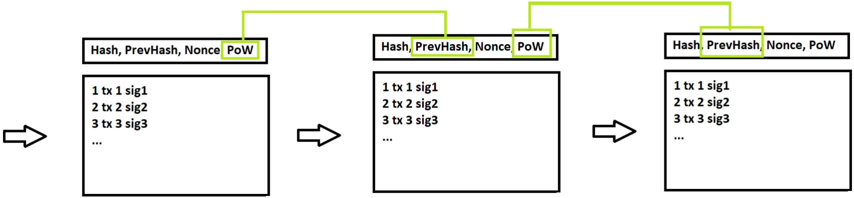
Identity hiding mechanisms in Monero

presentation by Patryk Kozieł
with a few extra slides by M Kutyłowski

Politechnika Wrocławska

January 15, 2021, notes for CS&C students

Blockchain



- Distributed, decentralized
- Used in cryptocurrencies

Discussion on problems with anonymity

Transaction:
user with this public key transfers this amount of money to the user with another public key

Required features:

Untraceability for each incoming transaction all possible senders are equiprobable

Unlinkability for any two outgoing transactions it is impossible to prove they were sent to the same person

MONERO

- Cryptocurrency originally based on CryptoNote protocol
- Initial release: 18 April 2014
- Blockchain + lot of crypto + many details (optimizations in terms of space requirements and performance, network layer security, encoding, ...)
- "Monero" means "coin" in Esperanto
- Ticker Symbol: XMR, 1 XMR = 470 PLN (02.l2.2020)
- Aims to provide full anonymity with respect to sender, recipient and transaction amount
- Dynamic development, lots and lots of major changes across the versions, many ideas in development, not easy to find reliable resources other than source code
- supports smart contracts (interesting topic but for another lecture)

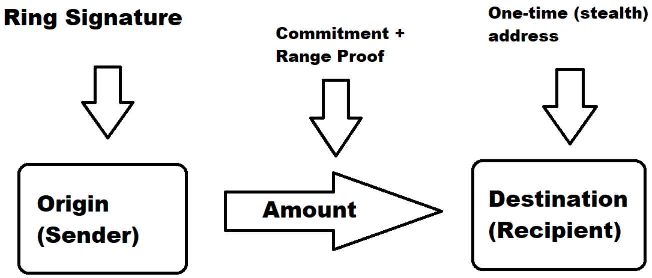
Preliminaries

- Cryptography in Monero is done over Ed25519, which is Twisted Edward Curve over prime field $\mathbb{F}_{2^{255}-19}$ by means of the following equation:

$$-x^2 + y^2 = 1 - \frac{121665}{121666}x^2y^2$$

- Ed25519 order l is 253-bits long, generator G
- $\mathcal{H}_p$ - hash function to point, $\mathcal{H}_n$ - hash function to scalar
- why this curve and not NSA? – avoid allegations about a trapdoor, positive: EdDSA, efficiency, ...

Monero transaction



The journey of Monero transaction

Ok, so I've decided I want to send some XMR to Bob. How can I make this happen?

The journey of Monero transaction - One-time (stealth) address

- First, I need to get Bob's address.
- Every Monero user has a pair of private/public keys: (k^v, K^v) and (k^s, K^s) - so called view and spend keys.
- Public address of a user consists of a tuple of public parts of the above pairs: (K^v, K^s)
- Public address is needed to create unique one-time address

The journey of Monero transaction - One-time (stealth) address

Generating one-time address:

1. Generate $r \in_R \mathbb{Z}_l$
2. Create address:

$$K^o = \mathcal{H}_n(rK^v)G + K^s$$

and add $R = rG$ to the transaction data.

The journey of Monero transaction - One-time (stealth) address

How does Bob know the transaction is for him?
Bob scans through all the transactions in newly mined block and for every output address, using k^v (viewing key) computes:

$$k^v R = rK^v$$

and

$$K'^s = K^o - \mathcal{H}_n(rK^v)G$$

and checks if $K'^s = K^s$. Equality means: it is for Bob!

The journey of Monero transaction - One-time (stealth) address

How does Bob know the transaction is for him?
Bob scans through all the transactions in newly mined block and for every output address, using k^v (viewing key) computes:

$$k^v R = rK^v$$

and

$$K'^s = K^o - \mathcal{H}_n(rK^v)G$$

and checks if $K'^s = K^s$. Equality means: it is for Bob!
Knowing that the output is for him, using k^s Bob can compute corresponding private key that he will need when he wants to spend XMRs received in the transaction:

$$k^o = \mathcal{H}_n(rK^v) + k^s$$
$$k^o G = \mathcal{H}_n(rK^v)G + k^s G = K^o$$

The journey of Monero transaction - One-time (stealth) address

Note: having p outputs in a transaction, to make sure all outputs are unique, we actually compute

$$k^o = \mathcal{H}_n(rK^v, t) + k^s$$
$$k^o G = \mathcal{H}_n(rK^v, t)G + k^s G = K^o$$

where t is an index of a transaction, $t \in \{0, \ldots, p-1\}$.

The journey of Monero transaction - hidden amounts

Now, let's look at the amounts of XMR sent.

The journey of Monero transaction - hidden amounts

Now, let's look at the amounts of XMR sent.

- Transaction amounts in Monero are hidden.
- However, miners need a way to verify that transactions are valid!
- This is achieved using two techniques:
  - Using Pedersen commitments to amounts in a way that miners can verify $\sum inputs = \sum outputs$
  - Using range proofs - Bulletproofs

The journey of Monero transaction - hidden amounts

Every transaction output in Monero has two corresponding values - encoded amount and Pedersen commitment to that amount.

The journey of Monero transaction - hidden amounts

Every transaction output in Monero has two corresponding values - encoded amount and Pedersen commitment to that amount.
This output becomes an input to some future transaction when Bob wants to spend received money.

Definition 1
Pedersen commitment to the amount a with mask x is defined as:

$$C(x, a) = xG + aH,$$

where $H = \text{to\_point}(\mathcal{H}(G)) = \gamma G$

Note that:

$$\begin{aligned} C(x, a) + C(y, b) &= xG + aH + yG + bH = \\ &= (x + y)G + (a + b)H = C(x + y, a + b) \end{aligned}$$

The journey of Monero transaction - hidden amounts

Usually (but not always) there's more than one output in Monero transaction, because inputs have to be spent entirely, it's common to include one output for myself with the change.

Creating commitment

For every output $t \in \{0, \ldots, p-1\}$ and amount b_t create commitment:

$$C(y_t, b_t) = y_t G + b_t H,$$

where b_t is 8-byte amount value and

$$y_t = \mathcal{H}_n(\text{"commitment mask"}, \mathcal{H}_n(rK^v t))$$
$$amount_t = b_t \oplus_8 \mathcal{H}_n(\text{"amount"}, \mathcal{H}_n(rK^v t))$$

$amount_t$ is published with the output.

Note that we are able to get masks and amount for our input commitments.

The journey of Monero transaction - hidden amounts

Now, having a set of n input commitments $C_{i,in} = C(x_i, a_i)$ and m output commitments $C_{j,out} = C(y_j, b_j)$ we could create a commitment to 0 (proving that the sum of inputs is equal to the sum of outputs):

$$\sum_i C_{i,in} - \sum_j C_{j,out} =$$

$$= (a_1 + \ldots + a_n - (b_1 + \ldots + b_m))H +$$

$$(x_1 + \ldots x_n - (y_1 + \ldots + y_m))G =$$

$$= zG$$

we know z, to we can prove it's a commitment to 0.

The journey of Monero transaction - hidden amounts

Now, having a set of n input commitments $C_{i,in} = C(x_i, a_i)$ and m output commitments $C_{j,out} = C(y_j, b_j)$ we could create a commitment to 0 (proving that the sum of inputs is equal to the sum of outputs):

$$\begin{aligned} \sum_i C_{i,in} - \sum_j C_{j,out} = \\ = (a_1 + \ldots + a_n - (b_1 + \ldots + b_m))H + \\ (x_1 + \ldots x_n - (y_1 + \ldots + y_m))G = \\ = zG \end{aligned}$$

we know z, to we can prove it's a commitment to 0.

But, there's a problem...

We would have to reuse the exact commitments from our input transactions and we don't want that (linkability), so instead we'll create pseudo-commitments and prove that the balance is ok in a slightly different way.

The journey of Monero transaction - hidden amounts - pseudo commitments

Pseudo-commitment - published instead of "real" commitments

Having $C(x_i, a_i) = x_i G + a_i H$ we create $C'(x_i', a_i) = x_i' G + a_i H$. Now $C(x_i, a_i) - C'(x_i', a_i)$ is a commitment to 0 (zG) and we have $z = x_i - x_i'$.

The journey of Monero transaction - hidden amounts - pseudo commitments

Pseudo-commitment - published instead of "real" commitments

Having $C(x_i, a_i) = x_i G + a_i H$ we create $C'(x_i', a_i) = x_i' G + a_i H$. Now $C(x_i, a_i) - C'(x_i', a_i)$ is a commitment to 0 (zG) and we have $z = x_i - x_i'$.

Proof that nothing sketchy is going on

For n inputs, we choose randomly $n - 1$ x_i's and set the remaining one in such a way that: $\sum_i C_{in,i}(x_i', a_i) - \sum_j C_{out,j}(y_j, b_j) = 0$

The journey of Monero transaction - hidden amounts - pseudo commitments

Pseudo-commitment - published instead of "real" commitments

Having $C(x_i, a_i) = x_i G + a_i H$ we create $C'(x_i', a_i) = x_i' G + a_i H$. Now $C(x_i, a_i) - C'(x_i', a_i)$ is a commitment to 0 (zG) and we have $z = x_i - x_i'$.

Proof that nothing sketchy is going on

For n inputs, we choose randomly $n-1$ x_i's and set the remaining one in such a way that: $\sum_i C_{in,i}(x_i', a_i) - \sum_j C_{out,j}(y_j, b_j) = 0$

Important note

Note that here, it's **plainly** 0 (in Ed25519), not a **commitment** to zero. Thanks to that miners can verify that sum of the inputs is equal to the sum of the outputs.

The journey of Monero transaction - Ring Confidential Transactions

What we've done for now:

1. We created outputs with one-time addresses to our recipients, encoded amounts using public transaction key and one-time addresses
2. We included pseudo-commitments to our inputs, that together with outputs prove that input amount in total is equal to output amount in total of the transaction

The journey of Monero transaction - Ring Confidential Transactions

What we haven't done so far:

1. Range proof - without it we can create any amount of XMRs in any transaction (out of scope, complicated, not relevant in this case)
2. We did not prove it's our money we spend
3. We did not provide any proof that our commitments are valid
4. How do we know that tx has not been tampered with?
5. What about the fee for a miner?

The journey of Monero transaction - Miner's Fee

What we haven't done so far: Let's comment on the fee first.

The journey of Monero transaction - Miner's Fee

What we haven't done so far: Let's comment on the fee first.

Including fee in the transaction

Fee f for the miner is mandatory (the amount depends on a couple variables). It is made public in the transaction, the commitment is not masked - fG and two conditions must be met:

$$\sum_i a_i - \sum_j b_j - f = 0$$

and

$$\sum_i C'_{in,i} - \sum_j C_{out_j} - fG = 0$$

(we have that already).

The journey of Monero transaction - Ring Confidential Transactions

Three issues that we have to deal with yet:

1. We did not prove it's our money we spend
2. We did not provide any proof that our commitments are valid
3. How do we know that tx has not been tampered with?

The journey of Monero transaction - Ring Confidential Transactions

Three issues that we have to deal with yet:

1. We did not prove it's our money we spend
2. We did not provide any proof that our commitments are valid
3. How do we know that tx has not been tampered with?

Plus one more, very important for blockchain:
How to prevent double-spending?

The journey of Monero transaction - Ring Confidential Transactions

Three issues that we have to deal with yet:

1. We did not prove it's our money we spend
2. We did not provide any proof that our commitments are valid
3. How do we know that tx has not been tampered with?

Plus one more, very important for blockchain:
How to prevent double-spending?
The Multilayer Linkable Spontaneous Anonymous Group (**ML-SAG**) signatures.

MLSAG (slightly simplified)

We define a ring of signers, where π is our secret index and we know private keys for $\{K_{\pi,\mathbf{1}}, K_{\pi,\mathbf{2}}\}$

$$\mathcal{R} = \{\{K_{\mathbf{1},\mathbf{1}}, K_{\mathbf{1},\mathbf{2}}\}, \{K_{\mathbf{2},\mathbf{1}}, K_{\mathbf{2},\mathbf{2}}\}, \ldots, \{K_{\pi,\mathbf{1}}, K_{\pi,\mathbf{2}}\}, \ldots, \{K_{\boldsymbol{n},\mathbf{1}}, K_{\boldsymbol{n},\mathbf{2}}\}\}$$

For j = 1,2 compute *key images*:

$$\tilde{K_{\pi,\boldsymbol{j}}} := \tilde{K_{\boldsymbol{j}}} = k_{\pi,\boldsymbol{j}}\mathcal{H}_{\boldsymbol{p}}(K_{\pi,\boldsymbol{j}})$$

Signature:

1. $\alpha_{\mathbf{1}}, \alpha_{\mathbf{2}} \in_{\boldsymbol{R}} \mathbb{Z}_{\boldsymbol{l}}$, $r_{\boldsymbol{i},\boldsymbol{j}} \in_{\boldsymbol{R}} \mathbb{Z}_{\boldsymbol{l}}$ for $i = 1, \ldots, \hat{\pi}, \ldots, n$ and j = 1,2
2. $c_{\pi+\mathbf{1}} = \mathcal{H}_{\boldsymbol{n}}(m, \alpha_{\mathbf{1}}G, \alpha_{\mathbf{1}}\mathcal{H}_{\boldsymbol{p}}(K_{\pi,\mathbf{1}}), \alpha_{\mathbf{2}}G, \alpha_{\mathbf{2}}\mathcal{H}_{\boldsymbol{p}}(K_{\pi,\mathbf{2}}))$
3. for $i = \pi + 1, \pi + 2, \ldots, n, 1, \ldots, \pi - 1$
   $c_{\boldsymbol{i}+\mathbf{1}} =$
   $\mathcal{H}_{\boldsymbol{n}}(m, r_{\boldsymbol{i},\mathbf{1}}G + c_{\boldsymbol{i}}K_{\boldsymbol{i},\mathbf{1}}, r_{\boldsymbol{i},\mathbf{1}}\mathcal{H}_{\boldsymbol{p}}(K_{\boldsymbol{i},\mathbf{1}}) + c_{\boldsymbol{i}}(\tilde{K_{\mathbf{1}}}), r_{\boldsymbol{i},\mathbf{2}}G + c_{\boldsymbol{i}}K_{\boldsymbol{i},\mathbf{2}}, r_{\boldsymbol{i},\mathbf{2}}\mathcal{H}_{\boldsymbol{p}}(K_{\boldsymbol{i},\mathbf{2}}) + c_{\boldsymbol{i}}(\tilde{K_{\mathbf{2}}}))$
4. Define $r_{\pi,\boldsymbol{j}} = \alpha_{\boldsymbol{j}} - c_{\pi}k_{\pi,\boldsymbol{j}}$ for j =1,2
5. Output signature $\sigma(m) = (c_{\mathbf{1}}, r_{\mathbf{1},\mathbf{1}}, r_{\mathbf{1},\mathbf{2}}, \ldots, r_{\boldsymbol{n},\mathbf{1}}, r_{\boldsymbol{n},\mathbf{2}})$

Verification:

1. Check if $l\tilde{K_{\boldsymbol{j}}} = 0$
2. Compute $c_{\boldsymbol{i}}'$'s for i $= 2, \ldots, n, 1$ using $c_{\mathbf{1}}$ and check if $c_{\mathbf{1}} = c_{\mathbf{1}}'$.

The journey of Monero transaction - Ring Confidential Transactions - MLSAG

MLSAG in Monero:

1. For every input of our own, we pull n other inputs (address + commitment) from blockchain and we hide among those inputs - anonymity
2. Our input is connected with the one-time address and one-time spend key we derive from that address that we need to use to sign - proof we own the money
3. Signature uses key image that comes to a pond of spent key images - double spending prevented
4. Finally - we use the "private-key" for pseudo-commitments in the signature - commitments are valid

Borromean signatures

- a signature for m-digit number, for each digit two keys per user
- public keys $K_{i,j}$ for $i \leq m$
- the signer knows k_{i,π_i} - private key for K_{i,π_i} for each $i \leq m$

Signature:

1. for $i = q, \ldots, n$
   1.1 generate α_i at random
   1.2 $c_i := H(m, \alpha_i G, i, \pi_i)$
   1.3 for $j = \pi_i + 1, \ldots, m - 1$ choose $r_{i,j}$ at random and set
   $$c_{i,j+1} := H(m, r_{i,j}G - c_{i,j}K_{i,j}, i, j)$$
2. for $i = 1, \ldots, n$ choose $r_{i,m}$ at random and set
   $$c_1 := H(m, r_{1,m}G - c_{1,m}K_{1,m}, \ldots, r_{n,m}G - c_{n,m}K_{n,m})$$
3. for $i = 1, \ldots, n$
   3.1 for $j = 1, \ldots, \pi_i - 1$ generate $r_{i,j}$ at random and set
   $$c_{i,j+1} := H(m, r_{i,j}G - c_{i,j}K_{i,j}, i, j)$$
   (where as $c_{i,1}$ we take c_1
   3.2 $r_{i,\pi_i} := \alpha_i + k_{i,\pi_i}c_{i,\pi_i}$

the signature: $(c_1, \text{all numbers } r_{i,j})$

Real-life example of a transaction

Literature: Zero to Monero: Second Edition a technical guide to a private digital currency; for beginners, amateurs, and experts Published April 4, 2020 (v2.0.0) koe1, Kurt M. Alonso2, Sarang Noether3