

Суперкороткая подпись Шнорра

Сергей Агиевич

НИИ прикладных проблем математики и информатики

Белорусский государственный университет

2026-06-04 [Минск]



Подпись Шнорра

Параметры:

- $\mathbb{G}$ — циклическая группа с образующим G (аддитивная запись);
- q — порядок $\mathbb{G}$, простое 2ℓ -битовое число;
- $\mathcal{S}_0$ — подмножество $\mathbb{Z}_q$;
- φ — функция хэширования $\mathbb{G} \times \{0, 1\}^* \rightarrow \mathcal{S}_0$,
 $\varphi(R, X) = \varphi(-R, X)$.

Ключи: $d \in \mathbb{Z}_q$ — личный, $Q = dG$ — открытый.

Подпись: $(s_0, s_1) \in \mathcal{S}_0 \times \mathbb{Z}_q$.

Проверочное соотношение:

$$\varphi(s_1 G + s_0 Q, X) \stackrel{?}{=} s_0.$$

Короткая подпись

Регулярная подпись:

$\mathcal{S}_0 = \mathbb{Z}_q \Rightarrow$ длина подписи $\approx 4\ell$ битов.

Короткая подпись (оригинальная статья Шнорра):

$|\mathcal{S}_0| \approx 2^\ell \Rightarrow$ длина подписи $\approx 3\ell$ битов.

Время решения основных криптоаналитических задач:

Подпись	KeyRecovery (определение d)	Forge (подделка подписи)
Регулярная	$\Omega(2^\ell)$	$\Omega(2^\ell)$
Короткая	$\Omega(2^\ell)$	$\Omega(2^\ell)$

При переходе от регулярной подписи к короткой ничего не меняется!

Суперкороткая подпись

Что делать, если короткую подпись нужно уложить в $n = 3\ell - \delta$ битов ($\delta > 0$)?

Вариант 1 (снижение уровня стойкости): уменьшить ℓ на $\approx \delta/3$.

Вариант 2 (суперкороткая подпись): сократить S_0 так, что $|S_0| \approx 2^{\ell-\delta}$.

Время решения основных задач:

Подпись	KeyRecovery	Forge
Пониженного уровня	$\Omega(2^{\ell-\delta/3})$	$\Omega(2^{\ell-\delta/3})$
Суперкороткая	$\Omega(2^\ell)$	$\Omega(2^{\ell-\delta})$

Подделка отдельной суперкороткой подписи в $\approx 2^\delta$ менее затратна, чем компрометация личного ключа с последующей подделкой любой подписи.

Но это нормальная ситуация для разумно небольших δ !

Как повысить сложность Forge?

Решение KeyRecovery: ρ -метод, порядка 2^ℓ сложений в $\mathbb{G}$.

Решение Forge: перебор (s_0, s_1) , порядка $2^{\ell-\delta}$ сложений в $\mathbb{G}$ и столько же операций хэширования (при разумной организации перебора).

Как навязать противнику дополнительные вычисления при решении Forge?

Набросок

- Изменить проверочное соотношение на

$$\varphi(s_1^2 G + 2s_1 \varphi_1(s_0, X) Q, X) \stackrel{?}{=} s_0.$$

Здесь $\varphi_1: S_0 \times \{0, 1\}^* \rightarrow \mathbb{Z}_q$ — дополнительная функция хэширования.

- Идея — противнику при подборе (s_0, s_1) всякий раз требуется применять φ_1 , а также вычислять кратный элемент $\mathbb{G}$ вместо однократного сложения.
- Выработка подписи:
 - 1) $k \xleftarrow{R} \mathbb{Z}_q$, $R \leftarrow kG$, $s_0 \leftarrow \varphi(R, X)$;
 - 2) решить относительно s_1 любое из двух квадратных уравнений

$$s_1^2 + 2s_1 \varphi_1(s_0, X) d \equiv \pm k \pmod{q}.$$

Решение будет найдено, если одно из чисел $(\varphi_1(s_0, X) d)^2 \pm k$ является квадратичным вычетом $\pmod{q}$;

- 3) если решение не найдено (вероятность $\approx 1/4$), то повторить генерацию k .