конференция РусКрипто

QApp

«Гиперикум» 2.0: проект квантово-устойчивой схемы цифровой подписи

Сергей Гребнев
Руководитель направления
прикладных исследований

qapp.tech

Sk Участник

Схема «Гиперикум»

Несколько слов о схеме подписи «Гиперикум»:

- Является схемой на хэшах без сохранения состояния
- Представляет собой модификацию схемы SPHINCS$^+$C из-за чего обладает лучшими характеристиками, чем стандартизированная схема SLH-DSA (бывший SPHINCS$^+$)
- Первая редакция проекта была представлена в ТК26 Росстандарта и получены замечания

Вторая редакция

Дополнения к проекту методических рекомендаций:

- Изменена реализация псевдослучайной функции (PRF)
- Добавлены оценки стойкости против классического злоумышленника
- Описан способ выбора наборов параметров
- Расширено описание промежуточных алгоритмов и обоснование стойкости

Замена реализация псевдослучайной функции (PRF)

QApp

В первой редакции используется PRF на базе HMAC [1]. Во второй редакции в качестве PRF предлагается использовать хэш-функцию параметризованную ключом, также известную как Streebog–K [2]. Согласно работам [2,3], функция Streebog–K обеспечивает не меньший уровень стойкости, чем HMAC (у HMAC-256 хуже граница числа обрабатываемых блоков)

[1] Technical Committee 26. Cryptography and security mechanisms: Information technology. Cryptographic algorithms to accompany the usage of digital signature and hash function algorithms
[2] Kiryukhin, V.: Keyed Streebog is a secure PRF and MAC. Cryptology ePrint Archive, Paper 2022/972
[3] V. A. Kiryukhin, “About «k-bit security» of MACs based on hash function Streebog”, Mat. Vopr. Kriptogr., 15:2 (2024), 47–68

Преимущества замены PRF

Согласно проведенным бенчмаркам для набора «Быстрая подпись» на процессоре Intel Core i7-8700 замена HMAC на Streebog–K дает следующие преимущества:

- Время выработки подписи «Гиперикум» уменьшается с 376 до 250 мс
- Время выработки ключа уменьшается с 10.1 до 8 мс

Оценки стойкости против классического злоумышленника

Стойкость «Гиперикума» сводится к ряду различных свойств хэш-функций. Однако, для свойств SM-UD и SM-TCR(+C) существует оценка * только против квантового злоумышленника

* A. Hulsing, M. Kudinov, E. Ronen and E. Yogev, "SPHINCS+C: Compressing SPHINCS+ With (Almost) No Cost," in 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2023, pp. 1435-1453

Свойство SM-UD

SM-UD представляет собой свойство необнаружимости (undetectability), в котором атакующий должен отличить хэш-код, сформированный на основе открытых параметров и случайного входного значения, от случайной строки

$$\mathrm{Adv}^{\mathrm{SM-UD}}_{\mathbf{Th},p}(A) = \left|\Pr[P \leftarrow_{\$} \mathcal{P}; S \leftarrow A_1^{\mathcal{O}_P(\cdot,0)}(\,); 1 \leftarrow A_2(Q,S,P) \wedge \mathbf{DIST}(\{T_i\}_{i=1}^{p})] - \Pr[P \leftarrow_{\$} \mathcal{P}; S \leftarrow A_1^{\mathcal{O}_P(\cdot,1)}(\,); 1 \leftarrow A_2(Q,S,P) \wedge \mathbf{DIST}(\{T_i\}_{i=1}^{p})]\right|$$

Свойство S-TCR(+C)

S-TCR(+C) представляет собой свойство стойкости к нахождению второго прообраза противником, где прообраз должен удовлетворять специальным требованиям WOTS+C

$$\mathrm{Succ}^{\mathrm{S-TCR(+C)}}_{\mathbf{Th},p}(A) = \Pr[P \leftarrow_{\$} \mathcal{P}; S \leftarrow A_1^{\mathcal{O}(P,\cdot,\cdot)}();$$
$$(i, M, counter) \leftarrow A_2(Q, S, P, \mathbf{Th}) :$$
$$\mathbf{Th}(P, T_i, M_i||j_i) = \mathbf{Th}(P, T_i, M||counter)$$
$$\wedge \ M \neq M_i \wedge \mathbf{DIST}(\{T_i\}_{i\in[p]})]$$

Полученные оценки

- Свойство SM-UD:

$$\mathrm{Adv}^{\mathrm{SM-UD}}_{\mathbf{Th},p}(A) \leq q/2^n,$$

- Свойство S-TCR(+C):

$$\mathrm{Succ}^{\mathrm{S-TCR(+C)}}_{\mathbf{Th},p}(A) \leq \frac{q+1}{2^n} + \frac{q}{|\mathcal{P}|},$$

где q — число запросов, n — длина хэш-кода, а $|\mathcal{P}|$ — размер пространства открытых параметров

Оценка стойкости против классического злоумышленника

Указанные оценки и оценки из работ [1,2] позволяют оценить EU-CMA стойкость схемы «Гиперикум» в ROM против **классического** атакующего:

- Для набора «Быстрая подпись»: 245 бит
- Для набора «Универсальный»: 244 бит
- Для набора «Маленькая подпись»: 243 бит

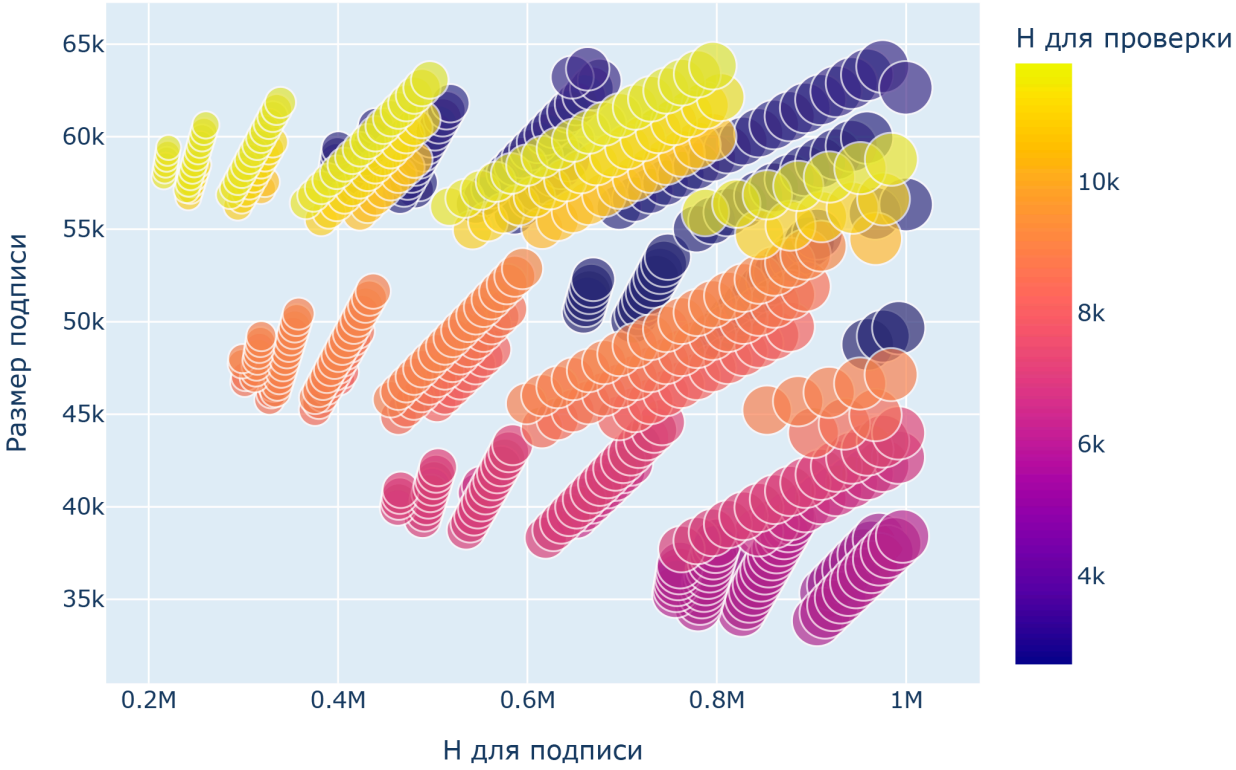
[1] Daniel J. Bernstein и др. «The SPHINCS+ Signature Framework». В: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security
[2] Andreas Hülsing, Joost Rijneveld и Fang Song. «Mitigating Multitarget Attacks in Hash Based Signatures». London, United Kingdom: Association for Computing Machinery, 2019.

Параметры «Гиперикум»

«Гиперикум» обладает широкими возможностями «настройки» характеристик. Сохраняя уровень стойкости, можно манипулировать временем выполнения и размером подписи

Основная идея — подобрать несколько наборов параметров под различные цели и задачи

Распределение наборов параметров



Сложность выбора наборов параметров

QApp

- Наивный подход с минимизацией по трём аргументам бесполезен — число хэшей для подписи варьируется от сотен тысяч до сотен миллионов, в то время как другие параметры варьируются только десятками тысяч
- Выбор точного метода оптимизации затруднителен, так как нет явной функции и явных критериев (что важнее — уменьшить подпись на 1 КБ или время выполнения подписи в 2 раза)
- Авторы SPHINCS и ряда статей посвященных схеме не приводят способ выбора параметров, а также не учитывают время проверки подписи

Способ выбора наборов параметров

Предлагается рассмотреть подход подбора параметров по нескольким критериям, в зависимости от сценария работы алгоритма с определением некоторой целевой функции y, например, минимизация суммы L2 норм основных характеристик:

$$y = c_1 \cdot Sig^2_{norm} + c_2 \cdot Hsign^2_{norm} + c_3 \cdot Hver^2_{norm},$$

где

Sig_{norm} — нормированный размер подписи в байтах,

$Hsign_{norm}$ — нормированное число хэшей для подписи,

$Hver_{norm}$ — нормированное число хэшей для проверки

* Коэффициенты c_1, c_2, c_3 подбираются эмпирически

Набор «Быстрая подпись»

Идея набора — быстрая подпись для высокопроизводительных систем, где подпись часто формируется (например, HSM)

Время выработки подписи является самым важным, время проверки учитывается, но с существенно меньшим коэффициентом, а размер подписи может быть произвольным (но не более 64 КБ)

Функция минимизации имеет вид:

$$y = 0 \cdot Sig^2_{norm} + 1 \cdot Hsign^2_{norm} + 5 \cdot 10^{-6} \cdot Hver^2_{norm}$$

Набор «Маленькая подпись»

Идея набора — маленькая подпись для устройств, где подпись формируется редко или однократно (например, в корнях доверия)

Размер подписи и время проверки являются важными, а время формирования подписи не учитывается

Функция минимизации имеет вид:

$$y = 1 \cdot Sig^2_{norm} + 0 \cdot Hsign^2_{norm} + 1 \cdot Hver^2_{norm}$$

Набор «Универсальный»

Идея набора — универсальный набор, являющийся промежуточным между двумя предыдущими

Приоритет по характеристикам от большего к меньшему: время подписи, время проверки, размер подписи

Функция минимизации имеет вид:

$$y = 1 \cdot Sig^2_{norm} + 5 \cdot 10^{-2} \cdot Hsign^2_{norm} + 5 \cdot 10^{-4} \cdot Hver^2_{norm}$$

Сравнение наборов параметров для максимального уровня стойкости

QApp

Набор	h	d	b	k	Размер, байт	H для под.	H для пров.
Быстрая подпись	66	22	9	36	58 164	215 900	11 645
SPHINCS$^+$-256f	68	17	9	35	47 944	313 289	9 078
Универсальный	64	8	14	21	28 068	2 770 136	4 440
SPHINCS$^+$-256s	64	8	14	22	28 548	2 802 890	4 454
Маленькая подпись	68	4	18	14	18 292	134 348 693	2 350

* Универсальный набор превосходит SPHINCS$^+$-256s по всем характеристикам

Соавторы доклада

QApp

Сергей Гребнев

Руководитель направления прикладных исследований

Олег Турченко

Криптограф-исследователь

QApp

Сергей Гребнев
Руководитель направления
прикладных исследований

sg@qapp.tech
@pikkunorsu

qapp.tech

Участник