

CryptoNote exploit

aneb proč se musí body na
Curve25519/Ed25519 validovat (pro
Monero, ByteCoin...)

Jak vyrobit \$1M z čistého vzduchu

limit_false • abclinux blog

Soukromí kryptoměn

- Bitcoin
 - ledger je veřejný, každou transakci vidět
- CryptoNote měny (Monero, ByteCoin...)
 - ring signatures, ringCT
 - „mixin“ mnoha výstupů (TXO), spending transakce neodkryje skutečný původní
 - Pedersen commitment – skrývá, kolik se minulo (varianta homomorphic hiding)
 - máme n skrytých obnosů $M_1..M_n$, ale umíme dokázat, že dávají správný součet

Soukromí kryptoměn

- ZCASH

- zkSNARK zero-knowledge proof protokol
- používá homomorphic hiding na zakrytí skutečných hodnot v transakci
- homomorphic hiding umožňuje operace jako lineární kombinace nad hodnotami
- nad tím vším je „Knowledge of Coefficient Assumption“ protokol, kde se přes „blind polynomials“ počítají „Quadratic Architecture Programs“
- „commitment“ se utratí přes „nullifier“

Soukromí kryptoměn prakticky

- Bitcoin

- mnoho free, komerčních i pro vlády určených nástrojů na sledování toků

- Monero

- mnoho transakcí nemá žádný mixin (66% do 31.1.2017)
- monerolink.com – dedukuje reálné výstupy (do 31.1.2017)
- několik změn v samplingu TXO do mixinů
 - uniform, triangular, „recent zone“

Soukromí kryptoměn prakticky

- Monero:

- statisticky nejnovější TXO je „pravé“
 - se změnou samplingu TXO na „recent zone“ už úplně neplatí
- změna signatur na ringCT od 1/2017

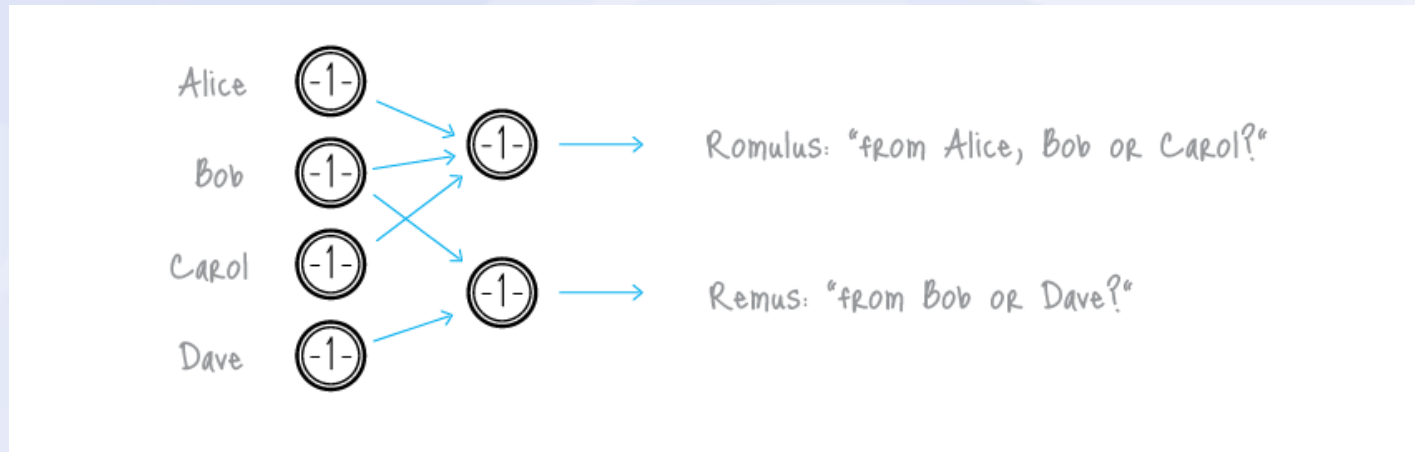
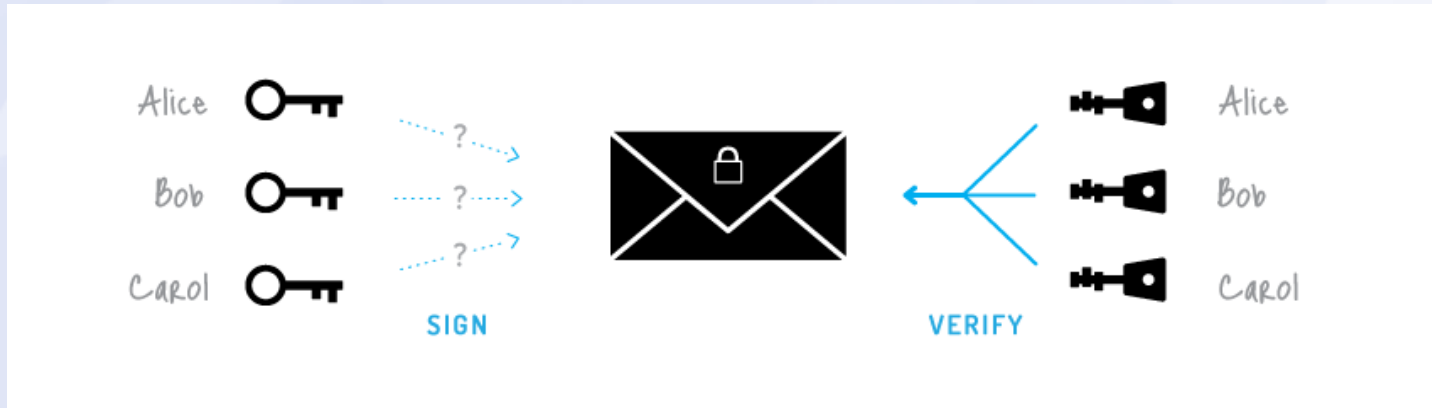
- ZCASH

- 2 typy transakcí: na „transparentní t-adresu“ a na „shielded z-adresu“
- výpočet zkSNARK pro shielded transakci vyžaduje ≥ 2 GB RAM

Bitcoin transakce

- řídí se skriptami, output transakce určuje skript, jak lze peníze minout
- nejběžnější:
 - P2PKH (pay to public key hash)
 - P2SH (pay to script hash)
 - nově segwit transakce (P2WSH, P2WPKH)

CryptoNote ring signatures



CryptoNote ring signatures

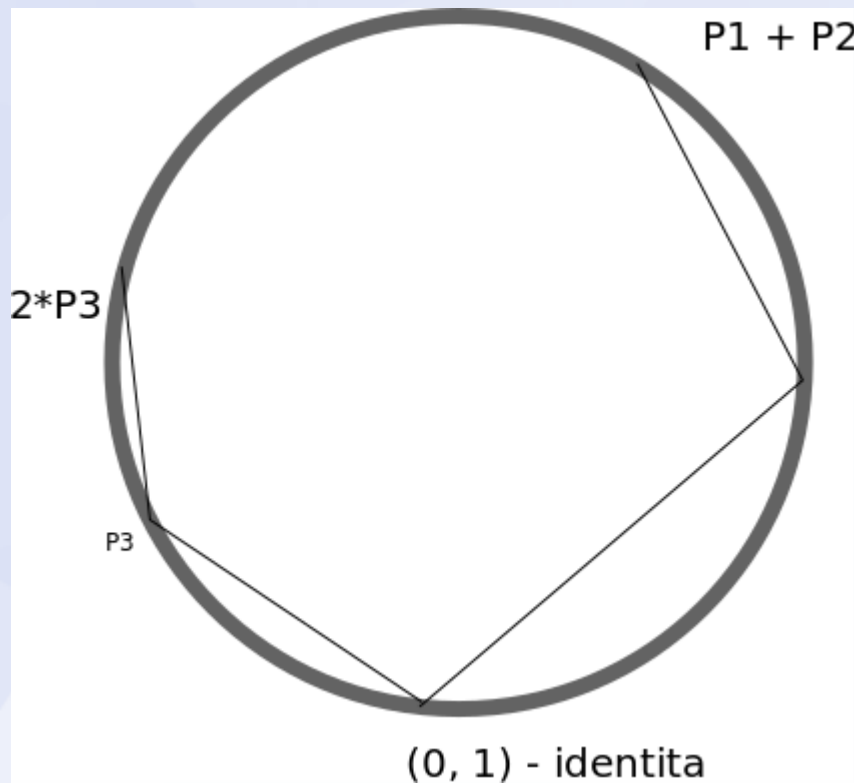
- obnosy měny jsou zaměnitelné (fungible)
 - rozděleny na „bankovky“ až na „dust“
- ring signature určí několik cílových public keys, které mohou utratit její obnos
- jenom jeden je pravý – ring signature ale neukáže, který
- jak zabránit double-spendu:
 - *key image* je něco jako sériové číslo, které zabraňuje utratit transakci 2 krát

Curve25519

- pro účely demonstrace si eliptickou křivku Curve25519 představte jako kruh
- má jeden bod identity („nula“)
- body lze sčítat nebo násobit skalárem
- budeme používat 2 typy dat
 - bod na křivce (x, y) , lze reprezentovat jen x souřadnicí nebo 256-bit stringem
 - skalár: nějaký integer
 - $\text{skalár} \cdot \text{bod} = \text{bod}$

Curve25519

- velmi zjednodušený náčrt operací $*$ a $+$



Ed25519

- jméno křivky i název podpisového schématu (hence the confusion)
- Ed25519 křivka „birationally equivalent“ s Curve25519 (Edwards twisted curve)
- body z Ed25519 lze převést na Curve25519
 - opačně ne, protože komprese bodů na Curve25519 zahazuje znaménko
 - nebo ztratíte znaménko

Řád prvku/grupy/křivky

- řád grupy je počet prvků grupy, řád křivky je počet bodů křivky. Řád křivky Curve25519/Ed25519 není prvočíselný (!!!)
 - je $8L$, $L=2^{252} + 27742317777372353535851937790883648493$
- řád prvku je počet krát, kolik se bod musí sčítat sám se sebou až bude výsledek nula
 - může být 1, 2, 4, 8, L, 2L, 4L, 8L

Curve25519 klíče

- privátní klíč je skalár $x \neq 0$
 - plus pár restrikcí (dělitelný 8,...)
- veřejný klíč je bod P , který vznikne násobením $P=x \cdot G$ (G je veřejně známý bod, tzv. generátor)
- G generuje grupu s prvočíselným řádem L
- tohle zaručuje, že bod veřejného klíče P nebude degenerovaný, tj jeho řád $order(P)=L$

„Špatné“ body na Curve25519

- pokud ale do protokolu vstupuje nějaký bod křivky zvenku, nemáte žádnou záruku, že nebude degenerovaný ($\text{order} < L$)
- na toto narazilo ověření ring signature v CryptoNote
- jmenuje se to „subgroup attack“
- Curve25519 má $\text{cofactor}=8$, tj. řád křivky je 8-násobek řádu G

One-time ring signature (full)

He picks a random $\{q_i \mid i = 0 \dots n\}$ and $\{w_i \mid i = 0 \dots n, i \neq s\}$ from $(1 \dots l)$ and applies the following *transformations*:

$$L_i = \begin{cases} q_i G, & \text{if } i = s \\ q_i G + w_i P_i, & \text{if } i \neq s \end{cases}$$
$$R_i = \begin{cases} q_i \mathcal{H}_p(P_i), & \text{if } i = s \\ q_i \mathcal{H}_p(P_i) + w_i I, & \text{if } i \neq s \end{cases}$$

The next step is getting the non-interactive *challenge*:

$$c = \mathcal{H}_s(m, L_1, \dots, L_n, R_1, \dots, R_n)$$

Finally the signer computes the *response*:

$$c_i = \begin{cases} w_i, & \text{if } i \neq s \\ c - \sum_{i=0}^n c_i \pmod{l}, & \text{if } i = s \end{cases}$$
$$r_i = \begin{cases} q_i, & \text{if } i \neq s \\ q_s - c_s x \pmod{l}, & \text{if } i = s \end{cases}$$

The resulting signature is $\sigma = (I, c_1, \dots, c_n, r_1, \dots, r_n)$.

VER: The verifier checks the signature by applying the inverse transformations:

$$\begin{cases} L'_i = r_i G + c_i P_i \\ R'_i = r_i \mathcal{H}_p(P_i) + c_i I \end{cases}$$

Finally, the verifier checks if $\sum_{i=0}^n c_i \stackrel{?}{=} \mathcal{H}_s(m, L'_0, \dots, L'_n, R'_0, \dots, R'_n) \pmod{l}$

One-time ring signature

- zjednodušený případ pro ring size = 1

```
#keygen - x privátní klíč, P veřejný:
```

```
x = random_scalar()
```

```
P = x*G
```

```
#podpis zprávy m:
```

```
I = x*hashp(P)
```

```
k = random_scalar()
```

```
e = hashes(m, k*G, k*hashp(P))
```

```
s = k - e*x
```

```
#publikuj podpis (P, e, s, I) - I zde ovládá útočník
```

```
#ověření podpisu (P, e, s, I)
```

```
e == hashes(m, s*G + e*P, s*hashp(P) + e*I)
```


Triviální útok

- key image I , který útočník posílá v spending transakci, může být jiný, než ten, s kterým se podpis vytvářel
- př. pokud použil původně I_1 řádu 4, při spending transakci použije jiný I_2 řádu 4
 - e musí být dělitelné 4-ma
 - pak v ověřovací rovnici $e \cdot I_1 = e \cdot I_2 = 0$
 - 2 transakce s různými $I \Rightarrow$ double-spend

Lepší útok

- double-spend lze udělat s libovolným $I'=I+B$, kde B je malého řádu > 1
 - e musí být dělitelné řádem B (velká pravděpodobnost)
- pak $e \cdot I' = e \cdot I + e \cdot B = e \cdot I + 0 = e \cdot I$ a ověřovací rovnice bude sedět

Bytecoin blockchain sieving

- lidi brzo našli několik doublespend transakcí
- chtěl jsem je najít všechny – napsal jsem si skript na prolezení blockchainu
- využili jenom ten triviální exploit
- <https://pastebin.com/TaFJBjBq>

Bytecoin – \$1M out of thin air

$$\text{order}(I_1) == \text{order}(I_2) == 8$$

From Block

Hash	4812091c56d146437476fe8d98988f3f132d8eb788fca311a042e2ecacff5dc
Height	1243490
Timestamp (UTC)	2017-04-14 17:29:16

Inputs (1)

Amount	Image	From transaction	Mixin count
188'999'999.99000000	26e8958fc2b227b045c3f489f2ef98f0d5dfac05d3c63339b13802886d53fc05	07a09e3c26d8ffc...	1

From Block

Hash	69990db3d9f696fe271e2f946b9e8708d28cd74b12748427bfcfda1de2f7f553
Height	1243492
Timestamp (UTC)	2017-04-14 17:29:16

Inputs (1)

Amount	Image	From transaction	Mixin count
188'999'999.99000000	26e8958fc2b227b045c3f489f2ef98f0d5dfac05d3c63339b13802886d53fc85	07a09e3c26d8ffc...	1

Bytecoin TX, ki

TXs:

```
cef289d7fab6e35ac123db8a3f06f7675b48067e0dff185c72b140845b8b3b23
7e418cc77935cc349f007cd5409d2b6908e4130321fa6f97ee0fee64b000ff85
5a3db49ef69e1f9dd9b740cabea7328cd3499c29fc4f3295bac3fa5e55384626
74298d301eb4b4da30c06989e0f7ff24a26c90bf4ffc4f2c18f34b7a22cf1136
85b477ca39976c1cf70c4bc5aa20c750859f9dbafcdba0a20aa79490cd9bdb1b
17320545c428fe7d67ff2c8140eef5c970adfc5eecab978986ac8b4b12a1dd84
f5e6754d7859ff4abf7a9733d5852d5ba35a77cab3dff4bb929c626cf1737b5a
```

ki:

```
26e8958fc2b227b045c3f489f2ef98f0d5dfac05d3c63339b13802886d53fc05 8
26e8958fc2b227b045c3f489f2ef98f0d5dfac05d3c63339b13802886d53fc85 8
0100000000000000000000000000000000000000000000000000000000000000 1
ecffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff7f 2
0000000000000000000000000000000000000000000000000000000000000000 4
0000000000000000000000000000000000000000000000000000000000000080 4
c7176a703d4dd84fba3c0b760d10670f2a2053fa2c39ccc64ec7fd7792ac037a 8
```

Aplikace na ringCT, ASNL

- ringCT (nástupce one-time ring signatures v Moneru – Borromean signatures) používá key image obdobným způsobem
 - taky se musí kontrolovat $order(I)$
- Aggregate Schnorr Non-linkable Ring Signature
 - předchozí kandidát na ringCT
 - sčítají se body z nedůvěryhodných zdrojů
 - zamítnuto i z dalších důvodů

Fin