О стойкости ключевых криптоалгоритмов на основе хэш-функции «Стрибог» к атакам со связанными ключами

В. А. Кирюхин

***Аннотация*—Бесключевая хэш-функция «Стрибог» служит основой для ряда ключевых алгоритмов – псевдослучайных функций (PRF), часто применяемых для защиты целостности (имитозащиты). Примером такой PRF служит Стрибог-К (CTCrypt 2022), его доказательство стойкости основано на сведении к специальным свойствам используемой в хэш-функции функции сжатия, а именно – функция сжатия должна быть стойкой, когда ключом является любой из двух её входов, а противник осуществляет атаки со связанными ключами (PRF-RKA).**

В настоящей работе доказано, что ни в одном из двух случаев требование к функции сжатия не может быть ослаблено с PRF-RKA до PRF. Кроме этого, если оба эти требования выполнены, то сам Стрибог-К является не только стойкой PRF, но и стойким к атакам со связанными ключами (PRF-RKA).

Аналогичные результаты представлены для стандартизированного криптоалгоритма HMAC-Стрибог.

***Ключевые слова*—Стрибог, HMAC-Стрибог, доказуемая стойкость, PRF, связанные ключи, PRF-RKA**

I. Введение

Российская стандартизированная бесключевая хэш-функция «Стрибог» (ГОСТ 34.11-2018) [1] основана на модифицированной схеме Меркла-Дамгарда (МД) [2, 3], согласно которой хэшируемое сообщение M дополняется специальным образом и разбивается на l блоков по n бит, затем к состоянию хэш-функции h и блоку сообщения m итеративно применяется функция сжатия $\mathsf{g}(h, m) = h'$. Начальное состояние хэш-функции является предопределённой константой, последнее состояние – результат хэширования (хэш-значение).

Хэш-функция «Стрибог» обладает двумя особенностями, отличающими её от «простой» схемы МД:

1) после обработки i-го блока к состоянию прибавляется по модулю 2 («$\oplus$») специальное значение Δ_i, а после обработки последнего l-го – $\tilde{\Delta}_l$, при этом $\Delta_l \neq \tilde{\Delta}_l$ [4, 5];

2) последний вызов функции сжатия выполняет «подмешивание» к состоянию контрольной суммы всех блоков сообщения (по модулю 2^n – «$\boxplus$»).

Указанные особенности играют особенно важную роль, когда «Стрибог» используется в качестве основы для *ключевого* криптоалгоритма (с секретным ключом) – псевдослучайной функции (PRF). Такие алгоритмы часто используются для защиты целостности (имитозащиты – message authentication code – MAC).

Статья получена 13 ноября 2023

Виталий Александрович Кирюхин, ООО «СФБ Лаб», АО «ИнфоТеКС» (email: vitaly.kiryukhin@sfblaboratory.ru).

Примерами служат, в частности, HMAC-Стрибог [6] (двойное хэширование) и вычислительно более эффективный Стрибог-К [4] (см. описание в разделе III). Оба алгоритма не вносят изменений в саму хэш-функцию.

Существующие доказательства стойкости [7—11] конструкции HMAC предполагают, что в качестве хэш-функции используется «простая» схема МД. Следовательно, для обоснования стойкости криптоалгоритма HMAC-Стрибог эти результаты неприменимы. Первичный анализ алгоритма был, среди прочего, представлен в [12]. В [4] за счёт сведе́ния к свойствам функции сжатия было доказано, что HMAC-Стрибог и Стрибог-К – стойкие псевдослучайные функции (PRF). Позднее было представлено несколько более сложное доказательство [13], позволяющее получить *точные* оценки преобладания (вероятности успеха) противника.

Результаты анализа [4, 13] продемонстрировали, что особенности хэш-функции «Стрибог» приводят к появлению «внутри» криптоалгоритма Стрибог-К (а равно HMAC-Стрибог) так называемых *связанных ключей* (related keys), даже когда ключи для самого криптоалгоритма («внешние») выбираются случайно и равновероятно. Под связанностью ключей K и K' понимается ситуация, когда противник не знает ни один из них, но, к примеру, знает или даже выбирает их сумму $\phi = K \oplus K'$.

Отсюда требование (точнее два) к функции сжатия g – стойкая PRF в условиях атак со связанными ключами ($PRF\text{-}RKA$ – см. раздел IV). Если секретным ключом для g является состояние h (обозначаем $\mathsf{g}_h^{\triangleright}$), то связь между ключами определяется операцией «$\oplus$» ($PRF\text{-}RKA_{\oplus}$), а когда роль секрета играет блок сообщения ($\mathsf{g}_m^{\triangledown}$) – связь задана операцией «$\boxplus$» ($PRF\text{-}RKA_{\boxplus}$).

В настоящей работе доказано, что ни одно из двух требований к функции сжатия не может быть ослаблено с $PRF\text{-}RKA$ до PRF (раздел V и VI). Доказательство заключается в построении специальных функций $\mathsf{w}^{\triangledown}$ и $\mathsf{w}^{\triangleright}$, которые сами являются стойкими PRF, но использование $\mathsf{w}^{\triangledown}$ в алгоритме Стрибог-К вместо $\mathsf{g}^{\triangledown}$ (или $\mathsf{w}^{\triangleright}$ вместо $\mathsf{g}^{\triangleright}$) приводит к эффективной атаке на алгоритм.

Таким образом, можно (несколько неформально) говорить, что требования, ранее предъявленные к g [4, 13], являются не только достаточными, но и *необходимыми*.

С практической точки зрения более интересно, что при неизменности этих требований, Стрибог-К и HMAC-Стрибог *сами стойки к атакам со связанными ключами* ($PRF\text{-}RKA$). Это даёт больше гибкости при их реализации в ограниченных условиях. Формирование совокупности из r связанных ключей может быть суще-

ственно проще, чем генерирование случайных и равновероятных. Подобное упрощение реализации приводит, однако, к r-кратному увеличению вероятности успеха противника, что делает такой подход *нежелательным*, но, повторимся, *допустимым* для двух анализируемых криптоалгоритмов.

Доказательство стойкости алгоритма Стрибог-К (PRF-$RKA_{\oplus}$ и PRF-$RKA_{\boxplus}$), обобщающее результаты работы [13], представлено в разделе VII. Далее показано, что стойкость (PRF-$RKA_{\oplus}$) алгоритма HMAC-Стрибог можно легко свести к стойкости алгоритма Стрибог-К, находящегося в таких же условиях (раздел VIII).

Совокупность результатов, полученных в настоящей работе, представлена на рисунке 1.

II. Обозначения и основные сведения

Обозначаем: n, k, τ – битовый размер состояния/блока, ключа и выхода соответственно ($n = 512$, $k \le n$, $\tau \le n$);

V^n – множество всех n-битных строк;

0^u – строка из u нулевых бит;

$||$ – конкатенация двоичных строк;

$\mathsf{msb}_u : V^n \to V^u$ – взятие u старших бит из n-битного блока;

$\oplus$ и $\boxplus$ – сложение по модулю 2 и 2^n соответственно;

$\mathrm{Func}(\mathbf{X}, \mathbf{Y})$ – множество всех функций, отображающих конечное множество $\mathbf{X}$ в конечное множество $\mathbf{Y}$, $\mathrm{Func}(\mathbf{X}) = \mathrm{Func}(\mathbf{X}, \mathbf{X})$;

$X \xleftarrow{\mathrm{R}} \mathbf{X}$ – случайный и равновероятный выбор элемента X из множества $\mathbf{X}$;

$\mathsf{F} : \mathbf{X} \to \mathbf{Y}$ – детерминированный алгоритм, отображение из множества входов $\mathbf{X}$ в множество выходов $\mathbf{Y}$.

Криптоалгоритмы обозначаются шрифтом без засечек: Alg, E. Параметризацию Alg преобразованием E обозначаем Alg[E], опуская параметр, когда он определён контекстом.

Под противником будем понимать интерактивный вероятностный алгоритм $\mathcal{A}$, взаимодействующий с другими алгоритмами (оракулами) [14]. В рамках модели угроз TM для криптоалгоритма Alg количественную характеристику возможностей (преобладание) противника $\mathcal{A}$ обозначаем $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A})$.

Максимум значения $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(\mathcal{A})$ среди противников, возможности которых ограничены вычислительно (числом операций t в некоторой модели вычислений) и информационно (числом запросов/ответов к оракулам q и другими определяемыми моделью TM и алгоритмом Alg параметрами $prms$) обозначаем $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(t, q, prms)$. Размер описания противника $\mathcal{A}$ (его исходный код) ограничен некоторым малым значением, что позволяет не рассматривать класс атак, связанных с «бесплатными» предвычислениями.

Результатом вычислений $\mathcal{A}$ после взаимодействия с оракулами $\mathcal{O}_1$, $\mathcal{O}_2$, ... $\mathcal{O}_w$, $w \in \mathbb{N}$, является значение $b \in \{0, 1\}$, обозначаем это $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \ldots \mathcal{O}_w} \Rightarrow b$. Без потери общности полагаем, что противник $\mathcal{A}$ всегда использует максимально возможное число запросов, и среди них нет совпадающих (нет «бессмысленных действий»).

Криптоалгоритм Alg неформально называем стойким в модели угроз TM (TM-стойким), если $\mathrm{Adv}^{TM}_{\mathsf{Alg}}(t, q, prms) \le \varepsilon$, где ε не превосходит некоторого малого значения, определяемого требованиями к стойкости криптосистемы, а ресурсы t, q и $prms$ сопоставимы с доступными противнику на практике.

Определение 1. Преобладанием противника $\mathcal{A}$ в модели PRF (PRF-CMA – неотличимость от случайной функции при атаке с выбранными сообщениями) для ключевой функции $\mathsf{F} : \mathbf{K} \times \mathbf{X} \to \mathbf{Y}$ назовём $\mathrm{Adv}^{PRF}_{\mathsf{F}}(\mathcal{A}) =$

$$= \Pr(K \xleftarrow{\mathrm{R}} \mathbf{K} : \mathcal{A}^{\mathsf{F}_K} \Rightarrow 1) - \Pr(\mathsf{R} \xleftarrow{\mathrm{R}} \mathrm{Func}(\mathbf{X}, \mathbf{Y}) : \mathcal{A}^{\mathsf{R}} \Rightarrow 1).$$

У всех рассматриваемых в работе алгоритмов множество $\mathbf{X}$ является конечным.

III. Хэш-функция «Стрибог»

Хэш-функция «Стрибог» [1] построена с использованием модифицированной схемы Меркла-Дамгарда.

Процесс хэширования сообщения выполняется следующим образом. Двоичное сообщение M произвольной длины (не более 2^n бит) дополняется битовой строкой $10\ldots0$ до кратности n, $M' = M||10\ldots0$. Дополнение выполняется даже если исходно длина M уже кратна n.

Дополненное сообщение M' разбивается на $(l+1)$ блоков по $n = 512$ бит $M' = m_0||m_1||m_2||\ldots||m_l$. Выполняется последовательная обработка блоков с помощью функции сжатия $\mathsf{g} : V^n \times V^n \times V^n \to V^n$,

$$h_{i+1} = \mathsf{g}(h_i, m_i, \mathbf{i}),\ i = 0, ..., l,\ \mathbf{i} = i \cdot n \in V^n.$$

Начальное состояние хэш-функции $h_0 = IV_\tau \in V^n$ зависит от длины выхода $\tau \in \{256, 512\}$.

После обработки блоков из M' выполняется финализация – «подмешиваются» L (битовая длина сообщения M) и контрольная сумма $\Sigma = \mathsf{sum}_{\boxplus}(M) = m_0 \boxplus \ldots \boxplus m_l$,

$$H = \mathsf{g}(\mathsf{g}(h_{l+1}, L, \mathbf{0}), \Sigma, \mathbf{0}).$$

Для 512-битной хэш-функции результатом хэширования является $\mathsf{H}_{512}(M) = H$, а для 256-битной – выполняется усечение H до старших 256 бит, $\mathsf{H}_{256}(M) = \mathsf{msb}_{256}(H)$.

Функция сжатия реализована с помощью 12-раундового блочного шифра XSPL-типа $\mathsf{E} : V^n \times V^n \to V^n$ с использованием конструкциии Миагучи-Пренеля

$$\mathsf{g}(h_i, m_{i+1}, \mathbf{i}) = \mathsf{E}(h_i \oplus \mathbf{i}, m_{i+1}) \oplus h_i \oplus m_{i+1} = h_{i+1}.$$

Эквивалентное представление [5] позволяет рассматривать функцию сжатия, зависящую только от двух входных параметров $\mathsf{g}(h, m) = y$, везде далее под функцией сжатия будет пониматься именно это преобразование. Хэш-функция примет вид (см. детали в [4]) $\mathsf{H}_{512}(M) =$

$$= \mathsf{g}(\mathsf{g}(\ldots(\mathsf{g}(\mathsf{g}(IV_{512}, m_0) \oplus \Delta_0, m_1) \oplus \Delta_1) \ldots \oplus \widetilde{\Delta}_l, L), \Sigma),$$

где $\Delta_i = \mathbf{i} \oplus (\mathbf{i} \boxplus \mathbf{1})$, $\widetilde{\Delta}_i = \mathbf{i}$, $\Delta_i \ne \widetilde{\Delta}_i$, $i \in \{0, 1, 2...\}$, аналогично для H_{256}.

На основе хэш-функции «Стрибог» в [6] определены два ключевых преобразования HMAC-Стрибог-256 и HMAC-Стрибог-512, там же задано ограничение на длину ключа $256 \le k \le 512$, HMAC-Стрибог-$\tau(K, M) =$

$$= \mathsf{H}_\tau(\overline{K} \oplus opad || \mathsf{H}_\tau(\overline{K} \oplus ipad || M)),$$

ключ $\overline{K} = K\,||\,0^{n-k}$, а $opad$ и $ipad$ – некоторые различные ненулевые константы.

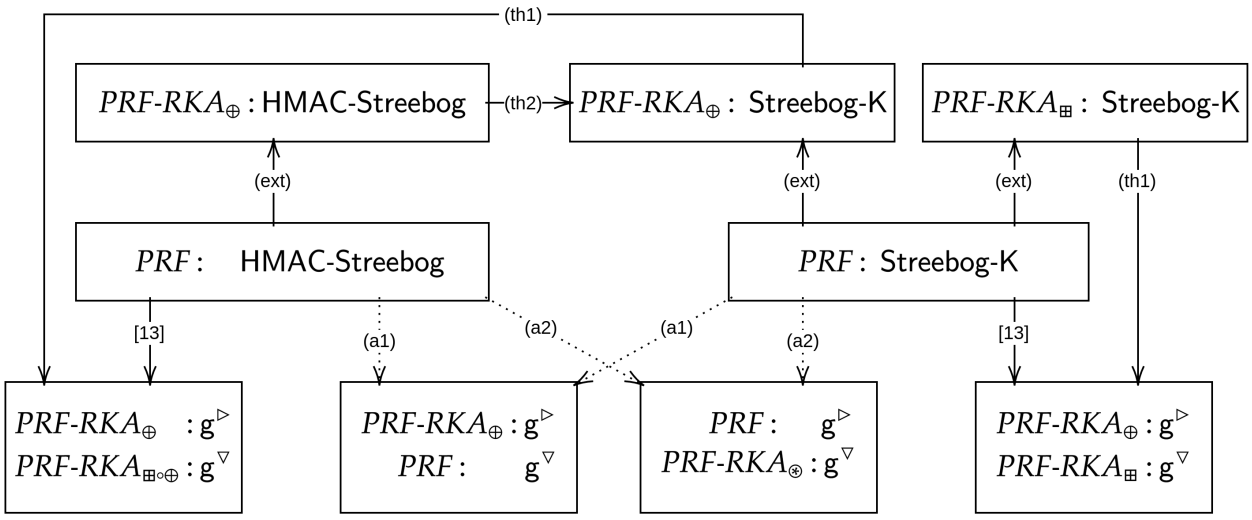

Рис. 1. Зависимости между моделями PRF и $PRF\text{-}RKA$ для криптоалгоритмов Стрибог-К (Streebog-K), HMAC-Стрибог (HMAC-Streebog) и функции сжатия g. Сплошными стрелками изображены *сведения*: (ext) – за счёт расширения модели; (th1), (th2) – согласно теореме 1 и 2; [13] – представлены в соответствующей работе. Пунктирными стрелками (a1) и (a2) изображены *невозможные* сведения (разделы V и VI).

Как показано в [4] особенности хэш-функции «Стрибог» позволяют построить ключевое преобразование более простым и эффективным способом

$$\text{Стрибог-К}(K, M) = \mathsf{H}_\tau(\overline{K} \,||\, M),\ \tau \in \{256, 512\}.$$

Для упрощения обозначений положим $\overline{K} = m_0$ и $M = m_1||...||m_l$, и определим каскадное преобразование следующим образом $\mathsf{Csc}(K_{\mathsf{Csc}}, M) =$

$$= \mathsf{g}(\ldots \mathsf{g}(\mathsf{g}(K_{\mathsf{Csc}} \oplus \Delta_0, m_1) \oplus \Delta_1, m_2) \ldots \oplus \widetilde{\Delta}_l, L),$$

$K_{\mathsf{Csc}} \in V^n$, полагая, что входные данные M с произвольной битовой длиной были дополнены строкой 10...0, а длина L увеличена на n из-за приписывания ключа. Ключевое преобразование примет вид

$$\text{Стрибог-К}(K, M) = \mathsf{g}(\mathsf{Csc}(\mathsf{g}(IV, \overline{K}), M), \overline{K} \boxplus \sigma),$$
$$\sigma = \mathsf{sum}_{\boxplus}(M) = m_1 \boxplus m_2 \boxplus \ldots \boxplus m_l.$$

Для краткости записи иногда сокращаем наименование Стрибог-К до KH, а HMAC-Стрибог до HMAC.

Если первый аргумент функции сжатия (блок состояния h) рассматривается в качестве секрета (ключа), обозначаем это $\mathsf{g}^{\triangleright}_h = \mathsf{g}(h, m)$, если секретом является второй аргумент (блок сообщения m), то – $\mathsf{g}^{\triangledown}_m = \mathsf{g}(h, m)$.

IV. Связанные ключи

Как упоминалось ранее, если хэш-функция «Стрибог» используется в качестве основы ключевого криптоалгоритма, то её особенности приводят к появлению связанных ключей. Дадим формальное определение этому понятию.

Определение 2. Преобладанием противника $\mathcal{A}$ в модели $PRF\text{-}RKA_{\circledast}$ для ключевого криптоалгоритма $\mathsf{F} : \mathbf{K} \times \mathbf{X} \to \mathbf{Y}$ назовём $\mathrm{Adv}^{PRF\text{-}RKA_{\circledast}}_{\mathsf{F}}(\mathcal{A}) =$

$$= \Pr\left(K \xleftarrow{\mathrm{R}} \overline{\mathbf{K}}; \mathcal{A}^{\mathsf{F}_{K\circledast}(\cdot)} \Rightarrow 1\right) -$$
$$- \Pr\left(K \xleftarrow{\mathrm{R}} \overline{\mathbf{K}}; \mathsf{R}_i \xleftarrow{\mathrm{R}} \mathrm{Func}(\mathbf{X}, \mathbf{Y}),\ \forall i \in \mathbf{K}; \mathcal{A}^{\mathsf{R}_{K\circledast}(\cdot)} \Rightarrow 1\right),$$

где $\mathbf{K}$, $\mathbf{X}$, $\mathbf{Y}$ – множества ключей, входов и выходов соответственно, $\overline{\mathbf{K}} \subseteq \mathbf{K}$. Символом «$\circledast$» обозначена бинарная операция над $\mathbf{K}$ – параметр модели. Запрос от $\mathcal{A}$ состоит из входа $x \in \mathbf{X}$ и «связи» $\kappa \in \mathbf{K}$. Ответ оракула – $y = \mathsf{F}_{K\circledast\kappa}(x)$ (соотв. $y = \mathsf{R}_{K\circledast\kappa}(x)$). Информационными ресурсами $\mathcal{A}$ являются: q запросов к оракулу; r «связей»; максимальное число «связей» (d), при которых преобразуется одно и то же x ($d \le r \le q$).

Отметим, что при $\mathbf{K} = \overline{\mathbf{K}}$ и использовании в качестве «$\circledast$» унарного тождественного преобразования модель $PRF\text{-}RKA_{\circledast}$ эквивалентна модели PRF.

Множество $\overline{\mathbf{K}}$ также является параметром модели, но введено исключительно из-за технических соображений, везде далее $\overline{\mathbf{K}} = \{\overline{K} = K||0^{n-k},\ K \in V^k\}$, $\mathbf{K} = V^n$, параметр k (длина ключа) полагаем при этом зафиксированным.

Для «$\circledast \in \{\oplus, \boxplus\}$» в предположении об отсутствии специфических уязвимостей, преобладание противника в задаче различения для произвольного криптоалгоритма F эвристически оценивается неравенствами [13]:

$$\mathrm{Adv}^{PRF\text{-}RKA_{\circledast}}_{\mathsf{F}}(t, q, r, d) \lessapprox \frac{t \cdot d}{2^k} \le \frac{t \cdot r}{2^k} \le \frac{t \cdot q}{2^k}, \qquad (1)$$

Однако, связь по ключам, определяемая композицией «$\oplus$» и «$\boxplus$», делает в общем случае (при $d > 1$) *любой* криптоалгоритм F уязвимым к атаке.

Утверждение 1. *В условиях атаки со связанными ключами, секретный ключ произвольного алгоритма* $\mathsf{F} : V^k \times \mathbf{X} \to \mathbf{Y}$ *может быть определён за* $q = 2k$ *запросов, если связь задаётся композицией операций «*$\oplus$*» и «*$\boxplus$*», а один и тот же вход* $x \in \mathbf{X}$ *может быть запрошен хотя бы дважды (*$d \ge 2$*).*

Доказательство. Выполняется серия из k пар запросов,

$$y_i = \mathsf{F}(K \oplus 0^k \boxplus I_i,\ x_i),$$
$$y'_i = \mathsf{F}(K \oplus I_i \boxplus 0^k, x_i),\ I_i = 0^{k-i}||1||0^{i-1},\ 1 \le i \le k.$$

Если $y_i = y'_i$, то i-й бит ключа равен нулю, если коллизии нет – равен единице. Ложноположительными срабатываниями критерия ($y_i = y'_i$) пренебрегаем. □

Отметим, что описанный алгоритм восстановления ключа, разумеется, влечёт существование различителя в модели PRF.

При анализе алгоритма HMAC-Стрибог [4, 13] описанная выше атака не влияла на эвристическую оценку (1) в силу того, что один и тот же $x = IV$ запрашивался только для двух заведомо различных ключей ($\overline{K} \oplus ipad$ и $\overline{K} \oplus opad$). В настоящей же работе рассматриваются сценарии, в которых число связанных ключей может быть произвольным, что требует дальнейшей детализации модели PRF-RKA.

Определение 3. В условиях определения 2 при тернарной операции «$\ominus \circ \circledast$» преобладанием противника $\mathcal{A}$ в модели $PRF\text{-}RKA_{\ominus\circ\circledast}$ назовём

$$\mathrm{Adv}^{PRF\text{-}RKA_{\ominus\circ\circledast}}_{\mathsf{F}}(\mathcal{A}) = \Pr\left(\mathcal{A}^{\mathsf{F}_{K\circledast\cdot}(\cdot),\mathsf{F}_{K\circledast\cdot\ominus\cdot}(\cdot)} \Rightarrow 1\right) - \\ - \Pr\left(\mathcal{A}^{\mathsf{R}_{K\circledast\cdot}(\cdot),\mathsf{R}_{K\circledast\cdot\ominus\cdot}(\cdot)} \Rightarrow 1\right).$$

Запрос от $\mathcal{A}$ к «левому» оракулу ($\mathsf{F}_{K\circledast\cdot}(\cdot)$ или $\mathsf{R}_{K\circledast\cdot}(\cdot)$) имеет вид (x, κ), а запрос к «правому» ($\mathsf{F}_{K\circledast\cdot\ominus\cdot}(\cdot)$ или $\mathsf{R}_{K\circledast\cdot\ominus\cdot}(\cdot)$) – (x, κ, ζ). Ресурсы $\mathcal{A}$ считаем суммарно по запросам к обоим оракулам. Значения x в запросах к «правому» оракулу *различны* и не совпадают с таковыми в запросах к «левому».

Ограничение, накладываемое определением, на запросы к «правому» оракулу позволяет использовать эвристические оценки (1). По сути, если запросы осуществляются только к «правому» оракулу, то $d = 1$.

Для пары операций «$\ominus$», «$\circledast$», выбранных из множества $\{\oplus, \boxplus\}$, верно

$$\mathrm{Adv}^{PRF}_{\mathsf{F}}(t, q) \le \mathrm{Adv}^{PRF\text{-}RKA_{\circledast}}_{\mathsf{F}}(t', q, r = 1, d = 1),$$
$$\mathrm{Adv}^{PRF\text{-}RKA_{\circledast}}_{\mathsf{F}}(t, q, r, d) \le \mathrm{Adv}^{PRF\text{-}RKA_{\ominus\circ\circledast}}_{\mathsf{F}}(t', q, r, d),$$

$t' \approx t$, т.к. каждая модель расширяет возможности предшествующей. На рисунке 1 это обозначено (ext).

В [4, 13] было доказано, что для PRF-стойкости алгоритма Стрибог-К достаточно одновременного выполнения двух условий:

1) стойкости $\mathsf{g}^{\triangledown}$ в модели $PRF\text{-}RKA_{\boxplus}$;

2) стойкости $\mathsf{g}^{\triangleright}$ в модели $PRF\text{-}RKA_{\oplus}$.

Для алгоритма HMAC-Стрибог требования аналогичны, но модель в первом условии заменяется на $PRF\text{-}RKA_{\boxplus\circ\oplus}$.

V. Необходимость PRF-$RKA_{\boxplus}$ для $\mathsf{g}^{\triangledown}$

Покажем, что стойкость преобразования $\mathsf{g}^{\triangledown}$ в модели $PRF\text{-}RKA_{\boxplus}$ является *необходимой* для PRF-стойкости алгоритма Стрибог-К. Более точно, если $\mathsf{g}^{\triangledown}$ является лишь PRF-стойким преобразованием, то Стрибог-К может НЕ быть PRF-стойким.

Чтобы доказать необходимость условия нужно построить такое преобразование $\mathsf{w}^{\triangledown}$, которое бы было само PRF-стойким, но при использовании в алгоритме Стрибог-К делало бы последний уязвимым к атаке. Как и большинство доказательств подобного рода, специально сформированное преобразование $\mathsf{w}^{\triangledown}$ имеет весьма специфический искусственный вид.

Преобразования $\mathsf{g}^{\triangledown}$ и $\mathsf{g}^{\triangleright}$ порождены одной и той же функцией сжатия g, но теорема о PRF-стойкости [13] не использует этот факт существенным образом, в качестве пары $(\mathsf{g}^{\triangledown}, \mathsf{g}^{\triangleright})$ допустимо использовать *независимые по построению преобразования*. Сделанное наблюдение существенно упрощает дальнейший анализ, при конструировании специального уязвимого $\mathsf{w}^{\triangledown}$ (играющего роль $\mathsf{g}^{\triangledown}$) можно быть уверенным в неизменности $\mathsf{g}^{\triangleright}$, сохранении «хороших» свойств последнего. В частности, такая независимость позволяет не переходить к эвристическим аргументам типа «идеальный шифр».

Пусть теперь Стрибог-К использует вместо $(\mathsf{g}^{\triangledown}, \mathsf{g}^{\triangleright})$ пару $(\mathsf{w}^{\triangledown}, \mathsf{g}^{\triangleright})$,

$$\text{Стрибог-К}(K, M) = \mathsf{w}^{\triangledown}(\mathsf{Csc}(\mathsf{w}^{\triangledown}(IV, K), M), K \boxplus \sigma),$$
$$\mathsf{Csc}(K_{\mathsf{Csc}}, M) = \mathsf{g}(\ldots(\mathsf{g}(K_{\mathsf{Csc}} \oplus \Delta_0, m_1)\ldots, L).$$

Фактически, рассматриваем более широкий класс преобразований, чем задаёт Стрибог-К при параметризации одной функцией сжатия $\mathsf{g}(h, m)$.

Построим PRF-стойкое преобразование $\mathsf{w}^{\triangledown} : V^n \times V^n \to V^n$, которое одновременно является слабым в модели $PRF\text{-}RKA_{\boxplus}$. Определим $\mathsf{w}^{\triangledown}$ на основе произвольного PRF-стойкого $\mathsf{h} : V^{n-1} \times V^n \to V^n$,

$$\mathsf{w}^{\triangledown}(0||K, X) = \mathsf{h}(K, X),$$
$$\mathsf{w}^{\triangledown}(1||K, X) = \begin{cases} \mathsf{h}(K, X), & X \notin \{X_0, X_1\} \\ & \text{или } IV \in \{X_0, X_1\}, \\ \mathsf{h}(K, X_0), & X = X_1, \\ \mathsf{h}(K, X_1), & X = X_0, \end{cases}$$
$$X_i = \mathsf{Csc}(\mathsf{h}(K, IV), P_i),\ i \in \{1, 2\},$$

P_0 и P_1 – произвольные тексты, контрольные суммы которых $\mathsf{sum}_{\boxplus}(P_0) = 0\ldots0 = \sigma_0$ и $\mathsf{sum}_{\boxplus}(P_1) = 10\ldots0 = \sigma_1$.

Преобразование $\mathsf{w}^{\triangledown}$ по сути дела разбивает все ключи на пары «почти» эквивалентных, а в редком случае $IV \in \{X_0, X_1\}$ – ключи в паре эквивалентны. Для пары ключей (K, K'), у которых различается лишь старший бит (а равно $K \boxplus 10\ldots0 = K \oplus 10\ldots0 = K'$) верно $\mathsf{w}^{\triangledown}(K, X) = \mathsf{w}^{\triangledown}(K', X)$, $X \in V^n \backslash \{X_0, X_1\}$, но что более важно, при $IV \notin \{X_0, X_1\}$ имеют место коллизии вида

$$\mathsf{w}^{\triangledown}(K, X_0) = \mathsf{w}^{\triangledown}(K', X_1),$$
$$\mathsf{w}^{\triangledown}(K, X_1) = \mathsf{w}^{\triangledown}(K', X_0).$$

Нетрудно видеть, что $\mathsf{w}^{\triangledown}$ не является стойким в модели $PRF\text{-}RKA_{\boxplus}$. Запросы к оракулу в ней, напомним, имеют вид (X, κ) – пара «значение, связь». Соответствующая атака различения с преобладанием, близким к единице:

– запросить $K_{\mathsf{Csc}} = \mathcal{O}(IV, 0)$;

– вычислить $X_0 = \mathsf{Csc}(K_{\mathsf{Csc}}, P_0)$ и $X_1 = \mathsf{Csc}(K_{\mathsf{Csc}}, P_1)$;

– запросить $Y_0 = \mathcal{O}(X_0, \sigma_0)$ и $Y_1 = \mathcal{O}(X_1, \sigma_1)$;

– если $Y_0 = Y_1$, то ответ – «$\mathsf{w}^{\triangledown}$», иначе – «R» .

Описанная атака требует три запроса ($q = 3$), но все они различны (наверняка, $X_0 \ne IV$, $X_1 \ne IV$), а значит, $d = 1$, число связанных ключей равно двум ($r = 2$).

Аналогичная атака различения ($\mathcal{D}_1$) будет иметь место для алгоритма Стрибог-К$[\mathsf{w}^{\triangledown}, \mathsf{g}^{\triangleright}]$, см. рис. 2. Процесс вычислений для запросов P_0 и P_1 будет следующим:

– каскадный ключ $K_{\mathsf{Csc}} = \mathsf{w}^{\triangledown}(\overline{K} \boxplus 0, IV)$;

– $X_0 = \mathsf{Csc}(K_{\mathsf{Csc}}, P_0)$ и $X_1 = \mathsf{Csc}(K_{\mathsf{Csc}}, P_1)$;

– $Y_0 = \mathsf{w}^{\triangledown}(\overline{K} \boxplus \sigma_0), X_0)$ и $Y_1 = \mathsf{w}^{\triangledown}(\overline{K} \boxplus \sigma_1), X_1)$.

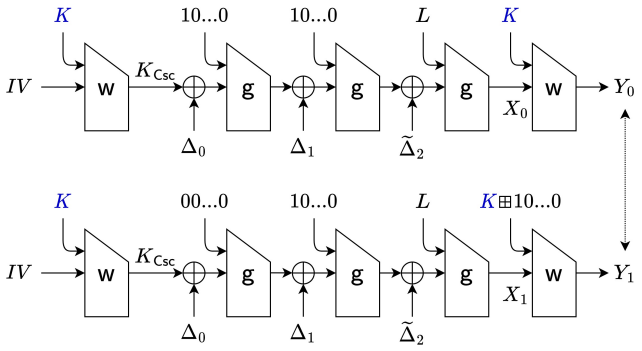

Рис. 2. Коллизия $Y_0 = Y_1$ при использовании «слабого» $\mathsf{w}^{\triangledown}$. С учётом дополнения $P_0 = 10...0||10...0$, $P_1 = 00...0||10...0$.

Коллизия $Y_0 = Y_1$ наверняка будет иметь место, исключение составляет случай, когда одно из значений X_0, X_1 равно IV. Вероятность коллизии, таким образом, можно оценить значением $\approx (1-2\cdot 2^{-n})$. Для случайного оракула вероятность коллизии равна 2^{-n},

$$\mathrm{Adv}^{PRF}_{\text{Стрибог-К}[\mathsf{w}^{\triangledown},\mathsf{g}^{\triangleright}]}(\mathcal{D}_1) \approx (1 - 2 \cdot 2^{-n}) - 2^{-n} \approx 1.$$

Осталось доказать, что $\mathsf{w}^{\triangledown}$ является PRF-стойким.

Утверждение 2. *Преобладание любого противника, атакующего* $\mathsf{w}^{\triangledown}$ *в модели* PRF*, ограничено*

$$\mathrm{Adv}^{PRF}_{\mathsf{w}^{\triangledown}}(t,q) \le \mathrm{Adv}^{PRF}_{\mathsf{h}}(t',q+1),\ t' = t + O(q).$$

Доказательство. Противник $\mathcal{A}$ решает задачу различения $\mathsf{w}^{\triangledown}$ от случайной функции $\mathsf{R} \in \mathrm{Func}(V^n)$.

Пусть $\mathcal{A}$ может различить $\mathsf{w}^{\triangledown} = \mathsf{w}^{\triangledown}[\mathsf{h}]$ и $\mathsf{w}^{\triangledown}[\mathsf{r}]$ – функцию $\mathsf{w}^{\triangledown}$, у которой h заменена на случайную функцию r из $\mathrm{Func}(V^n)$. Нетрудно построить алгоритм $\mathcal{B}$, который будет атаковать h в модели PRF с таким же преобладанием.

Алгоритм $\mathcal{B}$ выбирает случайным образом бит b_K, который играет роль старшего бита ключа. Если $b_K = 1$, то $\mathcal{B}$ делает запрос IV к оракулу $\mathcal{O} \in \{\mathsf{h}, \mathsf{r}\}$, получает ключ K_{Csc}, самостоятельно вычисляет $X_0 = \mathsf{Csc}(K_{\mathsf{Csc}}, P_0)$ и $X_1 = \mathsf{Csc}(K_{\mathsf{Csc}}, P_1)$.

Далее, каждый запрос X от алгоритма $\mathcal{A}$ алгоритм $\mathcal{B}$ обрабатывает с помощью своего оракула согласно описанию $\mathsf{w}^{\triangledown}$. Алгоритм $\mathcal{B}$ передаёт X оракулу и возвращает ответ $\mathcal{A}$. Исключением является случай $b_K = 1$, $IV \notin \{X_0, X_1\}$, $X \in \{X_0, X_1\}$, тогда вместо X_0 алгоритм $\mathcal{B}$ делает к оракулу запрос X_1, а если $X = X_1$, то запрос X_0.

Результат работы $\mathcal{B}$ равен результату работы $\mathcal{A}$. Если $\mathcal{B}$ взаимодействовал с h, то для $\mathcal{A}$ идеально симулировалось преобразование $\mathsf{w}^{\triangledown}[\mathsf{h}]$, а если с r, то $\mathsf{w}^{\triangledown}[\mathsf{r}]$, а значит

$$\Pr(\mathcal{A}^{\mathsf{w}^{\triangledown}[\mathsf{h}]} \Rightarrow 1) - \Pr(\mathcal{A}^{\mathsf{w}^{\triangledown}[\mathsf{r}]} \Rightarrow 1) \le \mathrm{Adv}^{PRF}_{\mathsf{h}}(\mathcal{B}).$$

Алгоритму $\mathcal{B}$ потребуется $t' = t + O(q)$ вычислений и не более $(q+1)$ запросов к оракулу.

Покажем теперь, что $\mathsf{w}^{\triangledown}[\mathsf{r}]$ неотличим от случайной функции $\mathsf{R} \in \mathrm{Func}(V^n)$.

При $b_K = 0$ это так в силу $\mathsf{w}^{\triangledown}[\mathsf{r}](X) = \mathsf{r}(X)$. При $b_K = 1$ рассмотрения требует случай $X \in \{X_0, X_1\}$, $IV \notin \{X_0, X_1\}$. В ответ на X_0 и X_1 формируется соответственно $\mathsf{r}(X_1)$ и $\mathsf{r}(X_0)$.

По сути дела, каждой функции $\mathsf{r} \in \mathrm{Func}(V^n)$ в зависимости от $\mathsf{r}(IV)$ ставится в соответствие некоторая перестановка, меняющая два её аргумента (ни одного, когда $IV \in \{X_0, X_1\}$), и что важно, не затрагивающая $\mathsf{r}(IV)$. Значение $\mathsf{r}(IV)$ позволяет разбить $\mathrm{Func}(V^n)$ на объединение из 2^n непересекающихся подмножеств. Каждому такому подмножеству поставлена в соответствие конкретная перестановка, применение которой к каждой функции из подмножества даёт такое же подмножество. Следовательно, при каждой функции r преобразованию $\mathsf{w}^{\triangledown}[\mathsf{r}]$ биективно соответствует функция R, а значит имеет место указанная неотличимость.

Поясним, что не имеет значения, по какому правилу, зависящему от $\mathsf{r}(IV)$, выбраны X_0 и X_1, но существенно то, что $X_0 \ne IV$ и $X_1 \ne IV$. Если убрать это условие, то можно, к примеру, $\mathsf{r}(IV)$ со старшим битом 0 поставить в соответствие $X_0 \ne IV$ и $X_1 \ne IV$, а $\mathsf{r}(IV)$ со старшим битом 1 – $X_0 = IV$, $X_1 \ne IV$. Тогда у $\mathsf{w}^{\triangledown}[\mathsf{r}](IV)$ с вероятностью 0.75 старший бит будет нулевым.

□

Таким образом, мы доказали, что стойкости алгоритма $\mathsf{g}^{\triangledown}$ в модели PRF недостаточно для стойкости алгоритма Стрибог-К в модели PRF. От $\mathsf{g}^{\triangledown}$ необходима стойкость в модели PRF-$RKA_{\boxplus}$ (хотя бы в условиях, когда все значения на входе различны, $d = 1$). Отметим, что полученный результат оставляет возможным некоторое дальнейшее уточнение параметров/ресурсов в модели PRF-$RKA_{\boxplus}$, что, как представляется, не имеет сколько-нибудь значимых последствий. Принципиально то, что к модели PRF сведение выполнить нельзя.

Все представленные в разделе соображения верны и для алгоритма HMAC-Стрибог. Для PRF-стойкости последнего недостаточно PRF-стойкости $\mathsf{g}^{\triangledown}$, требуется стойкость в модели PRF-$RKA_{\boxplus\circ\oplus}$. Описание «слабой» функции $\mathsf{w}^{\triangledown}$ и сама атака $\mathcal{D}_1$ для HMAC-Стрибог такие же, как для алгоритма Стрибог-К.

VI. Необходимость PRF-$RKA_{\oplus}$ для $\mathsf{g}^{\triangleright}$

Теперь покажем, что требование к $\mathsf{g}^{\triangleright}$ также является не только достаточным, но и необходимым. Рассмотрим Стрибог-К, в котором используется пара $(\mathsf{g}^{\triangledown}, \mathsf{w}^{\triangleright})$, а $\mathsf{w}^{\triangleright}$ является PRF-стойким, но не стойким в модели PRF-$RKA_{\oplus}$. Доказательство схоже с представленным в прошлом разделе, но обладает и собственной спецификой.

Построим $\mathsf{w}^{\triangleright}$ так, чтобы с высокой вероятностью каскадное преобразование порождало коллизию на двух специально выбранных сообщениях разной длины (см. рисунок 3): P и $P||L||$-L. Оба сообщения обладают одинаковой контрольной суммой – коллизия каскада означает коллизию для всего криптоалгоритма. Рассматривать сообщения одинаковой длины не имеет смысла, т.к. PRF-стойкости $\mathsf{g}^{\triangleright}$ в таком случае достаточно для PRF-стойкости криптоалгоритма [4, 15].

Для построения преобразования $\mathsf{w}^{\triangleright} : V^n \times V^n \to V^n$ на основе PRF-стойкого $\mathsf{h} : V^{n-1} \times V^n \to V^n$ поделим все ключи $\mathsf{w}^{\triangleright}$ на пары (K, K'), связанные соотношением $\phi = \Delta_i \oplus \widetilde{\Delta}_i = K \oplus K'$, для определённости выберем

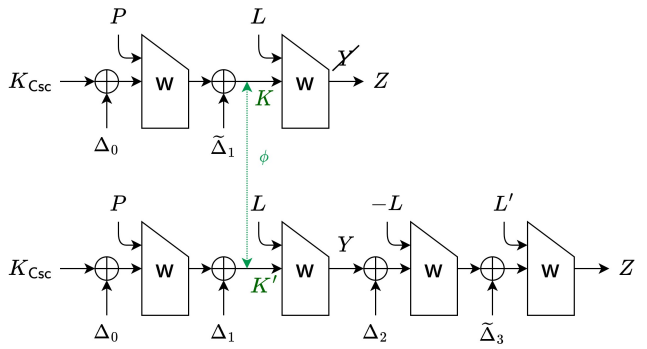

Рис. 3. Коллизия каскадного преобразования. Блоки «P» и «$-L$» включают в себя дополнение строкой 10...0. Контрольная сумма обоих сообщений равна P. Связь между ключами $\phi = \Delta_1 \oplus \widetilde{\Delta}_1$. Длина первого и второго сообщений L и L' бит соответственно.

$i = 1$, $\phi = \mathbf{2} = 2 \cdot n = 0...010...0$,
$K = \overline{K}||1||\underline{K}$, $K' = \overline{K}||0||\underline{K}$,

$$\mathsf{w}^{\triangleright}(\overline{K}||0||\underline{K}, X) = \mathsf{h}(\overline{K}||\underline{K}, X),$$

$$\mathsf{w}^{\triangleright}(\overline{K}||1||\underline{K}, X) = \begin{cases} \mathsf{h}(\overline{K}||\underline{K}, X), & X \neq L, \\ Z, & X = L, \end{cases}$$

$$Y = \mathsf{h}(\overline{K}||\underline{K}, L),$$

$$Z = \mathsf{h}(\mathsf{c}(\mathsf{h}(\mathsf{c}(Y \oplus \Delta_2), -L) \oplus \widetilde{\Delta}_3), L'),$$

а $\mathsf{c} : V^n \to V^{n-1}$ – удаление одного бита $b_K \in \{0, 1\}$, $\mathsf{c}(\overline{K}||b_K||\underline{K}) = \overline{K}||\underline{K}$.

Вычисление функции $\mathsf{w}^{\triangleright}$ на ключах K и K' даёт одинаковый результат для всех аргументов, кроме L.

Отметим, что свойства ключей в паре (K, K') не симметричны, пара сообщений P и $P||L||$-L порождает коллизию, когда $K = \overline{K}||1||\underline{K}$ соответствует первому сообщению, как на рисунке выше. Иными словами, коллизия будет наблюдаться в половине случаев, что можно использовать в качестве критерия различения, преобладание соответствующего алгоритма ($\mathcal{D}_2$) будет равно

$$\mathrm{Adv}^{PRF}_{\text{Стрибог-К}[\mathsf{g}^{\triangledown}, \mathsf{w}^{\triangleright}]}(\mathcal{D}_2) \approx \frac{1}{2} - 2^{-n} \approx \frac{1}{2}.$$

Осталось показать, что $\mathsf{w}^{\triangleright}$ является PRF-стойким, что в совокупности с различителем $\mathcal{D}_2$ продемонстрирует необходимость требования «стойкость в модели PRF-$RKA_{\oplus}$».

Утверждение 3. *Преобладание любого противника, атакующего* $\mathsf{w}^{\triangleright}$ *в модели* PRF*, ограничено*

$$\mathrm{Adv}^{PRF}_{\mathsf{w}^{\triangleright}}(t, q) \leq \mathrm{Adv}^{PRF}_{\mathsf{h}}(t', q) + 2 \cdot \mathrm{Adv}^{PRF}_{\mathsf{h}}(t', 1), \ t' \approx t.$$

Доказательство. Рассмотрим последовательность из преобразований $\mathsf{w}^{\triangleright}_0$, $\mathsf{w}^{\triangleright}_1$, $\mathsf{w}^{\triangleright}_2$, $\mathsf{w}^{\triangleright}_3$, где $\mathsf{w}^{\triangleright}_0 = \mathsf{w}^{\triangleright}$, а каждое следующее отличается от предыдущего.

У $\mathsf{w}^{\triangleright}_1$ вместо h используется случайная функция $\mathsf{r}_1 \in \mathrm{Func}(V^n)$, $\mathsf{w}^{\triangleright}_1(X) = \mathsf{r}_1(X)$, кроме случая $b_K = 1$, $X = L$, тогда $Y = \mathsf{r}_1(L)$, $\mathsf{w}^{\triangleright}_1(L) = Z$, а Z вычисляется также, как и для $\mathsf{w}^{\triangleright}$.

Преобразование $\mathsf{w}^{\triangleright}_2$ отличается от $\mathsf{w}^{\triangleright}_1$ тем, что $Z = \mathsf{h}(\mathsf{c}(\mathsf{r}_2(-L) \oplus \widetilde{\Delta}_3), L')$, а у $\mathsf{w}^{\triangleright}_3$, продолжая логику, $Z = \mathsf{r}_3(L')$, где r_2, r_3 – случайные функции из $\mathrm{Func}(V^n)$.

Если $\mathcal{A}$ может эффективно отличить $\mathsf{w}^{\triangleright}_0$ и $\mathsf{w}^{\triangleright}_1$, то легко построить алгоритм $\mathcal{B}_1$, который с таким же преобладанием отличит h от случайной функции r_1.

Алгоритм $\mathcal{B}_1$ формирует случайным образом бит ключа b_K и действует согласно описанию $\mathsf{w}^{\triangleright}$. Если запрос X от $\mathcal{A}$ не равен L, то $\mathcal{B}_1$ отправляет X своему оракулу $\mathcal{O} \in \{\mathsf{h}, \mathsf{r}_1\}$, возвращает ответ $\mathcal{A}$. Если $b_K = 1$ и $X = L$, алгоритм $\mathcal{B}_1$ получает от своего оракула Y и самостоятельно вычисляет Z, пользуясь тем, что описание алгоритма h ему известно, а секретный Y получен от оракула. Результат работы $\mathcal{B}_1$ равен результату работы $\mathcal{A}$, получаем

$$\Pr(\mathcal{A}^{\mathsf{w}^{\triangleright}_0} \Rightarrow 1) - \Pr(\mathcal{A}^{\mathsf{w}^{\triangleright}_1} \Rightarrow 1) \leq \mathrm{Adv}^{PRF}_{\mathsf{h}}(\mathcal{B}_1).$$

Алгоритм $\mathcal{B}_1$ выполняет $t' = t + O(q)$ операций, и делает к оракулу q запросов.

Пусть $\mathcal{A}$ может отличить $\mathsf{w}^{\triangleright}_1$ и $\mathsf{w}^{\triangleright}_2$. Тогда можно построить алгоритм $\mathcal{B}_2$, отличающий h от r_2. Алгоритм $\mathcal{B}_2$ также формирует b_K. На каждый запрос X от $\mathcal{A}$ алгоритм $\mathcal{B}_2$ возвращает случайное значение из V^n – симулирует r_1. Исключение составляет ситуация $X = L$ при $b_K = 1$, тогда $\mathcal{B}_2$ делает к своему оракулу $\mathcal{O} \in \{\mathsf{h}, \mathsf{r}_2\}$ запрос «$-L$», далее самостоятельно вычисляет Z, и возвращает его $\mathcal{A}$. Результат работы $\mathcal{B}_2$ равен результату работы $\mathcal{A}$,

$$\Pr(\mathcal{A}^{\mathsf{w}^{\triangleright}_1} \Rightarrow 1) - \Pr(\mathcal{A}^{\mathsf{w}^{\triangleright}_2} \Rightarrow 1) \leq \mathrm{Adv}^{PRF}_{\mathsf{h}}(\mathcal{B}_2),$$

алгоритм $\mathcal{B}_2$ делает к оракулу не более одного запроса.

Алгоритм $\mathcal{B}_3$ строится аналогично алгоритму $\mathcal{B}_2$. При $X = L$ и $b_K = 1$ запрос к оракулу $\mathcal{O} \in \{\mathsf{h}, \mathsf{r}_3\}$ равен L'.

Неравенство треугольника и произвольность алгоритма $\mathcal{A}$ дают доказываемое утверждение. □

Отметим, что от $\mathsf{g}^{\triangleright}$ можно потребовать стойкости к коллизиям (модель CR – Collision Resistance) вместо стойкости в модели PRF-$RKA_{\oplus}$. Модели PRF и CR не являются сравнимыми (также нельзя сравнить PRF-RKA и CR), существуют преобразования, которые стойки в одной модели и не стойки в другой. Однако, в типичном случае, преобладание противника в модели CR многократно больше, чем в PRF и PRF-RKA, а значит потребовать от $\mathsf{g}^{\triangleright}$ «быть стойкой к атакам на построение коллизий» означает по факту не ослабить исходное требование, а усилить его.

VII. Стрибог-К в модели PRF-$RKA_{\circledast}$

В двух предыдущих разделах показали, что для PRF-стойкости алгоритма Стрибог-К *необходима*:

1) PRF-$RKA_{\boxplus}$-стойкость $\mathsf{g}^{\triangledown}$;
2) PRF-$RKA_{\oplus}$-стойкость $\mathsf{g}^{\triangleright}$.

Теперь докажем, что в этих условиях Стрибог-К является не только стойкой псевдослучайной функцией (PRF), но и стоек к атакам со связанными ключами (PRF-$RKA_{\boxplus}$).

Для большей общности докажем стойкость в модели PRF-$RKA_{\circledast}$, где наибольший практический интерес представляют случаи «$\circledast = \boxplus$» (фактически используется PRF-$RKA_{\boxplus}$, требования к g совпадают с таковыми для PRF-стойкости Стрибог-К) или «$\circledast = \oplus$» (требования такие же, как для PRF-стойкости HMAC-Стрибог).

Представленная далее теорема является обобщением результата, представленного в [13], ход доказательства аналогичен.

Теорема 1. *Преобладание любого противника, атакующего Стрибог-К в модели $PRF\text{-}RKA_{\circledast}$, ограничено*

$$\mathrm{Adv}^{PRF\text{-}RKA_{\circledast}}_{\textit{Стрибог-К}}(t,q,l,r) \le \mathrm{Adv}^{PRF\text{-}RKA_{\boxplus\circ\circledast}}_{\mathsf{g}^{\nabla}}(t',q',r',d') + \\ + \sum_{i=1}^{r}\mathrm{Adv}^{PRF}_{\mathsf{Csc}}(t',q^{(i)},l') + \frac{q^2+q}{2^{n+1}},$$

где $t' = t + O(q\cdot l)$, $q' = r' = q + r$, $d' = r$, $l' = l + 1$, $\sum_{i=1}^{r} q^{(i)} = q$.

Доказательство. Обозначим результат применения каскадного преобразования к i-му сообщению

$$Y_i = \mathsf{Csc}(K_i^{\mathsf{Csc}}, M_i),$$
$$K_i^{\mathsf{Csc}} = \mathsf{g}^{\nabla}_{\overline{K}\circledast\phi_i\boxplus 0}(IV),\ i = 1,...,q.$$

Будем называть «коллизией» («С») совпадение любой пары элементов в последовательности

$$IV, Y_1, Y_2, ..., Y_q,$$

а противоположное событие обозначаем символом «$\overline{\mathrm{C}}$». «Коллизию», возникающую в результате взаимодействия противника $\mathcal{A}$ с KH, обозначаем $\mathcal{A}^{\mathsf{KH}} \Rightarrow (b, \mathrm{C})$. Значение бита $b \in \{0,1\}$ является непосредственным результатом, возвращаемым противником, а «коллизия» неявным «побочным эффектом» его вычислений и взаимодействий. Любой, кто знает каскадные ключи K_i^{Csc}, может определить, произошла «коллизия» или нет. Если символ b не указан в обозначениях, то подразумевается, что его значение может быть любым, т.е. имеет место равенство

$$\Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow \mathrm{C}) = \Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow (1,\mathrm{C})) + \Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow (0,\mathrm{C})).$$

Рассмотрим преобразование

$$\widetilde{\mathsf{KH}}(M_i) = \mathsf{f}_{\overline{K}\circledast\phi_i\boxplus\sigma_i}(\mathsf{Csc}(\mathsf{f}_{\overline{K}\circledast\phi_i\boxplus 0}(IV), M_i)),$$

полученное заменой в KH первого и последнего вызовов функции сжатия g^{∇} на семейство из 2^n случайных функций f. Если «коллизии» не происходит, то каскадные ключи $K_i^{\mathsf{Csc}} = \mathsf{f}_{\overline{K}\circledast\phi_i}(IV)$ не наблюдаются противником и являются случайными. Кроме этого, в тех же условиях, алгоритм $\widetilde{\mathsf{KH}}$ неотличим от случайной функции R,

$$\Pr(\mathcal{A}^{\mathsf{R}} \Rightarrow 1) = \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}} \Rightarrow (1,\overline{\mathrm{C}})),$$

благодаря тому, что независимо от σ_i и ϕ_i, запрашиваемые у f значения $IV, Y_1, ..., Y_q$ не повторяются.

Согласно определению модели PRF

$$\mathrm{Adv}^{PRF}_{\mathsf{KH}}(\mathcal{A}) = \Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow 1) - \Pr(\mathcal{A}^{\mathsf{R}} \Rightarrow 1).$$

По формуле полной вероятности $\Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow 1) =$

$$= \Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow (1,\mathrm{C})) + \Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow (1,\overline{\mathrm{C}})).$$

Группируя слагаемые и используя неравенство треугольника получим $\mathrm{Adv}^{PRF}_{\mathsf{KH}}(\mathcal{A}) \le$

$$\le \Big(\Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow (1,\overline{\mathrm{C}})) - \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}} \Rightarrow (1,\overline{\mathrm{C}}))\Big) + \\ + \Big(\Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow \mathrm{C}) - \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}} \Rightarrow \mathrm{C})\Big) + \\ + \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}} \Rightarrow \mathrm{C}) = \epsilon + \epsilon_{coll} + p_{coll}.$$

Построим алгоритм $\mathcal{B}_1$, преобладание которого будет равно *сумме* ϵ и ϵ_{coll}. Другими словами, в одном алгоритме используем:

1) способность $\mathcal{A}$ различать KH и $\widetilde{\mathsf{KH}}$, когда «коллизии» нет (ϵ);

2) способность $\mathcal{A}$ создавать «коллизии» в KH и $\widetilde{\mathsf{KH}}$ с разной вероятностью (ϵ_{coll}).

Полагаем, что значения неотрицательны ($\epsilon \ge 0$ и $\epsilon_{coll} \ge 0$), в противном случае инвертируем соответствующий результат работы алгоритма $\mathcal{B}_1$. Говоря более детально, можно построить четыре варианта алгоритма $\mathcal{B}_1$ ($\mathcal{B}_1^{00}, \mathcal{B}_1^{01}, \mathcal{B}_1^{10}, \mathcal{B}_1^{11}$), каждый из которых инвертирует результат работы $\mathcal{A}$ (фактический или побочный – построение «коллизии»). Так, биты 10 будут означать, что «коллизия» интерпретируется как взаимодействие с KH, а непосредственный результат работы $\mathcal{A}$ инвертируется.

$\mathcal{B}_1$ атакует g^{∇} в модели $PRF\text{-}RKA_{\boxplus\circ\circledast}$. При обработке запроса (M_i, ϕ_i) от $\mathcal{A}$, алгоритм $\mathcal{B}_1$:

– отправляет своему «левому» оракулу $\mathcal{O} \in \{\mathsf{g}^{\nabla}, \mathsf{f}\}$ запрос (IV, ϕ_i);

– получает i-й каскадный ключ $K_i^{\mathsf{Csc}} = \mathcal{O}(IV, \phi_i)$, а если ϕ_i ранее использовалось, то $\mathcal{B}_1$ уже обладает хранимым в памяти ключом K_i^{Csc};

– самостоятельно вычисляет значения $\sigma_i = \mathsf{sum}_{\boxplus}(M_i)$ и $Y_i = \mathsf{Csc}(K_i^{\mathsf{Csc}}, M_i)$;

– проверяет условие «коллизии», если $Y_i \in \{IV, Y_1, ..., Y_{i-1}\}$, то завершает работу $\mathcal{A}$ и возвращает 1 (в силу $\epsilon_{coll} \ge 0$ «коллизия» интерпретируется как взаимодействие с KH);

– делает запрос (Y_i, ϕ_i, σ_i) к «правому» оракулу $\mathcal{O} \in \{\mathsf{g}^{\nabla}, \mathsf{f}\}$ и возвращает его ответ $\mathcal{A}$.

Если после q запросов от $\mathcal{A}$ «коллизии» не произошло, то результатом работы $\mathcal{B}_1$ является результат работы $\mathcal{A}$.

Вычислительные ресурсы алгоритма $\mathcal{B}_1$ равны $t' = t + O(q\cdot l)$. К «левому» оракулу (связь по ключу определяется операцией «$\circledast$») делается не более r запросов, к «правому» (связь задана композицией «$\circledast$» и «$\boxplus$») – q запросов, всего не более $q' \le q + r$, совокупное число связанных ключей такое же $r' \le q + r$. Значение IV запрашивается r раз у «левого» оракула с разными связанными ключами, $d' = r$. Запросы к «правому» оракулу различны и не содержат IV.

До возникновения «коллизии» $\mathcal{B}_1$, взаимодействующий с g^{∇} или f, идеально симулирует для $\mathcal{A}$ оракул KH или $\widetilde{\mathsf{KH}}$ соответственно. Преобладание алгоритма $\mathcal{B}_1$ равно $\mathrm{Adv}^{PRF\text{-}RKA_{\boxplus\circ\circledast}}_{\mathsf{g}^{\nabla}}(\mathcal{B}_1) =$

$$= \Pr(\mathcal{B}_1^{\mathsf{g}^{\nabla}_{\overline{K}\circledast\cdot(\cdot)}, \mathsf{g}^{\nabla}_{\overline{K}\circledast\cdot\boxplus\cdot(\cdot)}} \Rightarrow 1) - \Pr(\mathcal{B}_1^{\mathsf{f}_{\overline{K}\circledast\cdot(\cdot)}, \mathsf{f}_{\overline{K}\circledast\cdot\boxplus\cdot(\cdot)}} \Rightarrow 1) =$$
$$= \big(\Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow (1,\overline{\mathrm{C}})) + \Pr(\mathcal{A}^{\mathsf{KH}} \Rightarrow \mathrm{C})\big) -$$
$$- \Big(\Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}} \Rightarrow (1,\overline{\mathrm{C}})) + \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}} \Rightarrow \mathrm{C})\Big) = \epsilon + \epsilon_{coll}.$$

Для завершения доказательства осталось оценить значение вероятности $p_{coll} = \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}} \Rightarrow \mathrm{C})$. Построим алгоритм $\mathcal{B}_2$, который будет отличать набор преобразований

$$\mathsf{Csc}(K_{(1)}^{\mathsf{Csc}}, \cdot), ..., \mathsf{Csc}(K_{(r)}^{\mathsf{Csc}}, \cdot),$$

от набора случайных функций $\mathsf{R}_{(1)}$,...,$\mathsf{R}_{(r)}$. Напомним, что среди q каскадных ключей $K_1^{\mathsf{Csc}}, ..., K_q^{\mathsf{Csc}}$ не более r различных. Алгоритм $\mathcal{B}_2$:

– получает запрос (M_i, ϕ_i) от $\mathcal{A}$;

– определяет номер оракула $j \in \{1, ..., r\}$ по ϕ_i;

– отправляет оракулу ($\mathsf{Csc}(K_{(j)}^{\mathsf{Csc}}, \cdot)$ или $\mathsf{R}_{(j)}$) запрос M_i;

– получает значение Y_i и сохраняет его в памяти;

– возвращает $\mathcal{A}$ случайную строку (симулирует f).

Если в процессе работы возникает «коллизия», то $\mathcal{B}_2$ выключает $\mathcal{A}$ и возвращает результат работы 1. Если за q запросов «коллизия» не произошла, то возвращает 0.

Все сообщения M_1, ..., M_q могут, вообще говоря, быть равны единственному значению (при $r = q$), но если $M_{i_1} = M_{i_2}$, то заведомо $\phi_{i_1} \neq \phi_{i_2}$, $1 \leq i_1 < i_2 \leq q$, а значит значения Y_{i_1} и Y_{i_2} будут получены в результате запроса к *различным* оракулам. Следовательно, если $\mathcal{B}_2$ взаимодействует с $\mathsf{R}_{(1)}$,...,$\mathsf{R}_{(r)}$, то все Y_1,...,Y_q формируются случайно равновероятно и независимо.

В силу вышесказанного, преобладание $\mathcal{B}_2$ в задаче различения оценим как $\epsilon_{Csc} =$

$$= \Pr(\mathcal{B}_2^{\mathsf{Csc}(K_{(1)}^{\mathsf{Csc}},\cdot),\ldots,\mathsf{Csc}(K_{(r)}^{\mathsf{Csc}},\cdot)} \Rightarrow 1) - \Pr(\mathcal{B}_2^{\mathsf{R}_{(1)},\ldots,\mathsf{R}_{(r)}} \Rightarrow 1) \geq$$

$$\geq \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}} \Rightarrow \mathrm{C}) - \left(\frac{q\cdot(q-1)}{2^{n+1}} + \frac{q}{2^n}\right),$$

где $\frac{q}{2^n}$ – оценка вероятности того, что $IV \in \{Y_1, ..., Y_q\}$, и $\frac{q\cdot(q-1)}{2^{n+1}}$ оценка вероятности коллизии среди значений $Y_1, ..., Y_q$, возвращаемых оракулами из набора R_1,...,R_r. Таким образом,

$$p_{coll} = \Pr(\mathcal{A}^{\widetilde{\mathsf{KH}}} \Rightarrow \mathrm{C}) \leq \epsilon_{Csc} + \frac{q^2+q}{2^{n+1}}.$$

Независимость ключей $K_{(1)}^{\mathsf{Csc}}$, ..., $K_{(r)}^{\mathsf{Csc}}$ и функций $\mathsf{R}_{(1)}$,...,$\mathsf{R}_{(r)}$, позволяет воспользоваться простым «гибридным аргументом» [14],

$$\epsilon_{Csc} \leq \sum_{i=1}^{r} \mathrm{Adv}_{\mathsf{Csc}}^{PRF}(\mathcal{B}_2^{(i)}),$$

при этом число запросов ограничено $\sum_{i=1}^{r} q^{(i)} = q$.

Алгоритм $\mathcal{B}_2^{(i)}$ перенаправляет к своему оракулу $\mathcal{O} \in \{\mathsf{Csc}, \mathsf{R}\}$ запросы от $\mathcal{B}_2$ к i-му оракулу ($\mathsf{Csc}(K_{(i)}^{\mathsf{Csc}},\cdot)$ или $\mathsf{R}_{(i)}$). В ответ на запросы к оракулам с номерами 1, ..., $(i-1)$ алгоритм $\mathcal{B}_2^{(i)}$ возвращает случайную строку, а на запросы к оракулам с номерами $(i+1), ..., r$ – самостоятельно вычисляет результат применения каскадного преобразования. Результат работы $\mathcal{B}_2^{(i)}$ равен результату работы $\mathcal{B}_2$.

Напомним, что блок, содержащий значение L, неявно присутствует в запросах, формируемых алгоритмами $\mathcal{B}_2$, $\mathcal{B}_2^{(1)}$, ... , $\mathcal{B}_2^{(r)}$ к каскадному преобразованию, в силу этого $l' = l + 1$.

Сумма верхних оценок на ϵ, ϵ_{coll}, ϵ_{Csc} и слагаемого $\frac{q^2+q}{2^{n+1}}$ дают доказываемое неравенство. □

PRF-стойкость каскадного преобразования Csc доказана в [4] сведением к стойкости функции сжатия $\mathsf{g}^{\triangleright}$ в модели PRF-$RKA_{\oplus}$. Эвристические оценки стойкости каскада, учитывающие специфику конкретного $\mathsf{g}^{\triangleright}$, были даны в [13]. С их учётом получаем для $\circledast \in \{\boxplus, \oplus\}$,

$$\mathrm{Adv}_{\text{Стрибог-К}}^{PRF\text{-}RKA_{\circledast}}(t,q,l,r) \lessapprox \frac{t'\cdot r}{2^k} + \frac{2\cdot t'\cdot q\cdot l'}{2^n} + \frac{q^2+q}{2^{n+1}}.$$

В условиях коллизий у каскадного преобразования, совпадения между связанными ключами не влияют на эвристическую оценку.

Если объем обрабатываемых данных $(q\cdot l)$ меньше, чем $r\cdot 2^{n-k}$, то наиболее эффективным способом нарушения свойств безопасности является универсальный – тотальное опробование ключей (первое слагаемое оценки). При этом достаточно угадать любой из r ключей, каждый новый связанный ключ увеличивает вероятность успеха противника.

При $r = 1$ (т.е. при использовании только одного ключа, отсутствии связанных) получаем в качестве следствия оценку PRF-стойкости

$$\mathrm{Adv}_{\text{Стрибог-К}}^{PRF}(t,q,l) \leq \mathrm{Adv}_{\text{Стрибог-К}}^{PRF\text{-}RKA_{\boxplus}}(t,q,l,r=1),$$

совпадающую с представленной в [13].

VIII. HMAC-Стрибог в модели PRF-$RKA_{\oplus}$

Показать стойкость HMAC-Стрибог в модели PRF-$RKA_{\oplus}$ можно за счёт сведения задачи к предыдущей (теорема 1), не строя напрямую сведéний к функции сжатия. Модель PRF-$RKA_{\boxplus}$ (и другие) не рассматривается для алгоритма HMAC-Стрибог в силу громоздкости порождаемой связи между ключами, которая будет в этом случае определяться тремя аргументами и композицией операций: $\boxplus$, $\oplus$, $\boxplus$.

Теорема 2. *Преобладание любого противника, атакующего* HMAC-*Стрибог в модели* PRF-$RKA_{\oplus}$ *ограничено*

$$\mathrm{Adv}_{\mathsf{HMAC}}^{PRF\text{-}RKA_{\oplus}}(t,q,l,r) \leq \mathrm{Adv}_{\textit{Стрибог-К}}^{PRF\text{-}RKA_{\oplus}}(t,2q,l,2r) + \frac{q^2}{2^{\tau}},$$

где $\tau \in \{\frac{n}{2}, n\}$ *– длина выхода хэш-функции «Стрибог».*

Доказательство. Нетрудно видеть, что

$$\mathsf{HMAC}_{\overline{K}\oplus\phi}(X) = \mathsf{KH}(\overline{K}\oplus opad\oplus\phi, \mathsf{KH}(\overline{K}\oplus ipad\oplus\phi, X)).$$

Рассмотрим преобразование HMAC', в котором согласно модели PRF-$RKA_{\oplus}$ вместо KH используется R – семейство из 2^n случайных функций, индексируемых «связью» $\phi \in V^n$.

Пусть существует эффективный алгоритм $\mathcal{A}$, который умеет отличать HMAC от HMAC', тогда легко построить алгоритм $\mathcal{B}$, эффективно атакующий KH в модели PRF-$RKA_{\oplus}$.

Алгоритм $\mathcal{B}$ на каждый запрос (X, ϕ) от $\mathcal{A}$:

– запрашивает $(X, \phi \oplus ipad)$ у оракула $\mathcal{O} \in \{\mathsf{KH}, \mathsf{R}\}$ и получает $H^I \in V^{\tau}$;

– запрашивает $(H^I, \phi \oplus opad)$ и получает $H^O \in V^{\tau}$;

– возвращает H^O алгоритму $\mathcal{A}$.

Алгоритм $\mathcal{B}$ делает не более $2q$ запросов и использует не более $2r$ связанных ключей. В силу идеальной симуляции оракулов HMAC и HMAC', получаем

$$\Pr(\mathcal{A}^{\mathsf{HMAC}} \Rightarrow 1) - \Pr(\mathcal{A}^{\mathsf{HMAC}'} \Rightarrow 1) \leq \mathrm{Adv}_{\mathsf{KH}}^{PRF\text{-}RKA_{\oplus}}(\mathcal{B}).$$

Противник $\mathcal{A}$ при взаимодействии с HMAC' порождает q наборов вида $(X_i, \phi_i^I, H_i^I, \phi_i^O, H_i^O)$, $\phi_i^I = \phi_i \oplus ipad$, $\phi_i^O = \phi_i \oplus opad$, $\phi_i^I \neq \phi_i^O$, $1 \leq i \leq q$. Значения H_i^I не наблюдаются противником непосредственно. До тех пор, пока не возникла любая из следующих коллизий,

$$\mathrm{C1}: \ (X_i, \phi_i^I) = (H_j^I, \phi_j^O),$$
$$\mathrm{C2}: \ (H_i^I, \phi_i^O) = (H_j^I, \phi_j^O),$$

$1 \leq i < j \leq q$, HMAC' неотличим от случайной функции. Коллизии $(X_i, \phi_i^I) = (X_j, \phi_j^I)$ при этом быть не может в силу того, что запросы противника не повторяются, коллизии $(X_i, \phi_i^I) = (H_i^I, \phi_i^O)$ также невозможны из-за $\phi_i^I \neq \phi_i^O$.

Вероятность события C1 не превосходит

$$\Pr(\mathrm{C1}) \le \binom{q}{2} \cdot 2^{-\tau} \le \frac{q^2}{2^{\tau+1}},$$

аналогично для C2. В силу «фундаментальной игровой леммы» [14] получаем

$$\Pr(\mathcal{A}^{\mathsf{HMAC}'} \Rightarrow 1) - \Pr(\mathcal{A}^{\mathsf{R}} \Rightarrow 1) \le \Pr(\mathrm{C1}) + \Pr(\mathrm{C2}),$$

а из неравенства треугольника и произвольности алгоритма $\mathcal{A}$ следует доказываемое утверждение. □

Теоремы 1 и 2 показывают, что PRF-стойкость криптоалгоритма HMAC-Стрибог можно рассматривать как следствие стойкости алгоритма Стрибог-К в модели $PRF\text{-}RKA_{\oplus}$.

Следствие 1. *Преобладание любого противника, атакующего* HMAC-*Стрибог в модели* PRF *ограничено*

$$\mathrm{Adv}^{PRF}_{\mathsf{HMAC}}(t,q,l,r) \le \mathrm{Adv}^{PRF\text{-}RKA_{\oplus}}_{\textit{Стрибог-К}}(t,2q,l,2) + \frac{q^2}{2^{\tau+1}},$$

где $\tau \in \{\frac{n}{2}, n\}$ *– длина выхода хэш-функции «Стрибог».*

Второе слагаемое оценки учитывает невозможность события C1 в модели PRF, $\phi_i^I \neq \phi_j^O$.

IX. Заключение

Основанные на хэш-функции «Стрибог» криптоалгоритмы Стрибог-К и HMAC-Стрибог являются, как было доказано в [13], стойкими псевдослучайными функциями (PRF) в двух предположениях о функции сжатия, используемой в алгоритме хэширования. Функция сжатия должна быть стойкой псевдослучайной функцией в условиях атак со связанными ключами ($PRF\text{-}RKA$), когда в качестве секретного ключа выступает любой из её двух входов.

В настоящей работе в таких же предположениях удалось показать, что Стрибог-К и HMAC-Стрибог сами являются стойкими к атакам со связанными ключами, что позволяет использовать их на практике при реализации в ограниченных условиях, когда генерирование случайных равновероятных независимых ключей является сложной технической задачей.

Подчеркнём, для любых криптоалгоритмов использование связанных ключей *нежелательно*. Полученные результаты лишь показывают, что для HMAC-Стрибог и Стрибог-К такое решение является *допустимым*.

Также было доказано, что ни одно из двух требований к функции сжатия не может быть ослаблено с $PRF\text{-}RKA$-стойкости до PRF-стойкости, тем самым получен ответ к нерешённой задаче [4, Open problem 1]. Результат означает, что оценки [13] PRF-стойкости криптоалгоритмов Стрибог-К и HMAC-Стрибог являются *точными* не только количественно (по численным значениям преобладания противника), но и *качественно* (по требуемым предположениям).

Библиография

[1] *ГОСТ 34.11-2018. Информационная технология. Криптографическая защита информации. Функция хэширования*, Москва: Стандартинформ, 2018.

[2] R. Merkle, «One way wash functions and DES», в *CRYPTO 1989*, сер. Lect. Notes Comput. Sci. Т. 435, 1990, с. 428—446.

[3] I. Damgård, «A design principle for hash functions», в *CRYPTO 1989*, сер. Lect. Notes Comput. Sci. Т. 435, 1990, с. 416—427.

[4] V. A. Kiryukhin, «Keyed Streebog is a secure PRF and MAC», *Mat. vopr. kriptogr.*, т. 14, № 2, с. 77—96, 2023.

[5] J. Guo, J. Jean, G. Leurent, T. Peyrin и L. Wang, «The usage of counter revisited: second-preimage attack on new Russian standardized hash function», в *SAC 2014*, сер. Lect. Notes Comput. Sci. Т. 8781, 2014, с. 195—211.

[6] *Р 50.1.113-2016 Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования*, Москва: Стандартинформ, 2016.

[7] M. Bellare, R. Canetti и H. Krawczyk, «Keying Hash Functions for Message Authentication», в *Crypto'96*, сер. Lect. Notes Comput. Sci. Т. 1109, 1996, с. 1—15.

[8] N. Koblitz и A. Menezes, «Another look at HMAC», *J. Math. Cryptol.*, т. 7:3, с. 225—251, 2013.

[9] M. Bellare, «New proofs for NMAC and HMAC: security without collision-resistance», в *CRYPTO 2006*, сер. Lect. Notes Comput. Sci. Т. 4117, April 2014, с. 602—619.

[10] P. Gaži, K. Pietrzak и M. Rybár, «The Exact PRF-Security of NMAC and HMAC», в *CRYPTO 2014*, сер. Lect. Notes Comput. Sci. Т. 8616, August 2014, с. 113—130.

[11] M. Nandi, «A New and Improved Reduction Proof of Cascade PRF», *Cryptology ePrint Archive: Report 2021/097*, 2021.

[12] Е. Алексеев, И. Ошкин, В. Попов и С. Смышляев, «О криптографических свойствах алгоритмов, сопутствующих применению стандартов ГОСТ Р 34.11-2012 и ГОСТ Р 34.10-2012», *Мат. вопр. криптогр.*, т. 7, № 1, с. 5—38, 2016.

[13] V. A. Kiryukhin, «About "k-bit security" of MACs based on hash function Streebog», *Cryptology ePrint Archive, Paper 2023/1305*, 2023.

[14] M. Bellare и P. Rogaway, *Introduction to Modern Cryptography*. 2005.

[15] M. Bellare, R. Canetti и K. H., «Pseudorandom Functions Revisited: The Cascade Construction and its Concrete Security», *IEEE*, с. 514—523, 1996.

On security of keyed cryptographic algorithms based on the Streebog hash function against related-key attacks

V. A. Kiryukhin

***Abstract*—The keyless hash function Streebog is a core of several keyed cryptographic algorithms that are used as pseudorandom functions (PRF) and message authentication codes (MAC). One example is Streebog-K proposed at CTCrypt 2022. The proof of its security is based on the reduction to the properties of the underlying compression function. The latter must be secure against related-key attacks (PRF-RKA) when keying through any of the two inputs.**

We prove that in both cases the compression function requirement cannot be relaxed from PRF-RKA to PRF. In addition, if both of these requirements are met, then Streebog-K itself is not only secure PRF, but also resistant to related-key attacks (PRF-RKA).

Similar results are presented for the standardized HMAC-Streebog cryptographic algorithm.

***Keywords*—Streebog, HMAC-Streebog, provable security, PRF, related keys, PRF-RKA**

References

[1] *GOST R 34.11-2012 – National standard of the Russian Federation – Information technology – Cryptographic data security – Hash function*, Moscow: Standartinform, 2012.

[2] R. Merkle, «One way wash functions and DES», in *CRYPTO 1989*, ser. Lect. Notes Comput. Sci. Vol. 435, 1990, pp. 428–446.

[3] I. Damgård, «A design principle for hash functions», in *CRYPTO 1989*, ser. Lect. Notes Comput. Sci. Vol. 435, 1990, pp. 416–427.

[4] V. A. Kiryukhin, «Keyed Streebog is a secure PRF and MAC», *Mat. vopr. kriptogr. [Mathematical Issues of Cryptography]*, vol. 14, no. 2, pp. 77–96, 2023.

[5] J. Guo, J. Jean, G. Leurent, T. Peyrin, and L. Wang, «The usage of counter revisited: second-preimage attack on new Russian standardized hash function», in *SAC 2014*, ser. Lect. Notes Comput. Sci. Vol. 8781, 2014, pp. 195–211.

[6] *R 50.1.113-2016 Informacionnaya tekhnologiya. Kriptograficheskaya zashchita informacii. Kriptograficheskie algoritmy, soputstvuyushchie primeneniyu algoritmov elektronnoj cifrovoj podpisi i funkcii heshirovaniya [R 50.1.113-2016 – Information technology – Cryptographic data security – Cryptographic algorithms accompanying the use of electronic digital signature algorithms and hash functions]*, Moscow: Standartinform, 2016.

[7] M. Bellare, R. Canetti, and H. Krawczyk, «Keying Hash Functions for Message Authentication», in *Crypto'96*, ser. Lect. Notes Comput. Sci. Vol. 1109, 1996, pp. 1–15.

[8] N. Koblitz and A. Menezes, «Another look at HMAC», *J. Math. Cryptol.*, vol. 7:3, pp. 225–251, 2013.

[9] M. Bellare, «New proofs for NMAC and HMAC: security without collision-resistance», in *CRYPTO 2006*, ser. Lect. Notes Comput. Sci. Vol. 4117, April 2014, pp. 602–619.

[10] P. Gaži, K. Pietrzak, and M. Rybár, «The Exact PRF-Security of NMAC and HMAC», in *CRYPTO 2014*, ser. Lect. Notes Comput. Sci. Vol. 8616, August 2014, pp. 113–130.

[11] M. Nandi, «A New and Improved Reduction Proof of Cascade PRF», *Cryptology ePrint Archive: Report 2021/097*, 2021.

[12] E. Alekseev, I. Oshkin, V. Popov, and S. Smyshlyaev, «On the cryptographic properties of algorithms accompanying the applications of standards GOST R 34.11-2012 and GOST R 34.10-2012», *Mat. vopr. kriptogr. [Mathematical Issues of Cryptography]*, vol. 7, no. 1, pp. 5–38, 2016.

[13] V. A. Kiryukhin, «About "k-bit security" of MACs based on hash function Streebog», *Cryptology ePrint Archive, Paper 2023/1305*, 2023.

[14] M. Bellare and P. Rogaway, *Introduction to Modern Cryptography*. 2005.

[15] M. Bellare, R. Canetti, and K. H., «Pseudorandom Functions Revisited: The Cascade Construction and its Concrete Security», *IEEE*, pp. 514–523, 1996.