Ring Signatures
Monero

Oct. 14, 2019

Overview

- Privacy Hierarchy
- Monero
    - Secretly signing a transaction
    - Secretly receiving a transaction

Privacy Hierarchy

Everything open	
Pseudonymous, amount open	**Bitcoin**

Privacy Hierarchy

Everything open	
Pseudonymous, amount open	**Bitcoin**
Pseudonymous, amount secret	**MimbleWimble**

Privacy Hierarchy

Everything open	
Pseudonymous, amount open	**Bitcoin**
Pseudonymous, amount secret	**MimbleWimble**
Pseudonymous, amount open miner/coordinator does not know input output	**CoinJoin**

Privacy Hierarchy

Everything open	
Pseudonymous, amount open	**Bitcoin**
Pseudonymous, amount secret	**MimbleWimble**
Pseudonymous, amount open miner/coordinator does not know input output	**CoinJoin**
amount open, transaction un-linkable Secret receive/secret sending	**Monero**

Privacy Hierarchy

Everything open	
Pseudonymous, amount open	**Bitcoin**
Pseudonymous, amount secret	**MimbleWimble**
Pseudonymous, amount open miner/coordinator does not know input output	**CoinJoin**
amount open, transaction un-linkable Secret receive/secret sending	**Monero**
amount secret, transaction un-linkable	**ZCash, ZeroCash, ZeroCoin**

Monero

- Anonymously receiving
  - Address is created from public key
  - Not possible to link target address to public key

- Anonymous sending
  - Owner selects an anonymity set $\mathcal{S} = \{T_1, T_2, \ldots, T_n\}$
  - Shows that one in $\mathcal{S}$ signed the transaction

Anonymous Sending

- Sending a transaction is one with a

one-time ring-signature

- o*ne-time*: signing a transaction more than once with the same key can be detected: Prevents double spend

- *ring-signature*: Given a set of public keys, show that one of the corresponding private keys signed it

Ring signature

tx 2:
(__, pk)

tx 4:
(__, pk)

tx 10:
(__, pk)

tx 9:
(__, pk)

tx 1:
(__, pk)

tx 3:
(__, pk)

tx 5:
(__, pk)

tx 12:
(__, pk)

tx 8:
(__, pk)

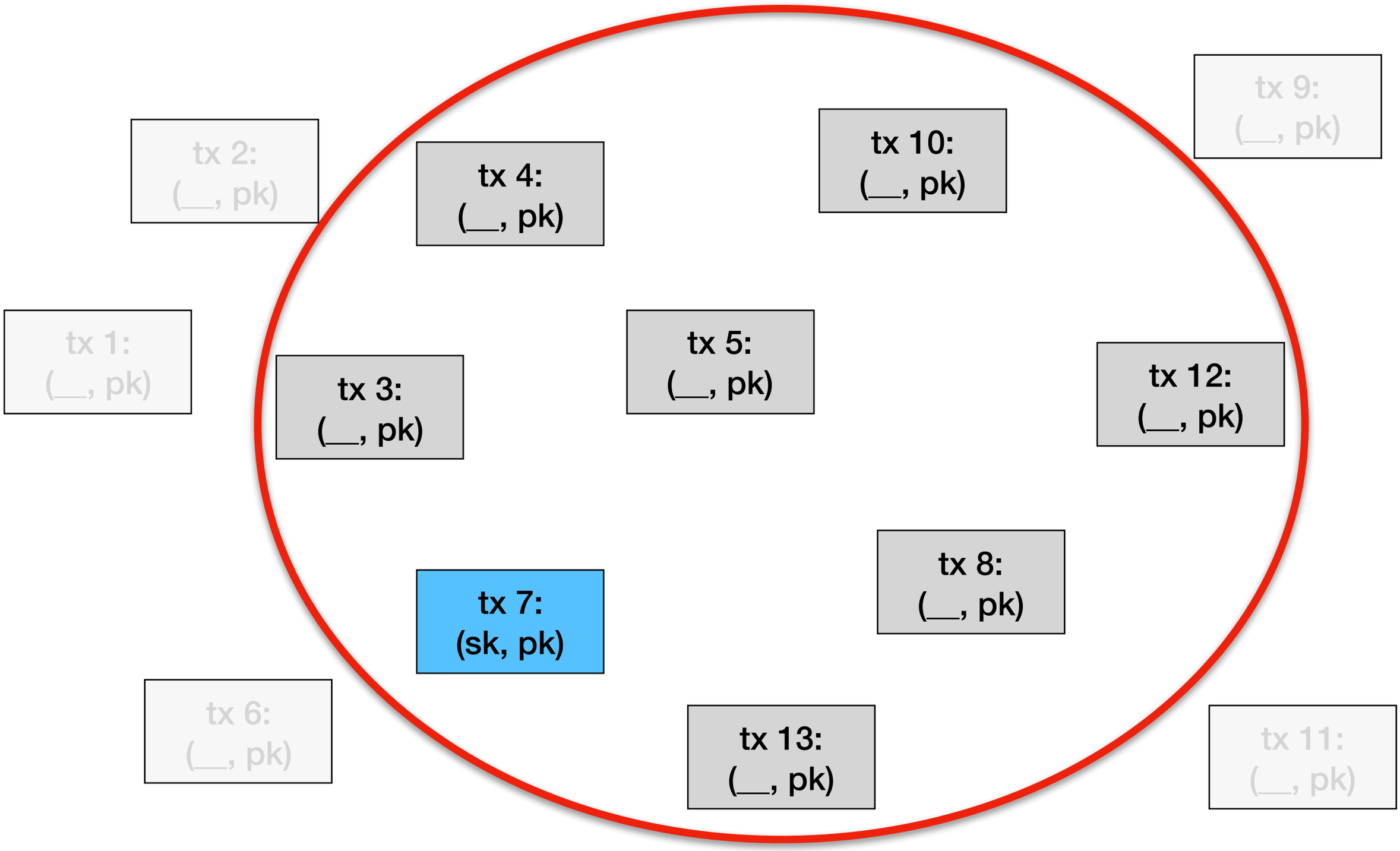
tx 7:
(sk, pk)

tx 6:
(__, pk)

tx 13:
(__, pk)

tx 11:
(__, pk)

Ring signature

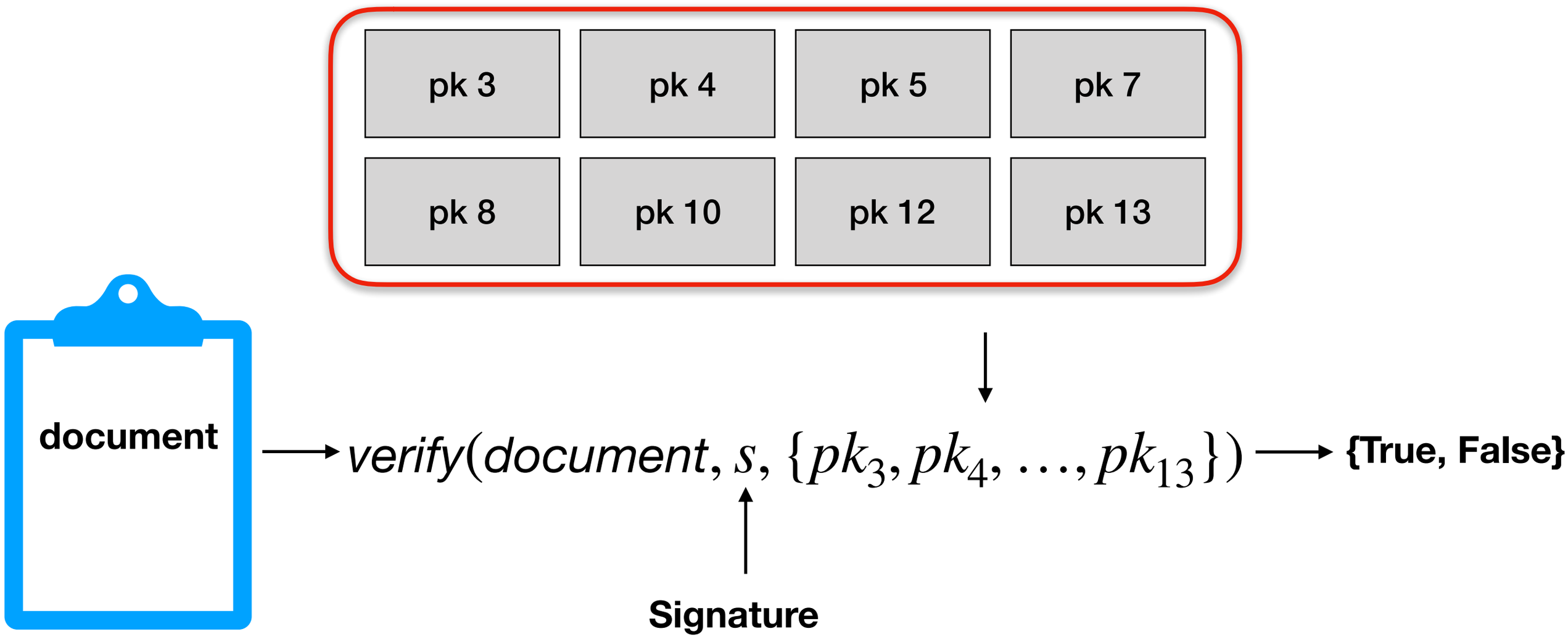


Ring signature

tx 3: (__, pk)	tx 4: (__, pk)	tx 5: (__, pk)	**tx 7: (sk, pk)**
tx 8: (__, pk)	tx 10: (__, pk)	tx 12: (__, pk)	tx 13: (__, pk)

document → $sign(document, sk, \{pk_3, pk_4, \ldots, pk_{13}\})$ → **Signature**

Ring signature



Ring Signature

- A signature that uses an anonymity set of public keys
- The signer hides his/her identity in this set
- Verifier can check whether someone of that set signed it
- Verifier cannot identify who exactly signed it
    - Possible application: whistleblowing

Single ECC signature

- common group element G
- public/private key $P = pG$

- Signature are 2 numbers

$$s = (c, d), \{c, d\} \in \mathbb{Z}$$

- verify via

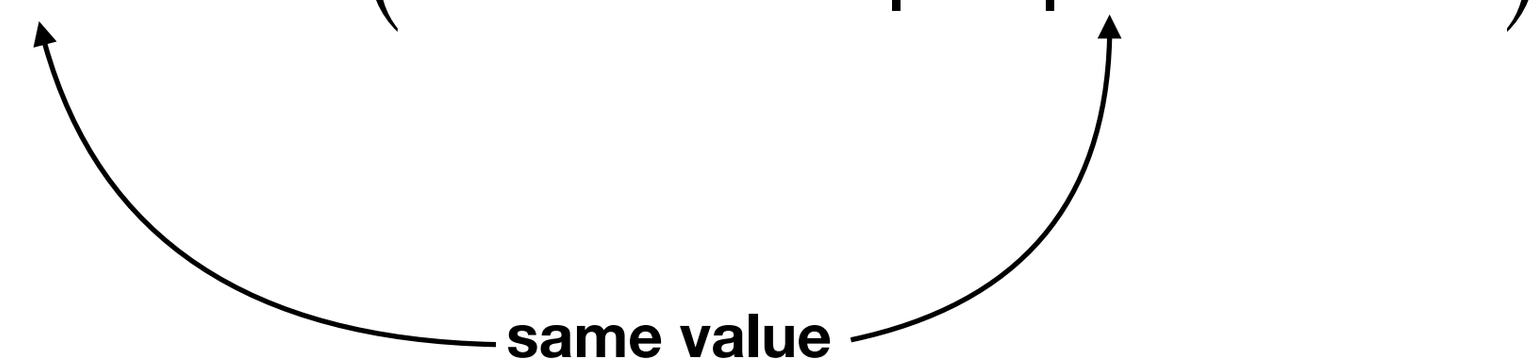
$$c = \mathscr{H}\left(\text{document} \,|\, G \,|\, cP + dG\right)$$

Single ECC signature

- common group element G, public/private key $P = pG$
- $s = (c, d), \{c, d\} \in \mathbb{Z}$

- verify via

$$c = \mathscr{H}\left(\text{document} \,|\, G \,|\, cP + dG\right)$$



Single ECC signature

- common group element G
- public/private key $P = pG$
- sign:
  - random r
  - $c = \mathscr{H}\left(\text{document} \,|\, G \,|\, rG\right)$
  - $rG = (cp + d)G \Rightarrow d = r - cp$
    $c = \mathscr{H}\left(\text{document} \,|\, G \,|\, cpG + dG\right)$
    $c = \mathscr{H}\left(\text{document} \,|\, G \,|\, cP + dG\right)$

2 element ECC ring signature

- common group element G
- public keys $\{P_1, P_2\}$
- signature $s_{\text{ring}_2} = (c_1, c_2, d_1, d_2)$

$$c_1 + c_2 = \mathscr{H}\left(\text{document} \,|\, G \,|\, c_1 P_1 + d_1 G \,|\, c_2 P_2 + d_2 G\right)$$

- Signer knows one of the private keys of $\{P_1, P_2\}$

2 element ECC ring signature

- common group element G
- own public/private key $P_1 = p_1 G$, other person's key P_2
- sign:
- random r, c_2, d_2
- $c = \mathscr{H}\left(\text{document} \,|\, G \,|\, rG \,|\, c_2 P_2 + d_2 G\right)$
- $c_1 = c - c_2 \Rightarrow c_1 + c_2 = \mathscr{H}(\ldots)$
- $rG = \left(c_1 p_1 + d_1\right) G \Rightarrow d_1 = r - c_1 p_1$

$$c_1 + c_2 = \mathscr{H}\left(\text{document} \,|\, G \,|\, c_1 p_1 G + d_1 G \,|\, c_2 P_2 + d_2 G\right)$$

General ECC ring signature

- common group element G
- public keys $\{P_1, P_2, \ldots, P_n\}$
- signature $s_{\text{ring}_2} = (c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_n)$

$$\sum_i c_i = \mathscr{H}\left(\text{document} \,|\, G \,|\, c_1 P_1 + d_1 G \,|\, \ldots \,|\, c_n P_2 + d_n G\right)$$

- Signer knows one of the private keys of $\{P_1, P_2, \ldots, P_n\}$

General ECC ring signature

- common group element G
- own public/private key $P_i = p_i G$, other keys P_j
- sign:
  - random $r, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_n$ (except c_i, d_i)
  - $c = \mathscr{H}\left(\text{document} \,|\, G \,|\, c_1 P_2 + d_1 G \,|\, \ldots \,|\, rG \,|\, \ldots \,|\, c_n P_n + d_n G\right)$
  - $c_i = c - \sum_{k:k\neq i} c_k \Rightarrow \sum_k c_k = \mathscr{H}(\ldots)$
  - $rG = \left(c_i p_i + d_i\right) G \Rightarrow d_i = r - c_i p_i$

$$\sum_i c_i = \mathscr{H}\left(\text{document} \,|\, G \,|\, \ldots \,|\, \ldots \,|\, (c_i p_i + d_i) G \,|\, \ldots \,|\, \ldots\right)$$

ECC Ring Signature

- We have now a method to sign a message anonymously
  - We pick a set of public keys
- The verifier can not determine who exactly signed it
  - It only knows “one of this group”, but not more

One-Time ECC Ring Sign.

- Assuming we have one key for each transaction
- We can send a transaction
    - No one knows which one was send
    - We can show that it was valid
- What keeps us from double spending?
    - If no one knows which transaction was send, why not send it twice

Unique Element: Key Image

- We add values to the signature to detect double spending
- Public/private key $p, pG = P$
- Compute the "*key image*" $I = p\mathcal{H}(P) \in \mathbb{Z}$
- This value I is unique to each key
    - Given I, neither the private key, nor the public key can be inferred

Unique Element: Key Image

- Public/private key $p, pG = P$, "*key image*" $I = p\mathscr{H}(P) \in \mathbb{Z}$
- Signature:
  - $s = (I, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_n)$
  - For each public key:
    - $L_i = c_i P_i + d_i G$
    - $R_i = c_i I + d_i \mathscr{H}(P_i)$

$$\sum c_i = \mathscr{H}\left(\text{document} \,|\, G \,|\, I \,|\, L_1 \,|\, L_2 \,|\, \ldots \,|\, L_n \,|\, R_1 \,|\, R_2 \,|\, \ldots \,|\, R_n\right)$$

One-Time ECC Ring Sign.

- Create the signature (similar to before):
  - Random value r for own public key
    - $L_{\text{own}} = rG$
    - $R_{\text{own}} = rI$
  - Random c_i, d_i for all other public keys P_i
    - $L_i = c_i P_i + d_i G$
    - $R_i = c_i I + d_i \mathcal{H}(P_i)$

One-Time ECC Ring Sign.

- Given:
  - $L_i = c_i P_i + d_i G$ $\qquad R_i = c_i I + d_i \mathscr{H}(P_i)$
  - $L_{\text{own}} = rG$ $\qquad R_{\text{own}} = rI$
- $c = \mathscr{H}\left(\text{document} \,|\, G \,|\, I \,|\, L_1 \,|\, \ldots \,|\, L_n \,|\, R_1 \,|\, \ldots \,|\, R_n\right)$
- $c_{\text{own}} = c - \sum_i c_i$ $\qquad d_{\text{own}} = r - c_{\text{own}} p$
- $L_{\text{own}} = c_{\text{own}} P + d_{\text{own}} G$

One-Time ECC Ring Sign.

- Signature:
  - $s = (I, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_n)$
- Verify via
  - $L_i = c_i P_i + d_i G \qquad\qquad R_i = c_i I + d_i \mathscr{H}(P_i)$

$$\sum_i c_i \stackrel{?}{=} \mathscr{H}\left(\text{document} \,|\, G \,|\, I \,|\, L_1 \,|\, \ldots \,|\, L_n \,|\, R_1 \,|\, \ldots \,|\, R_n\right)$$

One-Time ECC Ring Sign.

- $s = (I, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_n)$
- $\sum_i c_i \stackrel{?}{=} \mathscr{H}\left(\text{document} \,|\, G \,|\, I \,|\, L_1 \,|\, \ldots \,|\, L_n \,|\, R_1 \,|\, \ldots \,|\, R_n\right)$

- Properties:
  - Given a valid signature, we can not infer which private key was known
  - The key image I is tied to the private key
  - Two separate signatures using the same private key need to use the same key image I

One-Time ECC Ring Sign. Summary

- A user choses an anonymity set of public keys
- Signs a transaction, so that
  - It is impossible to identify who signed it
  - Signing more than once with same private key can be detected
  - If each transaction has its own public/private key, we can detect double-spending

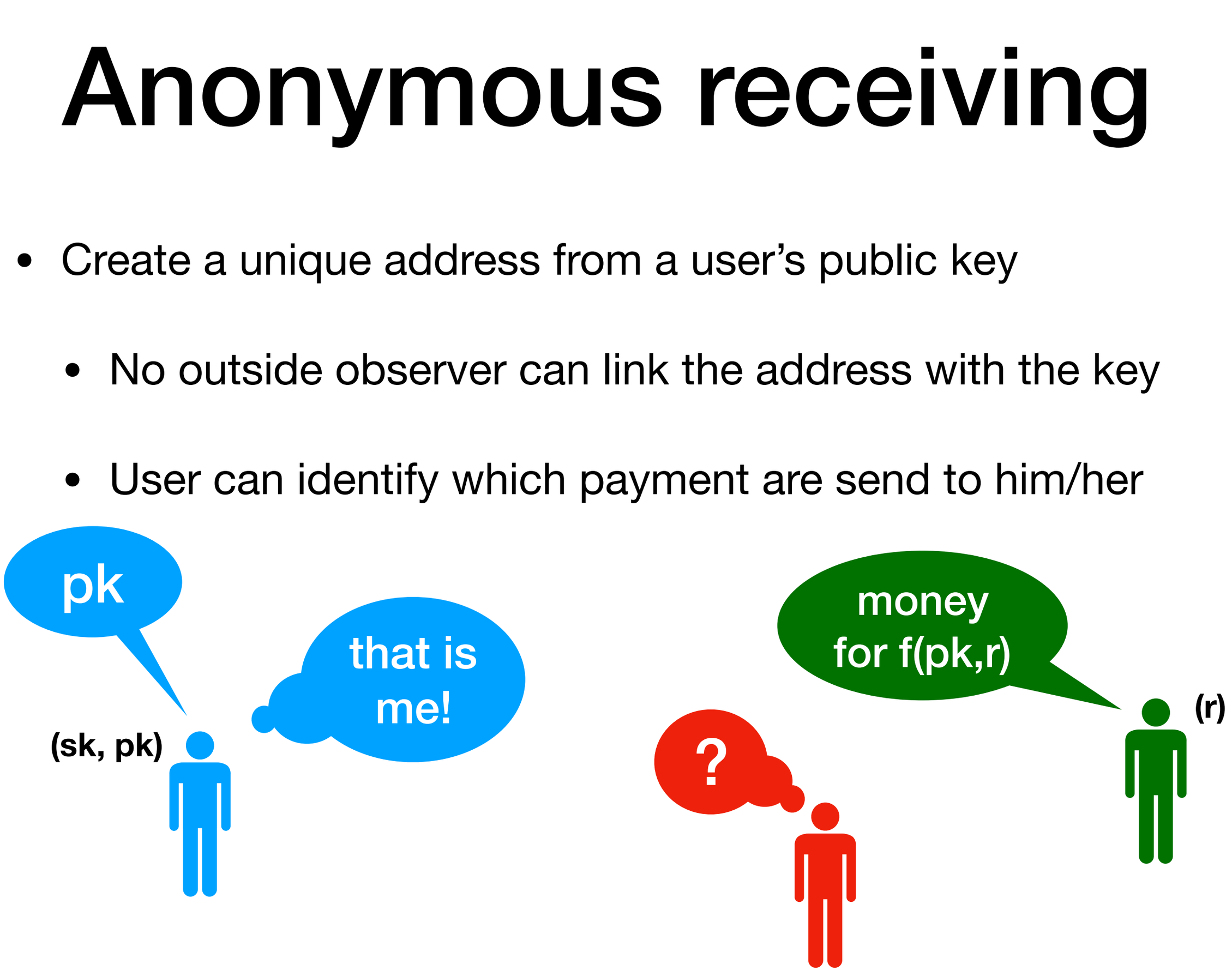
Monero

- Anonymously receiving
  - Address is created from public key
  - Not possible to link target address to public key

- Anonymous sending
  - Owner selects an anonymity set $\mathcal{S} = \{T_1, T_2, \ldots, T_n\}$
  - Shows that one in $\mathcal{S}$ signed the transaction

Anonymous receiving

- Create a unique address from a user's public key
  - No outside observer can link the address with the key
  - User can identify which payment are send to him/her



Keys in Monero

- Each user has an address template, consisting of 2 keys
- $a, b \in \mathbb{Z}\ a \neq b$ (*private key*)
- $(aG, bG) = (A, B)$ (*public key*)
- (a, B) (*tracking key*)

Receiving Money

- Alice wants to send money to Bob
  - r random value, one-time public key $R = rG$
  - one-time public key as address $P = \mathscr{H}(rA)G + B$
  - Alice sends transaction

```
Amount     :     1234 XMR
Public key:      R
Address    :     P
```

Receiving Money

- Alice wants to send money to Bob
- Alice creates $P = \mathscr{H}(rA)G + B$
- Bob sees

```
Amount     :     1234 XMR
Public key:      R
Address    :     P
```

- and can check if $P = \mathscr{H}(aR)G + B$
- His key (a, b), $(aG, bG) = (A, B)$

Receiving Money

- Alice creates $P = \mathscr{H}(rA)G + B$
- Bob's key: $(a, b), \quad (aG, bG) = (A, B)$

Alice can create this

$$P = \mathscr{H}(rA)G + B = \mathscr{H}(raG)G + B$$

$$= \mathscr{H}(arG)G + B = \mathscr{H}(aR)G + B$$

Bob can create this

Receiving Money

- Alice creates $P = \mathscr{H}(rA)G + B = \mathscr{H}(aR)G + B$
- Bob can detect payments directed to him
- No one else can see that this is a payment for Bob

Monero

- Anonymously receiving
  - Address is created from public key
  - Not possible to link target address to public key

- Anonymous sending
  - Owner selects an anonymity set $\mathcal{S} = \{T_1, T_2, \ldots, T_n\}$
  - Shows that one in $\mathcal{S}$ signed the transaction

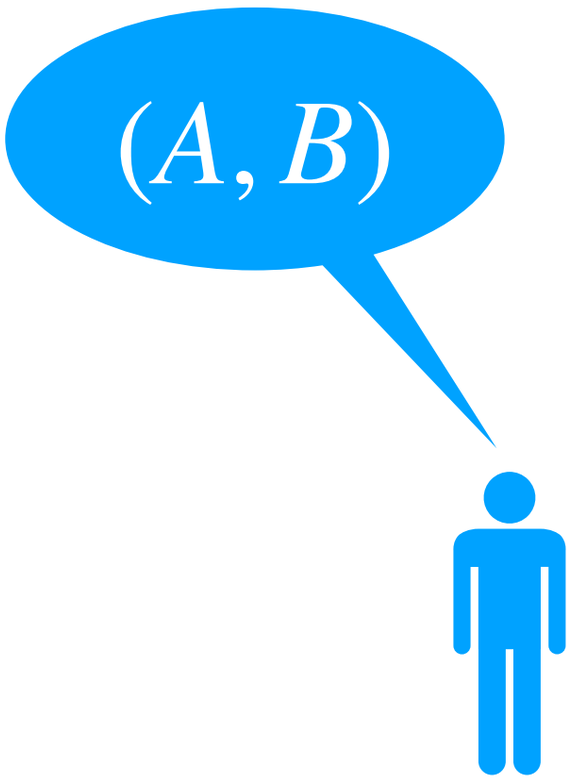
Taking ownership of a transaction

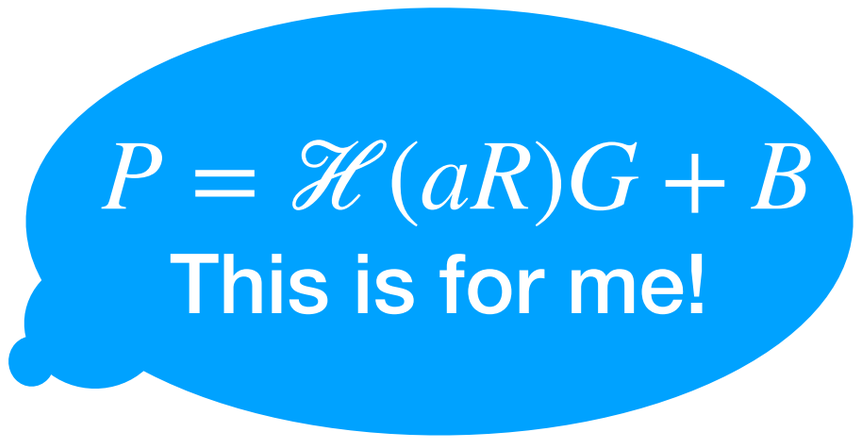
- The transaction's address P can be seen as a key
  - Everybody knows P
  - Alice knows how to construct $P = \mathscr{H}(rA)G + B$
  - Only Bob knows secret $p = \mathscr{H}(rA) + b$, so that $pG = P$

```
Amount     :     1234 XMR
Public key:     R
Address    :     P
```

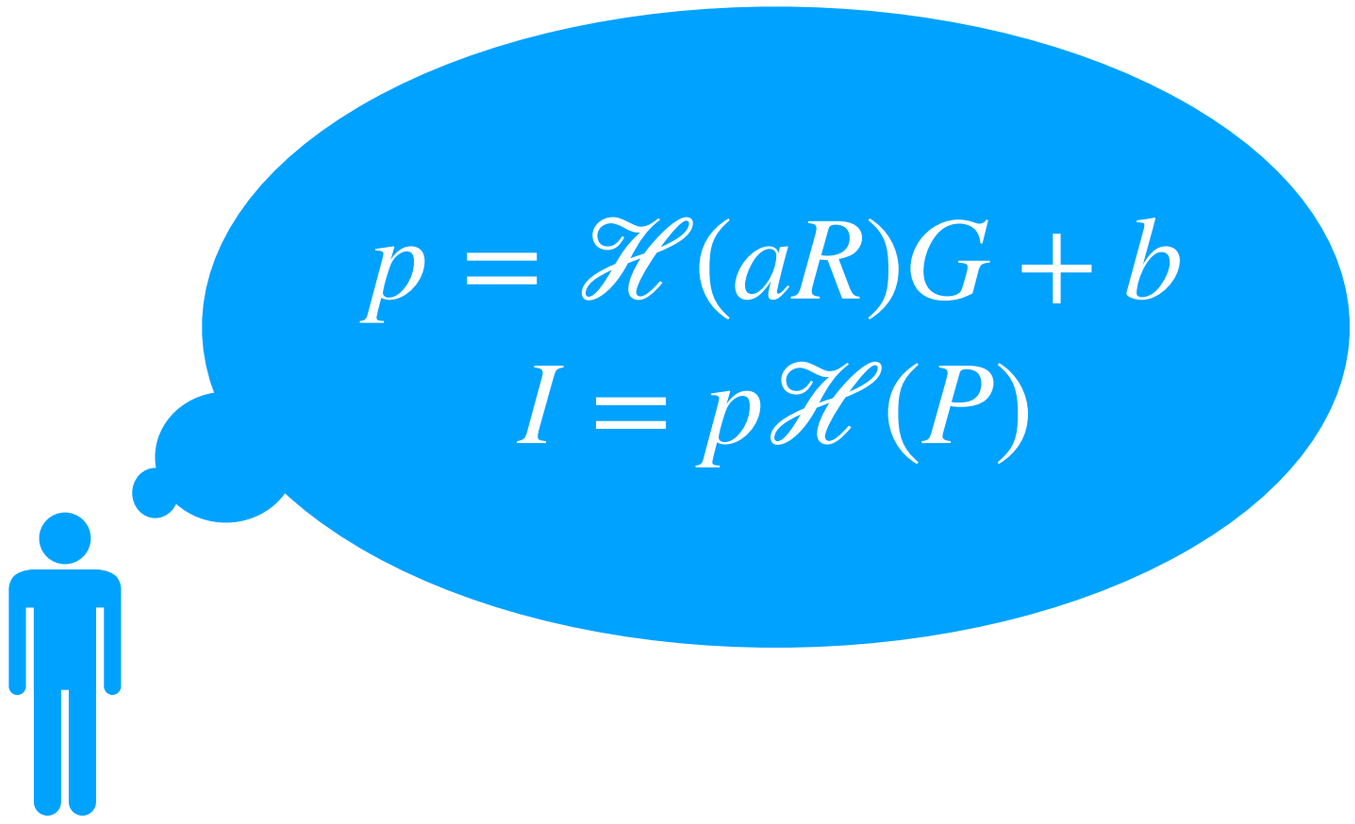
Lifetime of a Monero Tx

- Bob has secret (a, b) and publishes $(aG, bG) = (A, B)$
- Alice as secret r



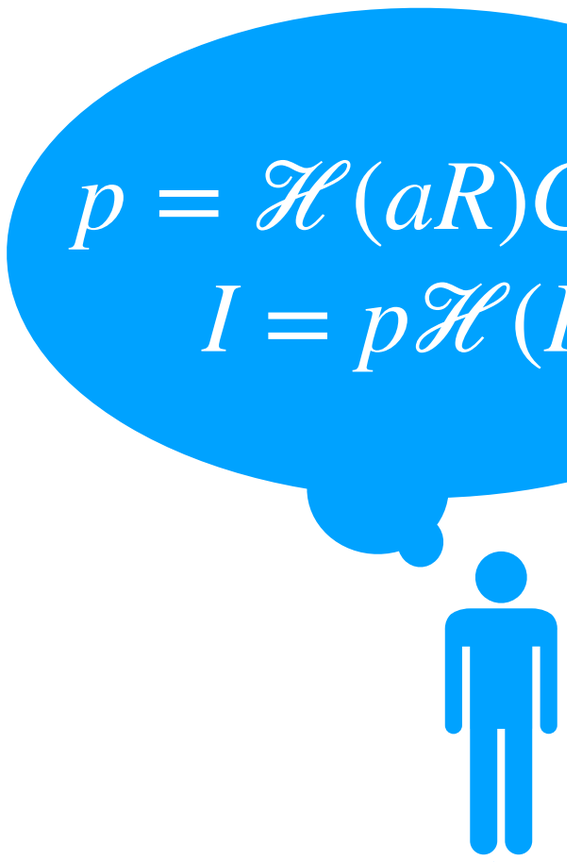


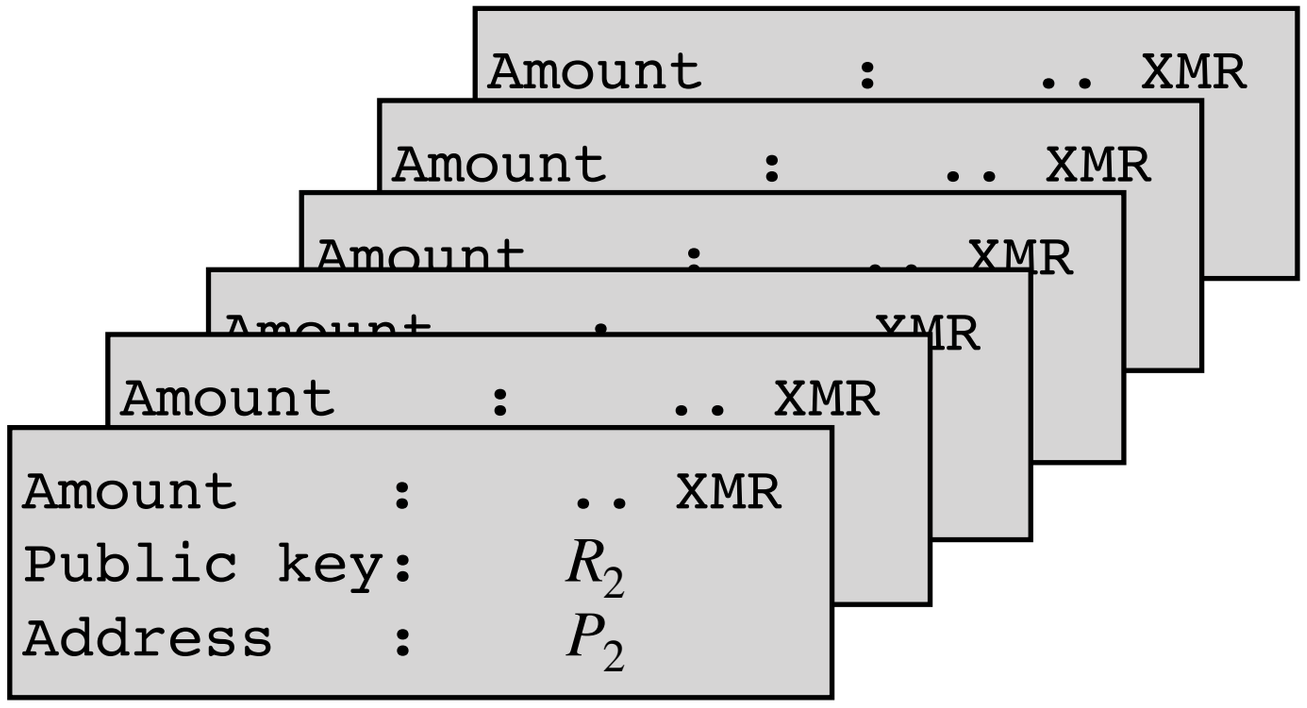
Lifetime of a Monero Tx



Lifetime of a Monero Tx

$$p = \mathcal{H}(aR)G + b$$
$$I = p\mathcal{H}(P)$$





other tx found on the Blockchain

$$\mathcal{S} = \{P, P_2, P_3, p_4, \ldots, P_n\}$$
$$s = \left(I, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_n\right)$$

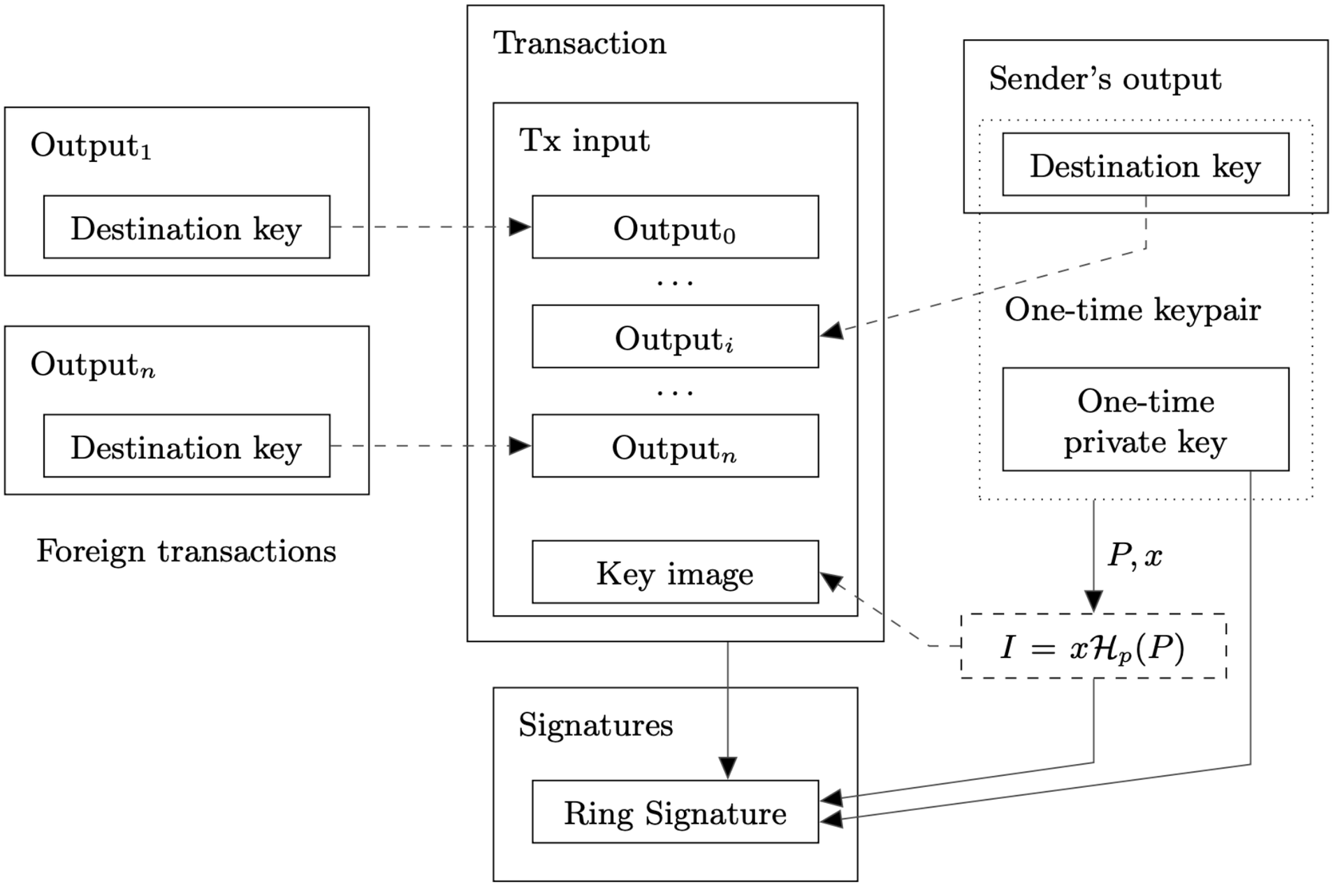
Lifetime of a Monero Tx

$$\mathcal{S} = \{P, P_2, P_3, p_4, \ldots, P_n\}$$
$$s = \left(I, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_n\right)$$

$$\sum_i c_i = \mathscr{H}\left(\mathbf{tx} \mid G \mid I \mid L_1 \mid \ldots \mid L_n \mid R_1 \mid \ldots \mid R_n\right)$$

tx is valid

A complete transaction



Problems with Monero

- Amounts are open
- User choses anonymity set
- Dust attack