«Спартиум» – постквантовая схема подписи с хранением состояния

Виталий Кирюхин

ООО «СФБ Лаб»

РусКрипто’2026
25 марта 2026

vitaly.kiryukhin@sfblaboratory.ru

СФБ ЛАБ

Общие сведения

Схемы подписи на хэш-функциях

«Любая схема ЭП использует хэш-функцию,
но только схемы подписи на хэшах
не используют ничего иного»

Схемы подписи на хэш-функциях

СФБ ЛАБ

«Любая схема ЭП использует хэш-функцию,
но только схемы подписи на хэшах
не используют ничего иного»

Достоинства

- В основе хорошо изученные базовые задачи – без стойких хэш-функций схемы подписи в принципе не существуют
- Короткие ключи подписи / проверки подписи (256-512 бит)
- Быстрая проверка подписи

Схемы подписи на хэш-функциях

СФБ ЛАБ

«Любая схема ЭП использует хэш-функцию,
но только схемы подписи на хэшах
не используют ничего иного»

Достоинства

- В основе хорошо изученные базовые задачи – без стойких хэш-функций схемы подписи в принципе не существуют
- Короткие ключи подписи / проверки подписи (256-512 бит)
- Быстрая проверка подписи

Недостатки

- Длина подписи (до 50 килобайт)
- Долгое время формирования подписи

Две категории ЭП на хэш-функциях

СФБ ЛАБ

Stateless (без состояния)	**Stateful (с состоянием)**
Алгоритмы	
Семейство SPHINCS, SPHINCS+ (SLH-DSA), Гиперикум	MSS, LMS, XMSS, XMSS-MT (RFC 8391), Спартиум
Компоненты	
Дерево Меркла (MT) Подпись Винтерница (WOTS) N-разовая подпись (FORS и др.)	Дерево Меркла (MT) Подпись Винтерница (WOTS)
Ограничения	
Отсутствуют – стандартные интерфейсы ЭП	Нужно хранить счётчик использования ключа подписи и другие значения

Целесообразность

Зачем нужна stateful-схема, налагающая доп.ограничения, если есть хорошие stateless-алгоритмы?

1. Размер подписи в 2-5 раз меньше
2. Скорость работы в 10-100 раз больше

Целесообразность

Зачем нужна stateful-схема, налагающая доп.ограничения, если есть хорошие stateless-алгоритмы?

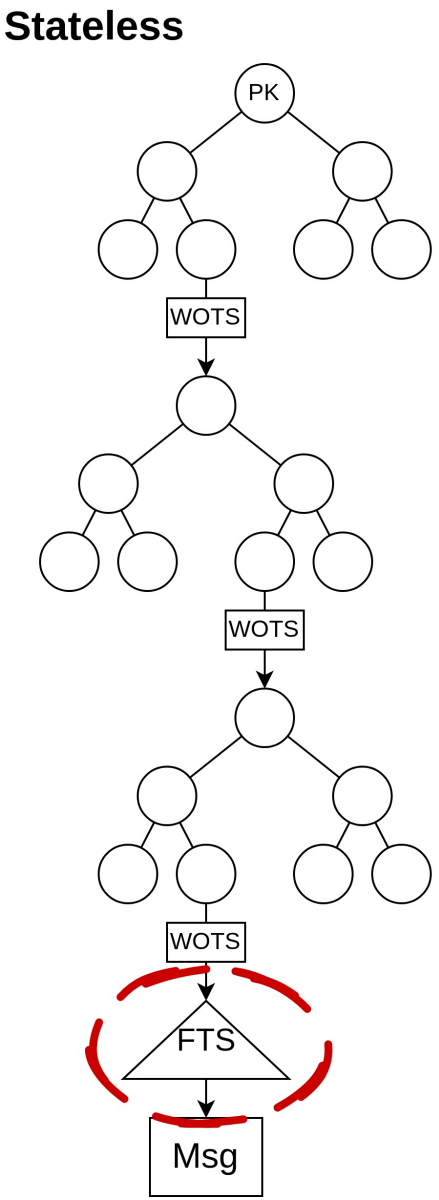
1. Размер подписи в 2-5 раз меньше
2. Скорость работы в 10-100 раз больше

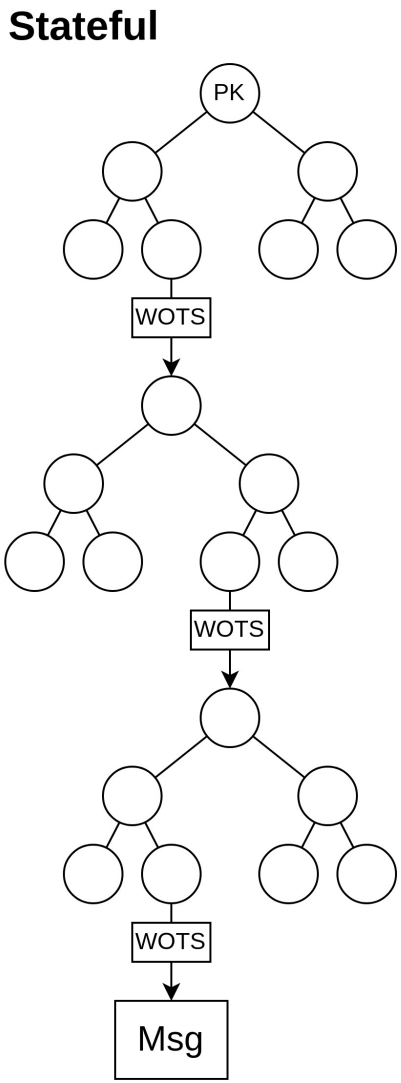
В ряде практических сценариев
хранение состояния можно обеспечить:

- использование токенов
- доверенная среда исполнения
- и т.д.

За счёт чего выигрыш по скорости и размеру?

Структуры схожи, но одно только отсутствие FTS (FORS, HORST или др.) снижает размер подписи на 20-50%

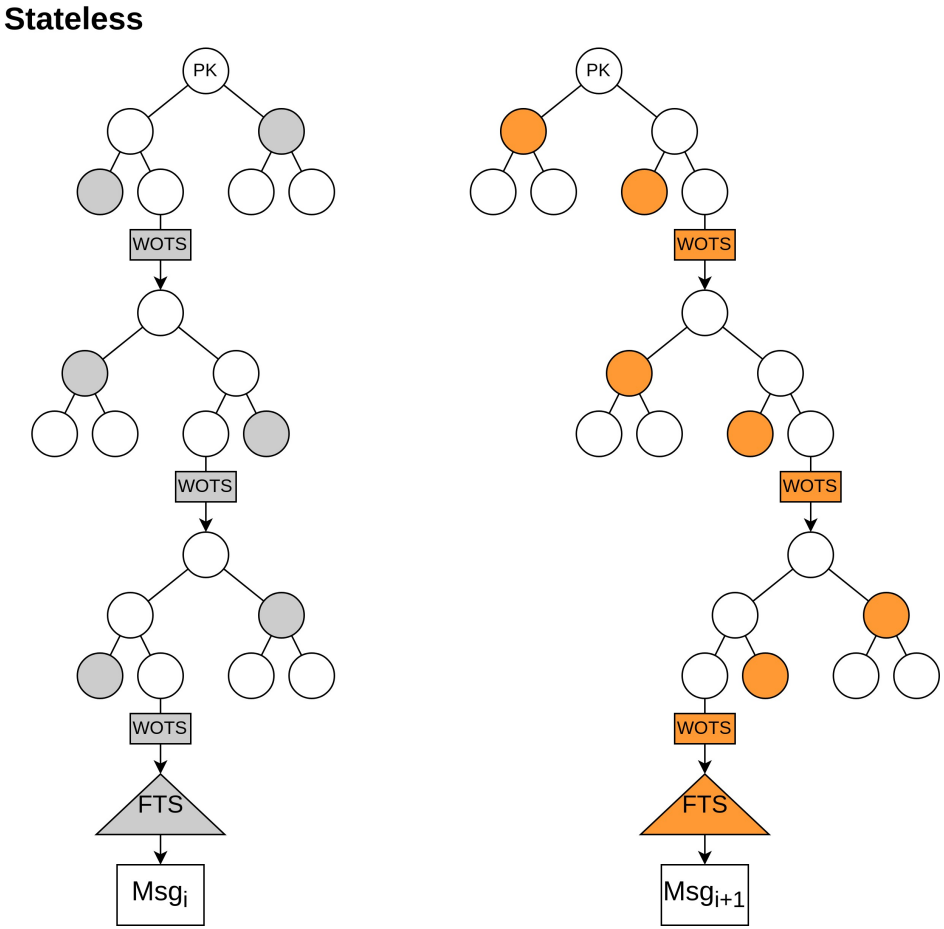




За счёт чего выигрыш по скорости и размеру?

Stateless-схемы

Псевдослучайный выбор номера листа. Подписи у Msg_i и Msg_{i+1} значимо различны – нужно ***заново*** строить все поддеревья на пути от листа к корню.

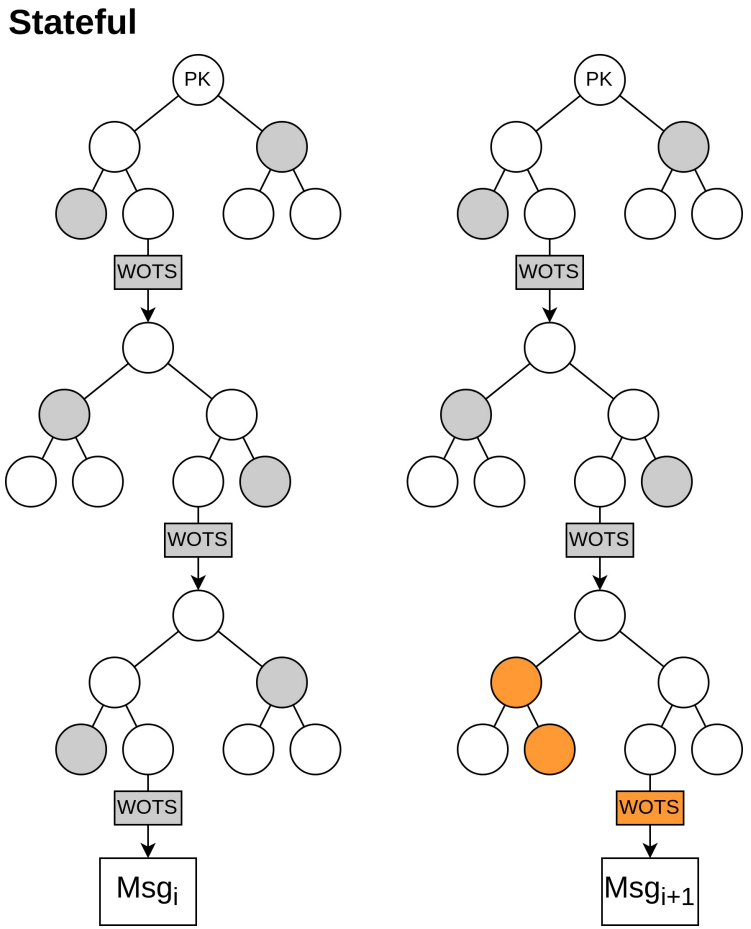


За счёт чего выигрыш по скорости и размеру?

Stateful-схемы

Для подписи выбирается следующий лист (по счётчику).

Подписи у Msg_i и Msg_{i+1} сильно схожи – нужно немного перестроить предыдущую.



Различия

СФБ ЛАБ

⇒ Качественно схожая структура,
но разные условия применения дают:

- разные количественные параметры (высота поддеревьев, длина цепочек и т.д.)
- разные синтезные решения

Ограничения

Формирование двух подписей при одном и том же значении счётчика – подделка. «Хрупкость stateful-схемы ЭП» ≈ «хрупкость режима гаммирования CTR».

Ограничения

Формирование двух подписей при одном и том же значении счётчика – подделка. «Хрупкость stateful-схемы ЭП» ≈ «хрупкость режима гаммирования CTR».

Stateful-схему *нельзя* использовать:

- в распределённой среде
  (один ключ подписи на разных ЭВМ)
- на виртуальных машинах, допускающих «клонирование»

Синтезные решения

Наименование

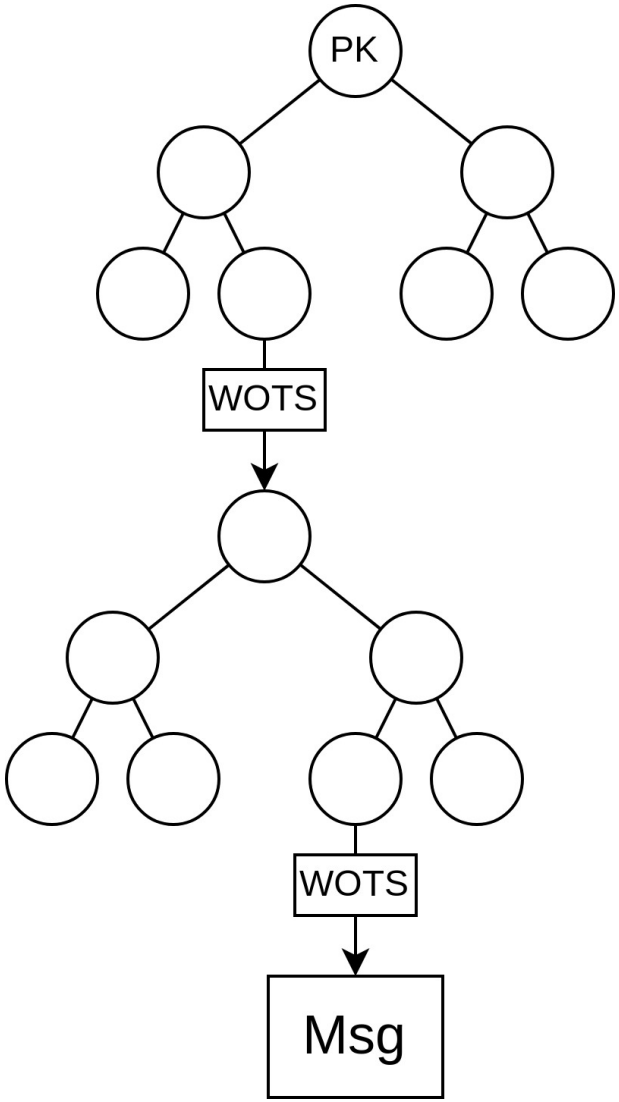
Спар**т**иу**м** (**М**е**т**ельник)

ключевые слова: Stateful, Signature, Tree, Merkle

Структура

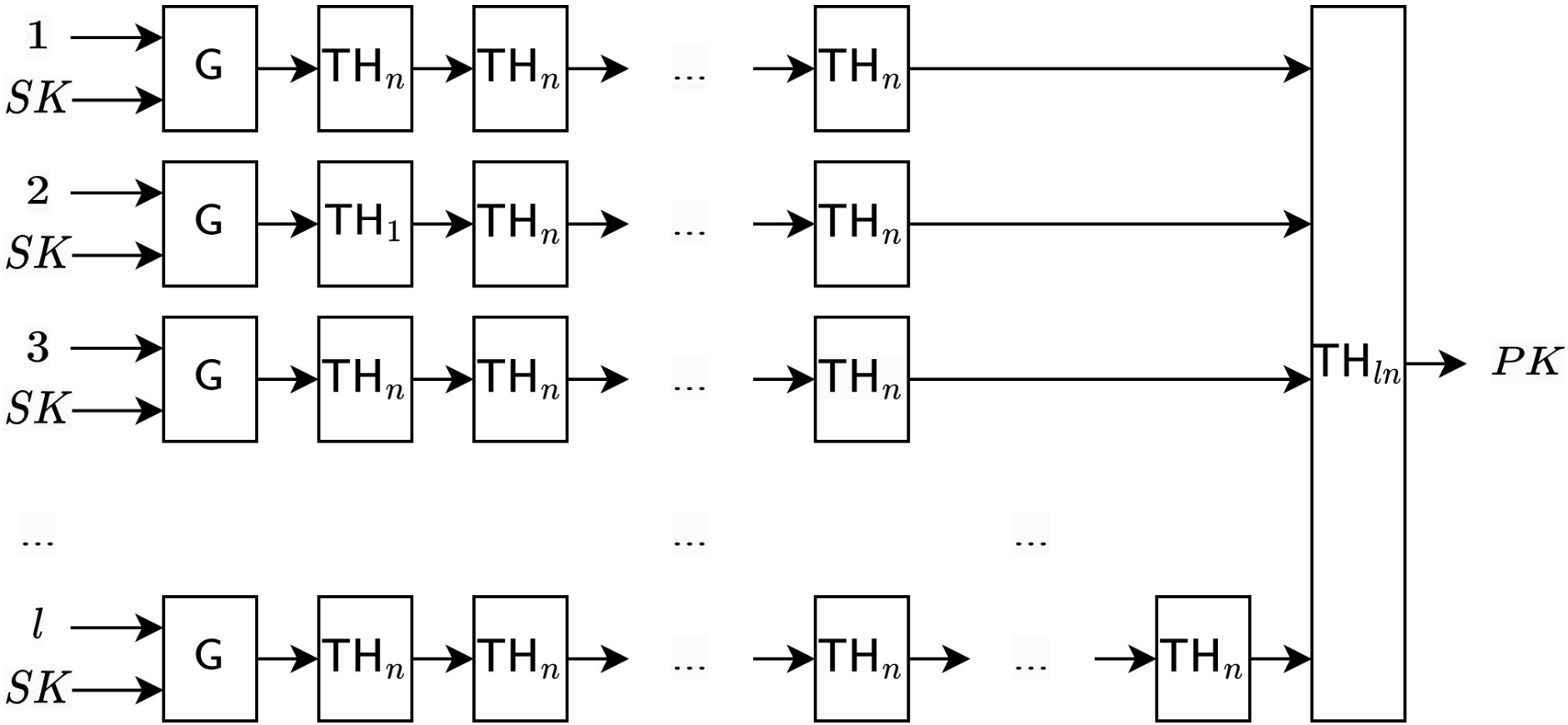
(Гипер)дерево Меркла

- Дерево для наборов малой ёмкости, $q \leq 1$ млн. подписей
- Гипердерево для остальных параметров – лист дерева «подписывает» корень другого дерева



Одноразовая подпись

Схема Винтерница – WOTS-TW



Одноразовая подпись

СФБ ЛАБ

Схема Винтерница – WOTS-TW

1. Публичный ключ – однократное хэширование
2. Опционально, контрольная сумма
   кодируется цепочками *иной* длины
   – сокращаем подпись на одну цепочку без потери скорости
3. Нет «майнинга» как в Гиперикуме и SPHINCS+C
   – для stateful-схем плох компромисс
   «сокращение размера» / «потеря скорости»
4. Нет α-кодирования как в SPHINCS-α
   – использование цепочек разной длины технически проще

Выработка производных ключей

Возможные варианты:

1. «Простой» одноуровневый KDF как в Гиперикуме/SPHINCS/XMSS
2. Древовидный KDF
   - защита от атак по побочным каналам
3. KDF на основе ПГСЧ с внутренним обновлением ключа и свойством Perfect Forward Security (CTR-SG [1])
   - защита от атак по побочным каналам
   - защита от компрометации – противник не сможет сформировать подпись со «старым» номером

[1] Цыпышев В.Н.
О стойкости режима шифрования с внутренним обновлением ключей
представлено в редакцию МВК

Хэш-функции

- «Стрибог-256», $\mathrm{Hash}_{256} : \{0,1\}^* \to \{0,1\}^{256}$
- функция сжатия, $g : \{0,1\}^{512} \times \{0,1\}^{512} \to \{0,1\}^{512}$
- функц. сж. с фикс. состоянием, $g' : \{0,1\}^{512} \to \{0,1\}^{512}$
- выход усекается до λ младших бит, $128 \leq \lambda \leq 256$

Назначение и вычислительная сложность

	Применение	Функция	Сложность (XSPL)
H	Сообщение	Hash_{256}	$25 \times (m + 3)$
G	Генерация ключей WOTS	g'	12
TH_n	Цепочки WOTS	g'	12
TH_{2n}	Деревья MT	g	25
$\mathrm{TH}_{\ell n}$	Сжатие ключей WOTS	Hash_{256}	$25 \times (\ell + 3)$

g в **три** раза быстрее Hash_{256}, g' в **шесть** раз быстрее Hash_{256}.

Ключи

СФБ ЛАБ

Ключ подписи (секретный)

- Секретный ключ ПГСЧ
- Несекретная случайная соль для Hash_{256}

Ключ проверки подписи (публичный)

- Корень (гипер)дерева
- Псевдослучайная соль для Hash_{256}
- Псевдослучайная соль для g
- Псевдослучайная соль для g'

Адресация в дереве

- При абстрактном описании –
  аналогично SPHINCS+ и Гиперикум
- На практике – короткий адрес (64-96 бит)
  – потенциальное использование блочных шифров

Тип операции	8 бит
Номер узла/листа гипердерева	32-64 бита
Номер цепочки	8 бит
Позиция в цепочке	16 бит

Целевой уровень практической стойкости

СФБ ЛАБ

Классические вычислители

λ_{classic} от 128 до 256 бит.
$\lambda_{\text{classic}} < 128$ бит может быть недостаточно.
Производительность сети Bitcoin $\approx 2^{95}$ хэшей в год
($1.1 \cdot 10^9$ TH/сек) при ежегодном двухкратном росте.

Квантовые вычислители

$\lambda_{\text{quantum}} \approx \lambda_{\text{classic}} / 2 \approx$ от 64 до 128 бит.
$\lambda_{\text{quantum}} \geq 64$, как предполагается, достаточно.

Samuel Jaques

Qunatum attacks on AES: When do we need to worry about a structureless, quantum, known plaintext attack against AES?

CHES 2024

Наборы параметров

СФБ ЛАБ

Условия

Параметры h (высота дерева), d (число уровней), w (длина цепочки) почти **не** влияют на стойкостные характеристики. Их выбор – компромисс между допустимой нагрузкой на ключ $q \leq 2^h$ и размером/скоростью формирования подписи.

Общий подход

- Формируем рекомендуемые наборы (λ, h, d, w).
- Допускаем использование иных наборов при: $128 \leq \lambda \leq 256$; $h < 64$; $h \bmod d = 0$; $w \in \{16, 256\}$.

Наборы параметров

СФБ ЛАБ

Пример 1

Стойкость $\lambda_{\text{classic}} = 160$ бит.

Дерево высоты $h = 20$, цепочки длины $w = 256$.

Число подписей $q \leq 2^{20} \approx 1$ млн.

Длина подписи 862 байта.

Длина ключей 40 и 80 байт.

До 2^{13} хэшированний на одну подпись

$\approx$ ГОСТ 34.10-2018 для файла на 1 мегабайт.

1,5 килобайта памяти для хранения состояния.

Наборы параметров

СФБ ЛАБ

Пример 2

Стойкость $\lambda_{\text{classic}} = 256$ бит.

Дерево высоты $h = 48$, уровней $d = 3$, цепочки длины $w = 16$.

Число подписей $q \leq 2^{48} \approx 300$ трлн.

Длина подписи 7910 байт.

Длина ключей 64 и 128 байт.

До 2^{13} хэширований на одну подпись.

6 килобайт памяти для хранения состояния.

Обоснование стойкости

Подходы к доказательству

СФБ ЛАБ

Модель	Тип док-ва	Оценка
SUF-naCMA	сведение	почти точная
SUF-CMA	сведение	неточная
SUF-CMA	RO для хэш-ния сообщ.	почти точная
SUF-CMA	RO	точная

- Почти точная оценка $\approx \lambda - \log_2(w) \approx \lambda$ бит
- Доказательство путём сведения допускает простую адаптацию для случая квантового нарушителя (Q1)

Требования к хэш-функции

СФБ ЛАБ

1. Хэш-функция Hash_{256}:
   - второй прообраз к одному из q сообщений со случайными префиксами и разными настройками (eTCR)
   - второй прообраз (TCR)
2. Функция сжатия g:
   - второй прообраз (TCR)
3. Функция сжатия с фикс. состоянием g':
   - псевдослучайный генератор (PRG)
   - недетектируемость, псевдослучайнойсть (UD)
   - второй прообраз (TCR)
   - прообраз (PRE)

Функция сжатия вне хэш-функции «Стрибог»

СФБ ЛАБ

Допустимо ли это?

Функция сжатия вне хэш-функции «Стрибог»

СФБ ЛАБ

Допустимо ли это?

Типичный подход к доказательству

Пусть противник может «сломать» хэш-функцию,
тогда можно «сломать» и ф.сж.,
а раз ф.сж. «стойкая», то «сломать» хэш-функцию нельзя.
Примеры:
– коллизия для схемы Меркла-Дамгарда
– прообраз для схемы Меркла-Дамгарда
– PRF-стойкость HMAC

Функция сжатия вне хэш-функции «Стрибог»

СФБ ЛАБ

Допустимо ли это?

Типичный подход к доказательству

Пусть противник может «сломать» хэш-функцию,
тогда можно «сломать» и ф.сж.,
а раз ф.сж. «стойкая», то «сломать» хэш-функцию нельзя.
Примеры:
– коллизия для схемы Меркла-Дамгарда
– прообраз для схемы Меркла-Дамгарда
– PRF-стойкость HMAC

Довод 1. **Элементарное преобразование служит «ядром доверия» для хэш-функции, а не наоборот!**

Функция сжатия вне хэш-функции «Стрибог»

СФБ ЛАБ

Довод 2. **Можно вывернуть наоборот**

– показать «почти эквивалентность».

Логика доказательства «в обратную сторону»

Пусть противник может «сломать» ф. сж.,
тогда можно «сломать» и хэш-функцию,
а раз хэш-функция стойкая, то «сломать» ф. сж. нельзя.

Неформальное утверждение

Пусть существует метод построения (второго) прообраза для g' или g со средней сложностью $2^{\lambda-\Delta}$ – стойкость снижена на Δ бит.
Тогда для хэш-функции «Стрибог-512» можно построить (второй) прообраз со средней сложностью $2^{512-\Delta}$.

(пока не доказано, но обосновывается атаками в типичных предположениях)

Хэш-функция и функция сжатия

Допустимо ли их совместное использование?

Хэш-функция и функция сжатия

Допустимо ли их совместное использование?

Да!

«Стандартная модель»

При сведении требования предъявляются по *отдельности* к хэш-функции и к ф.сж., а не к их совокупности.

Random Oracle

Обоснование сложнее, но возможно

за счёт различных значений «соли» и разделения доменов.

О результатах анализа хэш-функции

Хэш-функция Hash_{256}:

- стойкость подтверждается многочисленными результатами криптоанализа – отсутствуют нетривиальные атаки на второй прообраз для «Стрибог-256»
- стойкость в eTCR при гипотезе «Стрибог $\approx$ RO»

  L. R. Akhmetzyanova, A. A. Babueva, A. A. Bozhko

  Streebog as a random oracle – ПДМ – 2024
- в модели eTCR предложена эффективная атака со сложностью $2^{\tau-\log_2(q)} + 2 \cdot 2^{256+\log_2(q)/2}$
  - требуется построение мультиколлизий (сложность $> 2^{n/2}$)
  - применимость при длине выхода *более* $\tau > 256$ бит

Атака на Стрибог в модели eTCR

СФБ ЛАБ

eTCR

Известны хэши $H_i = \mathrm{Hash}(R_i||T_i||M_i)$, $1 \le i \le q$.

Сообщения M_i выбираются противником.

Рандомизатор R_i – (псевдо)случайный, T_i – номер сообщения.

Задача: построить второй прообраз (j, R', M') к одному из q сообщений

$$\mathrm{Hash}(R_i||T_i||M_i) = \mathrm{Hash}(R'||T_j||M').$$

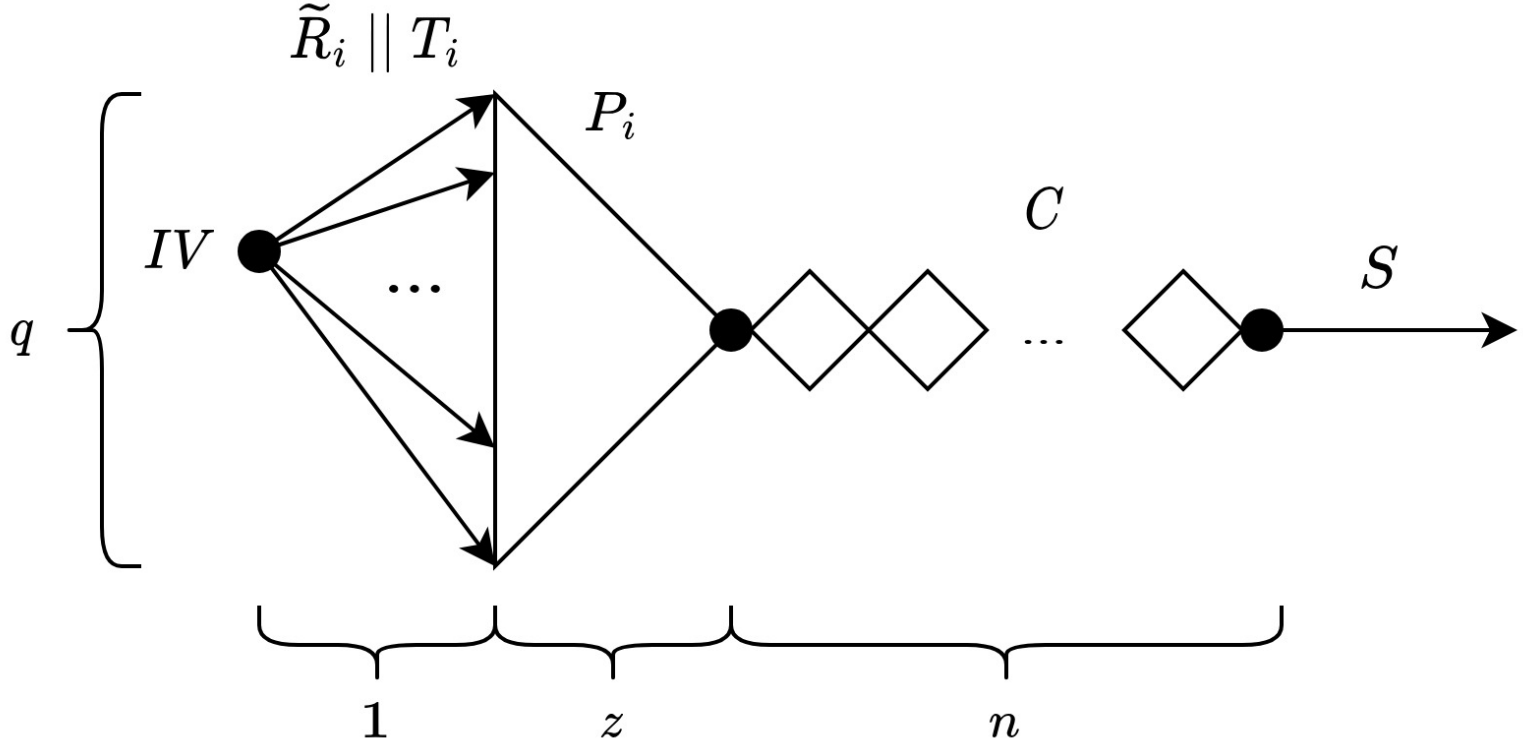
Атака на Стрибог в модели eTCR

Оффлайн этап

Выберем произвольные $\tilde{R}_1,\ldots,\tilde{R}_q$ и $T_1,\ldots,T_q$, хэшируем $\tilde{R}_i||T_i$.

Строим древовидную 2^z-коллизию, $q = 2^z$, сложность $2^{(n+z)/2}$.

Строим 2^n-коллизию, сложность $n \cdot 2^{n/2}$.



Атака на Стрибог в модели eTCR

Онлайн этап

Запросим произвольные сообщения $M_1,\ldots,M_q$.

Перебираем S до получения одного из хэшей $H_1,\ldots,H_q$.

При совпадении с H_i выбираем префикс $\tilde{R}_i||T_i||P_i$.

Сложность $2^\tau/q = 2^{\tau-\log_2(q)}$.

Выбираем блоки C в 2^n-коллизии так, чтобы подобрать контрольную сумму, сложность $2^{n/2+1}$.

Прообраз к $R_i||T_i||M_i$ имеет вид $\tilde{R}_i||T_i||P_i||C||S$.

Атака на Стрибог в модели eTCR

СФБ ЛАБ

Пример

Длина выхода $\tau = 256 + 32 = 288$ бит. Выберем $q = 2^{22}$.

Построение дерева $2^{256+11} = 2^{267}$.

Построение 2^n-коллизии $512 \cdot 2^{257} = 2^{266}$.

Сложность перебора $2^{288-22} = 2^{266}$. Сложность подбора КС 2^{257}.

Стойкость снижена на 20 бит.

Вывод

В схеме Спартиум (и подобных) нельзя обеспечить
более 256 бит классической стойкости
при использовании хэш-функции «Стрибог».
Аналогичное верно для схемы Гиперикум и задачи ITSR.

О результатах анализа хэш-функции

Функция сжатия g и g':

- стойкость подтверждается многочисленными результатами конструктивного криптоанализа (в т.ч. CTCrypt 2021, 2022)
- стойкость доказывается в модели «идеального шифра/подстановки» (Рускрипто 2022)
- возможности противника в целевых формальных моделях *меньше*, чем обычно предполагается при атаках (часть входов зафиксирована, длина выхода ≤ 256, а не 512)
- в рассматриваемых условиях, лучшие нетривиальные методы позволяют атаковать до 6 раундов из 12 полных

Описание и реализация

СФБ ЛАБ

Будут доступны в ближайшее время!

`https://gitflic.ru/project/vkir/spartium`

Благодарю за внимание!

Виталий Кирюхин

ООО «СФБ Лаб»

РусКрипто’2026

25 марта 2026

vitaly.kiryukhin@sfblaboratory.ru