нарушителя контролировать случайные значения и метки времени, используемые в процессе формирования подписи.

## II. Обозначения

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

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

Будем обозначать множество всех отображений из $A$ в $B$ через $Func(A,B)$. Факт того, что значение $s$ выбрано из некоторого множества $S$ в соответствии с равновероятным распределением $\mathcal{U}$, будем обозначать через $s \xleftarrow{\mathcal{U}} S$.

Через $x \leftarrow val$ будем обозначать присваивание значения $val$ переменной $x$. Аналогично, через $x \leftarrow y$ будем обозначать присваивание значения переменной $y$ переменной $x$.

Пусть $p$ — простое число. Будем обозначать через $\mathbb{Z}_p$ конечное поле характеристики $p$. Будем считать, что все элементы $\mathbb{Z}_p$ имеют каноническое представление в виде натуральных чисел из отрезка $[0; p-1]$. Будем обозначать через $\mathbb{Z}_p^*$ множество $\mathbb{Z}_p$ без нулевого элемента.

Обозначим группу точек эллиптической кривой над полем $\mathbb{Z}_p$ через $\mathbb{G}$, порядок простой подгруппы группы $\mathbb{G}$ через $q$, точку эллиптической кривой порядка $q$ через $P$. Определим вспомогательный параметр $\ell$ равным $\lceil \log q \rceil$.

Пусть $\mathsf{HMAC} : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^\ell$ — ключевая хэш-функция, определенная в [14], отображающая ключ $K$ длины $\kappa$ бит и строку $T$ произвольной длины в двоичный вектор длины $\ell$ бит. Будем считать, что размер ключа $\kappa$ может принимать любые значения от $256$ до $512$ бит. В рамках настоящей работы будем считать, что хэш-функция $\mathsf{H}_1$, используемая в схеме подписи, действует из $\{0,1\}^*$ в $\mathbb{Z}_q$.

Схема подписи (далее — SS) задается тремя алгоритмами: алгоритм генерации ключей KGen, алгоритм формирования подписи Sig и алгоритм проверки подписи Vf. Алгоритм KGen возвращает пару ключей $(\mathsf{sk}, \mathsf{pk})$, где sk – секретный ключ подписи, pk – открытый ключ проверки подписи. Алгоритм Sig принимает на вход ключ подписи sk и сообщение $m$ и возвращает значение подписи $sgn$. Алгоритм Vf принимает на вход ключ проверки подписи pk, сообщение $m$, значение подписи $sgn$ и возвращает единицу, если значение подписи верное, и ноль в противном случае. При этом для любой пары ключей $(\mathsf{sk}, \mathsf{pk})$ и для любого сообщения $m$ требуется, чтобы

$$\mathsf{SS.Vf}(\mathsf{pk}, m, \mathsf{SS.Sig}(\mathsf{sk}, m)) = 1.$$

## III. Короткая усиленная подпись на основе схемы GOST

Схема подписи GOST задается алгоритмами генерации ключей, формирования и проверки подписи. Приведем их описание в виде псевдокода:

**KGen( )**
$d \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
$Q \leftarrow dP$
**return** $(d, Q)$

**Vf**$(Q, m, (r, s))$
**if** $(s = 0) \vee (r = 0)$ : **return** 0
$e \leftarrow \mathsf{H}_1(m)$
**if** $e = 0 : e \leftarrow 1$
$R \leftarrow e^{-1}sP - e^{-1}rQ$
**if** $R.x \bmod q \neq r$ :
  **return** 0
**return** 1

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

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

Для укорочения длины значения подписи применим метод 1 из работы [12], в результате получим схему GOST-H. Настоящий метод заключается в изменении способа вычисления первой компоненты подписи $r$ из эфемерной точки $R$. Если в оригинальной схеме GOST значение $r$ полагается равным $R.x \bmod q$, то в модифицированной схеме оно вычисляется следующим образом:

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

где $\mathsf{H}_2$ отображает $\mathbb{Z}_p$ в $\{0,1\}^{\ell/2}$, и $\phi$ отображает $\{0,1\}^{\ell/2}$ в $\mathbb{Z}_q^*$. Функция $\phi$ определяется следующим образом: она ставит в соответствие значению $x \in \{0,1\}^{\ell/2} \setminus \{0\}$ значение $\text{int}(x)$ и переводит 0 в $2^{\ell/2}$. Таким образом, длина компоненты $r$ подписи не превосходит $\ell/2 + 1 = \lceil \log q \rceil / 2 + 1$ бит. Отметим, что в силу задания области значений функции $\phi$ значение $r$ никогда не можеть быть нулевым, а потому из алгоритмов формирования и проверки подписи можно исключить проверку на равенство $r$ нулю.

Заметим, что метод, предложенный в [12], является более общим и позволяет генерировать подписи, первая компонента $r$ которых имеет длину $b$ бит, $b < \lceil \log q \rceil$, где $b$ — настраиваемый параметр. Однако в настоящей работе фиксируется значение $b = \ell/2$ из следующих соображений. Оценка стойкости схемы GOST-H, полученная в работе [12], свидетельствует о том, что при уменьшении значения $b$ с $\ell$ до $\ell/2$ оценка стойкости схемы не изменяется, в то время как при дальнейшем уменьшении параметра оценка стойкости начинает ухудшаться. Таким образом, значение $b = \ell/2$ позволяет сохранить стойкость схемы, улучшив ее эксплуатационные характеристики.

Согласно работе [12] схема подписи GOST-H также является схемой подписи Эль-Гамаля, а потому для этой схемы критично использование доверенного источника случайности (предположение о том, что эфемерный ключ подписи $k$ выбирается в соответствии с равновероятным распределением). Действительно, если нарушитель вместе со значением подписи $(r, s)$ получает также значение $k$, он может восстановить ключ подписи $d$ из уравнения подписи. Применим к схеме GOST-H метод из работы [11], обеспечивающий защиту от вырождения случайности. Настоящий метод заключается в изменении способа выработки эфемерного ключа подписи $k$. В оригинальной схеме GOST и схеме GOST-H предполагается, что значение $k$ выбирается из множества $\mathbb{Z}_q^*$ в соответствии с равновероятным распределением. В модифицированной схеме значение $k$ вырабатывается из ключа подписи $d$ и хэш-значения сообщения $e$ в результате следующей последовательности действий:

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

где $time$ — текущее значение времени (в мс), прошедшее с полуночи 01.01.1970, представленное в виде битовой строки длины $\ell$ бит. Заметим, что в [11] настоящий метод определен для случая, когда ключ подписи хранится в маскированном виде, однако наличие масок существенно зависит от конкретной реализации и, вообще говоря, не является обязательным. Поэтому в настоящей работе определяется модифицированная схема в случае отсутствия масок. Кроме того, используется идея из раздела 7 [11]: в эфемерный ключ подписи замешивается текущее значение времени $time$.

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

$\underline{\mathsf{KGen}(\,)}$
$d \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
$Q \leftarrow dP$
**return** $(d, Q)$

$\underline{\mathsf{Vf}(Q, m, (r, s))}$
**if** $s = 0$ : **return** 0
$e \leftarrow \mathsf{H}_1(m)$
**if** $e = 0 : e \leftarrow 1$
$R \leftarrow e^{-1}sP - e^{-1}rQ$
**if** $\phi(\mathsf{H}_2(R.x)) \neq r$ :
  **return** 0
**return** 1

$\underline{\mathsf{Sig}(d, m)}$
$k' \xleftarrow{\mathcal{U}} \{0,1\}^\ell$
$e \leftarrow \mathsf{H}_1(m)$
$K \leftarrow \mathsf{HMAC}(0^{256}, \text{str}_\ell(d))$
$k'' \leftarrow \mathsf{HMAC}(K, \text{str}_\ell(e) \parallel k' \parallel time)$
$k \leftarrow \text{int}(k'') \bmod q$
**if** $k = 0$ : **return** $\perp$
**if** $e = 0 : e \leftarrow 1$
$R \leftarrow kP$
$r \leftarrow \phi(\mathsf{H}_2(R.x))$
$s \leftarrow ke + dr$
**if** $s = 0$ : **return** $\perp$
**return** $(r, s)$

Заметим, что методы из работ [11, 12] являются общими и применимы ко всем схемам подписи Эль-Гамаля.

## IV. Свойства безопасности

В данном разделе введем целевую для схемы подписи модель безопасности SUF-CMRA, которая предоставляет нарушителю возможность контролировать все случайные значения, использующиеся в алгоритме подписи. Кроме того, определим стандартную модель безопасности SUF-CMA для схемы подписи, вспомогательные модели безопасности для функции хэширования и функции HMAC и задачу дискретного логарифмирования.

Для формализации моделей безопасности используется алгоритмический подход, в рамках которого определяется порядок взаимодействия между экспериментатором и нарушителем. Экспериментатор и нарушитель моделируются с помощью согласованных интерактивных вероятностных алгоритмов. Экспериментатор моделирует для нарушителя функционирование исследуемой криптосистемы и предоставляет ему доступ к одному или более оракулам (для деталей см. [15]).

Экспериментатор и нарушитель описываются с помощью псевдокода с использованием следующих обозначений. Для вероятностного алгоритма $A$ через $A \xrightarrow{\$} x$ ($x \xleftarrow{\$} A$) будем обозначать присваивание результата его работы переменной $x$. В случае, когда требуется подчеркнуть детерминированность алгоритма $A$, будем использовать обозначение $A \to x$ ($x \leftarrow A$).

### A. *Модель* SUF-CMRA

Определим модель SUF-CMRA [16] со случайным оракулом для схемы подписи $\widetilde{\text{GOST-H}}$, которая рассматривает стойкость к угрозе нахождения подделки при атаке не только с выбором сообщений, но и с выбором случайных значений (chosen message and randomness attack). Для схемы $\widetilde{\text{GOST-H}}$ единственным случайным значением, использующимся в процессе формирования

подписи, является значение $k'$. При формализации предполагается, что это значение подается нарушителем на вход оракулу подписи вместе с значением сообщения. Кроме того, настоящая модель учитывает, что источник времени, вообще говоря, может быть недоверенным, а потому предоставляет нарушителю контролировать значение времени $time$, также подавая его на вход оракулу подписи.

В рамках настоящей модели нарушителю также предоставляется доступ к двум случайным оракулам: оракул $RO_1$ моделирует вычисление значения функции $\mathsf{HMAC}(0^{256}, \cdot)$, оракул $RO_2$ моделирует вычисление значения хэш-функции $\mathsf{H}_2$. Заметим, что в работах [11, 12] при обосновании стойкости также предполагалось, что соответствующие функции моделируются как случайные оракулы. Релевантность подобных предположений и интерпретация результатов, полученных в модели со случайным оракулом, подробно обсуждается в оригинальных работах.

**Определение IV.1.** Преимущество нарушителя $\mathcal{A}$ для схемы подписи $\widetilde{\mathsf{GOST\text{-}H}}$ в модели SUF-CMRA со случайным оракулом определяется следующим образом:

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

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

$\underline{\mathbf{Exp}^{\text{SUF-CMRA}}_{\widetilde{\mathsf{GOST\text{-}H}}}(\mathcal{A})}$
$\mathcal{F}_1 \xleftarrow{\mathcal{U}} Func(\{0,1\}^{\ell}, \{0,1\}^{\ell})$
$\mathcal{F}_2 \xleftarrow{\mathcal{U}} Func(\mathbb{Z}_p, \{0,1\}^{\ell/2})$
$d \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$
$Q \leftarrow dP$
$\mathcal{L} \leftarrow \varnothing$
$(m, sgn) \xleftarrow{\$} \mathcal{A}^{Sign, RO_1, RO_2}(Q)$
**if** $(m, sgn) \in \mathcal{L}$ : **return** $0$
**return** $\widetilde{\mathsf{GOST\text{-}H}}.\mathsf{Vf}(Q, m, sgn)$

Oracle $RO_1(x)$
$y \leftarrow \mathcal{F}_1(x)$

Oracle $RO_2(x)$
$y \leftarrow \mathcal{F}_2(x)$

Oracle $Sign(m, k', time)$
$sgn \leftarrow \widetilde{\mathsf{GOST\text{-}H}}.\mathsf{SigDet}(d, m, k', time)$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, sgn)\}$
**return** $sgn$

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

### *B. Модель* SUF-CMA

Для схемы подписи $\mathsf{SS}$ определим стандартную модель SUF-CMA, которая рассматривает стойкость схемы подписи к угрозе нахождения подделки при атаке с выбором сообщений.

**Определение IV.2.** Преимущество нарушителя $\mathcal{A}$ для схемы подписи $\mathsf{SS}$ в модели SUF-CMA определяется следующим образом:

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

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

$\underline{\mathbf{Exp}^{\text{SUF-CMA}}_{\mathsf{SS}}(\mathcal{A})}$
$(sk, pk) \leftarrow \mathsf{SS.KGen}(\,)$
$\mathcal{L} \leftarrow \varnothing$
$(m, sgn) \xleftarrow{\$} \mathcal{A}^{Sign}(Q)$
**if** $(m, sgn) \in \mathcal{L}$ : **return** $0$
**return** $\mathsf{SS.Vf}(pk, m, sgn)$

Oracle $Sign(m)$
$sgn \leftarrow \mathsf{SS.Sig}(sk, m)$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, sgn)\}$
**return** $sgn$

### *C. Псевдослучайность функции* HMAC *(модель* PRF*)*

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

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

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

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

Oracle $HMAC(m)$
**if** $b = 1$ :
  **return** $\mathsf{HMAC}(K, m)$
**else** :
  **return** $\mathcal{F}(m)$

### *D. Задача дискретного логарифмирования* ECDLP

Определим формально модель ECDLP, задачей нарушителя в которой является нахождение для некоторой случайно выбранной точки $Q$ ее дискретного логарифма по основанию образующей точки $P$.

**Определение IV.3.** Преимущество нарушителя $\mathcal{A}$ для группы $\mathbb{G}$ в модели ECDLP определяется следующим образом:

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

### *E. Свойства хэш-функции*

Будем рассматривать ключевые хэш-функции $\mathsf{H}_1$: $\{0,1\}^* \mapsto \mathbb{Z}_q$, неявно инициализированные соответствующим вектором; предполагается, что эксперименты в определениях моделей начинаются со случайного выбора вектора $IV \in \mathcal{IV}$ и передачи его нарушителю.

Определим для семейства хэш-функций $\mathsf{H}_1$ свойство устойчивости к коллизиям с точностью до знака (свойство SCR) и свойство устойчивости к поиску делителей с точностью до знака (свойство SDR).

**Определение IV.4** (свойство SCR)**.** Преимущество нарушителя $\mathcal{A}$ для семейства хэш-функций $\mathsf{H}_1$ в модели SCR определяется следующим образом:

$$\mathsf{Adv}^{\text{SCR}}_{\mathsf{H}_1}(\mathcal{A}) =$$
$$= \Pr\left[(m_1, m_2) \xleftarrow{\$} \mathcal{A} : \mathsf{H}_1(m_1) = \pm\mathsf{H}_1(m_2) \wedge m_1 \neq m_2\right]$$

**Определение IV.5** (свойство SDR)**.** Преимущество нарушителя $\mathcal{A}$ для семейства хэш-функций $\mathsf{H}_1$ в модели SCR определяется следующим образом:

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

## V. Оценка стойкости

Получим оценку стойкости схемы $\widetilde{\mathsf{GOST\text{-}H}}$ в модели SUF-CMRA, опираясь на результаты работ [11, 12].

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

- *нарушитель $\mathcal{D}$, решающий задачу* ECDLP *в используемой группе точек $\mathbb{G}$,*
- *нарушитель $\mathcal{C}$, решающий задачу* SCR *(поиска коллизий с точностью до знака) для хэш-функции $\mathsf{H}_1$,*
- *нарушитель $\mathcal{M}$, решающий задачу* SDR *(поиска делителей с точностью до знака) для хэш-функции $\mathsf{H}_1$,*
- *нарушитель $\mathcal{P}$ для схемы* HMAC *в модели* PRF*, делающий не более $Q_S$ запросов к оракулу, длина каждого из которых не превосходит $3\ell$ битов,*

*такие, что:*

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

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

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

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

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

*а) Оценка для метода усиления случайности (Теорема 1 из работы [11] для схемы подписи $\mathsf{SS} = \mathsf{GOST\text{-}H}$):*

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

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

Также заметим, что ограничение на длину запросов нарушителя $\mathcal{P}$ поменялось по сравнению с оценкой из работы [11]. Это связано с добавлением времени $time$ в аргументы функции HMAC при выработке значения $k''$. Поскольку на вход функции HMAC приходят значения $\mathrm{str}_\ell(e), k', time$, каждое из которых имеет длину $\ell$ бит, длина запросов нарушителя $\mathcal{P}$ составляет $3\ell$ бит. При этом, все остальное доказательство остается неизменным с добавлением $time$ и в точности повторяет доказательство [11].

*b) Оценка для метода укорочения (Теорема 1 из работы [12]):*

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

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

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

Вычислительные ресурсы нарушителей также определяются путем совмещения соответствующих оценок из работ [11, 12].

□

Аналогичная теорема может быть сформулирована для модифицированных схем подписи, основанных на других схемах подписи Эль-Гамаля. Этап а) доказательства в таком случае проводится идентично, поскольку оценка из работы [11] применима к произвольной схеме подписи Эль-Гамаля. При этом этап б) доказательства требует оценки стойкости укороченной схемы подписи в модели SUF-CMA, которая, в свою очередь, может быть получена с применением идей из работы [10] (в работе [12] настоящая оценка явно приводится только для схемы подписи GOST-H).

## VI. Заключение

В настоящей работе предложен способ модификации схем подписи Эль-Гамаля с целью повышения их защищенности в случае использования недоверенного источника случайности, а также уменьшения размера подписи. В качестве примера настоящая модификация применена к схеме подписи, определенной в ГОСТ Р 34.10-2012. Получена оценка стойкости результирующей схемы в модели безопасности SUF-CMRA, позволяющей нарушителю контролировать все случайные значения, используемые при формировании подписи. Предложенная схема может применяться в системах дистанционного электронного голосования, использующих протокол подписи вслепую.

Библиография


[1] A. Fujioka, T. Okamoto и K. Ohta, «A practical secret voting scheme for large scale elections,» в *Advances in Cryptology — AUSCRYPT '92*, J. Seberry и Y. Zheng, ред., Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, с. 244—251.

[2] T. Okamoto, «An electronic voting scheme,» в *Advanced IT Tools: IFIP World Conference on IT Tools 2–6 September 1996, Canberra, Australia*, N. Terashima и E. Altman, ред. Boston, MA: Springer US, 1996, с. 21—30.

[3] C.-I. Fan и W.-Z. Sun, «Uncoercible Anonymous Electronic Voting,» т. 2006, янв. 2006. DOI: 10.2991/jcis.2006.229.

[4] M. Chaieb, M. Koscina, S. Yousfi, P. Lafourcade и R. Robbana, *DABSTERS: Distributed Authorities using Blind Signature To Effect Robust Security in e-voting*, 2019. url: https://hal.science/hal-02145809.

[5] Q. He и Z. Su, «A new practical secure e-voting scheme,» в *SEC'98: international conference on information security*, 1998, с. 196—205.

[6] L. López-García, F. Rodríguez-Henríquez и M. A. L. Chávez, «An e-Voting Protocol based on Pairing Blind Signatures,» в *International Conference on Security and Cryptography*, 2008.

[7] D. Kirillov, V. V. Korkhov, V. Petrunin, M. Makarov, I. M. Khamitov и V. Dostov, «Implementation of an E-Voting Scheme Using Hyperledger Fabric Permissioned Blockchain,» в *Communication Systems and Applications*, 2019.

[8] *Дистанционное электронное голосование*, Рус-Крипто'22, 2022. url: https://www.ruscrypto.ru/resource/archive/rc2022/files/05_presentation.pdf.

[9] С. Смышляев, *Математические методы обоснования оценок уровня информационной безопасности программных средств защиты информации, функционирующих в слабодоверенном окружении*, Московский государственный университет имени М.В. Ломоносова, дис. ... д-ра ф.-м.н., 2022.

[10] M. Fersch, *The provable security of elgamal-type signature schemes*, Ruhr-Universität Bochum, Doct. Diss., 2018.

[11] Е. Алексеев, Л. Ахметзянова, А. Бабуева и С. Смышляев, «О повышении безопасности схем подписи Эль-Гамаля,» *Математические вопросы криптографии*, т. 12(3), 2021.

[12] L. Akhmetzyanova, E. Alekseev, A. Babueva и S. Smyshlyaev, «On methods of shortening ElGamal-type signatures,» *Математические вопросы криптографии*, т. 12(2), 2021.

[13] *ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи»*, Стандартинформ, 2012.

[14] H. Krawczyk, M. Bellare и R. Canetti, «HMAC: Keyed-Hashing for Message Authentication,» RFC 2104, 1997. url: https://www.rfc-editor.org/info/rfc2104.

[15] M. Bellare и P. Rogaway, «The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs,» в *Advances in Cryptology - EUROCRYPT 2006*, S. Vaudenay, ред., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, с. 409—426.

[16] Y. S. Ristenpart T., *When Good Randomness Goes Bad: Virtual Machine Reset Vulnerabilities and Hedging Deployed Cryptography*, NDSS, 2010.

# On ElGamal signature modification for application in one class of e-voting systems using the blind signature mechanism

Alexandra Babueva

***Abstract*—In this paper we consider the scenario of the signature scheme application in one class of e-voting systems using the blind signature mechanism. The specificities of this class of systems are the signature generation on the voter's mobile device and the inclusion of the signature value in the ballot. In this case the classical signature scheme may be required to provide a short signature length and remain secure even in case of using unreliable sources of randomness. Standard ElGamal signature schemes do not meet these requirements. The repeating of the random value used in the signature generation process leads to recovering the secret signing key. This paper shows that ElGamal signature schemes can be modified in order to ensure these properties. The proposed modification is described on the example of the standardized signature scheme, defined in the document GOST R 34.10-2012. It allows to reduce the length of the signature by a quarter and additionally uses the HMAC function. We obtain the security bound of the proposed scheme in the SUF-CMRA model, which allows the adversary to control random values, as well as timestamps used in the signature generation process.**

***Keywords*—ElGamal signature, GOST signature, short signature, blind signature**

## References

[1] A. Fujioka, T. Okamoto, and K. Ohta, «A practical secret voting scheme for large scale elections», in *Advances in Cryptology — AUSCRYPT '92*, J. Seberry and Y. Zheng, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 244–251.

[2] T. Okamoto, «An electronic voting scheme», in *Advanced IT Tools: IFIP World Conference on IT Tools 2–6 September 1996, Canberra, Australia*, N. Terashima and E. Altman, Eds. Boston, MA: Springer US, 1996, pp. 21–30.

[3] C.-I. Fan and W.-Z. Sun, «Uncoercible anonymous electronic voting», vol. 2006, Jan. 2006. DOI: 10.2991/jcis.2006.229.

[4] M. Chaieb, M. Koscina, S. Yousfi, P. Lafourcade, and R. Robbana, *Dabsters: Distributed authorities using blind signature to effect robust security in e-voting*, 2019. [Online]. Available: https://hal.science/hal-02145809.

[5] Q. He and Z. Su, «A new practical secure e-voting scheme», in *SEC'98: international conference on information security*, 1998, pp. 196–205.

[6] L. López-García, F. Rodríguez-Henríquez, and M. A. L. Chávez, «An e-voting protocol based on pairing blind signatures», in *International Conference on Security and Cryptography*, 2008.

[7] D. Kirillov, V. V. Korkhov, V. Petrunin, M. Makarov, I. M. Khamitov, and V. Dostov, «Implementation of an e-voting scheme using hyperledger fabric permissioned blockchain», in *Communication Systems and Applications*, 2019.

[8] *E-voting system*, RusCrypto'22, 2022. [Online]. Available: https://www.ruscrypto.ru/resource/archive/rc2022/files/05_presentation.pdf.

[9] S. Smyshlyaev, *Mathematical methods of proving security bounds for information security software running in the semi-trusted environment*, Lomonosov Moscow State University, Doct. Diss., 2022.

[10] M. Fersch, *The provable security of elgamal-type signature schemes*, Ruhr-Universität Bochum, Doct. Diss., 2018.

[11] L. Akhmetzyanova, E. Alekseev, A. Babueva, and S. Smyshlyaev, «Improving security of ElGamal-type signatures», *Matem. vopr. kriptogr.*, vol. 12(3), 2021.

[12] L. R. Akhmetzyanova, E. K. Alekseev, A. Babueva, and S. Smyshlyaev, «On methods of shortening ElGamal-type signatures», *Matem. vopr. kriptogr.*, vol. 12(2), 2021.

[13] *GOST R 34.10-2012. Information technology. Cryptographic data security. Signature and verification processes of electronic digital signature. National standard of the russian federation*, STANDARTINFORM, 2012.

[14] H. Krawczyk, M. Bellare, and R. Canetti, «HMAC: Keyed-hashing for message authentication», RFC 2104, 1997. [Online]. Available: https://www.rfc-editor.org/info/rfc2104.

[15] M. Bellare and P. Rogaway, «The security of triple encryption and a framework for code-based game-playing proofs», in *Advances in Cryptology - EUROCRYPT 2006*, S. Vaudenay, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 409–426.

[16] Y. S. Ristenpart T., *When good randomness goes bad: Virtual machine reset vulnerabilities and hedging deployed cryptography*, NDSS, 2010.