Ежегодная международная научно-практическая конференция
«РусКрипто’2021»

Доказательство с нулевым разглашением для аутентификации в Мастерчейн

Цветков Алексей,

руководитель разработки платформы Мастерчейн, Ассоциация ФинТех

Задача аутентификации в Мастерчейн

В распределённых реестрах аутентификация реализована

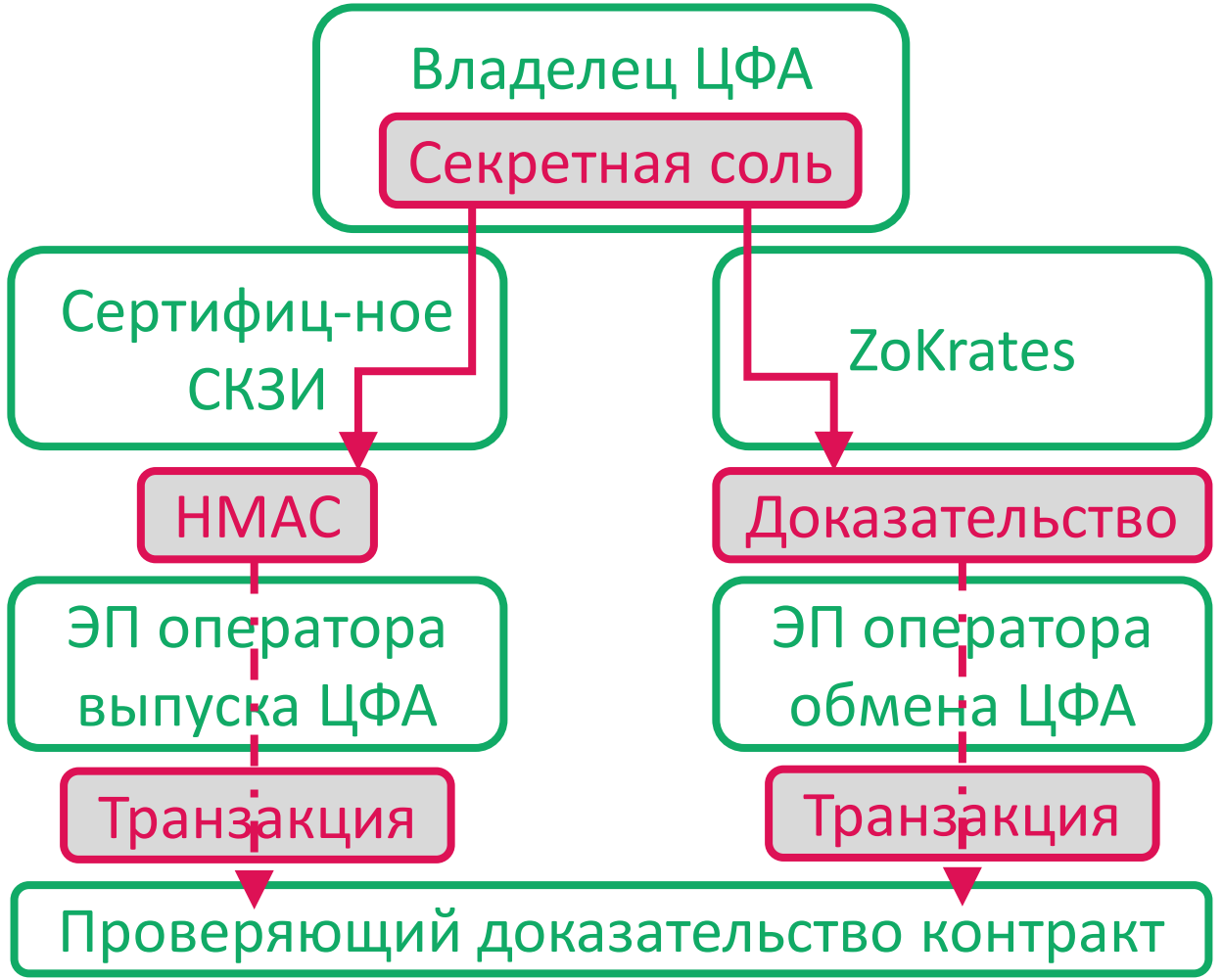
как **проверка электронной подписи** на транзакции (записи в реестре).

Такое решение не может быть использовано в случаях, когда:

- требуется сохранить **анонимность пользователя**;
- существуют технические или правовые **барьеры для обращения к УЦ**;
- **объём обращений превышает ресурсы** ключевой инфраструктуры.

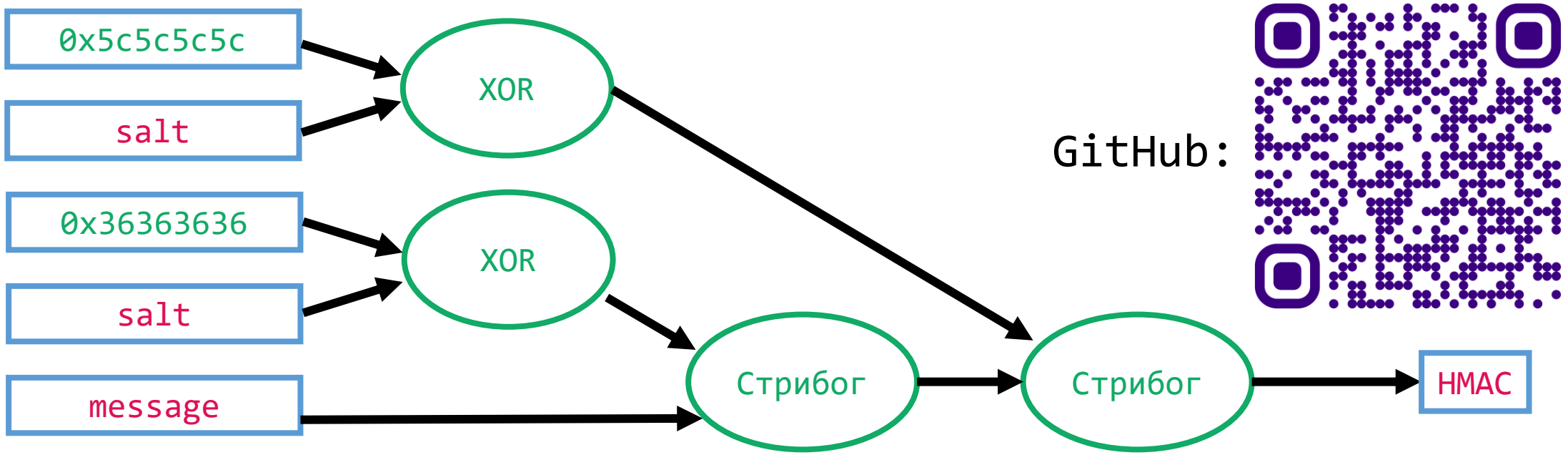
Аутентификация для ЦФА на Мастерчейн

1. При выпуске ЦФА владелец получает от оператора **номер ЦФА**.
2. Владелец генерирует HMAC по **секретной соли** и номеру ЦФА.
3. Оператор ЦФА **подписывает транзакцию** с HMAC владельца.
4. Смарт-контракт аутентифицирует владельца ЦФА по **доказательству знания соли** без её раскрытия.



Построение доказательства знания HMAC

```
def main(private u32[8] salt, u32[8] message) -> u32[8]:
    return H(0x5c5c5c5c, salt, H(0x36363636, salt, message))
```

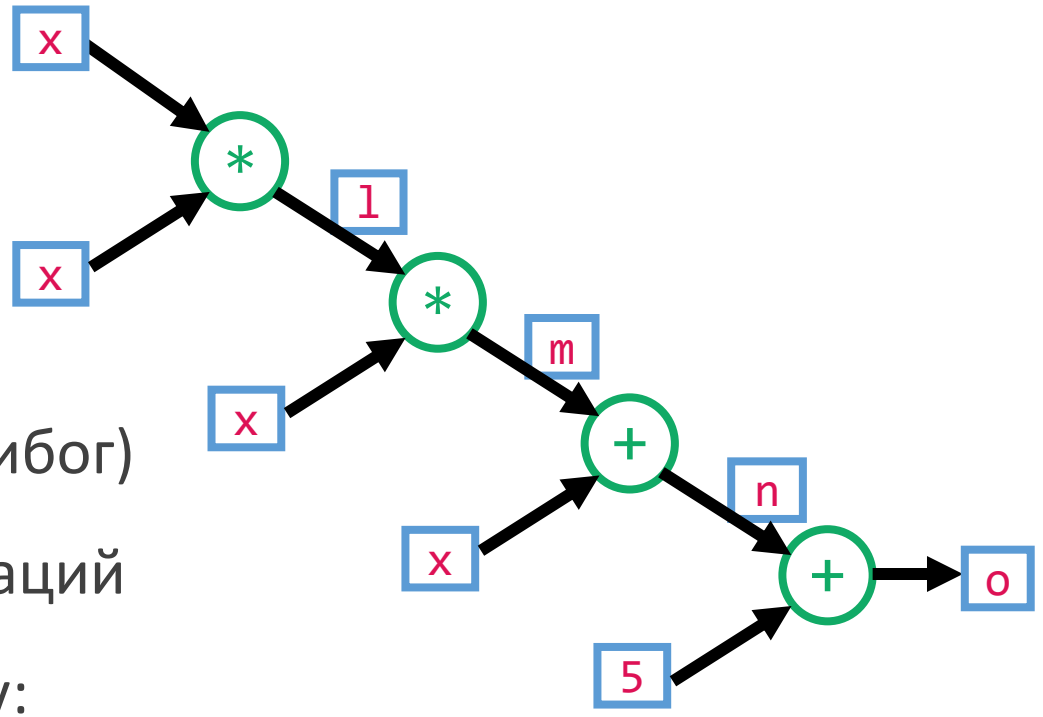


Представление в нормальной форме

Пример: $x^3 + x + 5 = 35$

Свидетельство решения:

W = [x = 3, l = 9, m = 27, n = 30, o = 35]



Вычисляемая функция (в нашем случае Стрибог) приводится к нормальной форме из ***n*** операций сложения и умножения, а затем к полиному:

$$T(c) = W \cdot A(c) \times W \cdot B(c) - W \cdot C(c),$$ который должен иметь нули в $c = 1..n$

Подстановка формы в гомоморфизм

$e : G_1 \times G_2 \rightarrow G_T$ – билинейное отображение для групп G_1, G_2 и G_T

с генераторами g, h и $e(g, h)$ соответственно такое, что:

$$e(g^a, h^b) = e(g, h)^{ab} \quad \text{и} \quad e(g, h)^{(a+b)} = e(g^a, h)\, e(g, h^b)$$

а также между G_1 и G_2 нет эффективно вычислимого изоморфизма.

Тогда рассмотрим равенство $W \cdot A(c) \times W \cdot B(c) = W \cdot C(c) + T(c)$ как:

$$e(g^{W \cdot A(c)}, h^{W \cdot B(c)}) = e(g^{W \cdot C(c)}, h)\; e(g, h)^{T(c)}$$

Вычисление доказательства

Из поля групп G_1 и G_2 изначально выбираются секретные значения α, β, δ, τ,

а при создании доказательства также выбираются слепые множители r и s.

Тогда доказательство определяется значениями: $A = W \cdot A(\tau) + \alpha + r\,\delta$,

$$B = W \cdot B(\tau) + \beta + s\,\delta, \quad C = A\,s + B\,r - r\,s\,\delta + \frac{W \cdot (\alpha\,A(\tau) + \beta\,B(\tau) + C(\tau) + H(\tau)\,Z(\tau))}{\delta}$$

а его проверка производится путём подстановки $\mathbf{g}^A$, $\mathbf{h}^B$, $\mathbf{g}^C$ в выражение:

$$e(\mathbf{g}^A, \mathbf{h}^B) = e(g, h)^{\frac{A(\tau)\,B(\tau) + D}{}} = e(g, h)^{\frac{C(\tau) + T(\tau) + D}{}} = e(g, h)^{\alpha\beta}\, e(\mathbf{g}^C, h^{\delta}),$$

где $D = \alpha\,\beta + \alpha\,B(\tau) + \beta\,A(\tau) + s\,\alpha\,\delta + s\,A(\tau)\,\delta + r\,\beta\,\delta + r\,B(\tau)\,\delta + s\,r\,\delta\,\delta$

Результаты тестирования

Сравнение доказательства знания соли от HMAC на разных хэш-функциях для схемы из статьи Jens Groth, Mary Maller (GM17): ia.cr / 2017 / 540

	Стрибог	SHA-256	Pedersen	MiMC
Количество уравнений	15 227 376	111 595	7 343	3 943
Размер ключа доказательства	11 832 Мб	88 Мб	5 Мб	3 Мб
Длительность построения док-ва	31 373 секунд	132 секунды	10 секунд	5 секунд
Требуемый объём оперативной памяти	~ 80 000 Мб	2 560 Мб	45 Мб	27 Мб

Алгебраические хэш-функции

Длительное время генерации доказательства и требования по объёму памяти вызваны представления XSPL-схемы Стрибога через сложения и умножения.

В стадии исследования существуют хэш-функции, оптимизированные для ДНР:

Pedersen hash: **Zcash** protocol specification, p.150: zips.z.cash/protocol/protocol.pdf

(…) Cryptographic Hashing with **Mi**nimal **M**ultiplicative **C**omplexity: ia.cr / 2016 / 492

Poseidon: A New Hash Function for Zero-Knowledge Proof Systems: ia.cr / 2019 / 458

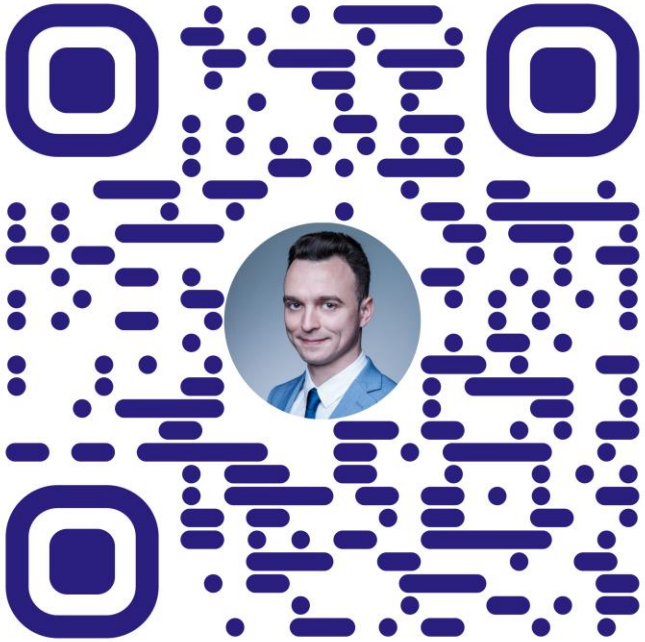
Заключение

Аутентификация в Мастерчейн через доказательство с нулевым разглашением технически выполнима, но затрагивает ряд открытых вопросов:

1. Построение **pairing-friendly** эллиптических кривых.
2. Разработка **хэш-функции**, оптимизированной для алгебраических вычислений.
3. Исследование **стойкости** протокола.

Спасибо за внимание

Добро пожаловать для вопросов и предложений!



Алексей Цветков

Руководитель разработки,

Развитие технологии

распределённых реестров

«Ассоциация ФинТех»

alexey.tsvetkov@fintechru.org