Blockchains and Distributed Ledgers Lecture 07

Aggelos Kiayias

THE UNIVERSITY of EDINBURGH

Lecture 07

- Anonymity & Privacy in blockchain protocols.
  - Bitcoin and CoinJoin transactions.
  - Mix-nets
  - group and ring signatures.
    - Cryptonote/Monero
  - Zero-knowledge proofs & SNARKs
    - Zcash.

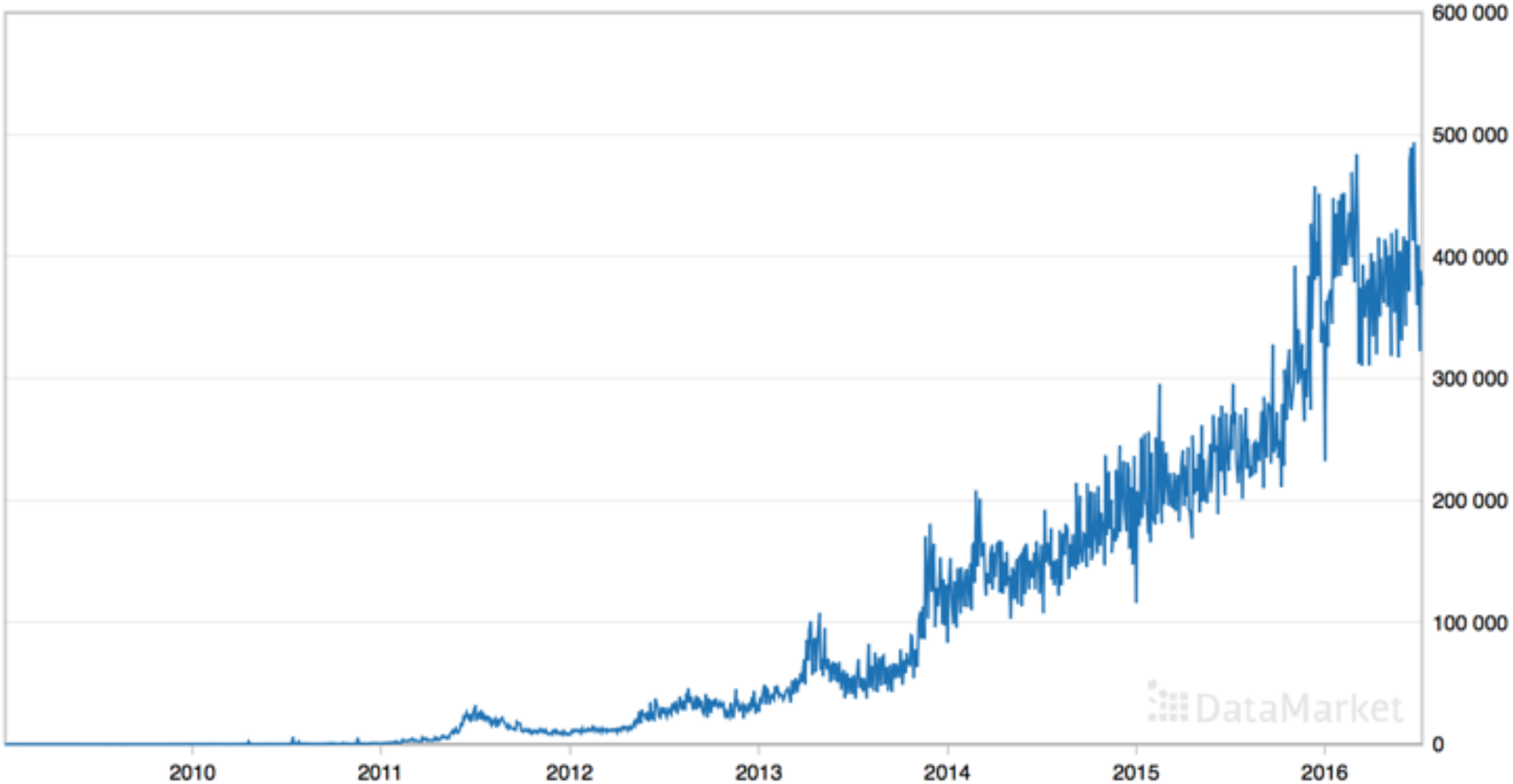
Pseudonymity

- Everything is public but identities are substituted by tags that are independently assigned to each identity.

Bitcoin

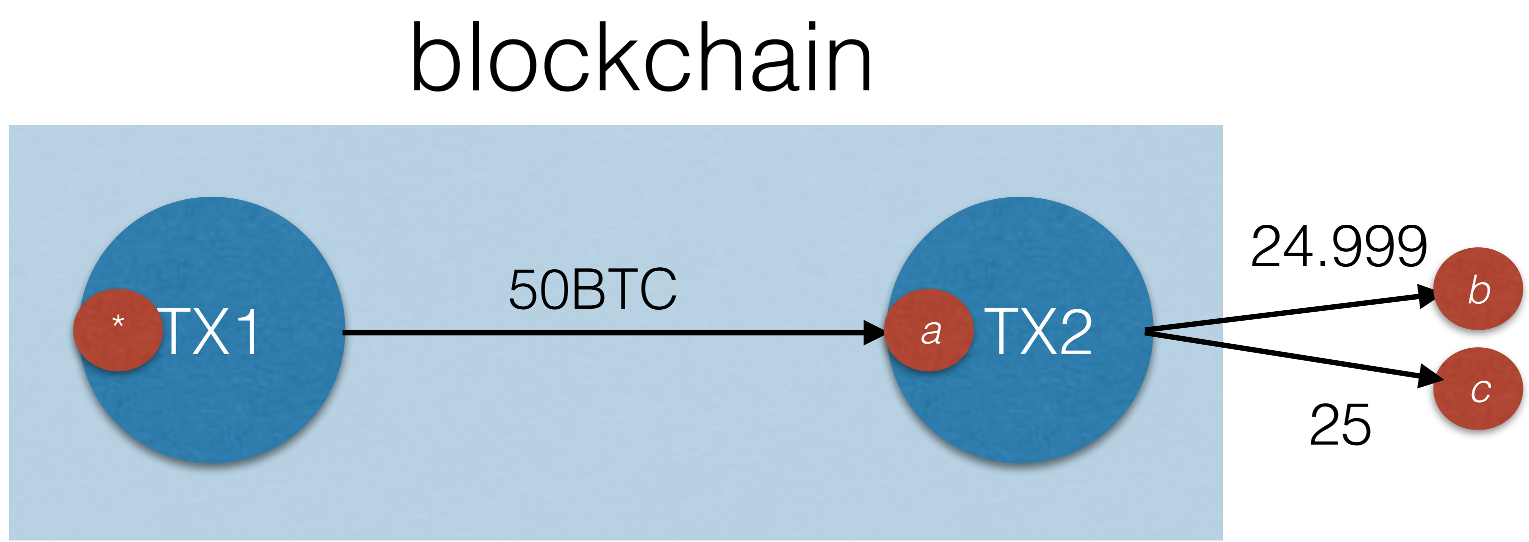
- Users can create accounts -practically- without cost and without association to previous accounts.

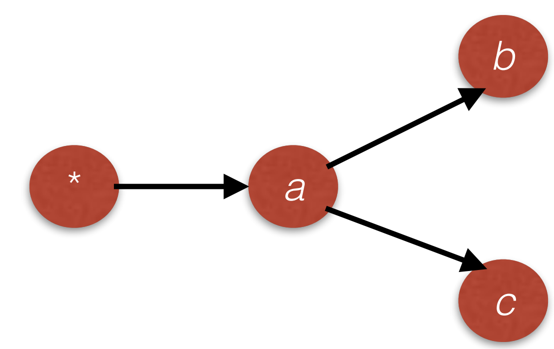
Accounts



graph from https://datamarket.com

Transaction Graph Analysis

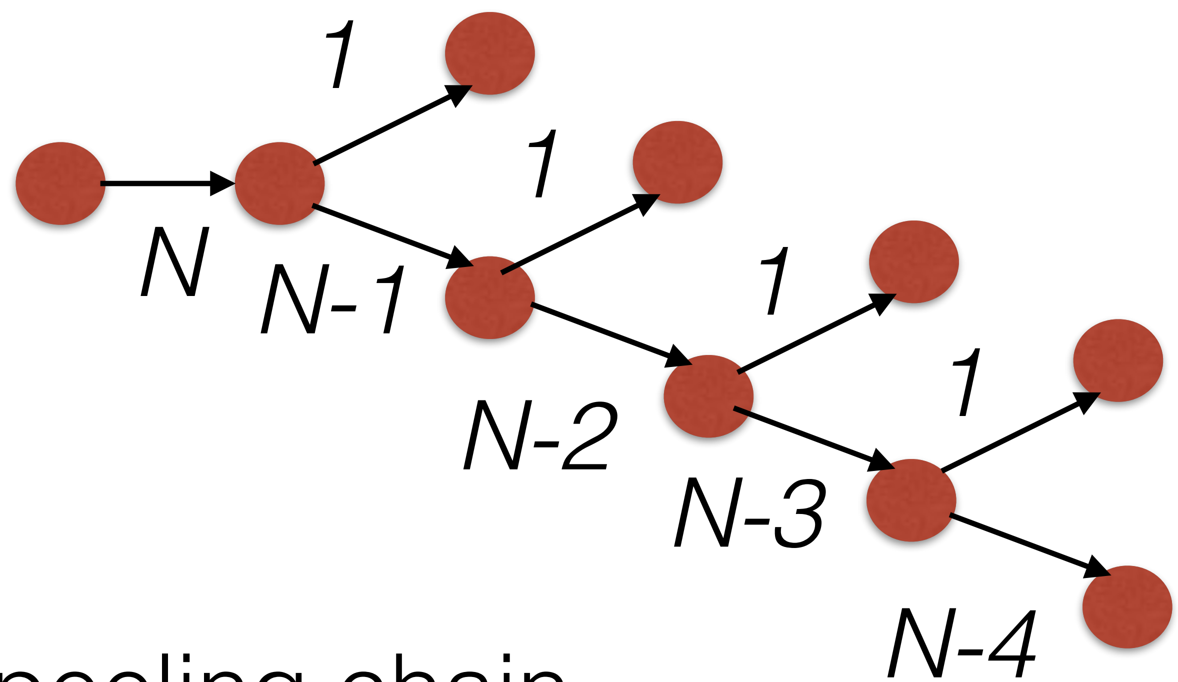




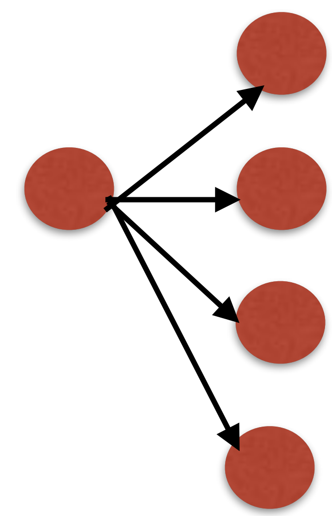
coinbase transaction

account *a* moves 50 BTC to accounts *b* and *c* (minus fees)

Common Behaviours



peeling chain

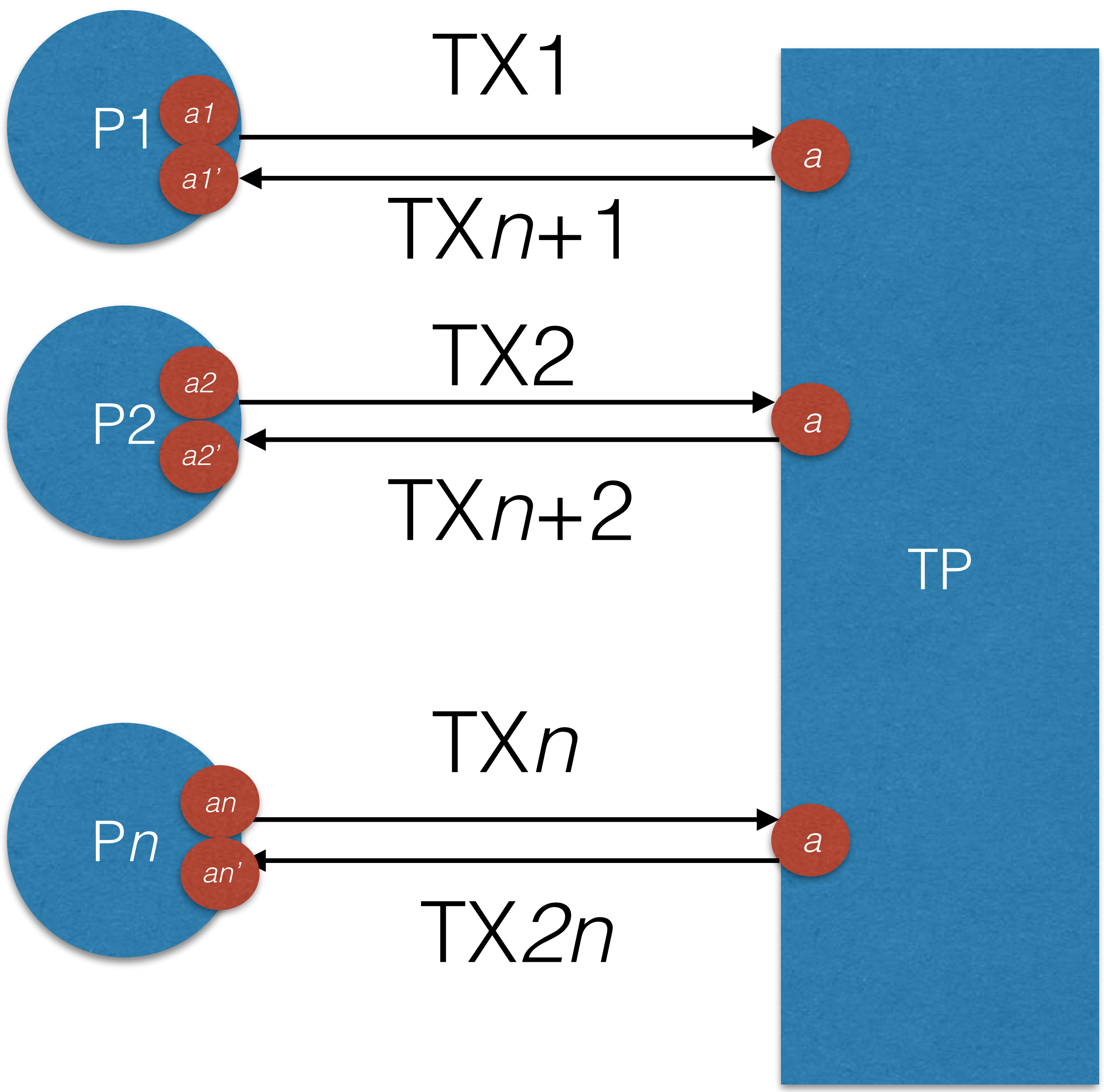


star

Fungibility & Privacy

- Coins are interchangeable.
- Since each “satoshi” has its whole history in the bitcoin blockchain, its fungibility is debatable.

Anonymizing Transactions

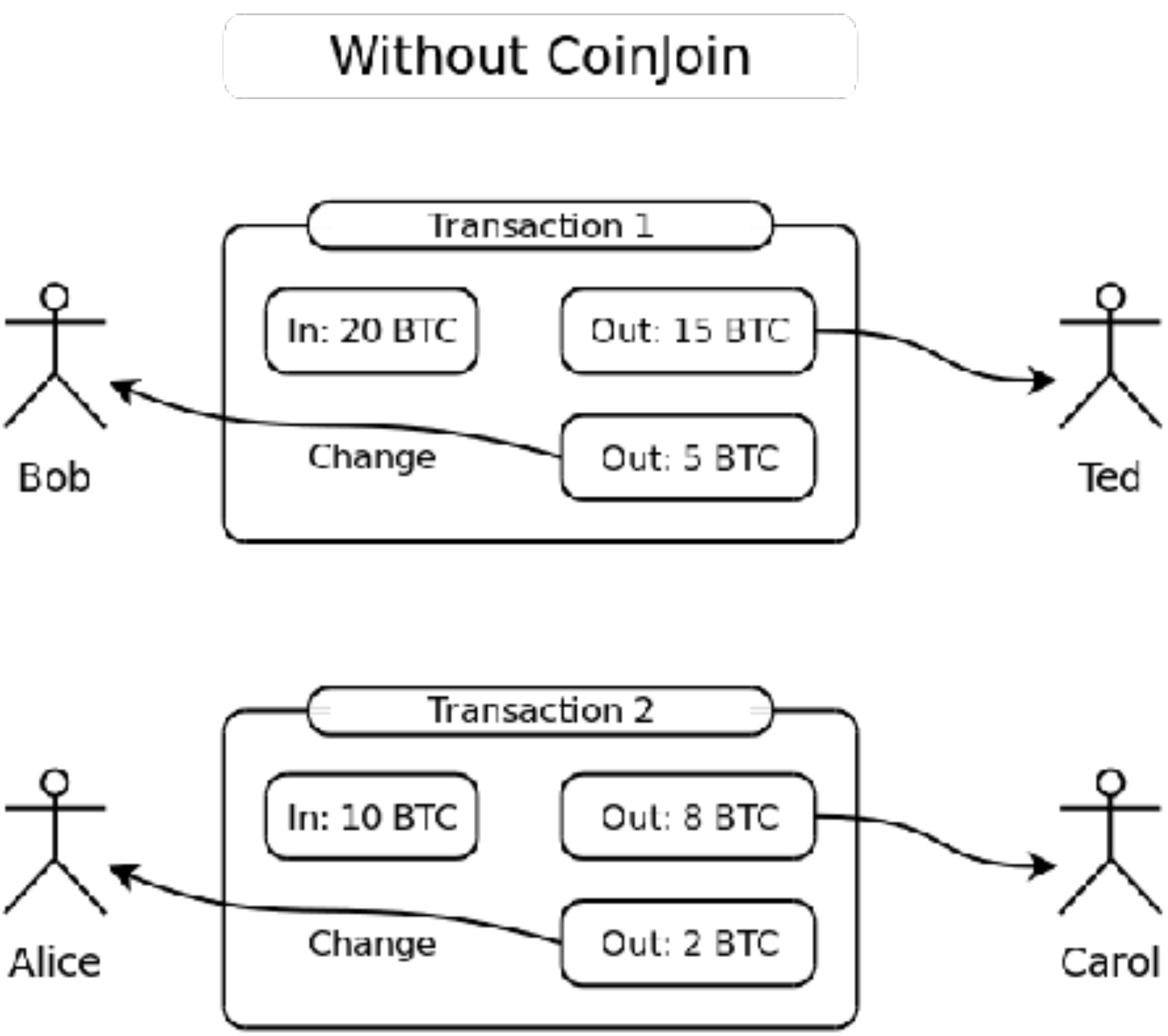


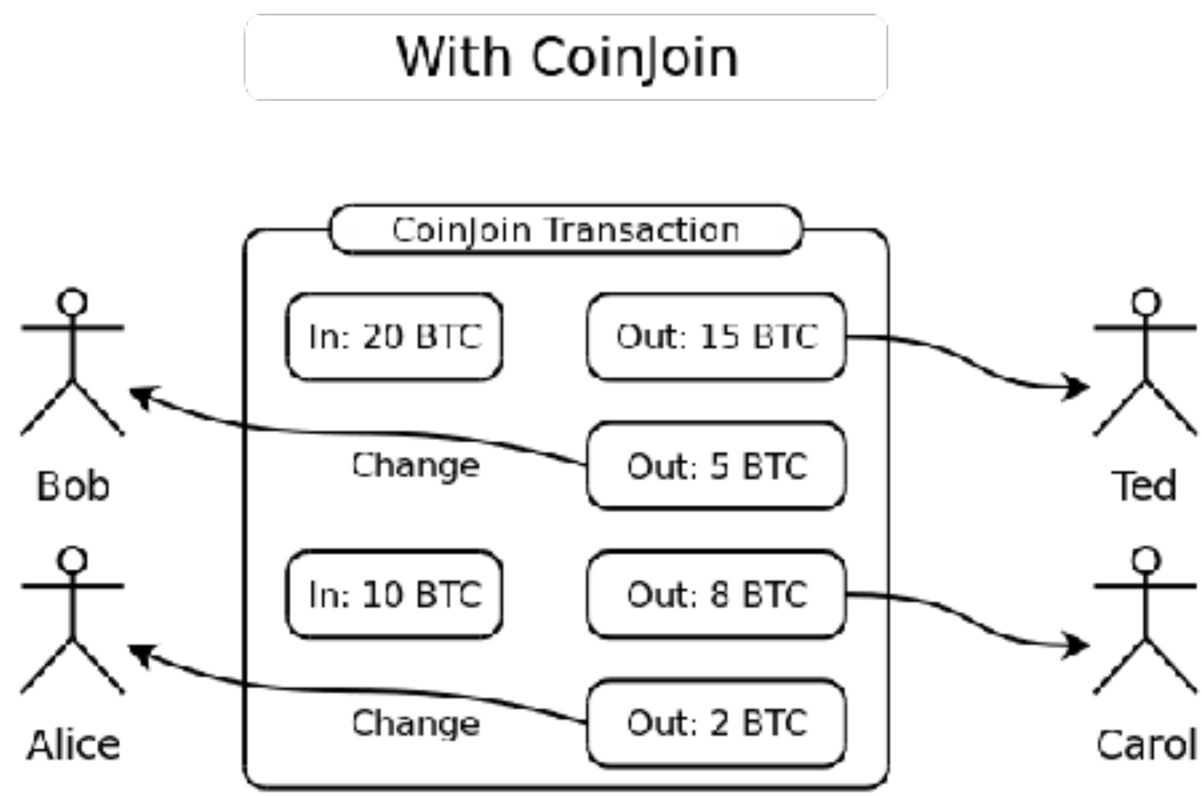
Anonymity set of size *n*

TP may disappear with the money

Multiple Input Transactions

Coinjoin transactions:





Multiple Input Transactions

- parties broadcast the a1',a2', …, a*n*' accounts.

- The *i*-th party broadcasts a signature from account a*i* to pay the account a'*i* from the set of accounts a1',a2', …, a*n*'. When all *n* signatures are broadcasted then the multiple input transaction can be posted on the blockchain.

- If any of the *n* parties abort the protocol the transaction cannot be validated.

- **Challenges**: how to ensure that the adversary cannot do a correlation between a*i* and a*i'*? in case of an abort how to restart the protocol without the offending party?

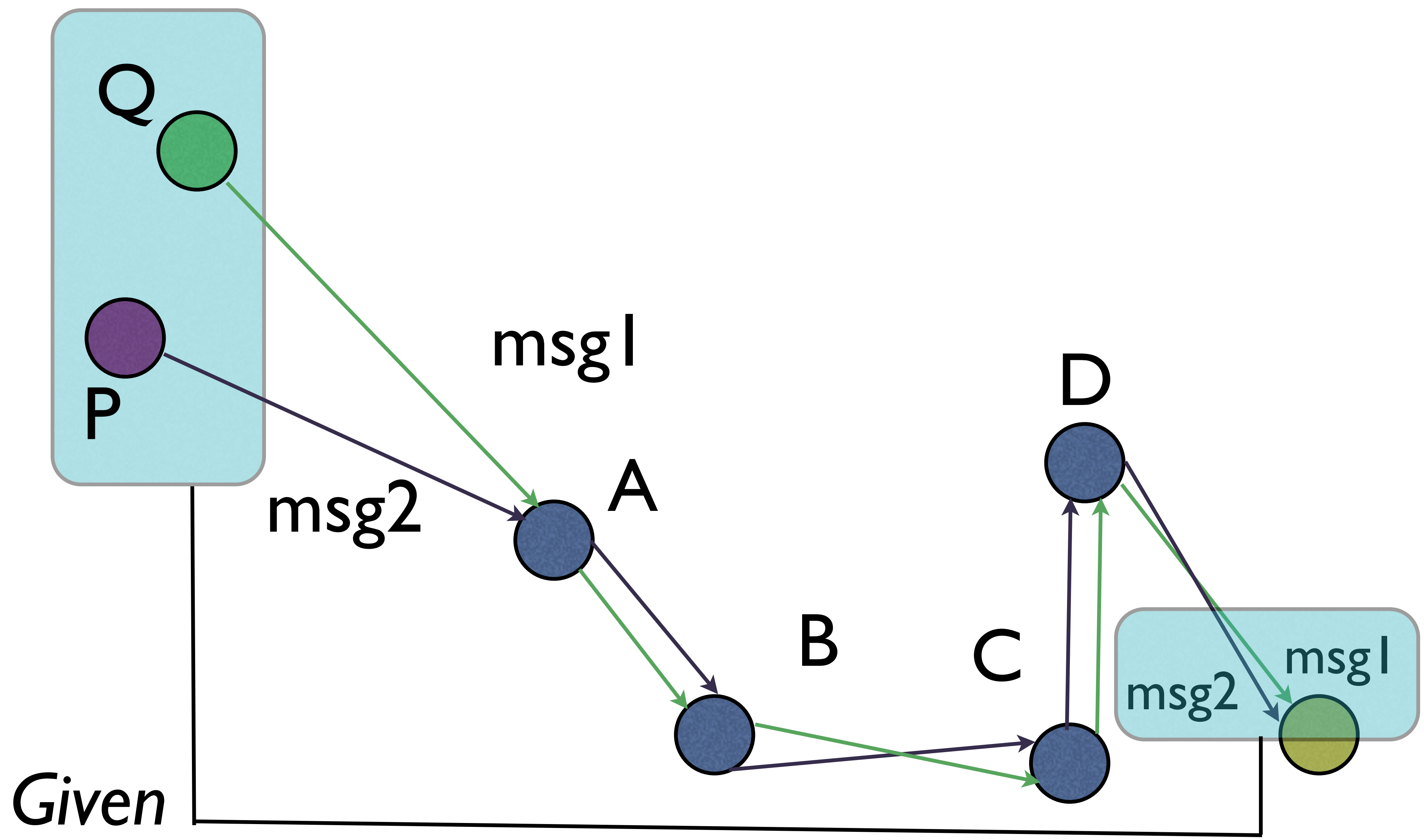
Passive vs. Active

- A “passive” adversary would just observe the transaction in the blockchain. In this case, an anonymity set of size *n* protects each participant.
- However, an “active" adversary participates in a protocol execution; the correlation between participants is apparent due to the broadcast.

Mix-net

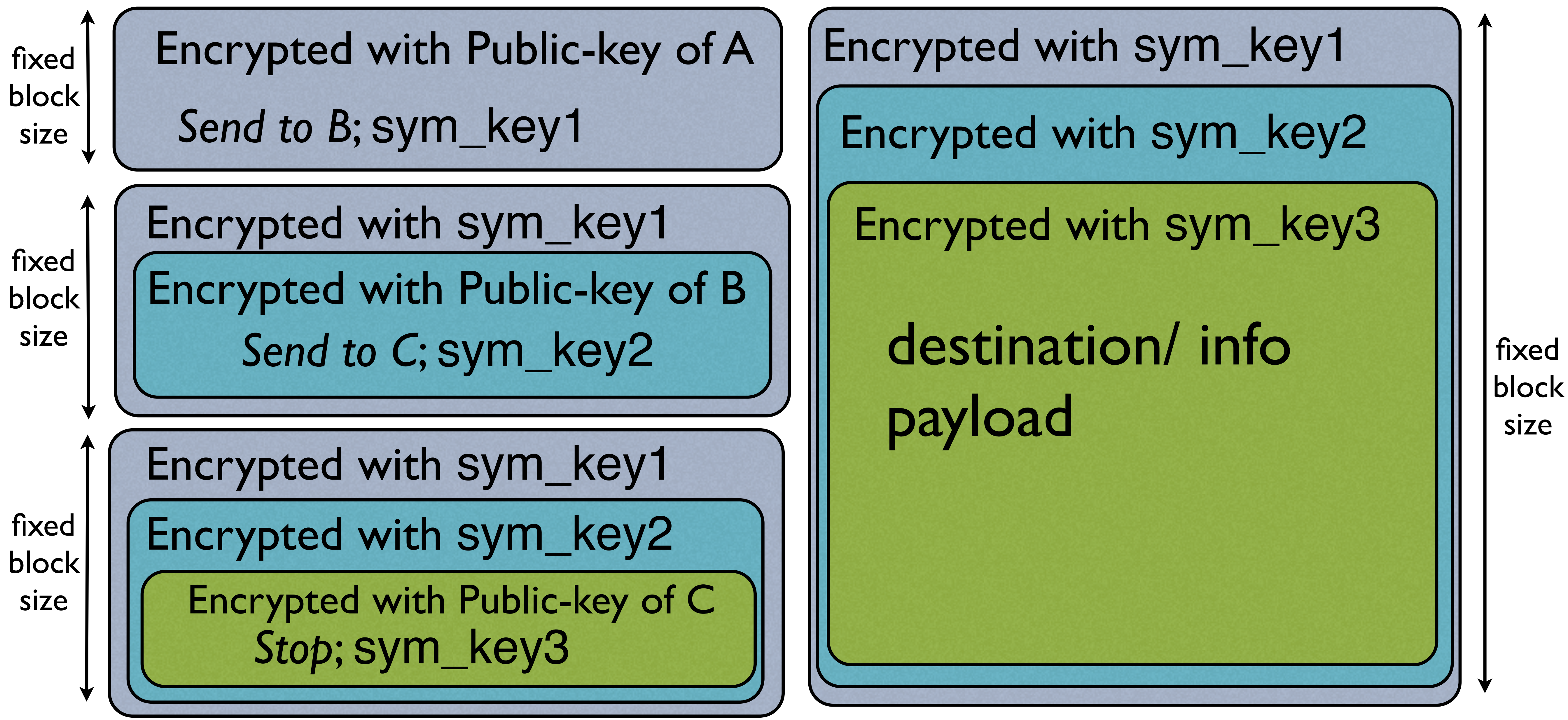
- Facilitates a sender-anonymous broadcast.
  - Decryption mix-nets.
  - Re-encryption mix-nets.

Mix-net

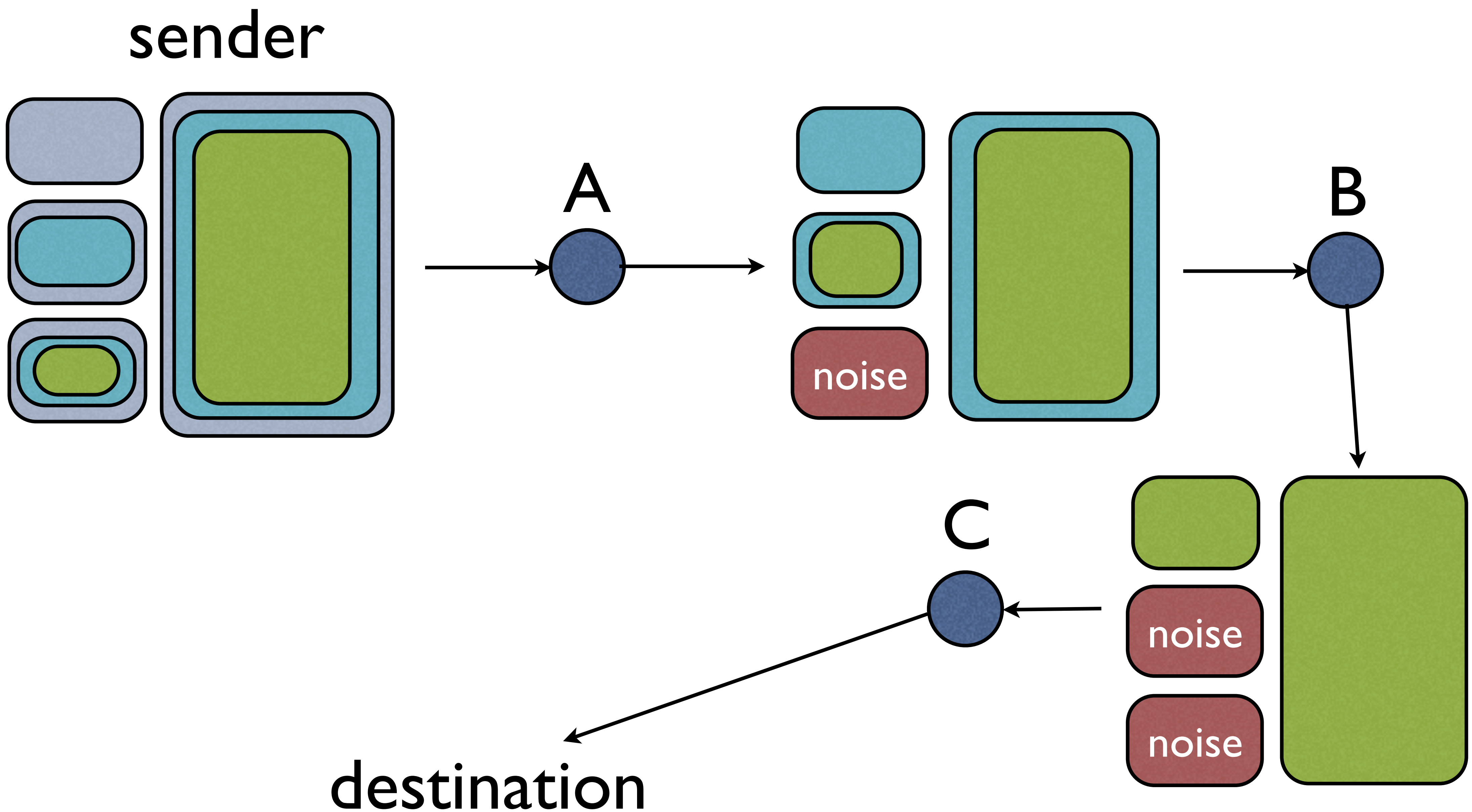


Not possible to relate whether P send msg1 or msg2 and similarly for Q (as long as there is one honest mix)

Decryption Mix-net



Routing via Mix-Net



Mix-Net for CoinJoin TX's

- Parties broadcast their public-keys (the association between public-keys and accounts a1,a2, …, an is public).

- Parties engage in a decryption mix-net in sequence so that the last party P*n* obtains the sequence of accounts a1',a2', …, a*n*'. P*n* broadcasts the accounts to all.

- Note that each step is performed by a designated party Pi, hence any abort can be attributed to that party. A repeat session may exclude the party Pi.

Re-encryption Mix-net

ciphertext re-rerandomization

$$\mathcal{E}(\rho; pk, M) \rightarrow \mathcal{E}(\rho'; pk, M)$$

ElGamal encryption:

$$\langle g, y = g^x \rangle$$

$$\mathcal{E}(\rho; pk, M) = \langle g^\rho, y^\rho M \rangle$$

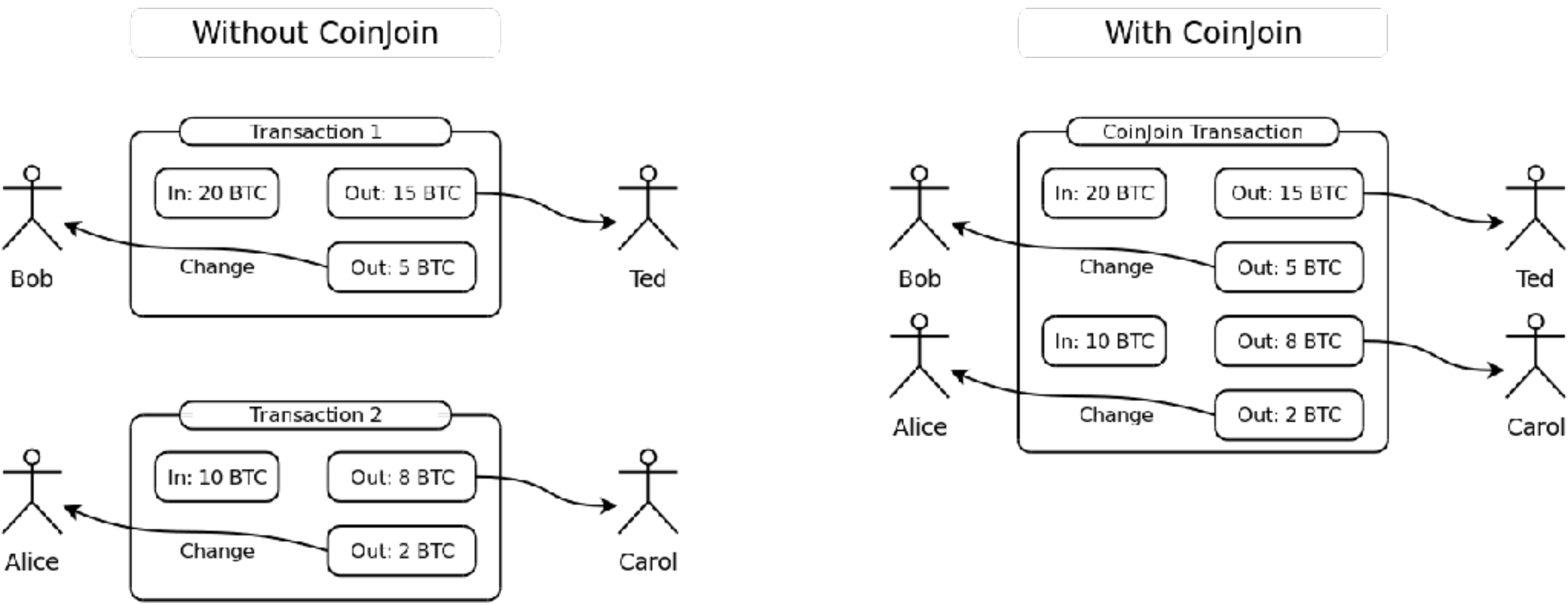
$$\mathcal{D}(sk, \langle G_1, G_2 \rangle) = G_1^{-x} G_2$$

Re-randomization:

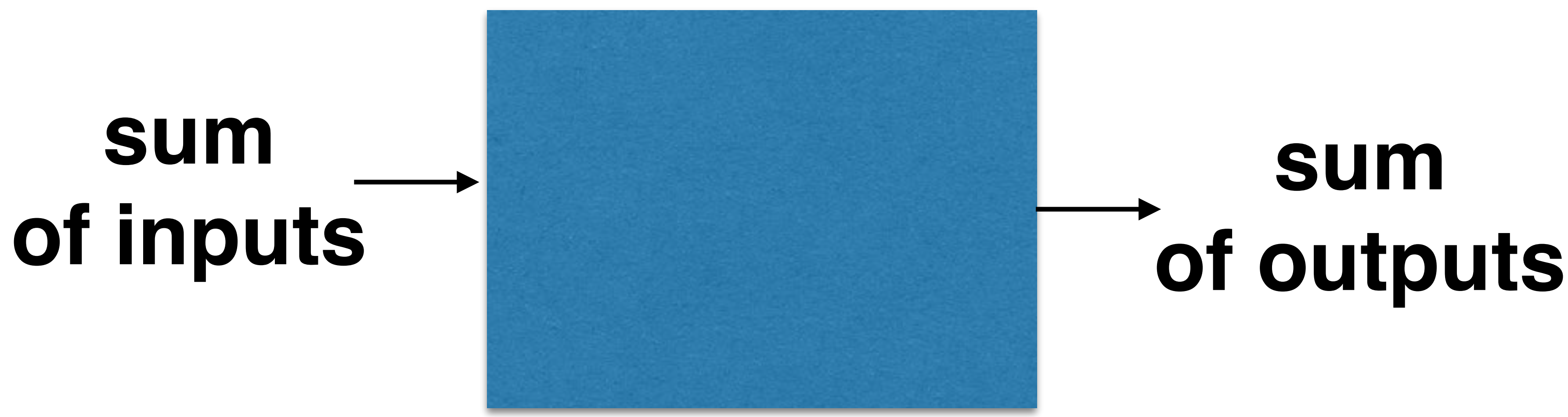
$$\langle G_1, G_2 \rangle \rightarrow \langle G_1 \cdot g^{\rho^*}, G_2 \cdot y^{\rho^*} \rangle, \rho^* \xleftarrow{R} \mathbb{Z}_m$$

Hiding Coin Balances

balances
are visible:

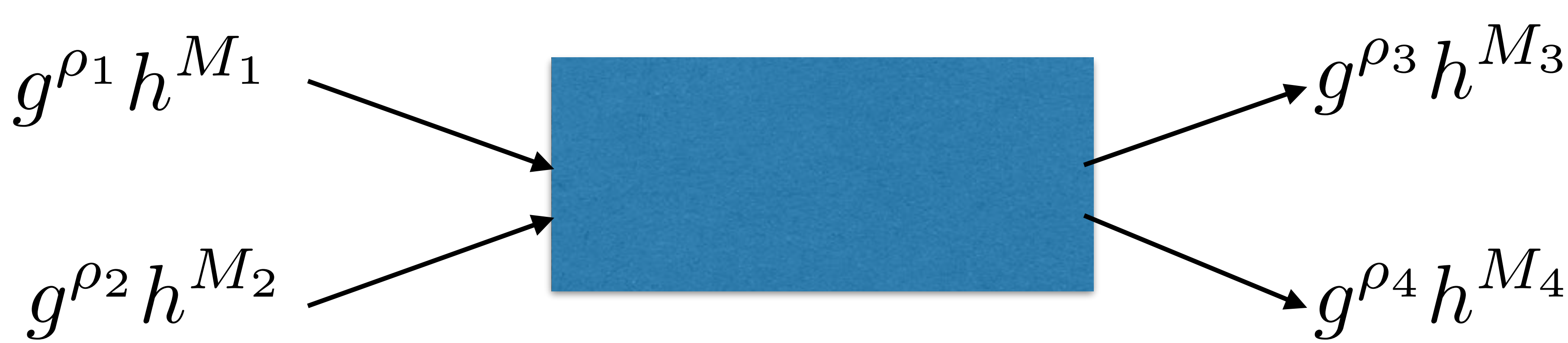


Using Commitment Schemes



Pedersen commitment

$$\mathrm{Commit}(\rho, M) = g^{\rho} h^{M}$$

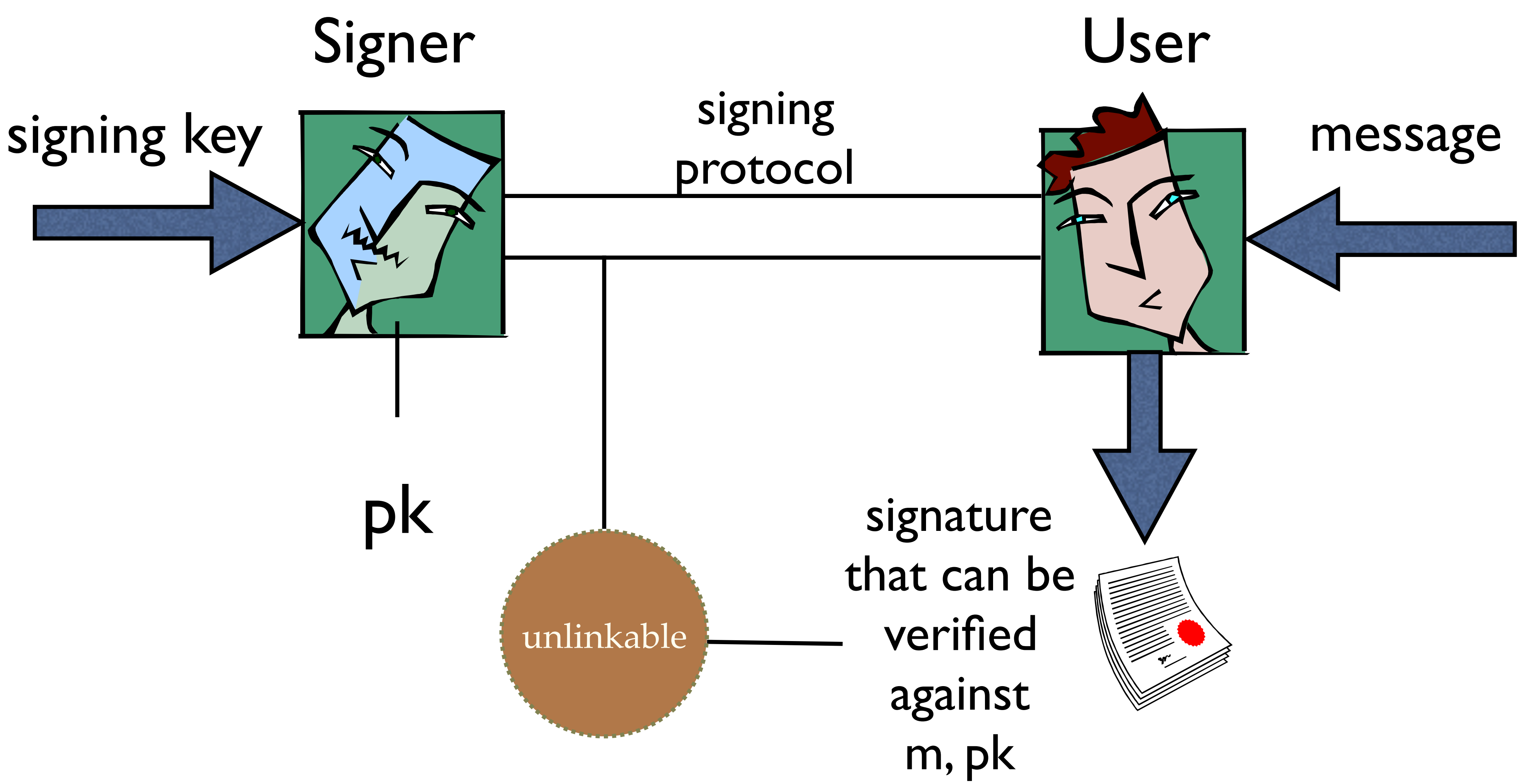


$$\psi_1 \psi_2 \psi_3^{-1} \psi_4^{-1} = g^{\rho_1 + \rho_2 - \rho_3 - \rho_4} h^{0}$$

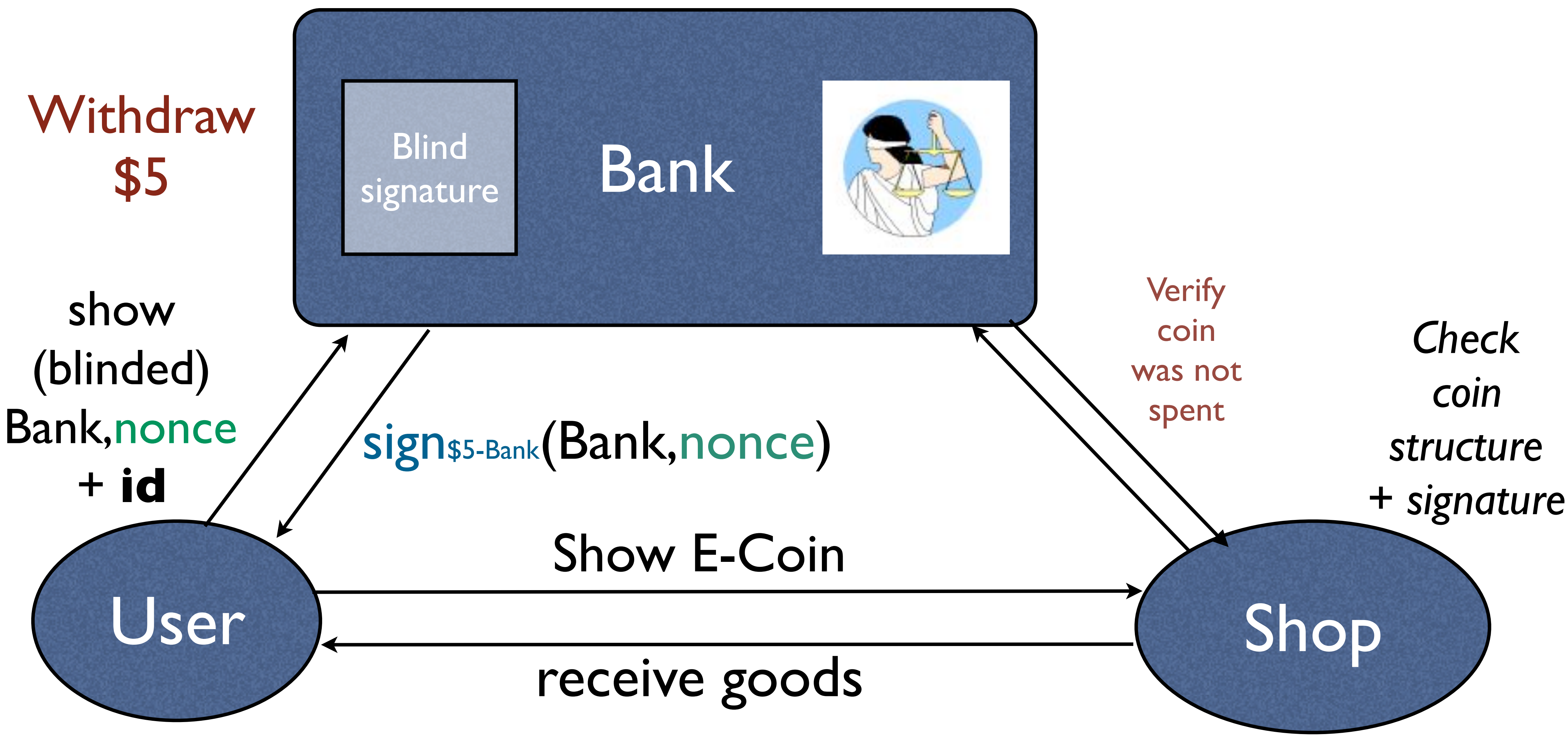
Challenges

- Coinjoin & similar techniques require coordination and message passing between the parties engaged in the transaction.
- Is it possible to improve that?

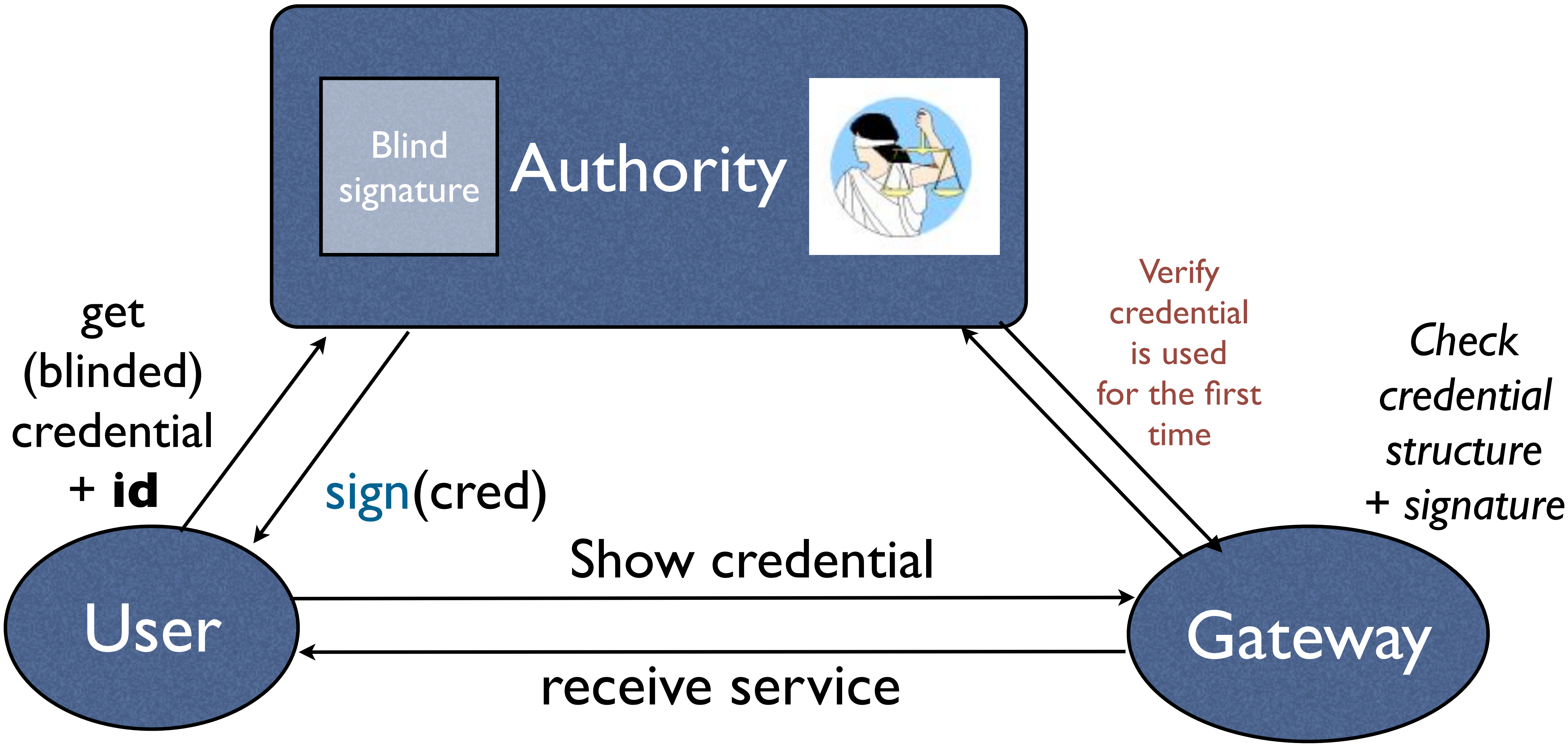
Blind Signatures



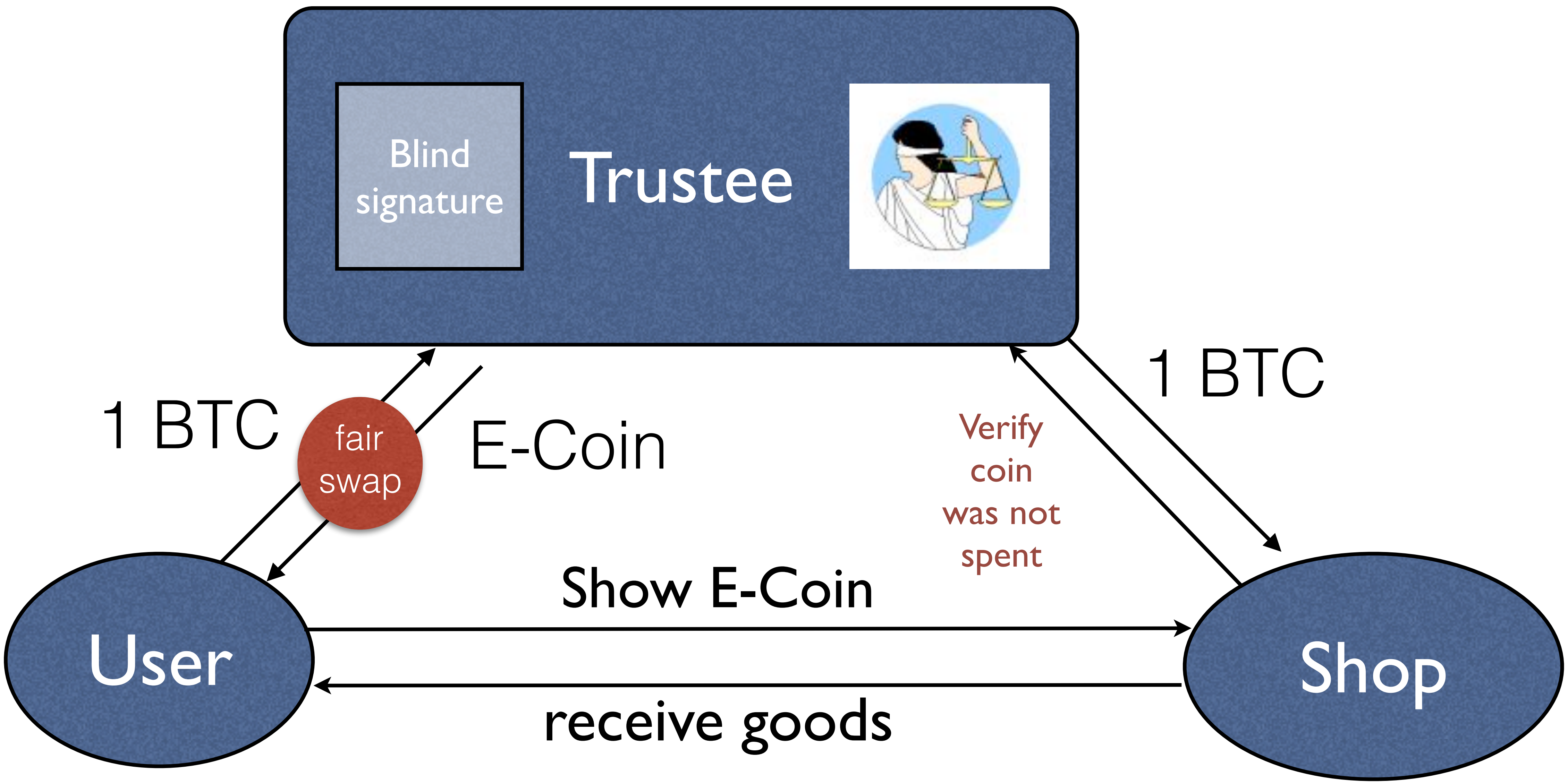
Chaum's E-cash



Anonymous Credentials



Anonymizing Bitcoin Payments via E-cash



Note: Trustee is trusted to honor its e-coins.

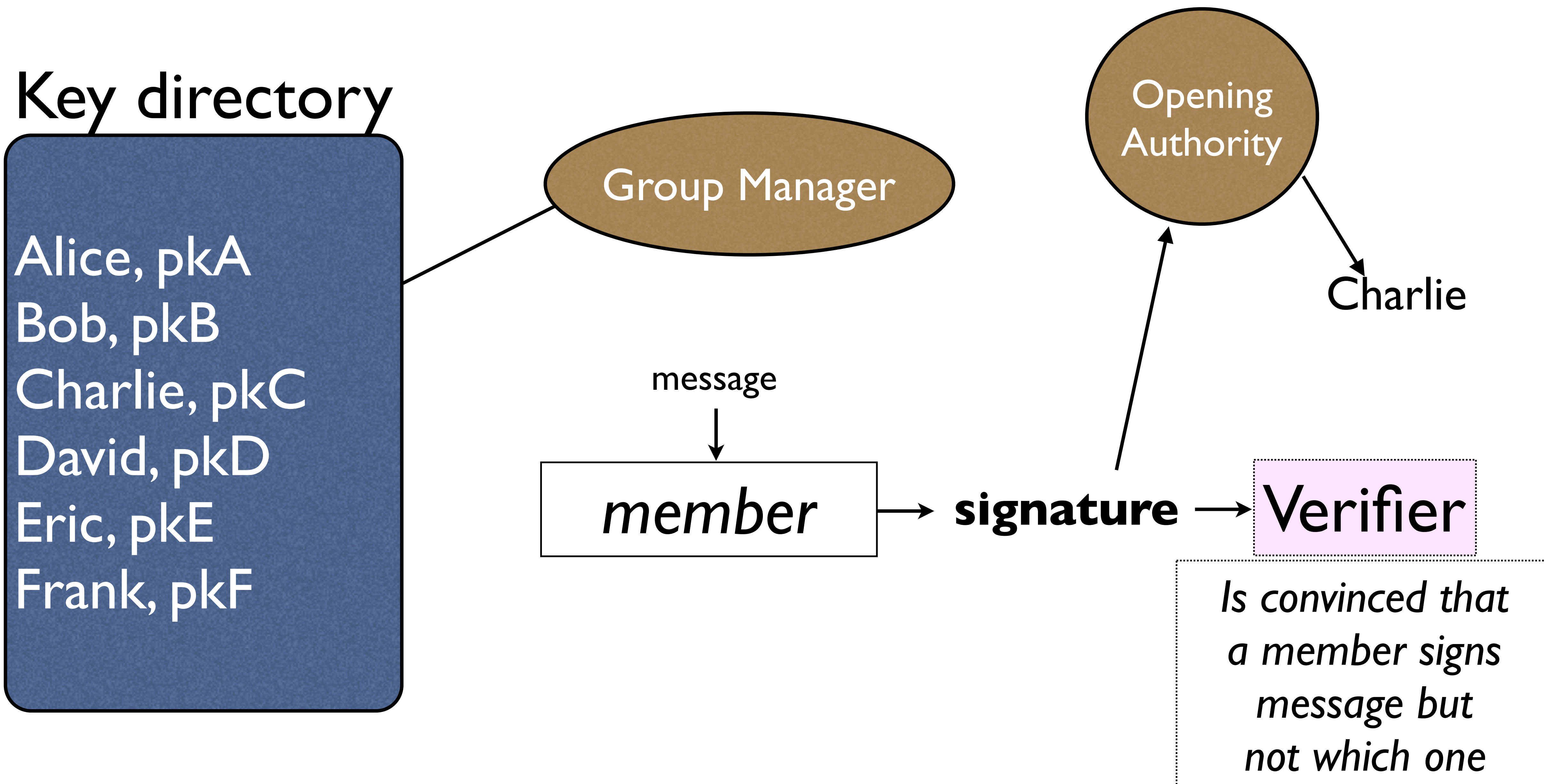
Challenges

- Using a trustee eases coordination requirements but introduces a single point of failure.
    - further enhance penalty mechanisms (along the lines of fair swaps of values) so that the trustee pays for any conceivable deviation.
    - or… use alternative techniques.

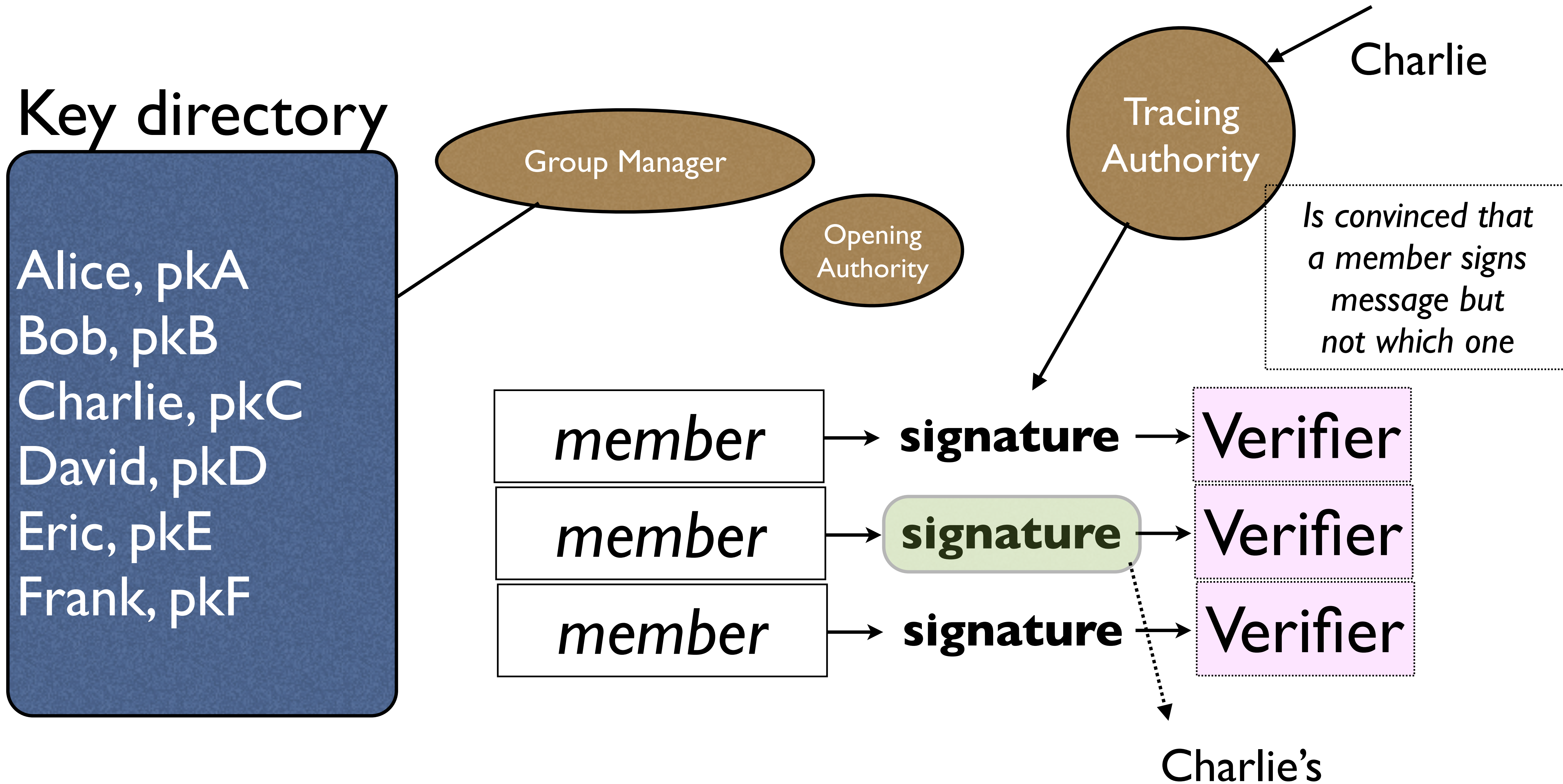
Anonymity Friendly Digital Signatures

- So far all digital signatures identify the signer.
    - Is it possible to hide the sender within a group?

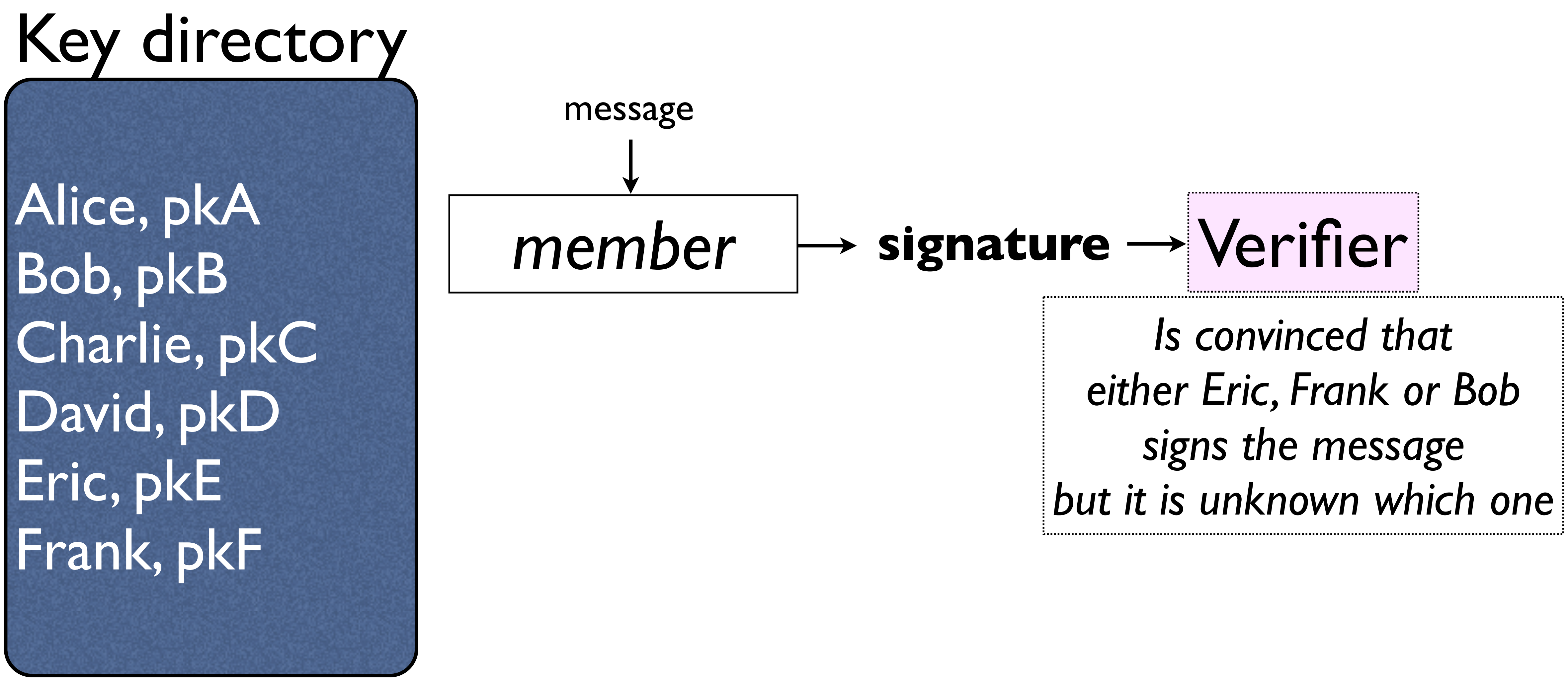
Group Signatures



Traceable Signatures



Ring Signatures

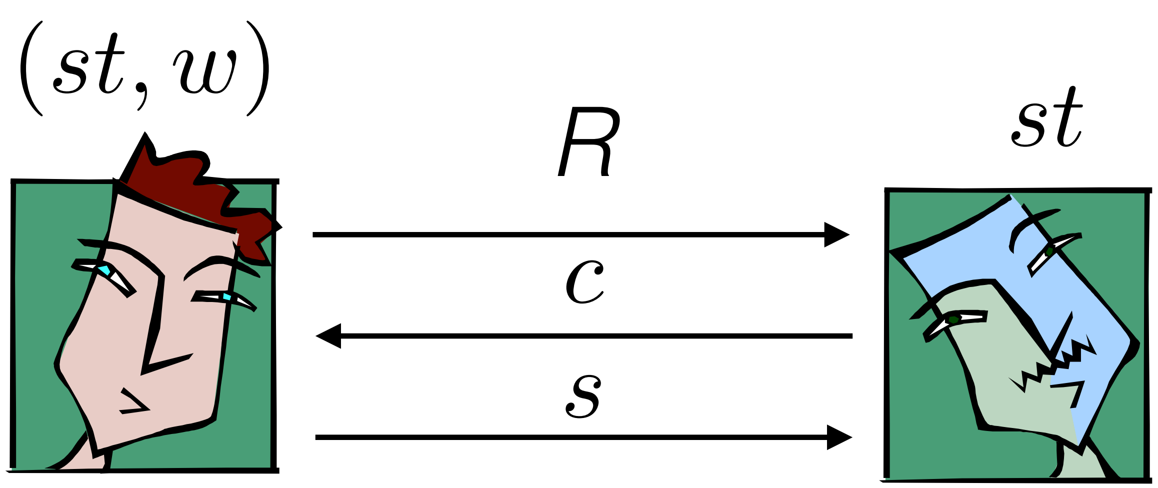


Constructing Signatures

- Start with a 3-move, public-coin, Zero-Knowledge Proof.
- Apply the Fiat-Shamir heuristic to make it a signature.

Zero-Knowledge Proofs

- 3-move, public-coin. Prove that $(st, w) \in R$



accepts or rejects

Properties:

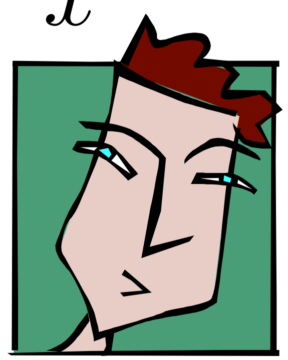
Completeness
Soundness
Zero-knowledge

Fiat-Shamir heuristic: $c \stackrel{?}{=} H(st, R, M)$

Schnorr Proof

I know $\log_g(Y)$

x

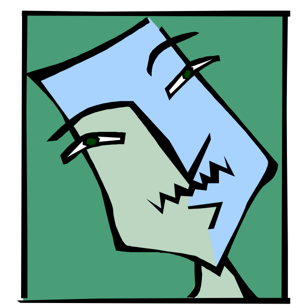


$$R = g^r$$

$$c$$

$$s = r + cx$$

$Y = g^x$



$$g^s \stackrel{?}{=} R \cdot Y^c$$

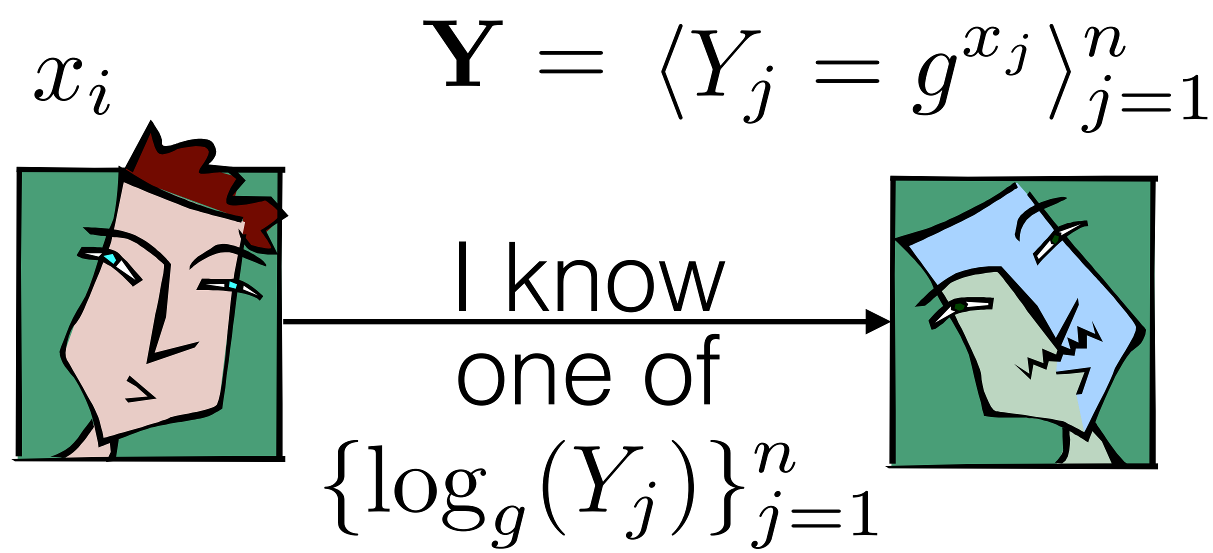
Schnorr Signature:

$$(c, s) = (H(Y, M, R), r + cx)$$

verification

$$c \stackrel{?}{=} H(Y, M, g^s Y^{-c})$$

OR Composition



Schnorr Ring Signature: pick random $d_j, s_j, j \neq i$

$(c, s_1, \ldots, s_n)$

$$d_i = c - \sum_{j \neq i}^{n} d_j$$

$R_j = g^{s_j} Y_j^{-d_j}$ for $j \neq i$

$c = H(\mathbf{Y}, M, R_1, \ldots, R_n)$

$s_i = r_i + d_i x_i \qquad R_i = g^{r_i}$

$$\sum_{j=1}^{n} d_j \stackrel{?}{=} H(\mathbf{Y}, M, g^{s_1} Y_1^{-d_1}, \ldots, g^{s_n} Y_n^{-d_n})$$

Monero/Cryptonote

- Uses “stealth” addresses and linkable ring signatures to provide better anonymity.
- For each payment an anonymity set is selected with accounts of the same monetary value.
- A ring signature is issued on behalf of that set, suitably restricted so that an account can only be used twice (linkable).

Stealth Addresses

- Addresses: $(A, B) = (g^a, g^b)$

To pay a party payment is issued to

$$R = g^r, P = g^{H(A^r)}B$$

Given a payment to (R^*, P^*)

To find out whether it is a payment made to $(A, B) = (g^a, g^b)$

test $P^* = g^{H((R^*)^a)}B$

Furthermore note that in such case:

$$\log_g(P^*) = H((R^*)^a) + b$$

Payments

- Consider if ring signatures are used to issue payments.

$$\begin{tikzcd} Y_1, Y_2 \arrow[d, "\text{knows } \log_g Y_1"] \\ (R, P) \end{tikzcd}$$

illegal

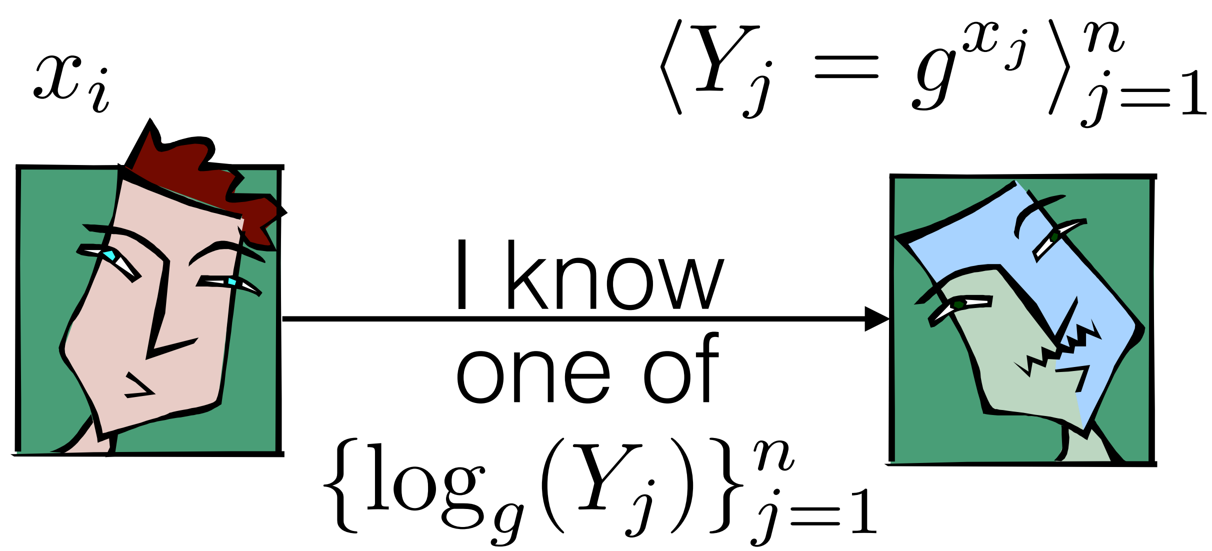
$$\begin{tikzcd} Y_1, Y_2 \arrow[d, "\text{knows } \log_g Y_1"] \\ (R, P) \end{tikzcd}$$

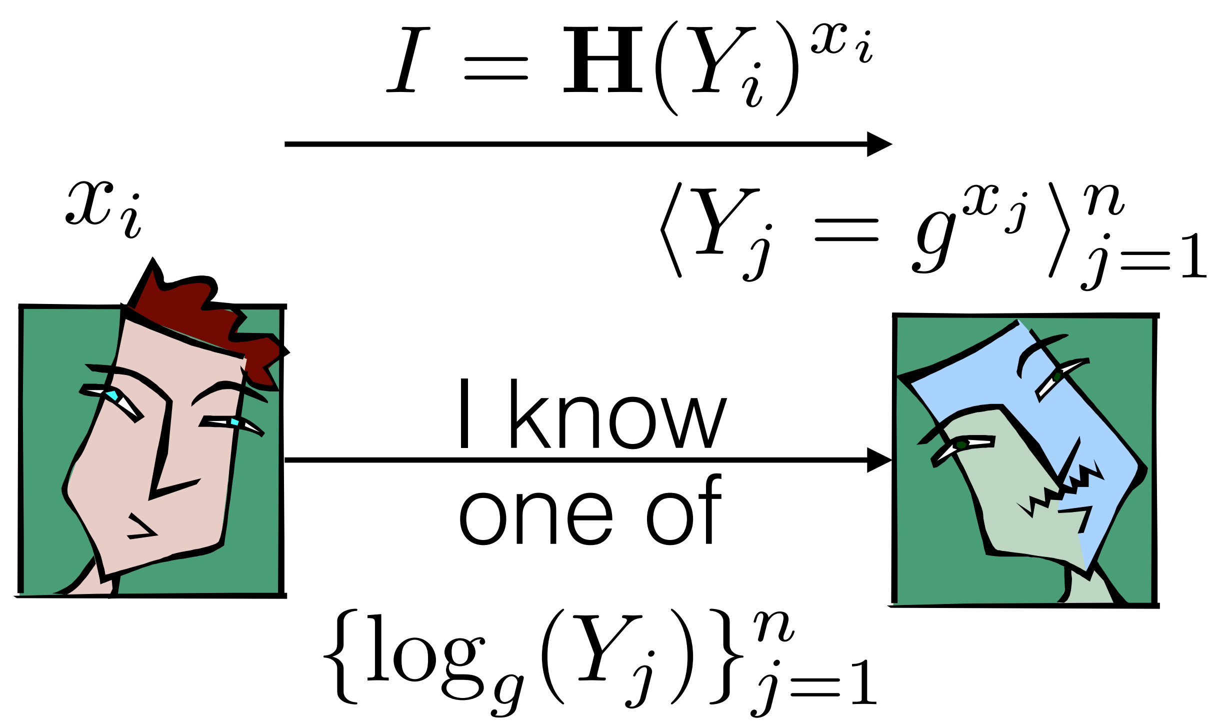
---

legal

$$\begin{tikzcd} Y_1, Y_2 \arrow[d, "\text{knows } \log_g Y_2"] \\ (R, P) \end{tikzcd}$$

Linkable Ring Signature





$\mathbf{H} : \{0,1\}^* \to \langle g \rangle$
$\mathbf{Y} = \langle Y_1, \ldots, Y_n \rangle$

and that value is also one of $\{\log_{\mathbf{H}(Y_j)}(I)\}_{j=1}^{n}$ for the same j

Construction

set $I = \mathbf{H}(Y_i)^{x_i}$

Ring Signature : pick random $d_j, s_j, j \neq i$

$(c, s_1, \ldots, s_n)$

$$d_i = c - \sum_{j \neq i}^{n} d_j$$

$R_j = g^{s_j} Y_j^{-d_j}$ for $j \neq i$

$R'_j = \mathbf{H}(Y_j)^{s_j} I^{-d_j}$

$s_i = r_i + d_i x_i$ $\quad R_i = g^{r_i}$

$R'_i = \mathbf{H}(Y_i)^{r_i}$

$c = H(\mathbf{Y}, M, R_1, \ldots, R_n, R'_1, \ldots, R'_n)$

---

verification

$$\sum_{j=1}^{n} d_j \stackrel{?}{=}$$

$$H(\mathbf{Y}, M, g^{s_1} Y_1^{-d_1}, \ldots, g^{s_n} Y_n^{-d_n}, \mathbf{H}(Y_1)^{s_1} I^{-d_1}, \ldots, \mathbf{H}(Y_n)^{s_n} I^{-d_n})$$

Linkability Argument

- Given a valid linkable ring signature the following can be inferred:
  - the signer knows one of the public-keys.
  - that one is committed in the *I* value which is **the same** each time the public-key is used.

Is Monero Anonymous?

- There is (potentially) more uncertainty in terms of transactions compared to a bitcoin-like blockchain.
- Nevertheless, it is not obvious how to quantify the level of anonymization.
- De-anonymization is feasible in a number of cases.

Increasing the anonymity set, I

- A larger anonymity set is most preferable.
- However in the techniques we have seen so far, transaction preparation work increases linearly with the anonymity set.
- **Ideal:** use the set of all possible unspent transaction outputs.

Increasing the anonymity set, II

$$\langle \rho, sn, \psi = \underline{\mathrm{Commit}(\rho, sn)} \rangle$$

public

The commitment value is associated with a deposit to the ledger (“minting” a coin for $1).

Spending a coin, requires announcing the *sn* and proving that it was committed before in the ledger; (withdrawing $1)

$$\exists i : \psi_i = \mathrm{Commit}(\rho, sn)$$

existential quantifier over all commitments in the blockchain

Increasing the anonymity set, III

Organize all commitments and serial numbers in a Merkle tree.

Prove that there is a leaf in the Merkle tree that contains the commitment

$$\psi_i = \mathrm{Commit}(\rho, sn)$$

Statement representation and witness size logarithmic in the number of coins.

Challenges

- How is it possible to prove efficiently statement referring to the leaf of a Merkle tree?
  - a possible solution: use “ZK-snarks”
- Transferring a coin from one user to another is not properly specified (one cannot simply transfer ρ).

ZK-SNARKs

- Zero-knowledge succinct arguments of knowledge.
  - like zero-knowledge proofs, but with:
    - computational soundness.
    - succinctness: the proof size and the verifier's running time is efficient proportionally to the *statement* only.

Constructing ZK-Snarks

$$\exists w : R(x, w) = 1$$

- There exist a SNARK for any NP-relation R.
- The actual proof sizes are small (hundreds of bytes)
- Verification **does not** depend on the running time of *R.*

Zerocash System

$\langle a_{\mathbf{pk}}, v, s \rangle$

value

random

$(a_{\mathbf{pk}}, a_{\mathbf{sk}})$

account public/secret key

$$k = \mathrm{Commit}(\rho, a_{\mathsf{pk}} || s)$$

$$sn = \mathsf{PRF}^{\mathsf{sn}}_{a_{\mathbf{sk}}}(s)$$

$$\psi = \mathrm{Commit}(\rho', v || k)$$

coin : $\langle a_{\mathsf{pk}}, v, s, \rho, \rho', \psi)$

The double commitment enables verifying that the value *v* is properly encoded in the coin without revealing information about the owner

Zerocash "Pour" operation

given coin $\langle a_{\mathsf{pk}}, v, s, \rho, \rho', \psi)$

produce two new coins with values $v_1 + v_2 = v$

$a^1_{\mathsf{pk}}, a^2_{\mathsf{pk}}$

set $k_i = \mathrm{Commit}(\rho_i, a^i_{\mathsf{pk}} || s_i)$

$\psi_i = \mathrm{Commit}(\rho'_i, v_i || k_i)$

Reveal ψ_1, ψ_2 and prove that the Merkle tree has a commitment corresponding to a coin $\langle a_{\mathsf{pk}}, v, s, \rho, \rho', \psi)$ that is split properly and a_{sk} is known

End of lecture 07

- Next lecture
  - Permissioned distributed ledgers.