Math-Net.Ru

Общероссийский математический портал

Е. К. Алексеев, И. Б. Ошкин, В. О. Попов, С. В. Смышляев, О криптографических свойствах алгоритмов, сопутствующих применению стандартов ГОСТ Р 34.11-2012 и ГОСТ Р 34.10-2012, *Матем. вопр. криптогр.*, 2016, том 7, выпуск 1, 5–38

DOI: <https://doi.org/10.4213/mvk172>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 54.92.164.108

26 апреля 2023 г., 14:26:55

**О криптографических свойствах алгоритмов,
сопутствующих применению стандартов
ГОСТ Р 34.11-2012 и ГОСТ Р 34.10-2012**

Е. К. Алексеев, И. Б. Ошкин, В. О. Попов, С. В. Смышляев

ООО «КРИПТО-ПРО», Москва

Получено 13.IX.2015

В ряде общепринятых моделей противника обосновываются криптографические свойства алгоритмов, сопутствующих применению стандартов ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012: алгоритма выработки кода аутентификации сообщений, алгоритма генерации псевдослучайных последовательностей и алгоритма выработки ключа обмена. Выводы о стойкости этих алгоритмов базируются на свойствах алгоритма ГОСТ Р 34.11-2012 и классических теоретико-сложностных задач (распознавательной и вычислительной задач Диффи – Хеллмана).

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 16-01-00226 А.

Ключевые слова: криптографические алгоритмы, аутентификация сообщений, псевдослучайные последовательности, выработка ключа

On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012

E. K. Alekseev, I. B. Oshkin, V. O. Popov, S. V. Smyshlyayev

CryptoPro LLC, Moscow

Abstract. For a number of commonly accepted models of eavesdropper the cryptographic properties of the algorithms accompanying the standards GOST R 34.11-2012 and GOST R 34.10-2012 are justified, namely, the algorithms generating the message authentication codes, pseudorandom sequences and the key exchange algorithm. The conclusions on the security of these algorithms are based on the properties of the GOST R 34.11-2012 algorithm and on classical complexity-theoretical problems (decisional and computational Diffie – Hellman problems).

Keywords: cryptographic algorithms, message authentication, pseudorandom sequence generators, key generation algorithms

Citation: *Mathematical Aspects of Cryptography*, 2016, v. 7, № 1, pp. 5–38 (Russian)

© Академия криптографии Российской Федерации, 2016 г.

1. Введение

Криптографические решения в современных информационных системах, как правило, базируются на криптографических алгоритмах, зафиксированных в качестве национальных стандартов. Разработка стандартов проводится под руководством и преимущественно внутри закрытых научно-исследовательских центров. Принятие алгоритма в качестве стандарта в определенном смысле гарантирует выполнение набора требований, предъявляемых к данным алгоритмам. Набор требований к функции хэширования иллюстрируется, например, списком задач, предложенных на конкурсе по исследованию свойств алгоритма ГОСТ Р 34.11-2012 [27].

Разработчики систем защиты информации предлагают высокоуровневые криптографические схемы, которые строятся на основе криптопримитивов. Такие схемы требуют проведения исследования их криптографических свойств. Так, например, необходимым условием включения высокоуровневых схем в рекомендации Технического комитета по стандартизации «Криптографическая защита информации» (ТК 26) Росстандарта является обоснование их стойкости, которое возлагается на разработчиков данных схем. Традиционный способ обоснования стойкости высокоуровневых схем заключается в сведении их свойств к свойствам базовых алгоритмов, используемых в данных схемах в качестве элементарных блоков.

Настоящая работа посвящена анализу криптографических свойств алгоритма выработки кода аутентификации сообщений, алгоритма генерации псевдослучайных последовательностей и алгоритма выработки ключа обмена. Показано, что стойкость указанных алгоритмов в различных моделях противника является следствием предположений о некоторых свойствах алгоритма хэширования ГОСТ Р 34.11-2012, а также предположений о сложности вычислительной и распознавательной задач Диффи – Хеллмана в группах точек эллиптических кривых, заданных в рекомендациях по стандартизации ТК 26 [20].

Статья содержит также элементы обзорного характера со ссылками на источники, содержащие известные результаты. Существенное влияние на данную работу оказали результаты, полученные Коблицем и Менезесом ([2, 3]).

2. Соглашения и обозначения

Далее $\mathbb{N} = \{1, 2, \dots\}$ — множество натуральных чисел, $\mathbb{N}_0 = \mathbb{N} \cup \{0\} = \{0, 1, 2, \dots\}$, $\mathbb{F}_2$ — конечное поле $\text{GF}(2)$, V_n — множество строк длины $n \in \mathbb{N}_0$ с элементами из $\mathbb{F}_2$ (далее — *битовые строки*). Для $u \in V_{n_1}$ и $v \in V_{n_2}$ через $u||v$ обозначим конкатенацию строк u и v , $u||v \in V_{n_1+n_2}$.

Элементы множества V_8^r для $r \in \mathbb{N}_0$ будем называть *байтовыми строками*. Любая строка $v \in V_n$ для n , кратного 8, может быть естественным образом интерпретирована как байтовая. Множество битовых строк, длины которых принадлежат множеству $\{a, a+1, \dots, b\}$, будем обозначать $V_{[a,b]} (= \bigcup_{r=a}^b V_r)$.

Пусть $LE_r : \{0, 1, \dots, 2^{8r} - 1\} \rightarrow V_8^r$ — некоторое произвольное фиксированное биективное отображение. *Байтовым представлением* $[n]_b$ произвольного числа $n \in \mathbb{N}$ будем называть строку $LE_r(n)$, где $8r$ — минимальное кратное 8 натуральное число, большее или равное $\log_2(n+1)$.

Будем обозначать GH_t ($t \in \{256, 512\}$) определенную в документе [16] хэш-функцию ГОСТ Р 34.11-2012 с длиной выхода, равной t битам. Заметим, что в документе [16] зафиксировано ограничение длины сообщения M , подлежащего хэшированию: $|M| < 2^{512}$. В связи с этим функция GH_t осуществляет отображение из множества $V_{[0, 2^{512}-1]}$ в V_t .

Через E будем обозначать подгруппу простого порядка q группы точек некоторой эллиптической кривой порядка μ , удовлетворяющей требованиям документа [17]. Порождающий элемент подгруппы E обозначим через P .

Если $\mathcal{U}$ — распределение вероятностей на некотором множестве S , то запись $x \in_{\mathcal{U}} S$ означает, что x выбирается случайно из S в соответствии с распределением $\mathcal{U}$.

Далее все входные параметры и выходные значения алгоритмов описываются байтовыми строками, если не указано обратное.

В случае когда рассуждения не зависят от индексов (верхних или нижних), приписанных исследуемым объектам, эти индексы опускаются.

3. Краткое описание рассматриваемых алгоритмов

Рассматриваемые в работе алгоритмы подробно описаны в документе [4]. Таблица (см. Дополнение 2) содержит наименования алгоритмов согласно [4] и их сокращенные обозначения, используемые в настоящей работе. Степень детализации приведенного ниже описания достаточна для обоснования их криптографических свойств.

Поясним смысл используемых аббревиатур:

- HMAC — Hash Based Message Authentication Code, код аутентификации сообщения, вычисляемый с помощью хэш-функции;
- PRF — Pseudo Random Function, функция выработки псевдослучайной строки;
- VKO — функция выработки ключа обмена (данная аббревиатура используется по историческим причинам, связанным с введенным в [13] классом процедур выработки ключа обмена под названием VKO);

- KDF — Key Derivation Function, функция выработки производных ключей;
- KDFTREE — Key Derivation Function Tree, функция выработки производных ключей, допускающая организацию иерархической древовидной зависимости ключей.

Везде далее значение параметра t принадлежит множеству $\{256, 512\}$, а значение длины ключа n удовлетворяет условию $256 \leq n \leq 512$.

Под блоком далее понимается битовая строка длины 512.

Для каждого $i \in \{1, 2, 3\}$ через $\text{PRF}_t^{(i)}$ обозначается семейство функций, параметризованное посредством натуральных величин m и r , которые для краткости не указываются в наименовании функции. Каждый раз при обсуждении свойств $\text{PRF}_t^{(i)}$ считается, что указанные значения m, r некоторым образом зафиксированы, т. е. речь идет уже о конкретной функции.

3.1. HMAC_t

Функции HMAC_t реализуют отображения

$$\text{HMAC}_t : V_n \times V_{[0, 2^{512}-1]} \rightarrow V_t.$$

Первым аргументом является ключ K , вторым — данные T . При необходимости ключ дополняется нулями до строки длины 512 бит: $K^* = K \parallel (0 \dots 0) \in V_{512}$. Окончательно

$$\begin{aligned} \text{HMAC}_t(K, M) &= \text{GH}_t(K^* \oplus \text{opad} \parallel \text{GH}_t(K^* \oplus \text{ipad} \parallel M)), \\ \text{ipad} &= \underbrace{(00110110) \parallel \dots \parallel (00110110)}_{64} \in V_{512}, \\ \text{opad} &= \underbrace{(01011100) \parallel \dots \parallel (01011100)}_{64} \in V_{512}. \end{aligned}$$

Алгоритмы HMAC_t определены в соответствии с документом [5]. Также HMAC описан в документах [6, 7, 8].

3.2. PRF_t⁽¹⁾

Функция $\text{PRF}_t^{(1)} : V_n \times V_m \rightarrow V_{r \cdot t}$ определена следующим образом:

$$\begin{aligned} \text{PRF}_t^{(1)}(K, M) &= \\ &= \underbrace{\text{HMAC}_t(K, A_1 \parallel M)}_{D_1} \parallel \underbrace{\text{HMAC}_t(K, A_2 \parallel M)}_{D_2} \parallel \dots \parallel \underbrace{\text{HMAC}_t(K, A_r \parallel M)}_{D_r}, \end{aligned}$$

где $A_1 = \text{HMAC}_t(K, M)$, $A_2 = \text{HMAC}_t(K, A_1), \dots$

Функция $\text{PRF}_t^{(1)}$ определена в соответствии с принципами документов [9, 10].

Отметим, что в настоящей работе вместо двух параметров, фигурирующих в [9, 10] и имеющих фиксированные длины, используется один параметр M . Это не нарушает общности рассуждений, поскольку при вычислении значений оригинальной функции используется лишь конкатенация этих двух параметров, а фиксированность их длин позволяет говорить об этой паре как об одной строке.

3.3. $\text{PRF}_t^{(2)}$

Функция $\text{PRF}_t^{(2)} : V_n \times V_m \rightarrow V_{r \cdot t}$ определена следующим образом:

$$\begin{aligned} \text{PRF}_t^{(2)}(K, M) &= \\ &= \underbrace{\text{HMAC}_t(K, M)}_{D_1} \parallel \underbrace{\text{HMAC}_t(K, D_1 \parallel M)}_{D_2} \parallel \dots \parallel \underbrace{\text{HMAC}_t(K, D_{r-1} \parallel M)}_{D_r}. \end{aligned}$$

Функции $\text{PRF}_t^{(2)}$ по схеме задания аргументов в итерациях аналогичны функции KEYMAT в [11].

3.4. $\text{PRF}_t^{(3)}$

Функция $\text{PRF}_t^{(3)} : V_n \times V_m \rightarrow V_{r \cdot t}$ определена следующим образом:

$$\begin{aligned} \text{PRF}_t^{(3)}(K, M) &= \underbrace{\text{HMAC}_t(K, M \parallel LE_1(1))}_{D_1} \parallel \underbrace{\text{HMAC}_t(K, D_1 \parallel M \parallel LE_1(2))}_{D_2} \parallel \\ &\dots \parallel \underbrace{\text{HMAC}_t(K, D_{r-1} \parallel M \parallel LE_1(r))}_{D_r}. \end{aligned}$$

Функции $\text{PRF}_t^{(3)}$ по схеме задания аргументов в итерациях аналогичны функции prf^+ в [12].

3.5. KDFTREE

Через KDFTREE обозначается семейство функций, параметризованное посредством натуральных величин m_1, m_2, r и R ($1 \leq r \leq 256 \cdot (2^{8R} - 1)$, $R \in \{1, 2, 3, 4\}$). В рассуждениях всегда будем полагать, что указанные параметры некоторым образом зафиксированы.

Функция $\text{KDFTREE} : V_n \times V_{m_1} \times V_{m_2} \rightarrow V_r$ определена следующим образом:

$$\text{KDFTREE}(K, \text{label}, \text{seed}) = K_1 \parallel K_2 \parallel K_3 \parallel \dots,$$

где $K_i = \text{HMAC}_{256}(K, LE_R(i) \parallel \text{label} \parallel LE_1(0) \parallel \text{seed} \parallel [r]_b)$.

Функция диверсификации KDFTREE использует общие принципы задания входных параметров и выхода для функций диверсификации, изложенные в документе [14].

Далее (без ограничения общности) нами будет рассматриваться функция $\text{KDFTREE} : V_n \times V_m \rightarrow V_r$, определенная следующим образом:

$$\text{KDFTREE}(K, M) = K_1 || K_2 || K_3 || \dots,$$

где $K_i = \text{HMAC}_{256}(K, LE_R(i) || M || [r]_b)$.

3.6. KDF

Функция $\text{KDF} : V_n \times V_m \rightarrow V_{256}$ определена следующим образом:

$$\text{KDF}(K, M) = \text{KDFTREE}(K, M) = \text{HMAC}_{256}(K, LE_1(1) || M || LE_1(1) || LE_1(0)).$$

Заметим, что KDF является частным случаем функции KDFTREE : длина выхода r зафиксирована и равна 256 битам, $R = 1$.

3.7. VKO_t

Функция $\text{VKO}_t : Z_q^* \times Z_q^* \times Z_q^* \rightarrow V_t$ определена следующим образом:

$$\text{VKO}_t(x, y, UKM) = \text{GH}_t \left(\left(\frac{\mu}{q} \cdot UKM \cdot x \right) (\bmod q) \cdot (yP) \right).$$

Алгоритмы VKO_t могут использоваться для согласования ключей длиной t бит. Параметр UKM (с учетом использования хэш-функции) позволяет многократно реализовывать VKO_t на долговременных закрытых ключах x, y (при этом UKM передается по каналу в открытом виде и от него требуется лишь неповторяемость). Множитель μ/q используется для того, чтобы кратная точка не могла принадлежать подгруппе малого порядка (в прикладных протоколах обычно проверяется, что кратная точка не является нейтральным элементом группы).

Алгоритмы определены по аналогии с разделом 5.2 документа [13].

4. Модели противника

В соответствии с терминологией, принятой в математической криптографии (см., например, [1]), модель противника состоит из следующих частей:

- *атака* — совокупность предположений о противнике и об информации, получаемой противником в ходе взаимодействия с исследуемым криптографическим объектом, в рамках которых анализируется стойкость этого объекта;

- *угроза* — задача, стоящая перед противником (например, задача нарушения конфиденциальности или целостности).
- *предположение о вычислительных ресурсах*, которыми располагает противник; под вычислительными ресурсами понимается сумма количества выполняемых противником операций и размера используемого кода.

В данной работе получены нижние оценки вычислительных ресурсов противника, при которых достижимо заданное значение вероятности успеха (или преобладания в задачах различения). Поэтому в настоящей работе модели противника содержат описания лишь первых двух пунктов — атаки и угрозы.

4.1. Модели противника для HMAC, PRF, KDF и KDFTREE

Пусть $F : V_n \times V_m \rightarrow V_{r,t}$, $F(K, M) = \underbrace{D_1 || D_2 || \dots || D_r}_D$, где $D_i \in V_t$, а

r — фиксированное значение, $r \in \mathbb{N}$. Описание функции F общедоступно.

Далее рассматривается *атака с адаптивным выбором открытых текстов*: противник имеет доступ к оракулу $\mathcal{O}_F$, который перед началом работы случайно и равновероятно выбирает и фиксирует ключ K из множества V_n . Оракул принимает на вход сообщения $M \in V_m$, в ответ на которые возвращает значение $F(K, M)$. Противник может адаптивно выбирать сообщения M_i и подавать их на вход оракулу, т. е. с учетом выбранных им ранее сообщений и полученных от оракула ответов.

Для некоторого оракула $\mathcal{O}$ через $\mathcal{O}(M)$ будем обозначать ответ этого оракула на строку M .

Вкупе с описанной атакой можно рассматривать следующие угрозы (далее используется устоявшаяся в работах отечественных авторов терминология, см., например, [1] или [30]):

- *восстановление ключа*: противник восстанавливает ключ K ,
- *селективная подделка*: для некоторого фиксированного M , выбранного противником перед осуществлением первого запроса к оракулу, противник строит образ $D = F(K, M)$,
- *экзистенциальная подделка*: противник строит такую пару (M, D) , что $D = F(K, M)$.

При реализации угрозы селективной или экзистенциальной подделки противник выдает в качестве результата пару (M, D) . Данные угрозы считаются реализованными успешно, только если сообщение M из результирующей пары (M, D) не встречалось среди запросов к оракулу $\mathcal{O}_F$.

Также рассматривается атака с адаптивным выбором открытых текстов для угрозы различения. При этом оракул $\mathcal{O}_F$, доступный противнику, перед началом работы случайно и равновероятно выбирает значение бита $b \in \{0, 1\}$. Если $b = 1$ (в этом случае пишем $\mathcal{O}_F = F$), то оракул случайно и равновероятно выбирает и фиксирует ключ $K \in V_n$ и на каждый запрос $M \in V_m$ возвращает значение $F(K, M)$. Если же $b = 0$ (в этом случае используется запись $\mathcal{O}_F = RF$), то при поступлении запроса $M \in V_m$, не встречавшегося ранее, оракул возвращает строку D , выбранную случайно и равновероятно из $V_{r,t}$, и запоминает пару (M, D) . При повторном поступлении запроса M оракул возвращает строку D из запомненной им пары (M, D) . Угроза формулируется следующим образом:

- *различение*: противник определяет значения бита b , используемого оракулом $\mathcal{O}_F$.

4.2. Модели противника для VKO

В криптографических исследованиях протокола Диффи–Хеллмана естественным образом возникают две задачи: по xP , yP найти xyP (вычислительная задача Диффи–Хеллмана, CDH) и по xP , yP и xyP найти x (эта задача эквивалентна задаче дискретного логарифмирования). Применительно к функции VKO возникают аналогичные задачи: по UKM , xP и yP найти $VKO(x, y, UKM)$ и по $VKO(x, y, UKM)$, xP и yP найти x .

Рассматриваемые ниже модели противника для функции VKO сформулированы в обобщенных условиях, для которых указанные выше задачи являются частными случаями, определяемыми выбором функций R_1 , R_2 и R_3 .

- **Модель 1.** *Атака*: противнику известны UKM , xP , yP , значение $K = VKO(x, y, UKM)$ и некоторая информация $R_1(x, y)$ о паре (x, y) .
Угроза: противник находит некоторую новую информацию $R_2(x, y)$ о паре (x, y) .
- **Модель 2.** *Атака*: противнику известны xP , yP и UKM .
Угроза: противник находит $VKO(x, y, UKM)$.

Представляет интерес рассмотрение модели противника, имеющего дополнительную информацию о результатах промежуточных вычислений:

- **Модель 2'.** *Атака*: противник знает открытые ключи xP , yP , параметр UKM алгоритма VKO и некоторую дополнительную информацию $R_3(xyP)$ о точке xyP .
Угроза: противник находит $VKO(x, y, UKM)$.

В Модели 2' разумными представляются следующие требования: сужение группы E , индуцируемое имеющейся у противника дополнительной информацией $R_3(xyP)$, является эффективно конструируемым, а распознавательная задача Диффи–Хеллмана (см. ниже с. 27, раздел 5.2) на этом сужении является труднорешаемой. Тогда обоснование стойкости алгоритма VKO в Модели 2' строится аналогично обоснованию для Модели 2 (эффективно конструируемым множеством, на котором вероятность коллизии для хэш-функции высока, является сужение группы E , индуцируемое дополнительной информацией $R_3(xyP)$).

5. Обоснование стойкости

Под противником, которого будем обозначать через A , далее понимается вероятностная машина Тьюринга. В дальнейших рассуждениях вероятностное пространство определяется случайной лентой противника и вероятностными свойствами используемого оракула. При этом всегда предполагается, что значения на случайной ленте выбираются случайно и равномерно.

Будем называть (T, q, u) -*противником* такого противника A , вычислительные ресурсы которого не превосходят T и который производит не более q запросов к своему оракулу, причем каждый запрос состоит не более чем из u блоков (учет количества блоков применим к функции HMAC).

Преобладанием противника A при реализации угрозы различения называется величина

$$\Pr[A = 1 | b_{\text{true}} = 1] - \Pr[A = 1 | b_{\text{true}} = 0],$$

где запись $A = b$ означает, что противник A дал ответ b , а b_{true} — истинное значение определяемого бита.

Функцию или задачу будем называть (ε, T, q, u) -*стойкой* (в некоторой модели противника, при этом, когда речь идет о задаче, модель может быть не указана), если вероятность успеха или преобладание в задаче различения для любого (T, q, u) -противника не превосходит ε . Такие параметры, как q и u , имеют смысл при использовании противником оракула, если же задача (например, DDH) не предполагает использования оракула, то эти параметры опускаются, а параметры ε и T сохраняют свое значение.

Функция $F(K, M)$ называется (ε, T, q, u) -*стойкой псевдослучайной функцией*, если преобладание любого (T, q, u) -противника при реализации угрозы различения меньше или равно ε .

5.1. Обоснование стойкости HMAC, PRF, KDF, KDFTREE

5.1.1. Сведение к псевдослучайности функции HMAC

Далее через PRF может обозначаться любая из функций $\text{PRF}_t^{(1)}$, $\text{PRF}_t^{(2)}$, $\text{PRF}_t^{(3)}$. При использовании обозначения PRF рассуждения справедливы для любой из этих функций.

Возможности противника по реализации угроз представляются ориентированным ациклическим графом (без ориентированных циклов), вершины которого помечены парами (*угроза, функция*).

Рис. Соотношения между функциями и моделями противника

Дуга, исходящая из вершины (U_1, f_1) в вершину (U_2, f_2) , означает, что осуществить угрозу U_2 по отношению к функции f_2 можно посредством противника, реализующего угрозу U_1 по отношению к функции f_1 . При этом вычислительные затраты и вероятность успеха при реализации U_2 по отношению к f_2 будут незначительно отличаться от соответствующих параметров, необходимых для осуществления угрозы U_1 по отношению к функции f_1 . Таким образом, существование дуги, исходящей из вершины (U_1, f_1) и входящей в вершину (U_2, f_2) , означает, что осуществить угрозу U_1 для функции f_1 не легче, чем осуществить угрозу U_2 для функции f_2 .

Обоснования соотношений, соответствующих направленным дугам графа, изображенного на рисунке, проводятся на основе небольшого количества типовых схем. Ниже доказаны общие утверждения, из которых можно легко выбрать утверждение, применимое к конкретной дуге.

Отметим, что доказанные в работе соотношения предназначены для того, чтобы иметь возможность делать выводы о стойкости высокоуровневых конструкций на основании свойств используемых криптопримитивов. Когда значения параметров приводят к неадекватным выводам (например, вычислительные ресурсы противника в итоговой оценке оказываются меньше нуля), следует говорить о том, что в этих условиях данное утверждение неприменимо.

Методы, которые используются ниже, являются стандартными для математической криптографии (см., например, [28, 29]).

Утверждение 5.1. *Если $F(K, M) : V_n \times V_m \rightarrow V_{r \cdot t}$ является (ε, T, q, u) -стойкой псевдослучайной функцией, то она является $(\varepsilon + 2^{-r \cdot t}, T-1, q-1, u)$ -стойкой относительно угрозы экзистенциальной подделки.*

Доказательство. Предположим, что существует $(T-1, q-1, u)$ -противник A , который реализует угрозу экзистенциальной подделки с вероятностью успеха, превосходящей $\varepsilon + 2^{-r \cdot t}$. Без ограничения общности будем считать, что противник A никогда не возвращает пару (M, D) с M , направлявшимся ранее в качестве запроса к оракулу. Действительно, в этом случае вероятность успеха возрастает, если возвращать случайную пару (M, D) с M не из множества запросов.

Построим противника A_F , работающего в модели с угрозой различения и принимающего решения на основе результата работы противника A . Противник A_F инициирует работу противника A и перенаправляет все его запросы на свой оракул $\mathcal{O}_F$, который с вероятностью $1/2$ реализует функцию F или случайную функцию. Ответы оракула возвращаются противнику A . В результате противник A строит пару (M, D) . Противник A_F посылает запрос M из указанной пары оракулу $\mathcal{O}_F$. Если ответ оракула $\mathcal{O}_F$ совпадает с D , то противник A_F возвращает 1, иначе — 0. Преобладание противника A_F в этом случае равно

$$\varepsilon_{A_F} = \underbrace{\Pr [A_F = 1 | \mathcal{O}_F = F]}_{\text{Вероятность успеха противника } A} - \Pr [A_F = 1 | \mathcal{O}_F = RF] > \varepsilon + \frac{1}{2^{r \cdot t}} - \frac{1}{2^{r \cdot t}} = \varepsilon.$$

Заметим, что событие $A_F = 1$ при условии $\mathcal{O}_F = RF$ эквивалентно тому, что результатом случайного и равновероятного выбора оракулом $\mathcal{O}_F$ элемента из V_r окажется строка D .

Вычислительные ресурсы противника A_F ограничены величиной T (ресурсы A и одно сравнение), и он осуществляет не более $q-1+1 = q$ запросов к оракулу O_F . Тот факт, что рассчитанное выше преобладание больше ε , противоречит условию. Следовательно, F является $(\varepsilon + 2^{-r \cdot t}, T-1, q-1, u)$ -стойкой относительно угрозы экзистенциальной подделки. $\square$

Утверждение 5.2. Если функция $F(K, M): V_n \times V_m \rightarrow V_{r \cdot t}$ является (ε, T, q, u) -стойкой относительно угрозы экзистенциальной подделки, то она является (ε, T, q, u) -стойкой относительно угрозы селективной подделки.

Доказательство. Предположим, что существует (T, q, u) -противник A , который реализует угрозу селективной подделки с вероятностью успеха, превосходящей ε .

Построим противника A_{EF} , который использует противника A для осуществления угрозы экзистенциальной подделки. Противник A_{EF} использует A в качестве черного ящика и возвращает то, что возвращает A , в качестве результата. В этом случае вероятность успеха противника A_{EF} совпадает с вероятностью успеха противника A . Это противоречит условию, следовательно, F является (ε, T, q, u) -стойкой относительно угрозы селективной подделки. $\square$

Утверждение 5.3. Если функция $F(K, M): V_n \times V_m \rightarrow V_{r \cdot t}$ является (ε, T, q, u) -стойкой относительно угрозы селективной подделки, то она является также $(\varepsilon \cdot (1 - 2^{-m})^{-q}, T - \tau_F, q, u)$ -стойкой относительно угрозы восстановления ключа, где τ_F — вычислительные ресурсы, необходимые для вычисления значения функции F .

Доказательство. Предположим, что существует $(T - \tau_F, q, u)$ -противник A , который восстанавливает ключ с вероятностью успеха, превосходящей значение $\varepsilon \cdot (1 - 2^{-m})^{-q}$.

Построим противника A_{SF} , осуществляющего селективную подделку. Сначала A_{SF} случайно и равновероятно выбирает строку $M \in V_m$. Затем A_{SF} использует противника A для восстановления ключа K . Получив результат K_A работы противника A , противник A_{SF} вычисляет значение $D = F(K_A, M)$ и возвращает пару (M, D) в качестве результата. Для вероятности успеха противника A_{SF} выполнены неравенства

$$\begin{aligned} \Pr [A_{SF} \text{ успешно реализует угрозу}] &\geq \\ &\geq \Pr [K_A = K] \cdot \Pr [\text{Противник } A \text{ не делал запрос } M \text{ к оракулу}] > \\ &> \frac{\varepsilon}{(1 - 2^{-m})^q} \cdot \left(1 - \frac{1}{2^m}\right)^q = \varepsilon. \end{aligned}$$

Это противоречит условию, следовательно, F является (ε, T, q, u) -стойкой относительно угрозы восстановления ключа. $\square$

Доказанные выше утверждения применимы к функциям HMAC, PRF, KDF и KDFTREE.

Пусть F вычисляется с использованием оракула, возвращающего значение HMAC_t на принятом им аргументе, в строгом соответствии с определением F . Через q_F будем обозначать количество обращений к функции HMAC_t , которые необходимо для этого осуществить. Так, например, если $F = \text{PRF}_t^{(1)}$ при $r=2$, то $q_F=4$: два обращения необходимо для вычисления A_1 и A_2 и два обращения нужно для вычисления D_1 и D_2 . Через τ_F обозначается объем остальных затрачиваемых на вычисление одного значения F вычислительных ресурсов (например, конкатенация аргументов).

Докажем, что псевдослучайность функции HMAC влечет за собой псевдослучайность функций PRF, KDF и KDFTREE. Следующее утверждение справедливо для любой из указанных выше функций, поэтому далее будем говорить о функции F , под которой в рассуждениях может подразумеваться любая из этих трех функций. Отметим, что в доказательстве следующего утверждения используется тот факт, что функция F может быть вычислена с помощью оракула, возвращающего значение функции HMAC.

Утверждение 5.4. Если HMAC_t является (ε, T, q, u) -стойкой псевдослучайной функцией, то функция F , принимающая значения в множестве $V_{r,t}$, является $(\varepsilon + \delta, T - q\tau_F, \tilde{q}, u - 1)$ -стойкой псевдослучайной функцией, где

$$\delta = \begin{cases} 0, & \text{если } F \in \{\text{PRF}_t^{(3)}, \text{KDFTREE}, \text{KDF}\}, \\ \tilde{q} \cdot \frac{(r-1)(r+2)}{2^{t+1}}, & \text{если } F = \text{PRF}_t^{(1)}, \\ \tilde{q} \cdot \frac{r(r-1)}{2^{t+1}}, & \text{если } F = \text{PRF}_t^{(2)}, \end{cases}$$

а $\tilde{q} = q/q_F$.

Доказательство. Предположим, что A — это $(T - q\tau_F, \tilde{q}, u - 1)$ -противник, отличающий F от случайной функции с преобладанием, превышающим $\varepsilon + \delta$. Построим (T, q, u) -противника A_{HMAC_t} , который отличает HMAC_t от случайной функции.

Противник A_{HMAC_t} использует A , перехватывая его запросы к оракулу, от которого противник A ожидает получить значения либо функции F , либо случайной функции. Каждый запрос M к такому оракулу противник A_{HMAC_t} обрабатывает следующим образом: делает q_F запросов к оракулу $\mathcal{O}_{\text{HMAC}_t}$ и работает τ_F шагов, самостоятельно вычисляя значение функции F на входе M .

Вычисленное таким образом значение противника A_{HMAC_t} возвращает противнику A в качестве ожидаемого им ответа от его оракула. В качестве результата A_{HMAC_t} выдает то, что выдает противник A .

Покажем, что преобладание построенного противника A_{HMAC_t} при реализации угрозы различения для функции HMAC_t превышает ε . Таким образом, необходимо оценить снизу величину

$$\Pr[A_{\text{HMAC}_t} = 1 | \mathcal{O}_{\text{HMAC}_t} = \text{HMAC}_t] - \Pr[A_{\text{HMAC}_t} = 1 | \mathcal{O}_{\text{HMAC}_t} = RF].$$

Заметим, что в том случае, когда оракул $\mathcal{O}_{\text{HMAC}_t}$ возвращает значения функции HMAC_t , вычисленные для случайно и равномерно выбранного ключа, противник A_{HMAC_t} честно реализует для противника A функцию F . Поэтому

$$\Pr[A_{\text{HMAC}_t} = 1 | \mathcal{O}_{\text{HMAC}_t} = \text{HMAC}_t] = \Pr[A = 1 | \mathcal{O}_F = F].$$

Остается оценить сверху величину $\Pr[A_{\text{HMAC}_t} = 1 | \mathcal{O}_{\text{HMAC}_t} = RF]$. Таким образом, далее рассуждения проводятся для случая, когда $\mathcal{O}_{\text{HMAC}_t}$ возвращает значения случайной функции.

Обозначим через $Z(M)$ множество тех запросов, направляемых оракулу $\mathcal{O}_{\text{HMAC}_t}$ противником A_{HMAC_t} , ответы на которые формируют непосредственно ответ на запрос M , сделанный противником A (например, при работе с функцией $\text{PRF}_t^{(1)}$ множество $Z(M) = \{A_1 || M, A_2 || M, \dots\}$). Для любой из указанных выше функций F при $M_1 \neq M_2$ множества $Z(M_1)$ и $Z(M_2)$ не пересекаются, поскольку в каждом таком запросе присутствует в качестве подстроки определяющая множество $Z(M)$ строка M . Таким образом, ответы оракула $\mathcal{O}_{\text{HMAC}_t}$ при обработке различных M_1 и M_2 независимы.

Рассмотрим случай, когда $F \in \{\text{PRF}_t^{(3)}, \text{KDFTREE}, \text{KDF}\}$. При обработке противником A_{HMAC_t} запроса M , сделанного противником A , ввиду наличия в аргументах HMAC_t номера i выходного блока D_i запросы к оракулу $\mathcal{O}_{\text{HMAC}_t}$ при $r > 1$ заведомо различны. Поэтому противник A находится в точности в условиях взаимодействия со своим оракулом, когда тот реализует случайную функцию. Из этого следует, что

$$\Pr[A_{\text{HMAC}_t} = 1 | \mathcal{O}_{\text{HMAC}_t} = RF] = \Pr[A = 1 | \mathcal{O}_F = RF],$$

т. е.

$$\begin{aligned} & \Pr[A_{\text{HMAC}_t} = 1 | \mathcal{O}_{\text{HMAC}_t} = \text{HMAC}_t] - \Pr[A_{\text{HMAC}_t} = 1 | \mathcal{O}_{\text{HMAC}_t} = RF] = \\ & = \Pr[A = 1 | \mathcal{O}_F = F] - \Pr[A = 1 | \mathcal{O}_F = RF] > \varepsilon. \end{aligned}$$

Рассмотрим случай, когда $F = \text{PRF}_t^{(1)}$. Если при обработке противником A_{HMAC_t} перехваченного запроса противника A не возникло совпадений среди A_i , то строка, которую A_{HMAC_t} возвращает A в качестве ответа на его запрос, также представляет собой случайную строку (т.е. она выбрана случайно и равновероятно из множества всех строк длины $r \cdot t$). Вероятность наличия повтора среди $A_1, A_2, \dots, A_r$ при обработке хотя бы одного запроса (далее это событие будем обозначать через Coll) оценивается сверху величиной $\tilde{q} \cdot (r-1)(r+2)/2^{t+1}$ (доказывается по индукции). Таким образом, вероятность возврата противником A_{HMAC_t} значения 1 при условии реализации оракулом $\mathcal{O}_{\text{HMAC}_t}$ случайной функции не более чем на $\tilde{q} \cdot (r-1)(r+2)/2^{t+1}$ превосходит вероятность возврата 1 противником A при реализации его оракулом случайной функции:

$$\begin{aligned}
 & \Pr[A_{\text{HMAC}_t} = 1 \mid \mathcal{O}_{\text{HMAC}_t} = RF] = \\
 &= \underbrace{\Pr[A_{\text{HMAC}_t} = 1 \mid \mathcal{O}_{\text{HMAC}_t} = RF \ \& \ \text{Coll}]}_{\leq 1} \cdot \underbrace{\Pr[\text{Coll}]}_{\leq \tilde{q} \cdot r(r+1)/2^{t+1}} + \\
 &+ \underbrace{\Pr[A_{\text{HMAC}_t} = 1 \mid \mathcal{O}_{\text{HMAC}_t} = RF \ \& \ \overline{\text{Coll}}]}_{= \Pr[A=1 \mid \mathcal{O}_F = RF]} \cdot \underbrace{\Pr[\overline{\text{Coll}}]}_{\leq 1} \leq \\
 &\leq \Pr[A=1 \mid \mathcal{O}_F = RF] + \tilde{q} \cdot \frac{(r-1)(r+2)}{2^{t+1}}.
 \end{aligned}$$

Следовательно,

$$\begin{aligned}
 & \Pr[A_{\text{HMAC}_t} = 1 \mid \mathcal{O}_{\text{HMAC}_t} = \text{HMAC}_t] - \Pr[A_{\text{HMAC}_t} = 1 \mid \mathcal{O}_{\text{HMAC}_t} = RF] \geq \\
 &\geq \Pr[A=1 \mid \mathcal{O}_F = F] - \Pr[A=1 \mid \mathcal{O}_F = RF] - \tilde{q} \cdot \frac{(r-1)(r+2)}{2^{t+1}} > \\
 &> \varepsilon.
 \end{aligned}$$

Рассмотрим случай, когда $F = \text{PRF}_t^{(2)}$. Далее при необходимости рассмотрения конкретного случая для краткости через Pr' будем обозначать вероятность в том случае, когда при работе используется оракул $\mathcal{O}_{\text{HMAC}_t}$, реализующий случайную функцию, а через Pr'' — вероятность в том случае, когда при работе используется оракул $\mathcal{O}_F$, также реализующий случайную функцию.

Для анализа данного случая будем использовать следующие соотношения (здесь $N = 2^t$, $\tilde{q}$ — количество запросов к оракулу, осуществляемых противником A , $\mathbf{M} = (M_1, \dots, M_{\tilde{q}}) \in V_m^{\tilde{q}}$, $\mathbf{D} = (D^1, \dots, D^{\tilde{q}}) \in V_{r,t}^{\tilde{q}}$, где D^v — строка, полученная противником A от оракула в ответ на запрос M_v):

$$\Pr[A=1] = \sum_{\mathbf{M}, \mathbf{D}} \Pr[A=1 | \mathbf{M}, \mathbf{D}] \cdot \Pr[\mathbf{M}, \mathbf{D}].$$

Здесь величина $\Pr[A=1 | \mathbf{M}, \mathbf{D}]$ зависит лишь от противника A . Величину $\Pr[\mathbf{M}, \mathbf{D}]$ можно представить следующим образом:

$$\begin{aligned} \Pr[\mathbf{M}, \mathbf{D}] &= \prod_{v=1}^{\tilde{q}} \Pr[M_v | M_1, \dots, M_{v-1}, D^1, \dots, D^{v-1}] \times \\ &\quad \times \Pr[D^v | M_1, \dots, M_v, D^1, \dots, D^{v-1}]. \end{aligned}$$

При этом величина $\Pr[M_v | M_1, \dots, M_{v-1}, D^1, \dots, D^{v-1}]$ зависит лишь от противника A , так как она равна вероятности того, что в качестве запроса с номером v при указанных условиях противник выберет строку M_v . Величина же $\Pr[D^v | M_1, \dots, M_v, D^1, \dots, D^{v-1}]$ зависит лишь от действий оракула.

Если среди блоков $D_1^v, D_2^v, \dots, D_r^v$ строки D^v нет одинаковых, то

$$\begin{aligned} \Pr'[D^v | M_1, \dots, M_v, D^1, \dots, D^{v-1}] &= \\ &= \Pr''[D^v | M_1, \dots, M_v, D^1, \dots, D^{v-1}] = \alpha, \end{aligned}$$

где

$$\alpha = \begin{cases} \frac{1}{N^r}, & \text{если } M_v \text{ не совпадает ни с одним } M_w, \quad w < v, \\ 0, & \text{если } M_v = M_w \text{ для некоторого } w < v \text{ и } D_v \neq D_w, \\ 1, & \text{если } M_v = M_w \text{ для некоторого } w < v \text{ и } D_v = D_w. \end{cases}$$

Отсюда следует, что если для любого $v \in \{1, \dots, \tilde{q}\}$ выполнено условие несовпадения блоков в D^v , то $\Pr'[\mathbf{M}, \mathbf{D}] = \Pr''[\mathbf{M}, \mathbf{D}]$.

Через $\mathcal{C}$ обозначим множество таких наборов $\mathbf{D} \in V_m^{\tilde{q}}$, в которых существует строка D^v , содержащая хотя бы два совпадающих блока D_i^v и D_j^v .

Через $\bar{\mathcal{C}}$ обозначим его дополнение. Тогда

$$\begin{aligned}
 & \Pr'[A = 1] = \\
 &= \sum_{\mathbf{M}, \mathbf{D} \in \mathcal{C}} \Pr[A = 1 | \mathbf{M}, \mathbf{D}] \Pr'[\mathbf{M}, \mathbf{D}] + \sum_{\mathbf{M}, \mathbf{D} \notin \mathcal{C}} \Pr[A = 1 | \mathbf{M}, \mathbf{D}] \Pr'[\mathbf{M}, \mathbf{D}] = \\
 &= \sum_{\mathbf{M}, \mathbf{D} \in \mathcal{C}} \Pr[A = 1 | \mathbf{M}, \mathbf{D}] \Pr'[\mathbf{M}, \mathbf{D}] + \\
 &+ \underbrace{\sum_{\mathbf{M}, \mathbf{D} \notin \mathcal{C}} \Pr[A = 1 | \mathbf{M}, \mathbf{D}] \Pr''[\mathbf{M}, \mathbf{D}] + \sum_{\mathbf{M}, \mathbf{D} \in \mathcal{C}} \Pr[A = 1 | \mathbf{M}, \mathbf{D}] \Pr''[\mathbf{M}, \mathbf{D}]}_{=\Pr''[A=1]} - \\
 &- \underbrace{\sum_{\mathbf{M}, \mathbf{D} \in \mathcal{C}} \Pr[A = 1 | \mathbf{M}, \mathbf{D}] \Pr''[\mathbf{M}, \mathbf{D}]}_{\geq 0} \leq \\
 &\leq \underbrace{\sum_{\mathbf{M}, \mathbf{D} \in \mathcal{C}} \Pr[A = 1 | \mathbf{M}, \mathbf{D}] \Pr'[\mathbf{M}, \mathbf{D}]}_{\leq \Pr'[\mathcal{C}]} + \Pr''[A = 1].
 \end{aligned}$$

Оценим сверху вероятность $\Pr'[\mathcal{C}]$. Справедливо соотношение

$$\Pr'[\mathcal{C}] = 1 - \Pr'[\bar{\mathcal{C}}] = 1 - \Pr''[\bar{\mathcal{C}}].$$

Будем оценивать снизу $\Pr''[\bar{\mathcal{C}}]$. Справедливо равенство

$$\Pr''[\bar{\mathcal{C}}] = \sum_{\mathbf{M}} \Pr''[\mathbf{M}] \sum_{\mathbf{D} \in \bar{\mathcal{C}}} \Pr''[\mathbf{D} | \mathbf{M}].$$

Через $\lambda(\mathbf{M})$ обозначим количество различных запросов среди $M_1, \dots, M_{\tilde{q}}$. Будем говорить, что $\mathbf{D}$ и $\mathbf{M}$ — *согласованные* наборы, если для тех v, w , для которых $M_v = M_w$, верно также равенство $D_v = D_w$. Заметим, что $\Pr''[\mathbf{D} | \mathbf{M}] = 0$ для $\mathbf{M}$ и $\mathbf{D}$, не являющихся согласованными. Далее для некоторого $\mathbf{M}$ через $\bar{\mathcal{C}}_{\mathbf{M}}$ будем обозначать множество тех $\mathbf{D} \in \bar{\mathcal{C}}$, которые являются согласованными с $\mathbf{M}$. Кроме того, для согласованных $\mathbf{M}$ и $\mathbf{D} \in \bar{\mathcal{C}}$ выполнено $\Pr''[\mathbf{D} | \mathbf{M}] = N^{-r \lambda(\mathbf{M})}$, где каждый множитель N^{-r} соответствует первому появлению каждого из $\lambda(\mathbf{M})$ различных запросов. Следовательно,

$$\Pr''[\bar{\mathcal{C}}] = \sum_{\mathbf{M}} \Pr''[\mathbf{M}] \sum_{\mathbf{D} \in \bar{\mathcal{C}}} \Pr''[\mathbf{D} | \mathbf{M}] = \sum_{\mathbf{M}} \Pr''[\mathbf{M}] \left(\frac{1}{N^r} \right)^{\lambda(\mathbf{M})} |\bar{\mathcal{C}}_{\mathbf{M}}|.$$

Оценим $|\bar{\mathcal{C}}_{\mathbf{M}}|$. Для каждого из $\lambda(\mathbf{M})$ различных запросов M_v ответ D^v выбирается случайно, равновероятно и независимо. Каждый ответ D^v представляет собой произвольную бесповторную строку из r блоков. Количество таких строк равно $N(N-1) \dots (N-r+1)$, следовательно, $|\bar{\mathcal{C}}_{\mathbf{M}}| = (N(N-1) \dots (N-r+1))^{\lambda(\mathbf{M})}$.

В итоге получаем, что

$$\begin{aligned} \Pr''[\bar{\mathcal{C}}] &= \sum_{\mathbf{M}} \Pr''[\mathbf{M}] \cdot \left(\frac{N(N-1) \dots (N-r+1)}{N^r} \right)^{\lambda(\mathbf{M})} \geq \\ &\geq \min_{\mathbf{M}} \left(\frac{N(N-1) \dots (N-r+1)}{N^r} \right)^{\lambda(\mathbf{M})} \underbrace{\sum_{\mathbf{M}} \Pr''[\mathbf{M}]}_{=1} = \\ &= \left(\frac{N(N-1) \dots (N-r+1)}{N^r} \right)^{\tilde{q}} \geq \\ &\geq \left(1 - \frac{r(r-1)}{2N} \right)^{\tilde{q}} \geq \\ &\geq 1 - \tilde{q} \cdot \frac{r(r-1)}{2N}. \end{aligned}$$

Переходя к изначальным обозначениям, находим

$$\Pr'[\mathcal{C}] = 1 - \Pr''[\bar{\mathcal{C}}] \leq \tilde{q} \cdot \frac{r(r-1)}{2^{t+1}}.$$

Окончательно

$$\begin{aligned} &\Pr[A_{\text{HMAC}_t} = 1 | \mathcal{O}_{\text{HMAC}_t} = \text{HMAC}_t] - \Pr[A_{\text{HMAC}_t} = 1 | \mathcal{O}_{\text{HMAC}_t} = RF] = \\ &= \Pr[A = 1 | \mathcal{O}_F = F] - \Pr'[A = 1] \geq \\ &\geq \Pr[A = 1 | \mathcal{O}_F = F] - \Pr''[A = 1] - \tilde{q} \cdot \frac{r(r-1)}{2^{t+1}} > \varepsilon. \end{aligned}$$

Таким образом, преобладание противника A_{HMAC_t} больше ε . Заметим, что длина запроса к оракулу (в блоках) увеличивается не более чем на единицу.

Противник A_{HMAC_t} совершает T операций, делая не более q запросов к оракулу, т. е. является (T, q, u) -противником, а это противоречит тому, что по условию HMAC_t является (ε, T, q, u) -стойкой псевдослучайной функцией. Полученное противоречие доказывает утверждение. $\square$

Использование номера i при вычислении значения D_i обеспечивает уникальность строк, подаваемых на вход HMAC, при формировании разных блоков. Неповторяемость входных данных HMAC при обработке разных запросов достигается за счет зависимости от M каждого блока D_i . Указанные

свойства позволяют избежать появления зависимости в блоках выходной строки D . Данная зависимость например, для $\text{PRF}_t^{(1)}$, состоит в том, что появление одинаковых строк $A_i = A_j$ приводит к многократному повторению блоков в выходной строке D . Эта особенность находит свое отражение и в выкладках, необходимых для обоснования стойкости различных функций. Так, доказательство стойкости функции $\text{PRF}_t^{(3)}$ намного проще и меньше по объему, чем рассуждения, касающиеся функции $\text{PRF}_t^{(2)}$. Заметим, что данный подход близок к методам, используемым для оптимизации метода Хеллмана [31].

Тот факт, что HMAC является стойкой псевдослучайной функцией (т.е. задача различения для HMAC является труднорешаемой), доказывается ниже (на основе результатов работ [2, 3]) при условии выполнения **свойств 1, 2 и 3**, которые сформулированы в разделе 6. Все требуемые свойства функций HMAC, PRF, KDF и KDFTREE, в соответствии с приведенным ориентированным графом, сводятся к доказываемому ниже свойству псевдослучайности HMAC.

5.1.2. Псевдослучайность функции HMAC

В разделе 4 было показано, что стойкость всех (кроме VKO) рассматриваемых в данной работе алгоритмов к атакам, интересным с точки зрения практики, основана на свойстве псевдослучайности функции HMAC. В данном разделе приводятся три свойства базовых для HMAC криптографических примитивов и обосновывается достаточность этих свойств для псевдослучайности HMAC.

Определение 5.1. Пусть функция F осуществляет отображение из $V_n \times V_{[0, 2^{512}-1]}$ в V_t . Пусть оракул $\mathcal{O}_{wcr}$ действует следующим образом: в начале работы он случайно и равномерно выбирает ключ $K \in V_n$, а на любой запрос $M \in V_{[0, 2^{512}-1]}$ возвращает строку $F(K, M) \in V_t$. Функция F называется (ε, T, q, u) -стойкой к коллизиям в слабом смысле, если любой (T, q, u) -противник, производящий запросы M длины не более u блоков к оракулу $\mathcal{O}_{wcr}$, возвращает в качестве результата (находит коллизию) такую пару $M \neq M'$, что $F(K, M) = F(K, M')$ с вероятностью, не превосходящей ε .

Через GH' обозначим ограничение функции GH на входы, состоящие из двух блоков. Через $\text{GH}'(K, \cdot)$ ($\text{GH}(K, \cdot)$) для $K \in V_{512}$ обозначим функцию GH' (GH соответственно) с фиксацией первого входного блока строкой K :

$$\text{GH}'(K, M) = \text{GH}'(K \| M) = \text{GH}(K \| M) \quad (\text{GH}(K, M) = \text{GH}(K \| M)).$$

Определение 5.2. Пусть оракул $\mathcal{O}_{rka}$ действует следующим образом. В начале работы он случайно и равновероятно выбирает бит $l \in \{0, 1\}$ и, если $l = 0$, то случайно и равновероятно выбирает ключ $K \in V_n$, иначе случайно, равновероятно и независимо выбирает ключи $K_1, K_2 \in V_n$. Далее, если $l = 0$, то оракул на запрос $(M, 1)$, где $M \in V_{[0, 2^{512}-1]}$, выдает $\text{GH}(r'(K), M)$, а на запрос $(M, 2)$, где $M \in V_{512}$, выдает $\text{GH}'(r''(K), M)$. Если же $l = 1$, то $\mathcal{O}_{rka}$ на запрос $(M, 1)$ отвечает $\text{GH}(K_1, M)$, а на запрос $(M, 2)$ отвечает $\text{GH}'(K_2, M)$.

Пару функций (GH, GH') будем называть (ε, T, q, u) -стойкой относительно атаки с (r', r'') -связанными ключами, где $r', r'' : V_n \rightarrow V_n$, $r'(K) = K \oplus \text{ipad}$, а $r''(K) = K \oplus \text{opad}$, если для любого (T, q, u) -противника, использующего оракул $\mathcal{O}_{rka}$, преобладание при реализации угрозы различения случаев $l = 0$ и $l = 1$ меньше или равно ε .

Через $\text{NMAC}(K_1, K_2, \cdot)$ будем обозначать функцию, определенную по аналогии с HMAC , но при независимых ключах K_1, K_2 :

$$\text{NMAC}_t(K_1, K_2, M) = \text{GH}_t(K_2 \oplus \text{opad} \parallel \text{GH}_t(K_1 \oplus \text{ipad} \parallel M)),$$

при этом K_1 называется ключом внутреннего хэширования или внутренним ключом, а K_2 — ключом внешнего хэширования или внешним ключом. Так, $\text{NMAC}(K, \cdot) = \text{NMAC}(K \oplus \text{ipad}, K \oplus \text{opad}, \cdot)$.

Теорема 5.1. Если функция GH является (ε_1, T, q, u) -стойкой к коллизиям в слабом смысле, функция GH' является $(\varepsilon_2, T + uqT_g, q, 1)$ -стойкой псевдослучайной функцией, а (GH, GH') является $(\varepsilon_3, T, 2q, u)$ -стойкой относительно атаки с (r', r'') -связанными ключами, то функция NMAC является $(\varepsilon_1 + \varepsilon_2 + \varepsilon_3, T, q, u)$ -стойкой псевдослучайной функцией.

Опишем предварительно общую структуру доказательства. Предположим, что есть противник A_{HMAC} , который может эффективно (с преобладанием, превышающим заявленное в условии теоремы) отличать HMAC от случайной функции. Если этот противник существенно использует тот факт, что в HMAC внутренний и внешний ключи являются зависимыми, то можно с помощью A_{HMAC} построить такого противника A_{rka} , который будет отличать пару (GH, GH') на (r', r'') -связанных ключах от этой же пары функций на независимых ключах. Однако из условия теоремы следует, что не существует противника, который эффективно решает эту задачу. Следовательно, противник A_{HMAC} не использует (поскольку не может) при своей работе тот факт, что ключи в HMAC зависимы.

Поскольку при независимых ключах НМАС превращается в NMAC, противник $A_{\text{НМАС}}$ будет примерно с тем же преобладанием отличать NMAC от случайной функции, т. е. с его помощью можно построить эффективного противника $A_{\text{НМАС}}$ для решения задачи различения NMAC и случайной функции. Используем теперь $A_{\text{НМАС}}$, чтобы построить противника $A_{\text{GH}'}$, который эффективно отличает функцию GH' от случайной функции. Для того чтобы построить такого противника, будем вычислять внутренний хэш из функции NMAC на случайно выбранном ключе и передавать результат оракулу, который, если он не реализует случайную функцию, выдаст в итоге окончательное значение функции NMAC. Если же оракул реализует случайную функцию, то с точки зрения $A_{\text{НМАС}}$ он будет отличаться от истинно случайной функции только тем, что некоторые входы будут приводить к совпадающим значениям на входе оракула $\mathcal{O}_{\text{GH}'}$, поскольку они будут образовывать коллизию для внутренней хэш-функции GH . По условию теоремы вероятность такого события мала, поэтому преобладание противника $A_{\text{GH}'}$ будет велико (так как он основан на эффективном противнике $A_{\text{НМАС}}$), что противоречит несуществованию противника, который может эффективно отличать функцию GH' от случайной функции. Неэффективность $A_{\text{GH}'}$ влечет за собой неэффективность $A_{\text{НМАС}}$, которая, в свою очередь, влечет за собой неэффективность $A_{\text{НМАС}}$.

Отметим, что представленное ниже доказательство является адаптированной (на случай использования хэш-функции ГОСТ Р 34.11-2012) версией доказательства аналогичной теоремы из работы [3].

Доказательство. Предположим противное. Пусть $A_{\text{НМАС}}$ — некоторый (T, q, u) -противник, отличающий функцию НМАС от случайной функции с преобладанием $\varepsilon > \varepsilon_1 + \varepsilon_2 + \varepsilon_3$.

Построим (T, q, u) -противника $A_{\text{НМАС}}$, который отличает функцию NMAC от случайной функции. Оракул $\mathcal{O}_{\text{НМАС}}$ противника $A_{\text{НМАС}}$ сначала выбирает случайный бит и, в зависимости от его значения, на любой запрос выдает либо значение случайной функции, либо значение функции NMAC при случайно и независимо выбранных перед началом взаимодействия с противником ключах K_1, K_2 . Преобладание $A_{\text{НМАС}}$ обозначим через γ . Противник $A_{\text{НМАС}}$ использует $A_{\text{НМАС}}$ в качестве черного ящика, перехватывая его запросы к оракулу. На любой запрос M противника $A_{\text{НМАС}}$ противник $A_{\text{НМАС}}$ возвращает ему строку, которую вернул ему оракул $\mathcal{O}_{\text{НМАС}}$ в ответ на запрос M . В результате противник $A_{\text{НМАС}}$ возвращает то же, что возвращает используемый им противник $A_{\text{НМАС}}$ ¹⁾.

¹⁾ Если некий метод отличения НМАС от случайной функции не основывается на свойствах связанных ключей (а, например, только на свойствах функции НМАС), то этот же метод можно применить для того, чтобы отличать NMAC от случайной функции.

Построим теперь $(T, 2q, u)$ -противника A_{rka} для решения задачи различения случаев связанных и независимых ключей для пары (GH, GH') . Этот противник использует A_{HMAC} в качестве черного ящика и на любой его запрос M отвечает строкой $\mathcal{O}_{rka}(\mathcal{O}_{rka}(M, 1), 2)$. Если противник A_{HMAC} в результате решает, что оракул, от которого он получает ответы на свои запросы, является случайной функцией, то A_{rka} отвечает, что $\mathcal{O}_{rka}$ использует независимые ключи, иначе — связанные. Пусть δ — преобладание A_{rka} .

Справедливы следующие соотношения:

$$\gamma = \Pr[A_{HMAC} = 1 | \mathcal{O}_{HMAC} = NMAC] - \Pr[A_{HMAC} = 1 | \mathcal{O}_{HMAC} = RF],$$

$$\begin{aligned} \delta = \Pr \left[A_{HMAC} = 1 \mid \underbrace{\mathcal{O}_{rka} \text{ использует связанные ключи}}_{\text{в этом случае } A_{rka} \text{ моделирует работу } NMAC} \right] - \\ - \Pr \left[A_{HMAC} = 1 \mid \underbrace{\mathcal{O}_{rka} \text{ использует независимые ключи}}_{\text{в этом случае } A_{rka} \text{ моделирует работу } NMAC} \right]. \end{aligned}$$

Поскольку величина

$$\gamma + \delta = \Pr[A_{HMAC} = 1 | \mathcal{O}_{HMAC} = NMAC] - \Pr[A_{HMAC} = 1 | \mathcal{O}_{HMAC} = RF]$$

является преобладанием противника A_{HMAC} , имеет место неравенство $\gamma + \delta > \varepsilon_1 + \varepsilon_2 + \varepsilon_3$. По условию преобладание δ противника A_{rka} не превосходит ε_3 , поэтому $\gamma > \varepsilon_1 + \varepsilon_2 + \varepsilon_3 - \delta \geq \varepsilon_1 + \varepsilon_2$.

Теперь с помощью A_{HMAC} построим $(T + uqT_g, q, 1)$ -противника $A_{GH'}$ (T_g — трудоемкость вычисления функции сжатия g отображения GH), который может отличать функцию $GH'(K, \cdot)$ от случайной. В начале своей работы $A_{GH'}$ случайно и равновероятно выбирает ключ K_1 . Противник $A_{GH'}$ использует A_{HMAC} в качестве черного ящика, передавая ему в качестве ответа на любой его запрос M строку $D = \mathcal{O}_{GH'}(GH(K_1, M))$. Если $\mathcal{O}_{GH'} = GH'(K, \cdot)$, то возвращаемая противником $A_{GH'}$ в ответ на запрос M строка D равна $NMAC(K_1, K, M)$. В случае когда $\mathcal{O}_{GH'} = RF$, распределение возвращаемых $A_{GH'}$ строк D отличается от равномерного тем, что разные входы M_1 и M_2 могут образовать коллизию для внутренней функции $GH(K_1, \cdot)$, делая равными (следовательно, нарушая условие независимости их выбора) возвращаемые оракулом $\mathcal{O}_{GH'}$ строки D_1 и D_2 . Эта вероятность не превосходит вероятности успеха произвольного противника A_{wcr} , который пытается построить коллизию для функции $GH(K, \cdot)$ с неизвестным ключом. Напомним, что по условию теоремы эта вероятность меньше или равна ε_1 .

Рассмотрим преобладание противника $A_{GH'}$:

$$\begin{aligned}
 & \underbrace{\Pr [A_{GH'} = 1 | \mathcal{O}_{GH'} = GH']}_{=\Pr[A_{NMAC}=1 | \mathcal{O}_{NMAC}=NMAC]} - \Pr [A_{GH'} = 1 | \mathcal{O}_{GH'} = RF] \geq \\
 & \geq \Pr [A_{NMAC} = 1 | \mathcal{O}_{NMAC} = NMAC] - \\
 & - (\Pr [A_{NMAC} = 1 | \mathcal{O}_{NMAC} = RF] + \Pr [A_{wcr} = 1]) \geq \\
 & \geq \gamma - \varepsilon_1 > \varepsilon_2.
 \end{aligned}$$

Это противоречит тому, что по условию теоремы преобладание любого $(T+uqT_{GH}, q)$ -противника $A_{GH'}$, отличающего GH' от случайной функции, не превосходит ε_2 . Следовательно, НМАС является $(\varepsilon_1 + \varepsilon_2 + \varepsilon_3, T, q, u)$ -стойкой псевдослучайной функцией. $\square$

Заметим, что использование в формулировке теоремы пары функций (GH, GH') объясняется тем, что именно они участвуют в вычислении целевой для обоснования функции НМАС. Использование чуть более естественной пары (GH, GH) привело бы к неоправданному усилению противника.

5.2. Обоснование стойкости VKO

Напомним определение *распознавательной задачи Диффи–Хеллмана* (DDH) [18]: на вход подаются точки xP , yP и Q , причем xP , yP выбираются случайно и равновероятно из группы E , точка Q с вероятностью $1/2$ равна xyP , а с вероятностью $1/2$ выбирается случайно и равновероятно из группы E ; требуется выдать 1, если $Q = xyP$, и 0, если $Q \neq xyP$.

Приведем общую схему обоснования стойкости алгоритма VKO в моделях, описанных в подразделе 4.2.

Рассмотрим *Модель 1*. Предположим, что существует противник A , который для некоторых произвольных фиксированных отображений $R_1(x, y)$, $R_2(x, y)$ может по $R_1(x, y)$, xP , yP , UKM и $VKO(x, y, UKM)$ эффективно вычислять $R_2(x, y)$ с вероятностью p , не являющейся пренебрежимо малой. Поскольку хэш-функция GH является эффективно вычислимой, на основе этого алгоритма можно построить эффективный алгоритм, который с не меньшей вероятностью успеха вычисляет $R_2(x, y)$ по $R_1(x, y)$, xP , yP и xyP (при сведении UKM выбирается самим противником случайно и равновероятно). Поэтому стойкость алгоритма VKO в Модели 1 противника не меньше стойкости задачи дискретного логарифмирования с дополнительной информацией. Под задачей дискретного логарифмирования с дополнительной информацией здесь понимается задача нахождения x по xP , yP , xyP и $R_1(x, y)$, эквивалентная задаче нахождения x по xP и $R_1(x, y)$.

Рассмотрим *Модель 2*. Пусть есть противник A , который может по открытым ключам xP и yP эффективно угадывать $GH(xyP)$ с вероятностью p , не являющейся пренебрежимо малой. Пусть при этом вероятность коллизии у хэш-функции GH на элементах E является пренебрежимо малой. В таком случае противник A может решать задачу DDH следующим образом: противник отличает xyP от случайной точки по значению хэш-функции, которую для xyP он может угадывать эффективно и с вероятностью, не являющейся пренебрежимо малой. Это противоречит предположению о труднорешаемости распознавательной задачи Диффи – Хеллмана [18]. Отметим также, что рекомендуемые к использованию для работы с ГОСТ Р 34.10-2012 кривые ([13, 20]) имеют простой порядок ([18, 19]).

Предположим теперь, что вероятность коллизии у хэш-функции GH на элементах E не является пренебрежимо малой. Это приводит к тому, что существует эффективно конструируемое множество (множество с определенным для него описанием элементов в некотором алфавите, всякий элемент которого возможно восстановить по номеру за ограниченное полиномом от размера время; в практических случаях — возможно восстановить за короткое время), а именно, группа E , на которой вероятность коллизии хэш-функции не является пренебрежимо малой. Это противоречит базовому **свойству 3'**, сформулированному в разделе 6.

Перейдем к строгому формальному обоснованию криптографических свойств алгоритма VKO.

Заметим, что Модель 1 на самом деле является семейством моделей, которое параметризовано парой функций R_1, R_2 . Области значений функций R_1 и R_2 не фиксируются, поскольку могут быть заданы в той или иной конкретике и не существенны для дальнейших рассуждений. Поясним смысл значений $R_1(x, y)$ и $R_2(x, y)$ на примерах.

Дополнительно в Модели 1 не будем фиксировать равномерным распределение, в соответствии с которым выбирается параметр UKM . Таким образом, эта модель дополнительно параметризуется посредством распределения параметра UKM , которое далее будем обозначать через $\mathcal{D}$.

Пример 5.1. Для противника, имеющего доступ лишь к каналу связи, $R_1(x, y) = 0$, а $R_2(x, y) = x$.

Пример 5.2. Пусть противник получает некоторую информацию о значении x по побочному каналу. В этом случае $R_1(x, y)$ — информация об x , полученная по побочному каналу.

Утверждение 5.5. Если для функций R_1 , R_2 и распределения $\mathcal{D}$ задача получения $R_2(x, y)$ по известным xP , yP , xyP и $R_1(x, y)$ является (ε, T) -стойкой, то функция $VKO(x, y, UKM)$ является $(\varepsilon, T - T_{GH} - T_m)$ -стойкой в Модели 1, где T_{GH} — количество операций, необходимых для вычисления хэш-функции GH , а T_m — количество операций, необходимых для умножения элемента группы E точек эллиптической кривой на число.

Доказательство. Предположим, что существует такой противник A_1 , совершающий $T - T_{GH} - T_m$ операций, что

$$\Pr [A_1(UKM, K, xP, yP, R_1(x, y)) = R_2(x, y)] > \varepsilon,$$

где x и y выбираются случайно и равномерно, UKM выбирается в соответствии с распределением $\mathcal{D}$, а $K = VKO(x, y, UKM)$. Построим противника A_2 , который решает задачу получения $R_2(x, y)$ по известным $R_1(x, y)$, xP , yP и xyP с трудоемкостью T и вероятностью успеха, превышающей ε . Противник A_2 вначале выбирает UKM в соответствии с распределением $\mathcal{D}$, вычисляет

$$K = GH\left(\frac{\mu}{q} \cdot UKM \cdot xyP\right)$$

(затрачивая на это $T_{GH} + T_m$ операций) и $U = A_1(UKM, K, xP, yP, R_1(x, y))$ (за $T - T_{GH} - T_m$ операций). В качестве результата работы A_2 возвращает U . В таком случае

$$\begin{aligned} \Pr [A_2(xP, yP, xyP, R_1(x, y)) = R_2(x, y)] &= \\ &= \Pr [A_1(UKM, K, xP, yP, R_1(x, y)) = R_2(x, y)] > \varepsilon, \end{aligned}$$

причем A_2 совершает T операций.

Получено противоречие, которое доказывает утверждение. $\square$

Согласно утверждению 5.5 решение задачи обращения VKO не легче задачи дискретного логарифмирования с дополнительной информацией.

Замечание 5.1. Существуют методы (см., например, [25]) дискретного логарифмирования в простых подгруппах точек эллиптических кривых, основанные на вложении этих групп в подгруппы мультипликативных групп конечных полей. Для того чтобы трудоемкость вычисления указанных вложений была велика (что влечет за собой высокую трудоемкость алгоритмов дискретного логарифмирования), необходимо, чтобы порядок элемента p ($\text{ord}(p)$) в группе $\mathbb{Z}_q^*$ был большим (здесь p — характеристика поля, q — порядок простой подгруппы точек эллиптической кривой). В документе [26] приводится следующая рекомендация: $\text{ord}(p)$ должен быть больше $(q-1)/100$.

Заметим, что с помощью указанных вложений можно решать распознавательную задачу DDH с трудоемкостью, линейной относительно трудоемкости вычисления вложения.

Проверка указанного выше условия на $\text{ord}(p)$ осуществляется по следующей схеме:

- Разложить $q-1$ на множители: $q-1 = q_1^{s_1} \cdots q_r^{s_r}$, где q_i — различные простые числа.
- Проверить, что существует такое число $d = q_1^{s'_1} \cdots q_r^{s'_r} > (q-1)/100$, что $p^d = 1 \pmod q$, и для любого такого i , что $s'_i \neq 0$, выполнено сравнение $p^{d/q_i} \neq 1 \pmod q$.

В разделе 8 приведены результаты вычислений $\text{ord}(p)$ и показано, что для кривых, определенных в [13, 20], выполнено условие $\text{ord}(p) \geq (q-1)/6$.

Рассмотрим Модель 2, определенную в подразделе 4.2.

Заметим, что применение ρ -метода Полларда (см. [21]), позволяющее решить задачу дискретного логарифмирования за $O(2^{t/2})$ операций, дает возможность осуществить указанную в Модели 2 угрозу при известных xP , yP и UKM .

Утверждение 5.6. Если задача DDH является (ε, T) -стойкой, то функция $VKO(x, y, UKM)$ является $(\varepsilon + s, T - T_{\text{GH}} - T_m)$ -стойкой в Модели 2, где T_{GH} — количество операций, необходимых для вычисления хэш-функции GH , T_m — количество операций, необходимых для умножения элемента группы E точек эллиптической кривой на число, а

$$s = \max_{h \in V_t, Q \in_R E} \Pr [\text{GH}(Q) = h].$$

Доказательство. Предположим, что существует такой противник A_3 , совершающий $T - T_{\text{GH}} - T_m$ операций, что

$$\Pr [A_3(xP, yP, UKM) = VKO(x, y, UKM)] > \varepsilon + s.$$

Построим противника A_4 , решающего задачу DDH. Получив на вход xP , yP и Q , противник A_4 случайно и равномерно выбирает UKM , затем вычисляет значения $K = A_3(xP, yP, UKM)$ (за $T - T_{\text{GH}} - T_m$ операций) и $K' = \text{GH}(\frac{p}{q} \cdot UKM \cdot Q)$ (за $T_{\text{GH}} + T_m$ операций) и возвращает 1 в случае, когда $K = K'$, и 0 в противном случае.

Оценим величину преобладания

$$\gamma = \Pr [A_4(xP, yP, xyP) = 1] - \Pr [A_4(xP, yP, Q) = 1]$$

для противника A_4 (x и y выбираются случайно и равномерно из $\mathbb{Z}_q$, точка Q выбирается случайно и равномерно из E).

Справедлива цепочка неравенств

$$\begin{aligned}
 \Pr [A_4(xP, yP, Q) = 1] &= \Pr [A_3(xP, yP, UKM) = GH(UKM \cdot Q)] = \\
 &= \sum_{h \in V_t} \Pr [GH(UKM \cdot Q) = h] \Pr [A_3(xP, yP, UKM) = h] \leq \\
 &\leq \max_{h \in V_t, Q \in_R E} \Pr [GH(Q) = h] \sum_{h \in V_t} \Pr [A_3(xP, yP, UKM) = h] \leq \\
 &\leq \max_{h \in V_t, Q \in_R E} \Pr [GH(Q) = h] = s.
 \end{aligned}$$

Так как $\Pr[A_4(xP, yP, xyP) = 1] = \Pr[A_3(xP, yP, UKM) = VKO(x, y, UKM)] > \varepsilon + s$, то $\gamma > \varepsilon$. При этом трудоемкость работы противника A_4 составляет T операций. $\square$

Исследуем зависимость трудоемкости построения коллизии для хэш-функции GH от величины вероятности s .

Утверждение 5.7. Если $s = \max_{h \in GH(E), P \in_R E} \Pr [GH(P) = h]$, то для любого $c \in \mathbb{N}$ существует выполняющий не более T операций, где $T = \lceil c/s \rceil + 1$, и использующий память, размер которой не превосходит T , алгоритм, который находит коллизию для хэш-функции GH с вероятностью, не меньшей $\varepsilon = 1 - (c+1)e^{-c}$.

Доказательство. Укажем алгоритм, параметры которого соответствуют тем, что указаны в условии.

Случайным образом сформируем множество $U \subseteq E$, $|U| = T$. Пусть для $P \in_R E$ строка $h^* \in GH(E)$ такова, что

$$\Pr[GH(P) = h^*] = \max_{h \in GH(E), P \in_R E} \Pr[GH(P) = h],$$

Справедливы соотношения

$$\begin{aligned}
 \Pr[\text{в } U \text{ есть две такие точки } P \text{ и } Q, \text{ что } GH(P) = GH(Q)] &\geq \\
 &\geq \Pr[\text{в } U \text{ есть две такие точки } P \text{ и } Q, \text{ что } GH(P) = GH(Q) = h^*] = \\
 &= \Pr[\xi \geq 2] = 1 - \Pr[\xi = 0] - \Pr[\xi = 1] = 1 - (1-s)^T - T(1-s)^{T-1}s = \\
 &= 1 - (1-s)^{T-1}(1-s+sT) = 1 - (1-s)^{T-1}(1+s(T-1)) = \\
 &= 1 - \left(1 - \frac{c}{T-1}\right)^{T-1}(1+c) > 1 - (c+1)e^{-c}, \tag{5.1}
 \end{aligned}$$

где $\xi \sim Bin(T, s)$. Оцененная снизу в (5.1) вероятность является вероятностью успеха алгоритма поиска коллизии. $\square$

Заметим, что если строка h^* известна, то коллизию можно найти, используя алгоритм, аналогичный описанному в доказательстве утверждения 5.7, но с использованием памяти: в памяти необходимо хранить строку h^* и две точки, образующие коллизию.

Рассмотрим для примера следующий случай. Пусть $s = r \cdot 2^{-n/2}$ и $c = 8$. Тогда $T = 8 \cdot 2^{n/2}/r$, а $\varepsilon \approx 0,99698$.

Из сложности задачи построения коллизий для хэш-функции GH (см. **свойство 3'** в разделе 6) следует, что вероятность s является пренебрежимо малой, что позволяет говорить (утверждение 5.6) о стойкости функции VKO в Модели 2.

Замечание 5.2. Условие стойкости задачи DDH для стойкости алгоритма VKO не представляется необходимым. В силу того что алгебраические структуры, в которых работают алгоритмы, входящие в состав VKO, не связаны друг с другом, представляется правдоподобным, что стойкость алгоритма VKO эквивалентна стойкости вычислительной задачи Диффи – Хеллмана.

6. Фундаментальные свойства

Обоснование стойкости рассматриваемых схем в описанных выше моделях противника основывается на следующих свойствах алгоритмов, используемых в качестве базовых.

Свойство 1. Псевдослучайность хэш-функции ГОСТ Р 34.11-2012. Функция GH' является $(\varepsilon_2, T, q, 1)$ -стойкой псевдослучайной функцией.

Свойство 2. Стойкость хэш-функции ГОСТ Р 34.11-2012 к атакам со связанными ключами. Пара функций (GH, GH') является (ε_3, T, q, n) -стойкой относительно атаки с (r', r'') -связанными ключами (определение 5.2).

Свойство 3: Стойкость хэш-функции ГОСТ Р 34.11-2012 к коллизиям в слабом смысле. Функция GH является (ε_1, T, q, n) -стойкой к коллизиям (определение 5.1).

Свойство 3'. Ни один из существующих методов криптографического анализа хэш-функций не позволяет строить коллизии для функции ГОСТ Р 34.11-2012 эффективнее универсальных методов.

Отметим, что свойство 3' предъясняется как представляющийся единственно возможным невырожденный вариант свойства 3, спроецированный на свойства бесключевой функции GH , используемой в алгоритме VKO.

7. Заключение

Для каждого из представленных в документе [4] алгоритмов вопрос стойкости рассмотрен в наиболее сильных моделях противника, при обоснованиях эксплуатировался стандартный набор предположений о стойкости задач Диффи–Хеллмана в группах точек эллиптических кривых и базовых криптографических свойствах введенного стандартом ГОСТ Р 34.11-2012 [16] алгоритма хэширования. Для всех рассмотренных сопутствующих алгоритмов сделан вывод об их стойкости в соответствующих моделях противника.

Приведем пример, иллюстрирующий возможность использования VKO в качестве базового алгоритмического блока при конструировании протоколов (аутентифицированной) выработки общего ключа в конкретных моделях взаимодействия многих абонентов и моделях противника.

Заметим, что с использованием алгоритма VKO можно реализовать не только базовое расширение протокола Диффи–Хеллмана, но и протокол *HMQRV* (см. [24]), который, в свою очередь, является модификацией протокола *MQV* (см. [22, 23]) и позволяет производить аутентифицированную выработку ключа при помощи долговременных и эфемерных ключей. Классический вариант протокола *HMQRV* реализуется с использованием реализации

$$\overline{\text{VKO}}(x, Y, UKM) = \text{GH} \left(\left(\frac{\mu}{q} \cdot UKM \cdot x \bmod q \right) \cdot Y \right)$$

функции VKO в качестве черного ящика следующим образом (при использовании *HMQRV* на основе протокола Диффи–Хеллмана без кофактора; пример для простоты изложения приводится в предположении использования кривой с группой точек простого порядка; иначе следует в качестве *UKM* положить $\left(\frac{\mu}{q}\right)^{-1} \bmod q$):

- Абонент *A* (идентификатор ID_A) передает абоненту *B* сертификат своего открытого ключа $A = aP$ и открытый эфемерный ключ $X = xP$;
- Абонент *B* (идентификатор ID_B) передает абоненту *A* сертификат своего открытого ключа $B = bP$ и открытый эфемерный ключ $Y = yP$;
- Абонент *A* вычисляет секретный ключ *K* по формуле

$$\begin{aligned} K &= \overline{\text{VKO}}(ad + x, Y + eB, 1) = \\ &= H((ad + x) \cdot (Y + eB)) = \\ &= H((ady + xy + xbe + abed) \cdot P); \end{aligned}$$

- Абонент B вычисляет секретный ключ K по формуле

$$K = \overline{\text{VKO}}(be + y, X + dA, 1) = H((be + y) \cdot (Y + dA)) = \\ = H((ady + xy + xbe + abed) \cdot P).$$

Параметры e и d вычисляются так: $e = H(yP || ID_A)$, $d = H(xP || ID_B)$.

Необходимо добавить, что VKO является алгоритмом, свойства которого являются абстрагированными от модели взаимодействия пользователей протоколов, строящихся на его основе. Другими словами, при описании и обосновании алгоритма VKO не осуществляется анализ свойств, используемых в протоколах ключей (в частности, вопроса о порядке использования ключей в качестве эфемерных или долговременных). В то же время схема $HMQR$ является базовым протокольным решением, исследование криптографических свойств которого выходит за рамки данной работы и необходимо при анализе криптографических протоколов, использующих в своей структуре $HMQR$ (MQV).

Авторы статьи выражают благодарность А. М. Зубкову, И. В. Лаврикову, В. И. Рудскому и участникам семинара по математической криптографии, проходящего на факультете ВМК МГУ, за ценные замечания и конструктивную критику.

Список литературы

- [1] Анохин М. И., Варновский Н. П., Сидельников В. М., Яценко В. В., *Криптография в банковском деле*, М.: МИФИ, 1997.
- [2] Koblitz N., Menezes A., "Another look at HMAC", *J. Math. Cryptol.*, 7:3 (2013), 225–251.
- [3] Koblitz N., Menezes A., "Another look at security theorems for 1-key nested MACs", <http://eprint.iacr.org/2013/248.pdf>.
- [4] *Рекомендации по стандартизации. Использование криптографических алгоритмов, сопутствующих применению стандартов ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012*, Утверждены решением технического комитета по стандартизации ТК 26, М.: РОССТАНДАРТ, 2014.
- [5] ISO/IEC 9797-2:2011 *Information technology — Security techniques — Message Authentication Codes (MACs). Part 2: Mechanisms Using Dedicated Hash-Function*.
- [6] Krawczyk H., Bellare M., Canetti R., "HMAC: Keyed-hashing for message authentication", *RFC*, № 2104 (1997).
- [7] *The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standards Publication FIPS PUB 198*, Gaithersburg, MD: National Institute of Standards and Technology, March 2002.
- [8] ANSI X9.71, *Keyed Hash Message Authentication Code*, Washington, DC–N. Y.: American National Standards Institution, 2000.
- [9] Dierks T., Allen C., "The TLS Protocol Version 1.0", *RFC*, № 2246 (January 1999).
- [10] Dierks T., Rescorla E., "The Transport Layer Security (TLS) Protocol Version 1.2", *RFC*, № 5246 (August 2008).

- [11] Harkins D., Carrel D., “The Internet Key Exchange (IKE)”, *RFC*, № 2409 (November 1998).
- [12] Kaufman C., “Internet Key Exchange (IKEv2) Protocol”, *RFC*, № 4306 (December 2005).
- [13] Popov V., Kurepkin I., Leontiev S., “Additional cryptographic algorithms for use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 algorithms”, *RFC*, № 4357 (January 2006).
- [14] *Recommendation for Key Derivation Using Pseudorandom Functions*, NIST SP 800-108, Gaithersburg, MD: National Institute of Standards and Technology, October 2009.
- [15] *Алгоритм криптографического преобразования ГОСТ 28147-89*, Государственный Комитет СССР по стандартам, М: ИПК Изд-во стандартов, 1989.
- [16] *Национальный стандарт Российской Федерации ГОСТ Р 34.11-2012*, Федеральное агентство по техническому регулированию и метрологии, М.: Стандартиформ, 2013.
- [17] *Национальный стандарт Российской Федерации ГОСТ Р 34.10-2012*, Федеральное Агентство по техническому регулированию и метрологии, М.: Стандартиформ, 2012.
- [18] Boneh D., “The decision Diffie–Hellman problem”, *Lect. Notes Comput. Sci.*, **1423** (1998), 48–63.
- [19] Gennaro R., Krawczyk H., Rabin T., “Secure hashed Diffie–Hellman over non-DDH groups”, <http://eprint.iacr.org/2004/>.
- [20] *Техническая спецификация по заданию параметров эллиптических кривых в соответствии с ГОСТ Р 34.10-2012*, М.: Федеральное агентство по техническому регулированию и метрологии (РОССТАНДАРТ). Технический комитет № 26, 2013.
- [21] Василенко О. Н., *Теоретико-числовые алгоритмы в криптографии*, М.: МЦНМО, 2003, 328 с.
- [22] Menezes A., Qu M., Vanstone S., “Some new key agreement protocols providing mutual implicit authentication”. In: “*Second Workshop on Selected Areas in Cryptography (SAC 95) (Ottawa, May 18–19, 1995)*”, Ottawa, ON: Carleton Univ., 1995, 22–32.
- [23] Law L., Menezes A., Qu M., Solinas J., Vanstone S., “An efficient protocol for authenticated key agreement”, *Des., Codes and Cryptogr.*, **28**:2 (2003), 119–134.
- [24] Krawczyk H., “HMQV: A high-performance secure Diffie–Hellman protocol”, <http://eprint.iacr.org/2005/176.pdf>.
- [25] Menezes A., Vanstone S., “Reducing elliptic curve logarithms to logarithms in a finite field”, *IEEE Trans. Inform. Theory*, **IT-39**:5 (1993), 1639–1646.
- [26] Lochter M., Merkle J., “Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation”, *RFC*, № 5639 (March 2010).
- [27] *Положение об Открытом конкурсе научных работ по исследованию хэш-функции ГОСТ Р 34.11-2012*, <http://www.streebog.info>.
- [28] Goldreich O., *Foundations of Cryptography — Basic Tools*, Cambridge: Cambridge Univ. Press, 2001.
- [29] Goldreich O., *Foundations of Cryptography — Basic Applications*, Cambridge: Cambridge Univ. Press, 2004.
- [30] Варновский Н. П., “Стойкость схем электронной подписи в модели с защищенным модулем”, *Дискретная математика*, **20**:3 (2008), 147–159.
- [31] Hellman M. E., “A cryptanalytic time-memory trade-off”, *IEEE Trans. Inform. Theory*, **IT-26**:4 (1980), 401–406.

8. Дополнение 1. Порядки элементов p в некоторых мультипликативных группах $\mathbb{Z}_q^*$

Везде в данном разделе под p понимается характеристика некоторого простого поля, а q — порядок группы точек эллиптической кривой, заданной над полем $\text{GF}(p)$ (везде q — простое число).

В данном разделе приводятся результаты вычислений $\text{ord}(p)$ в мультипликативных группах $\mathbb{Z}_q^*$ для параметров эллиптических кривых

id-GostR3410-2001-CryptoPro-A-ParamSet,

id-GostR3410-2001-CryptoPro-B-ParamSet,

id-GostR3410-2001-CryptoPro-C-ParamSet,

описанных в RFC 4357 [13], а также

id-tc26-gost-3410-12-512-paramSetA,

id-tc26-gost-3410-12-512-paramSetB,

приведенных в [20]. Вычисления проводились в соответствии с процедурой, описанной в замечании 5.1.

Эллиптическая кривая id-GostR3410-2001-CryptoPro-A-ParamSet.

Параметры:

- $p = 2^{256} - 617 =$
 $= 115792089237316195423570985008687907853269984665640564039457584007913129639319$
- $q =$
 $= 115792089237316195423570985008687907853073762908499243225378155805079068850323$
- $q - 1 = 2 \cdot 3 \cdot 7 \cdot 17 \cdot 37 \cdot 127 \cdot 121493 \cdot 5592900119 \cdot$
 $\cdot 50791017540450015071456350284045037169936855765408748581$
- $\text{ord}(p)$ в группе $\mathbb{Z}_q^*$: $\text{ord}(p) = \frac{q-1}{2}$

Эллиптическая кривая id-GostR3410-2001-CryptoPro-B-ParamSet.

Параметры:

- $p = 2^{255} + 3225 =$
 $= 57896044618658097711785492504343953926634992332820282019728792003956564823193$
- $q =$
 $= 57896044618658097711785492504343953927102133160255826820068844496087732066703$
- $q - 1 = 2 \cdot 47336631894758162101 \cdot$
 $\cdot 611535319489737880361765400867656717933201264526640746251$
- $\text{ord}(p)$ в группе $\mathbb{Z}_q^*$: $\text{ord}(p) = \frac{q-1}{2}$

Эллиптическая кривая id-GostR3410-2001-CryptoPro-C-ParamSet.

Параметры:

- $p = 70390085352083305199547718019018437841079516630045180471284346843705633502619$
- $q = 70390085352083305199547718019018437840920882647164081035322601458352298396601$
- $q - 1 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot 47 \cdot 207130852417 \cdot 15398703602419036183 \cdot$
 $\cdot 260862815097120313262827914162129492639311$
- $\text{ord}(p)$ в группе $\mathbb{Z}_q^*$: $\text{ord}(p) = q - 1$

Эллиптическая кривая id-tc26-gost-3410-12-512-paramSetA.

Параметры:

- $p = 2^{512} - 569 =$
 $= 134078079299425970995740249982058461274793658205923933777235614437217640300735$
 $46976801874298166903427690031858186486050853753882811946569946433649006083527$
- $q = 134078079299425970995740249982058461274793658205923933777235614437217640300734$
 $449232318290585817636498049628612556596899500625279906416653993875474742293109$
- $q - 1 = 2 \cdot 2 \cdot 19 \cdot 41 \cdot 257 \cdot 619681 \cdot 56230387 \cdot 3067250436697090551527 \cdot$
 $\cdot 1566530004958535144270949664750568664934770494453380896639546221087773012$
 $98738109970223050607345391305841500928811$
- $\text{ord}(p)$ в группе $\mathbb{Z}_q^*$: $\text{ord}(p) = q - 1$

Эллиптическая кривая id-tc26-gost-3410-12-512-paramSetB.

Параметры:

- $p = 2^{511} + 111 =$
 $= 67039039649712985497870124991029230637396829102961966888617807218608820150367$
 $73488400937149083451713845015929093243025426876941405973284973216824503042159$
- $q = 67039039649712985497870124991029230637396829102961966888617807218608820150369$
 $2258541985374819038361506291094774340556751014839882071710028285687776119229$
- $q - 1 = 2 \cdot 2 \cdot 3 \cdot 2389 \cdot$
 $\cdot 2338462384879063258611348018383885539186438855272846619527619897398103116$
 $728380977194032980392906242173472494678179701238366261622965362174848917$
 87921
- $\text{ord}(p)$ в группе $\mathbb{Z}_q^*$: $\text{ord}(p) = \frac{q-1}{6}$

9. Дополнение 2. Наименования исследуемых алгоритмов в соответствии с [4]

Таблица. Краткие обозначения исследуемых алгоритмов

Наименование алгоритма	Используемое обозначение
HMAC_GOSTR3411_2012_256	HMAC ₂₅₆
HMAC_GOSTR3411_2012_512	HMAC ₅₁₂
PRF_TLS_GOSTR3411_2012_256	PRF ₂₅₆ ⁽¹⁾
PRF_TLS_GOSTR3411_2012_512	PRF ₅₁₂ ⁽¹⁾
PRF_IPSEC_KEYMAT_GOSTR3411_2012_256	PRF ₂₅₆ ⁽²⁾
PRF_IPSEC_KEYMAT_GOSTR3411_2012_512	PRF ₅₁₂ ⁽²⁾
PRF_IPSEC_PRFLPLUS_GOSTR3411_2012_256	PRF ₂₅₆ ⁽³⁾
PRF_IPSEC_PRFLPLUS_GOSTR3411_2012_512	PRF ₅₁₂ ⁽³⁾
KDF_GOSTR3411_2012_256	KDF
KDF_TREE_GOSTR3411_2012_256	KDFTREE
VKO_GOSTR3410_2012_256	VKO ₂₅₆
VKO_GOSTR3410_2012_512	VKO ₅₁₂