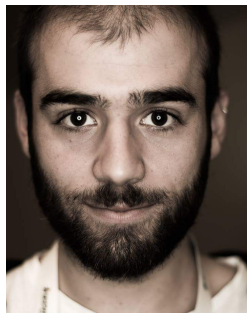
The security of Mimblewimble

Georg Fuchsbauer

joint work with

Michele Orrù

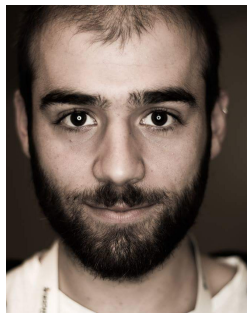


and Yannick Seurin

The security of Mimblewimble

Georg Fuchsbauer

joint work with

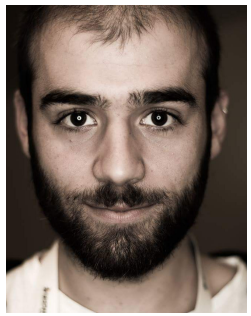
Michele Orrù  and Yannick Seurin 

F, Orrù, Seurin: **Aggregate cash systems: A cryptographic investigation of Mimblewimble.** EUROCRYPT'19

The security of Mimblewimble

Georg Fuchsbauer

joint work with

Michele Orrù  and Yannick Seurin 

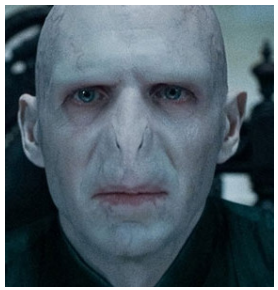
F, Orrù, Seurin: **Aggregate cash systems: A cryptographic investigation of Mimblewimble.** EUROCRYPT'19

F, Orrù: **Non-interactive Mimblewimble transactions, revisited.** ASIACRYPT'22

What is it?

- **Cryptocurrency scheme**

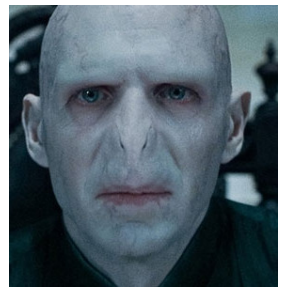
- proposed by "Tom Elvis Jedusor" in 2016



What is it?

- **Cryptocurrency scheme**

- proposed by "Tom Elvis Jedusor" in 2016

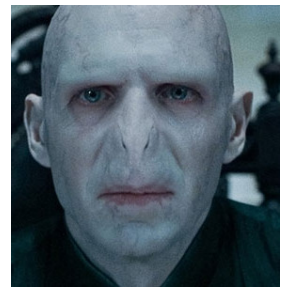


```
MIMBLEWIMBLE
Tom Elvis Jedusor
19 July, 2016

\****/
Introduction
/****\

Bitcoin is the first widely used financial system for which all the necessary
data to validate the system status can be cryptographically verified by anyone
However, it accomplishes this feat by storing all transactions in a public
database called "the blockchain" and someone who genuinely wishes to check
this state must download the whole thing and basically replay each transaction
check each one as they go. Meanwhile, most of these transactions have not
```

What is it?

- **Cryptocurrency scheme**

- proposed by "Tom Elvis Jedusor" in 2016

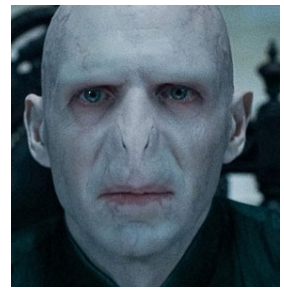


- uses ideas from Gregory Maxwell

What is it?

- **Cryptocurrency scheme**

- proposed by "Tom Elvis Jedusor" in 2016

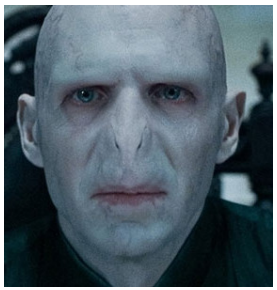


- uses ideas from Gregory Maxwell
- further developed by Andrew Poelstra

What is it?

- **Cryptocurrency scheme**
  - **Privacy** (all amounts hidden; input/output relation blurred)

- proposed by "Tom Elvis Jedusor" in 2016

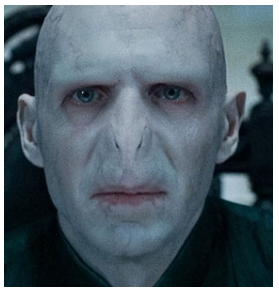


- uses ideas from Gregory Maxwell
- further developed by Andrew Poelstra

What is it?

- **Cryptocurrency scheme**
  - **Privacy** (all amounts hidden; input/output relation blurred)
  - **Scalability** (forget about spent tx's)

- proposed by "Tom Elvis Jedusor" in 2016



- uses ideas from Gregory Maxwell
- further developed by Andrew Poelstra

Applications

Implemented by several cryptocurrencies (since 2019):

#	Name	Price	1h %	24h %	7d %	Market Cap
1273	Beam BEAM	$0.03445	▲0.28%	▼0.51%	▼11.47%	$5,194,030
1435	Grin GRIN	$0.03167	▼0.08%	▼1.60%	▼6.82%	$3,110,211

Non-interactive TXs

Main **drawback**: transactions are *interactive*

2020: David Burkett, Gary Yu:
Non-interactive transactions

Non-interactive TXs

Main **drawback**: transactions are *interactive*

2020: David Burkett, Gary Yu:
Non-interactive transactions

2021: Fixed by Burkett, F, Orrù

Analyzed by F, Orrù

Non-interactive TXs

#	Name	Price	1h %	24h %	7d %	Market Cap
1273	Beam BEAM	$0.03445	▲0.28%	▼0.51%	▼11.47%	$5,194,030
1435	Grin GRIN	$0.03167	▼0.08%	▼1.60%	▼6.82%	$3,110,211

Non-interactive TXs

#	Name	Price	1h %	24h %	7d %	Market Cap
247	MimbleWimbleCoin MWC	$16.77	▲0.76%	▲0.42%	▼3.92%	$183,788,914
1273	Beam BEAM	$0.03445	▲0.28%	▼0.51%	▼11.47%	$5,194,030
1435	Grin GRIN	$0.03167	▼0.08%	▼1.60%	▼6.82%	$3,110,211

Non-interactive TXs

2022: Implemented in

Litecoin ("Mimblewimble extension blocks")

Non-interactive TXs

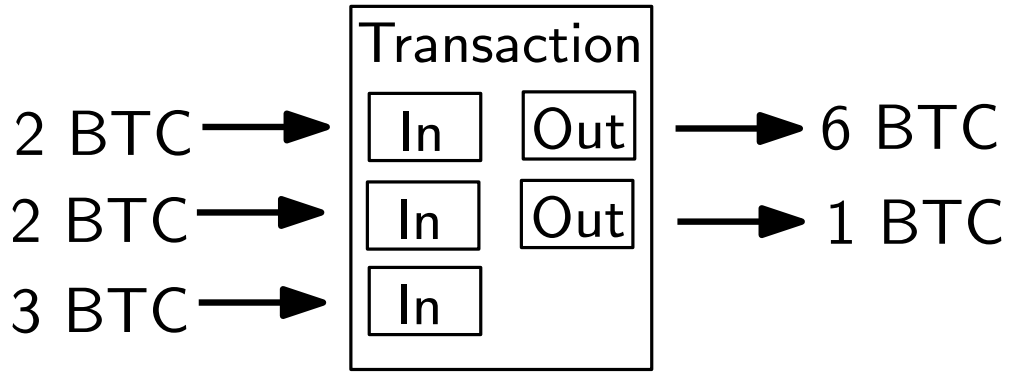
2022: Implemented in

Litecoin ("Mimblewimble extension blocks")

#	Name	Price	1h %	24h %	7d %	Market Cap
1	Bitcoin BTC	$63,990.10	▼0.12%	▼0.44%	▼3.86%	$1,261,608,323,848
2	Ethereum ETH	$3,468.64	▼0.12%	▼0.99%	▼3.53%	$424,131,839,729
⋮						
19	Polygon MATIC	$0.5682	▼0.88%	▼0.66%	▼8.52%	$5,627,473,007
20	Litecoin LTC	$74.63	▼0.17%	▲0.50%	▼5.63%	$5,573,047,076

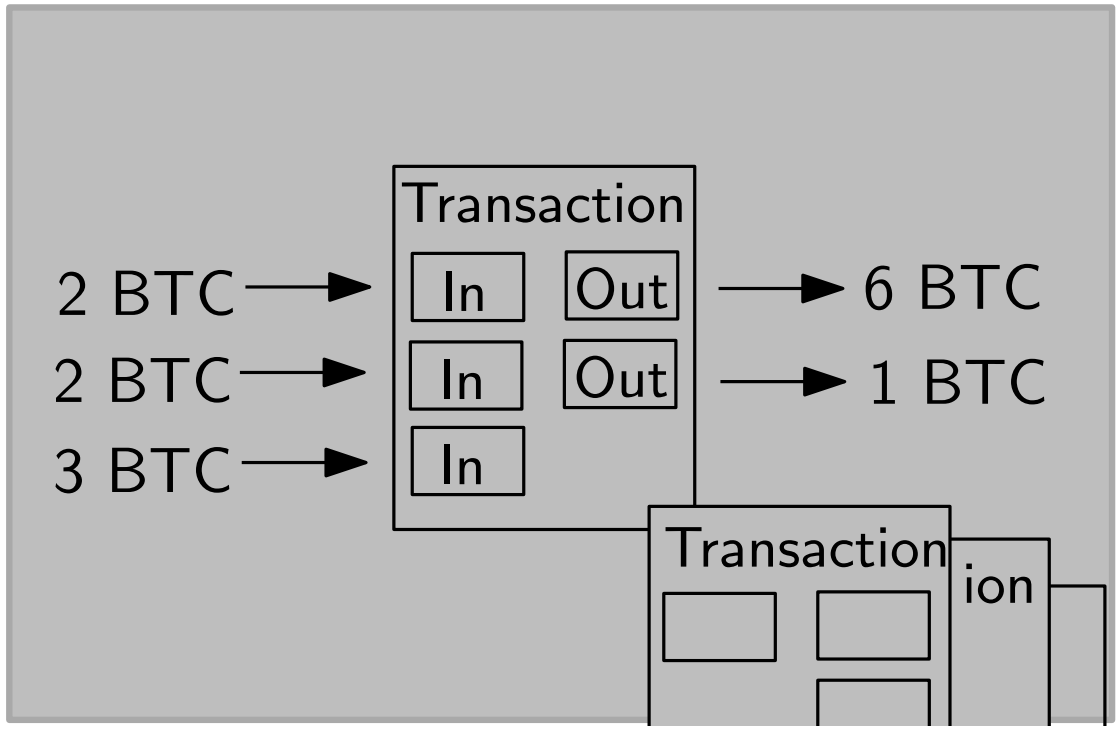
Bitcoin

- **Transactions**



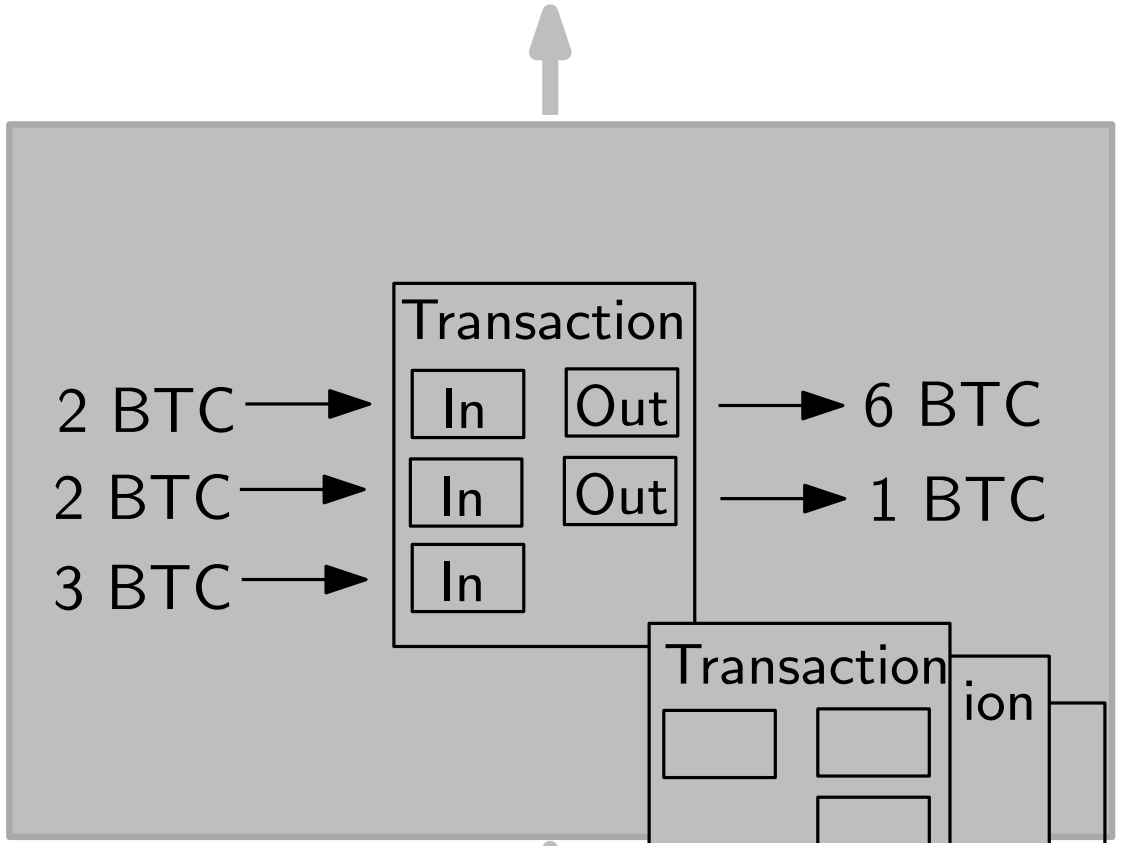
Bitcoin

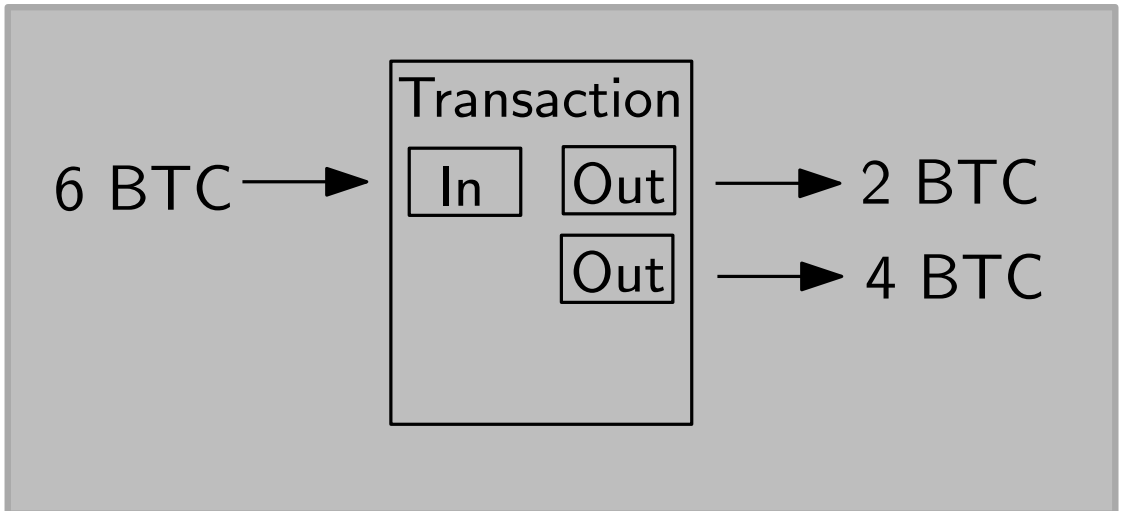
- **Block**



Bitcoin

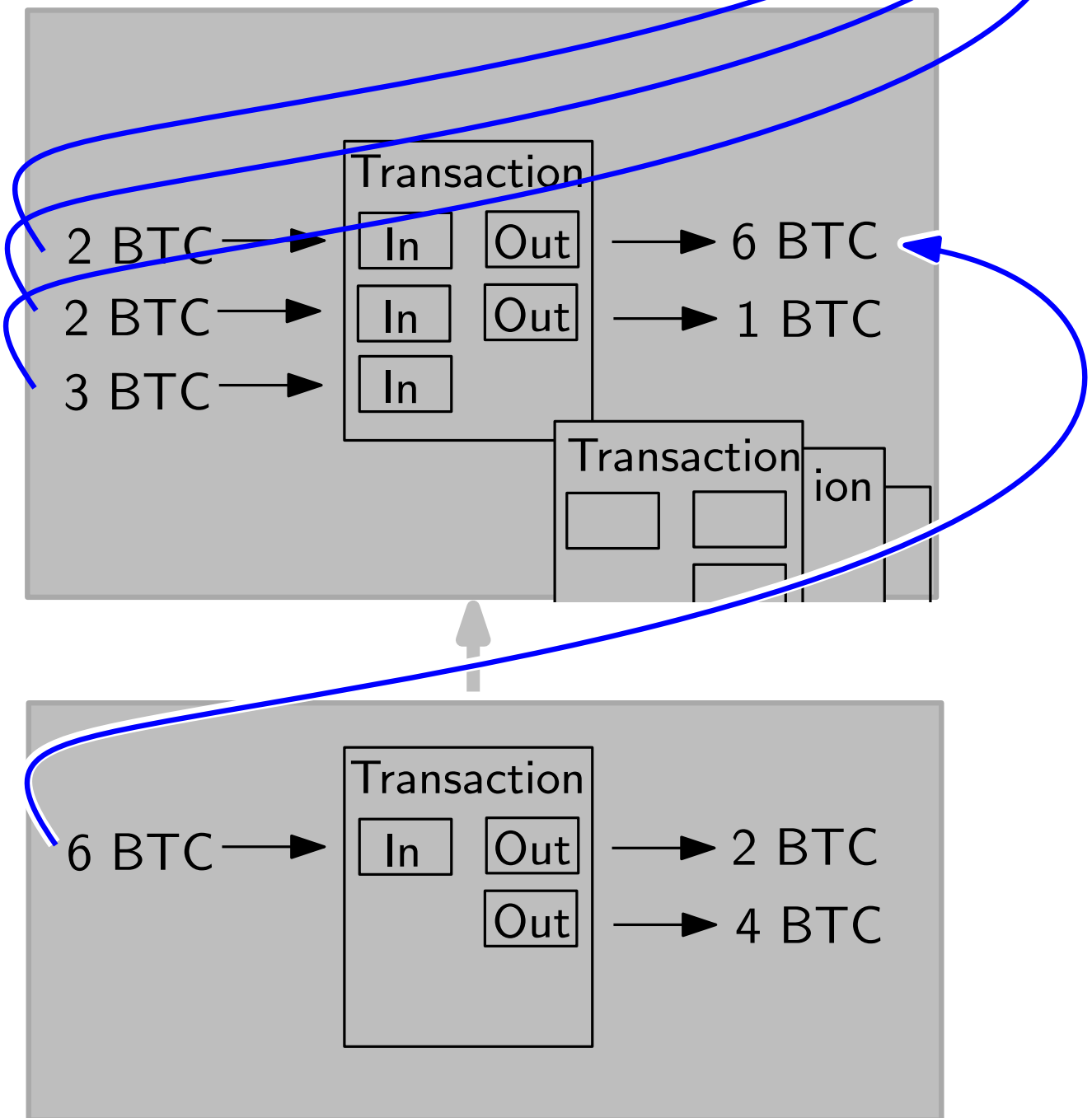
- **Blockchain**





Bitcoin

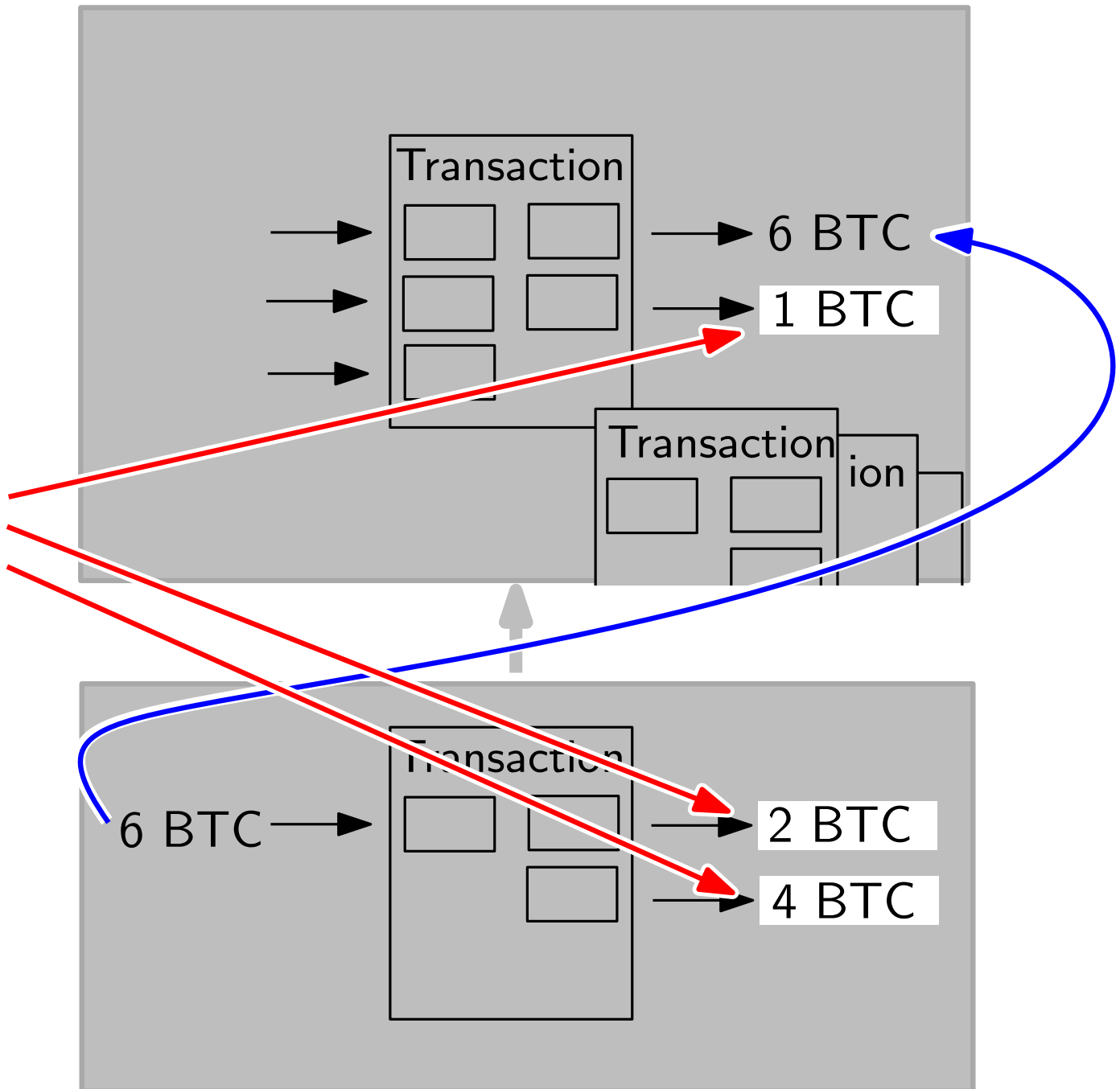
- Reference to previous output



Bitcoin

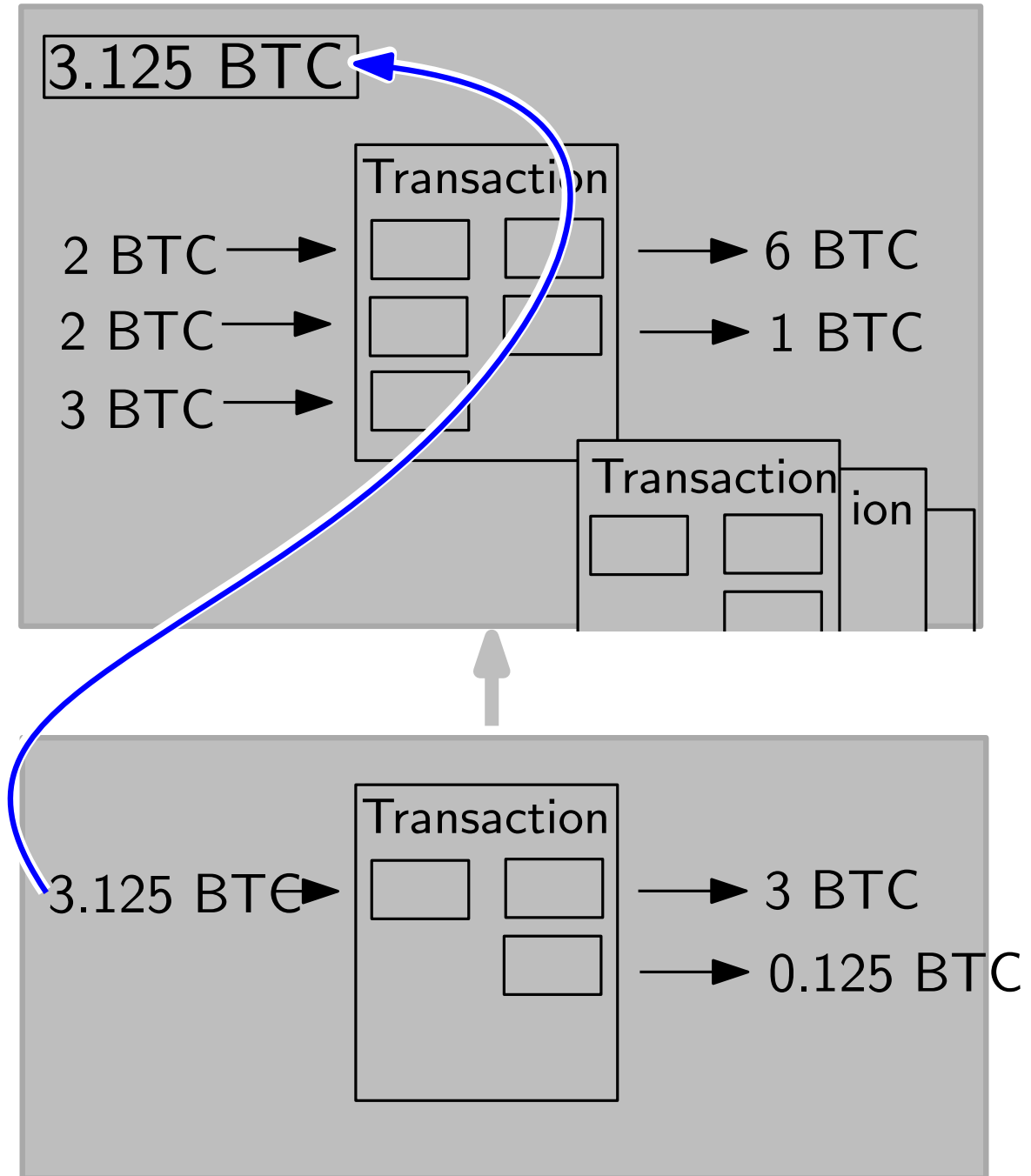
Unspent transaction outputs (UTXO's)

= existing money in system



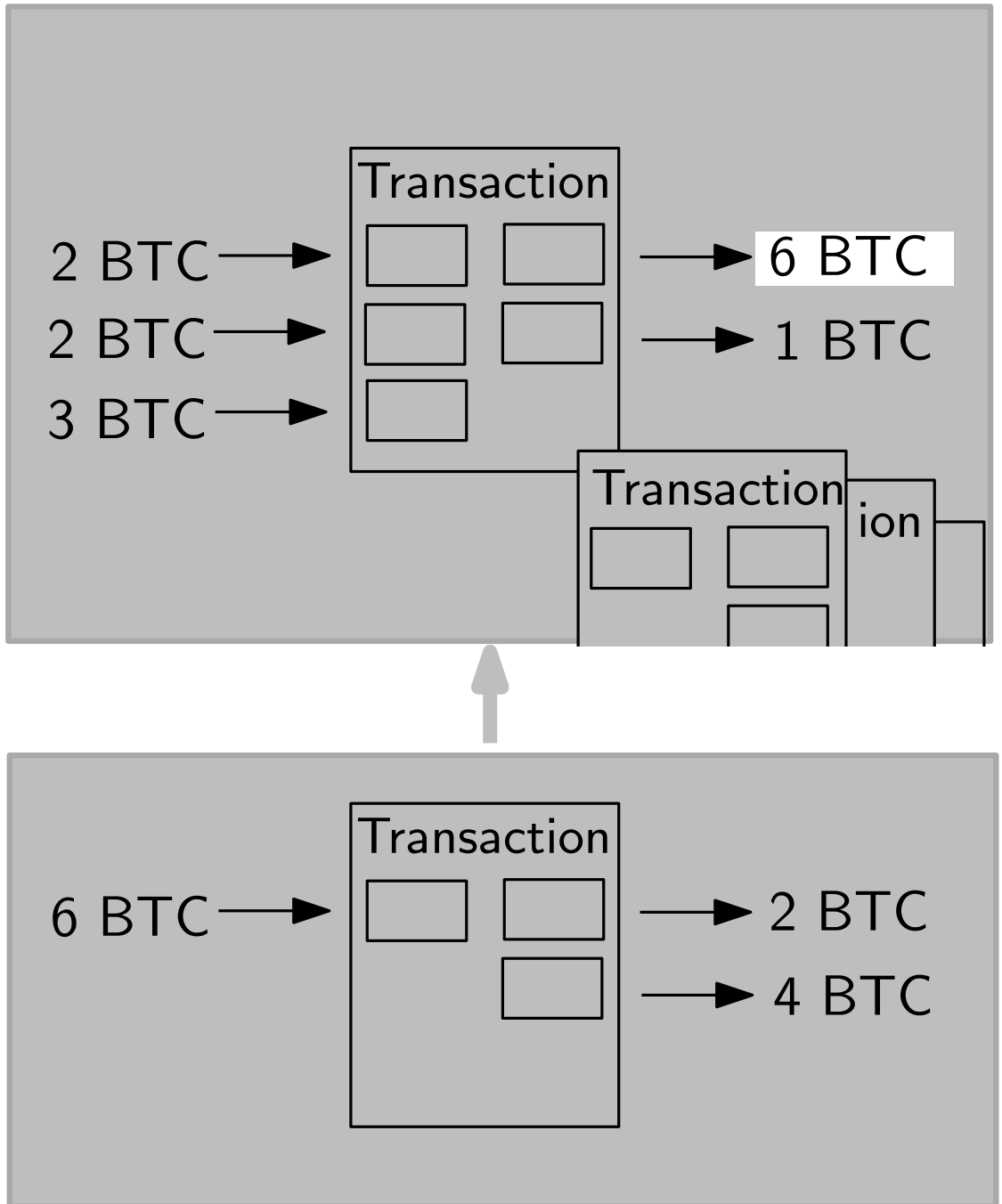
Bitcoin

- **Coinbase transaction**



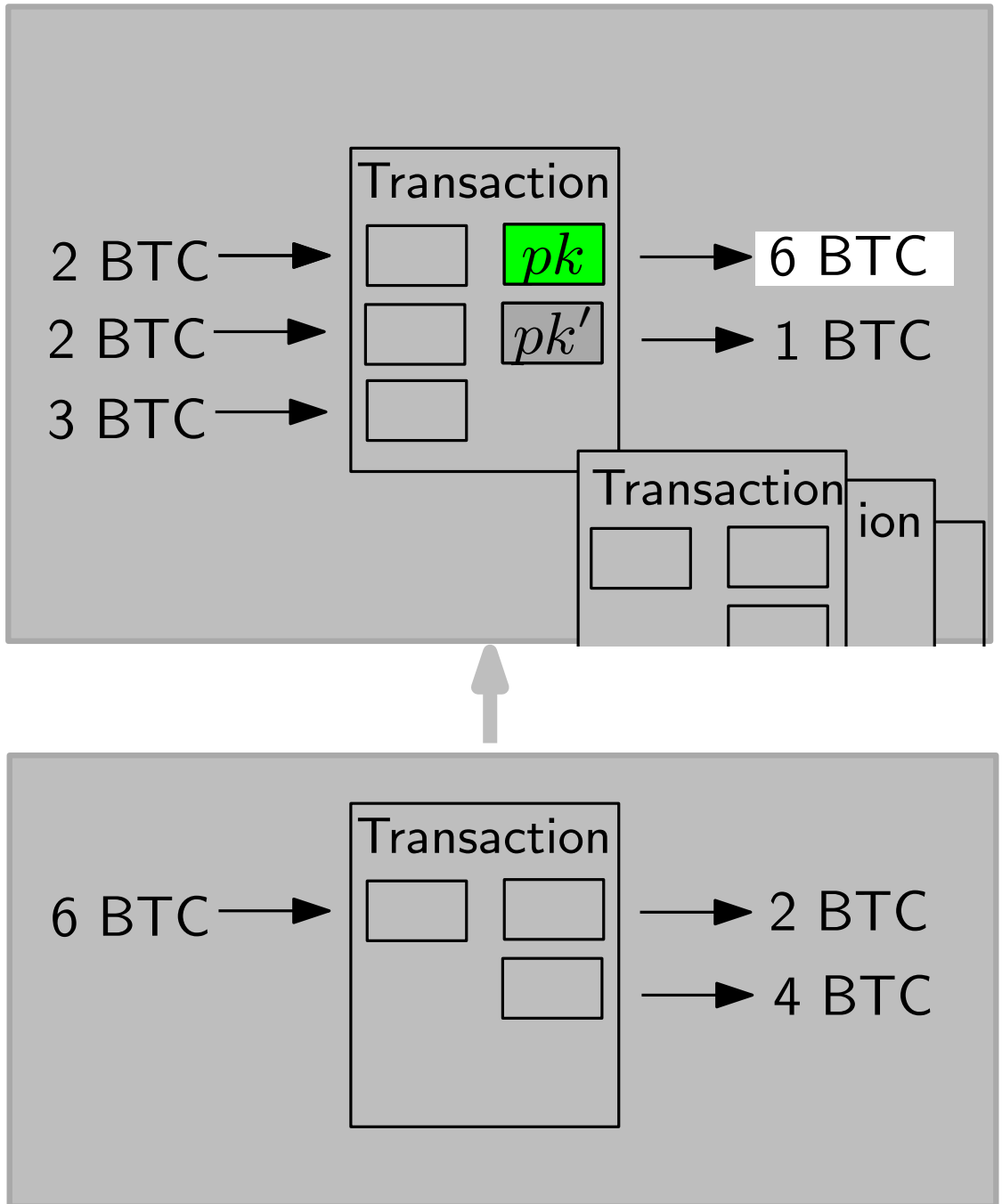
Bitcoin

- **Owning**
  an output



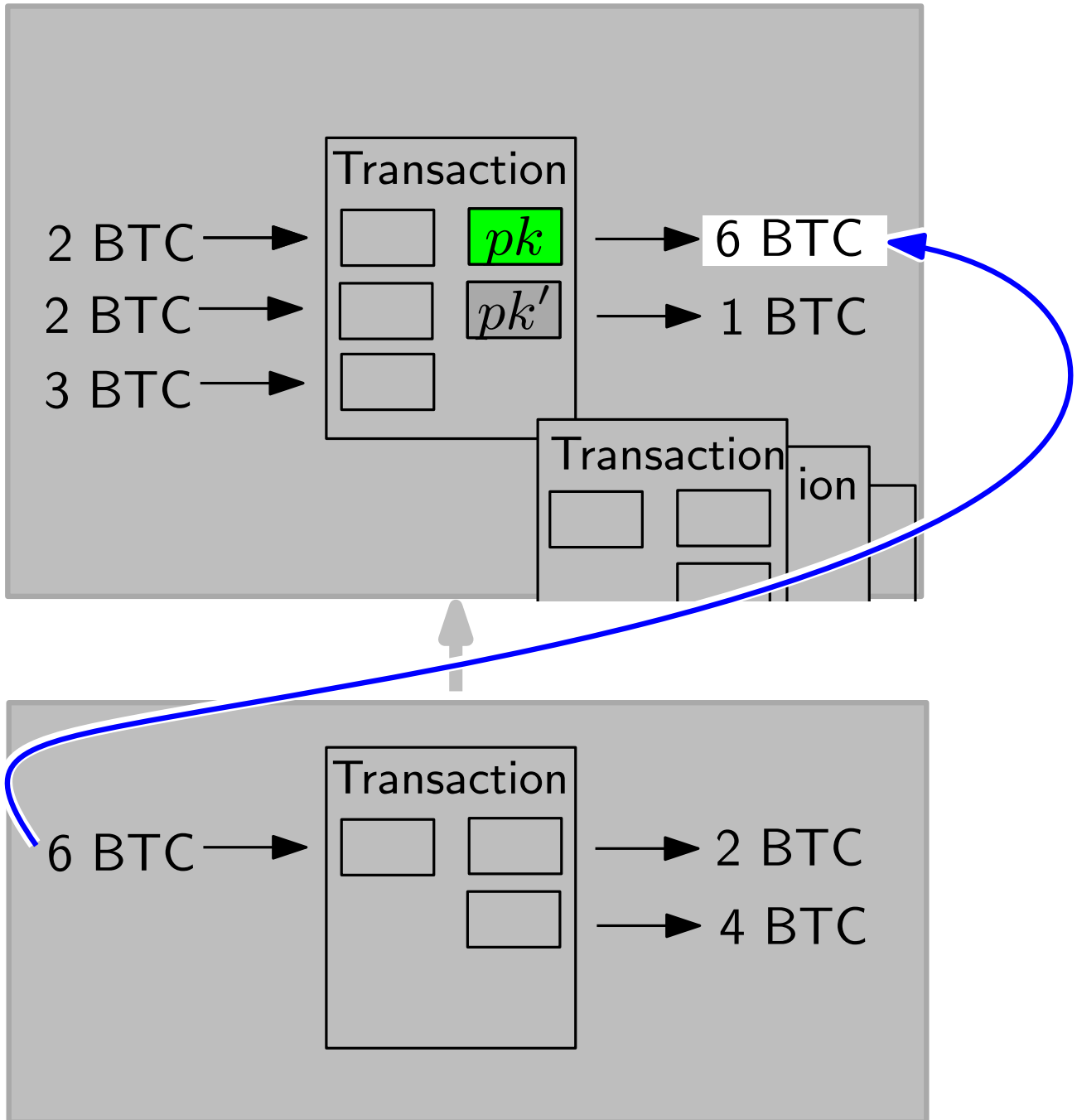
Bitcoin

- **Owning**
  an output

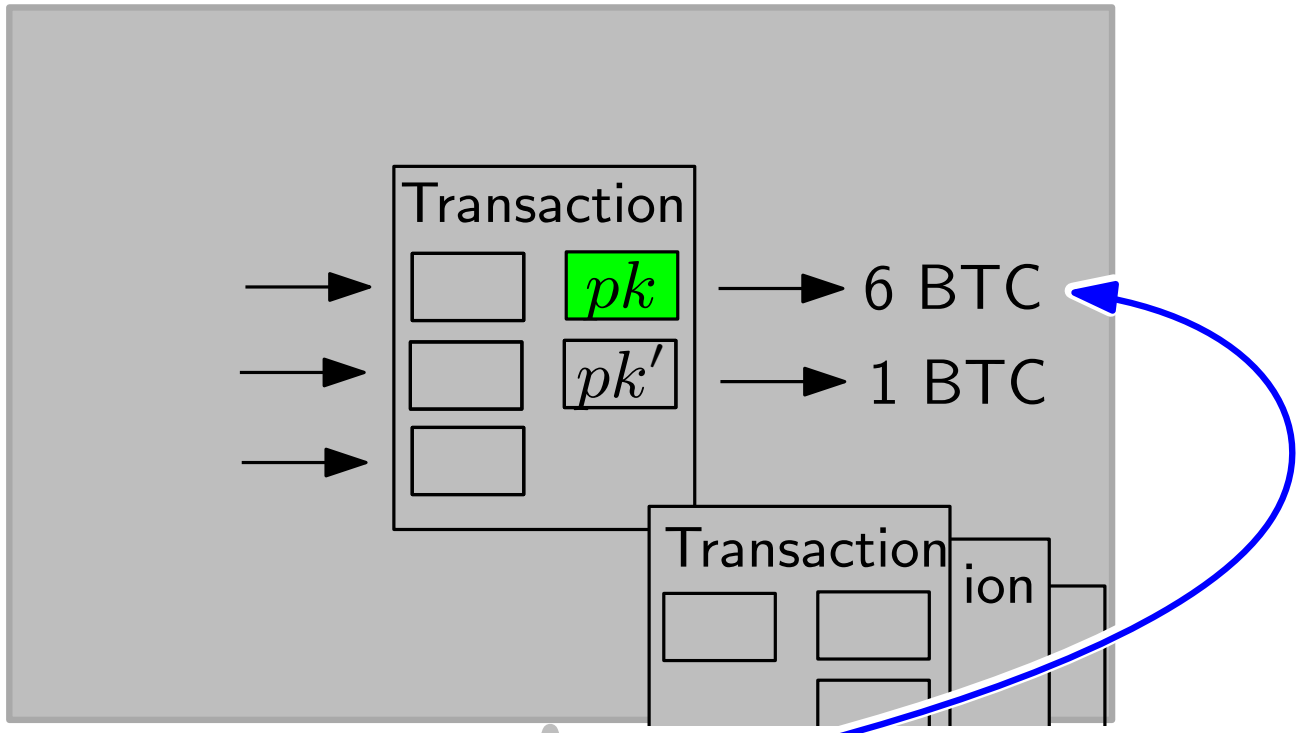


Bitcoin

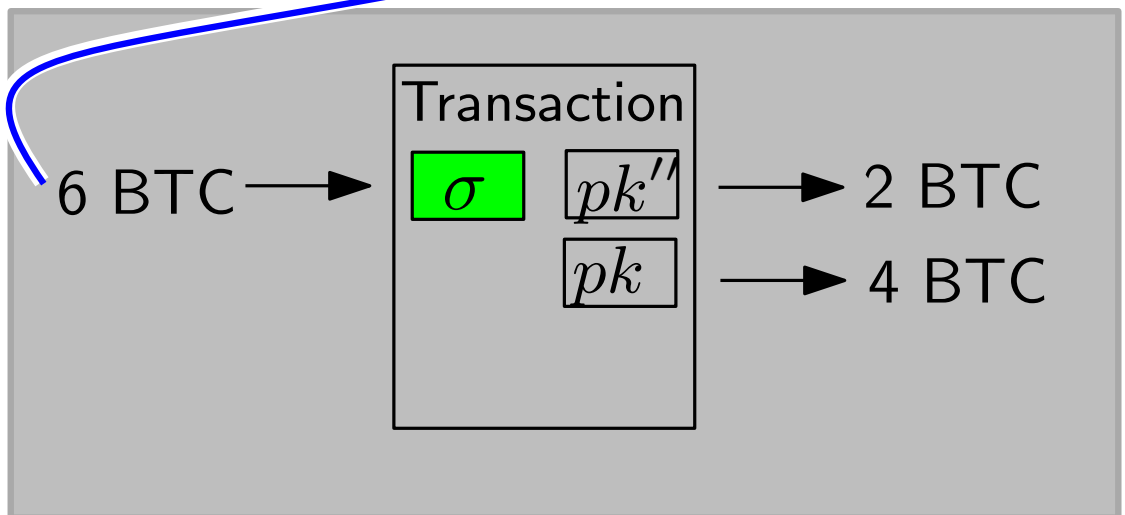
- **Owning**
  an output



Bitcoin



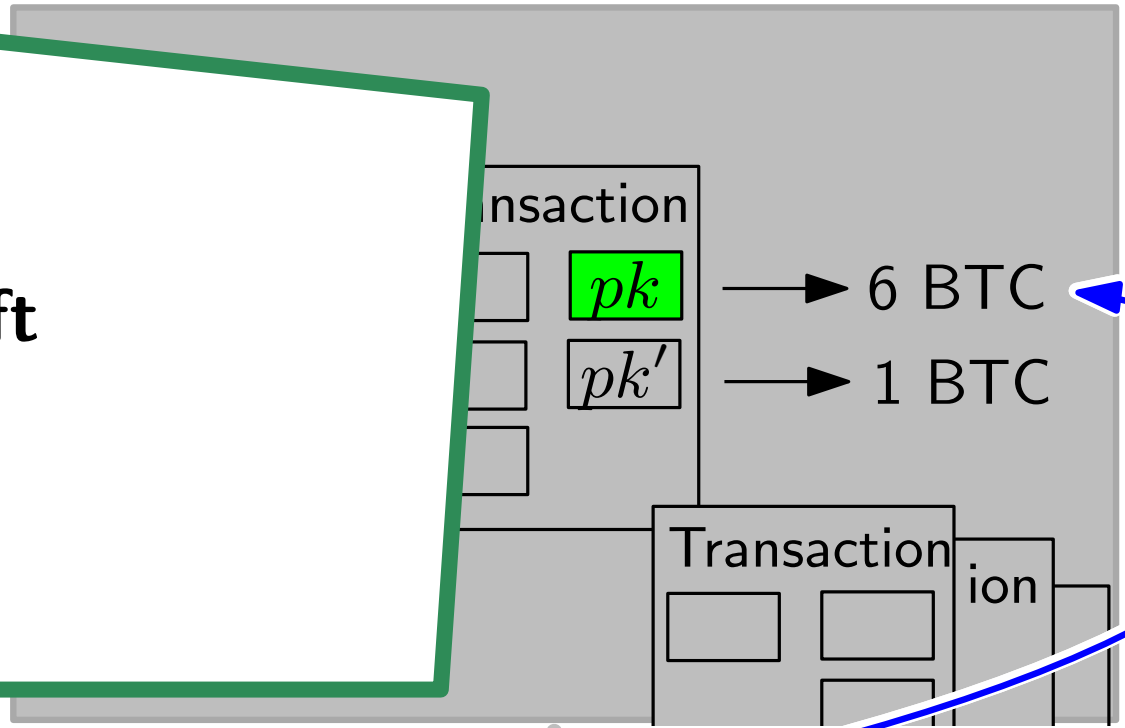
σ is **signature** under pk on tx



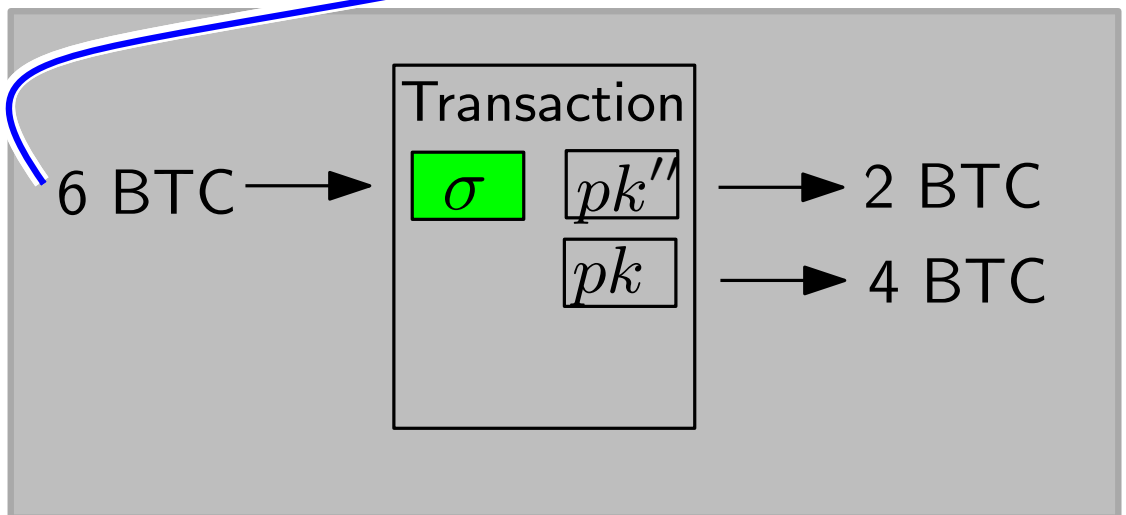
Bitcoin

Security

- signatures
  $\Rightarrow$ **no theft**



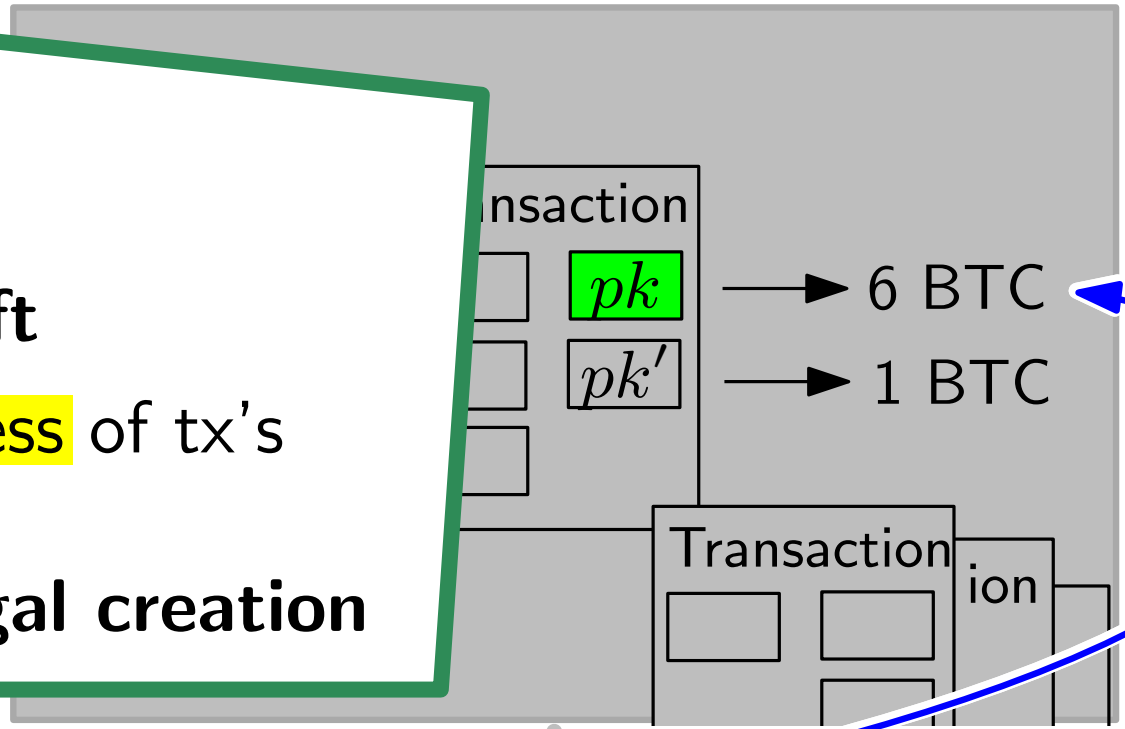
σ is **signature** under pk on tx



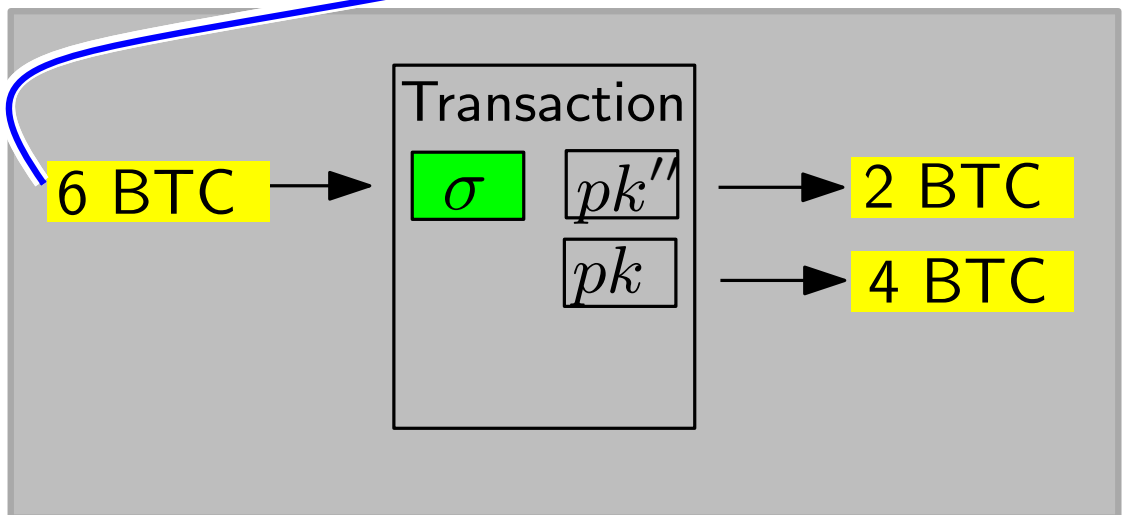
Bitcoin

Security

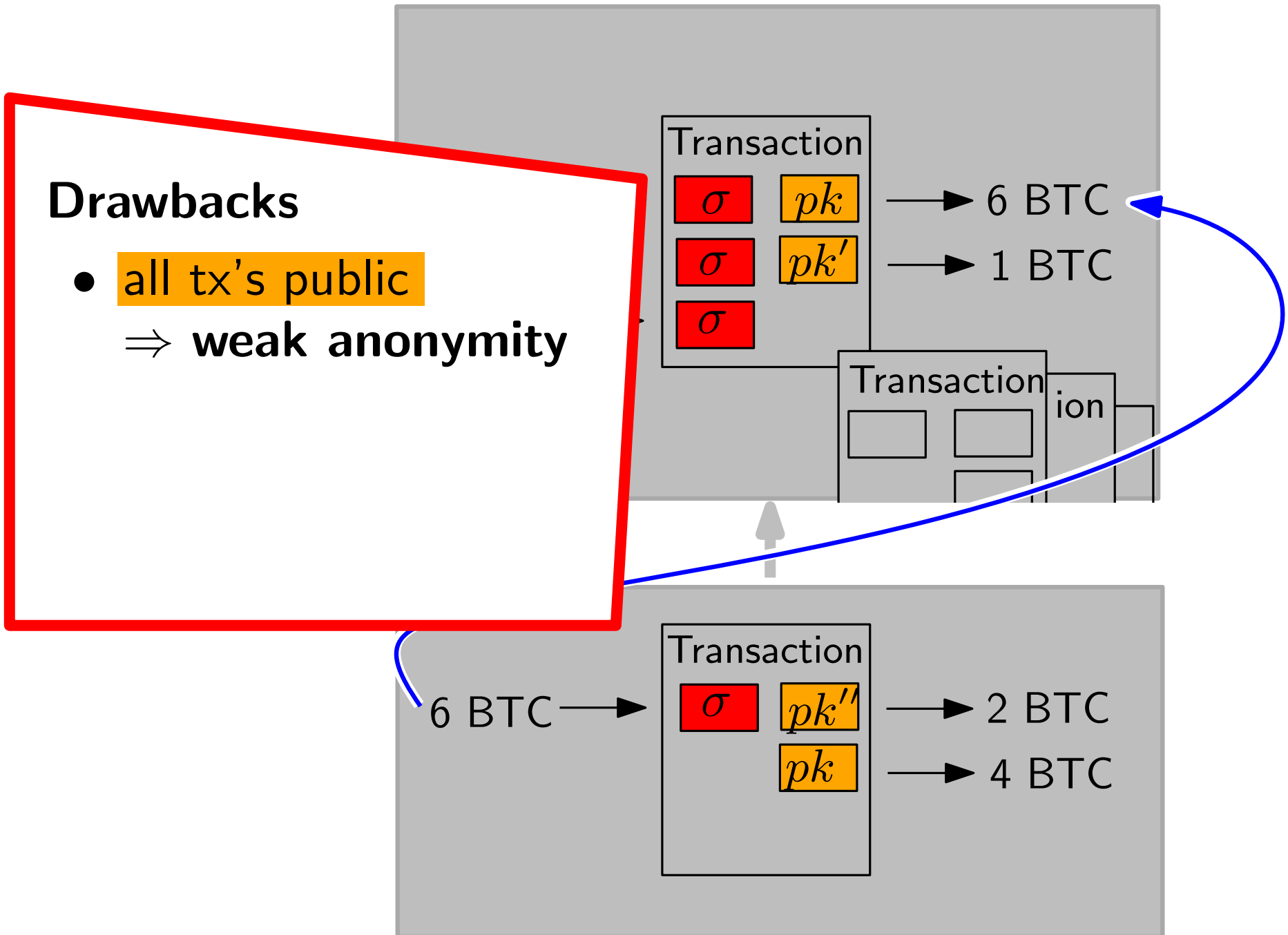
- signatures
  $\Rightarrow$ **no theft**
- balancedness of tx's checkable
  $\Rightarrow$ **no illegal creation**



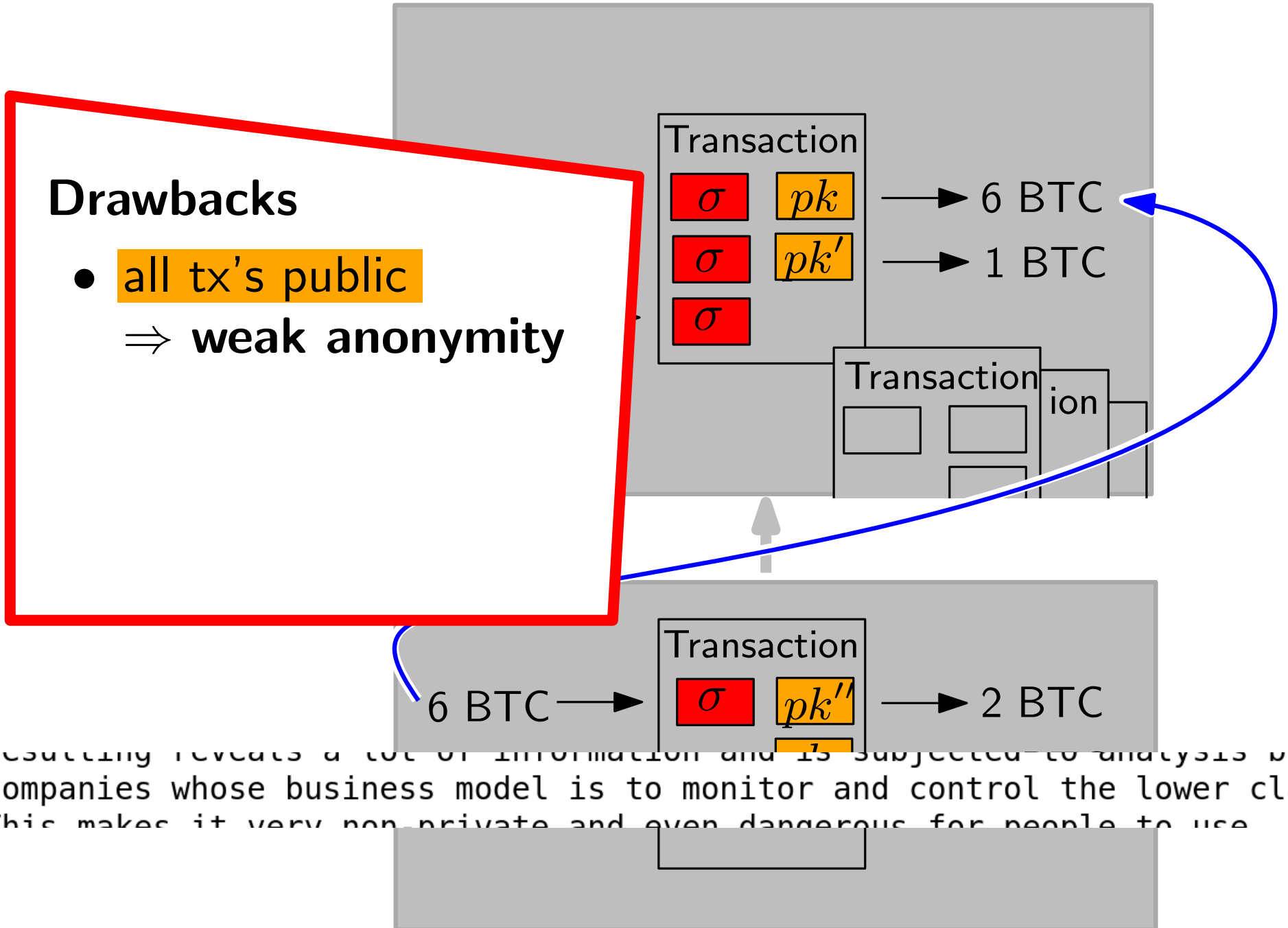
σ is **signature** under pk on tx



Bitcoin


Drawbacks
• all tx's public
⇒ weak anonymity
Transaction
σ
σ
σ
pk
pk'
6 BTC
1 BTC
Transaction
ion
6 BTC
Transaction
σ
pk''
pk
2 BTC
4 BTC

Bitcoin

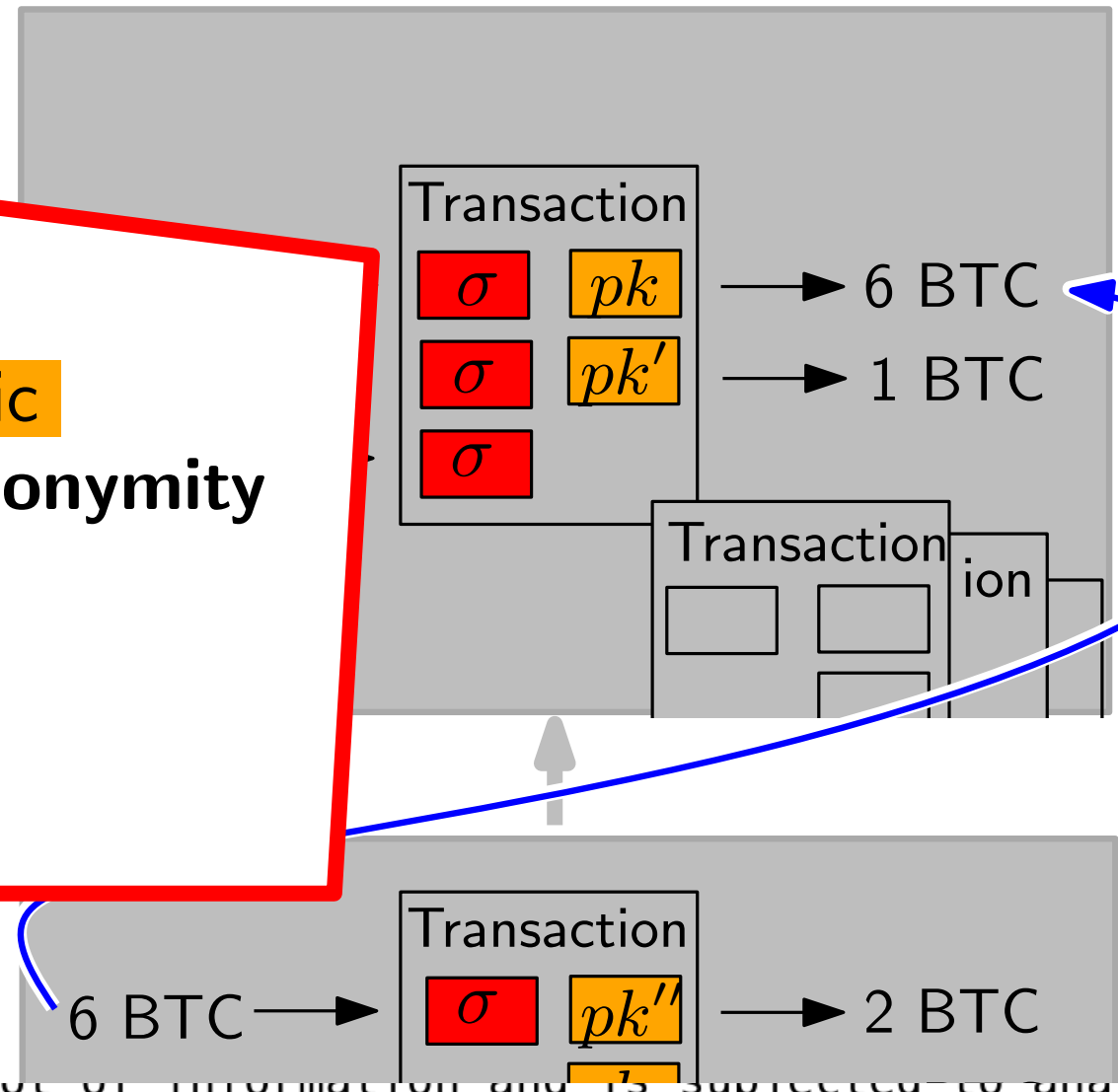


resulting reveals a lot of information and is subjected to analysis by many companies whose business model is to monitor and control the lower classes. This makes it very non-private and even dangerous for people to use

Bitcoin

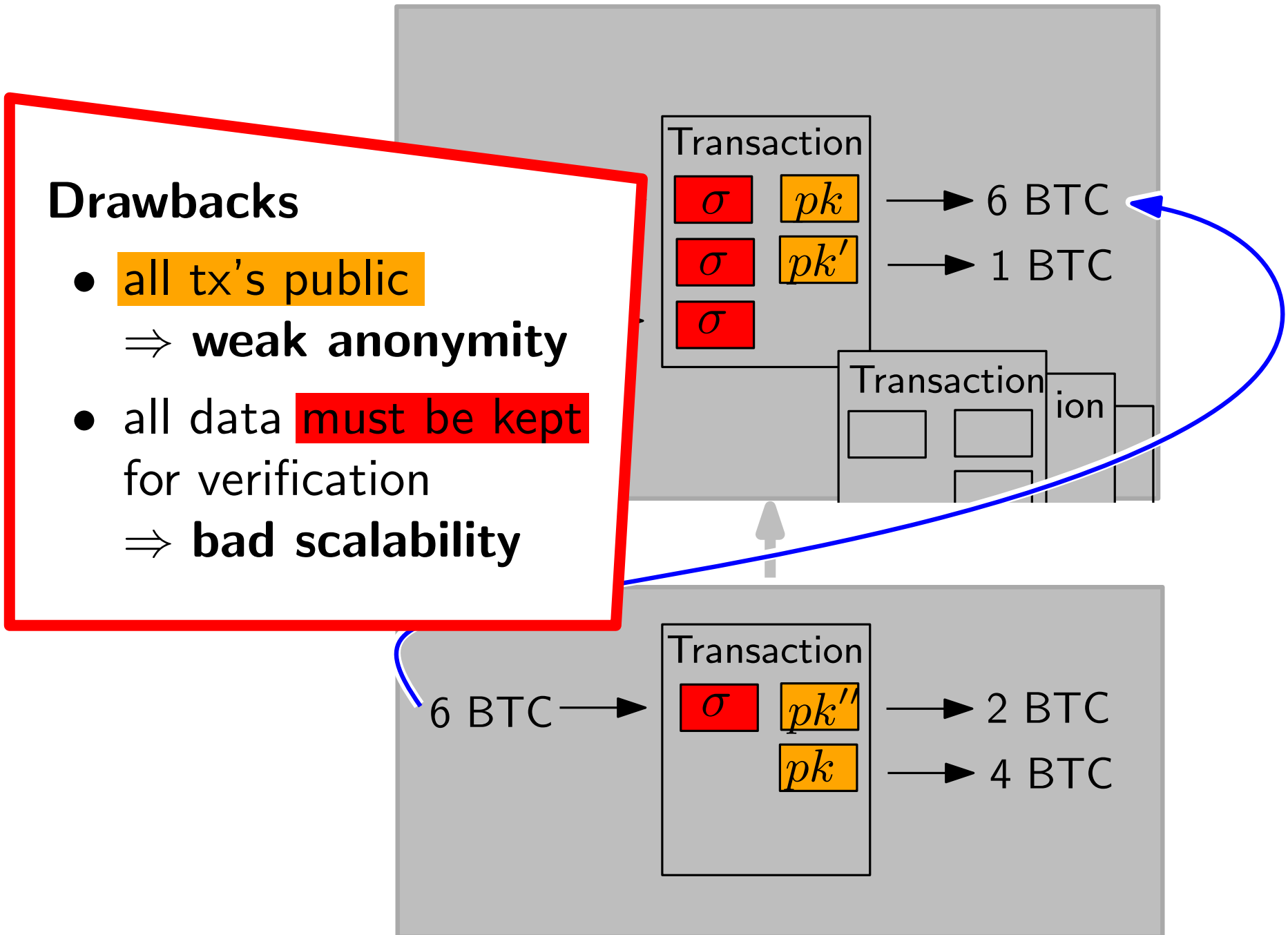
Drawbacks

- all tx's public
  $\Rightarrow$ **weak anonymity**



resulting reveals a lot of information and is subjected to analysis by many companies whose business model is to monitor and control the lower classes. This makes it very non-private and even dangerous for people to use

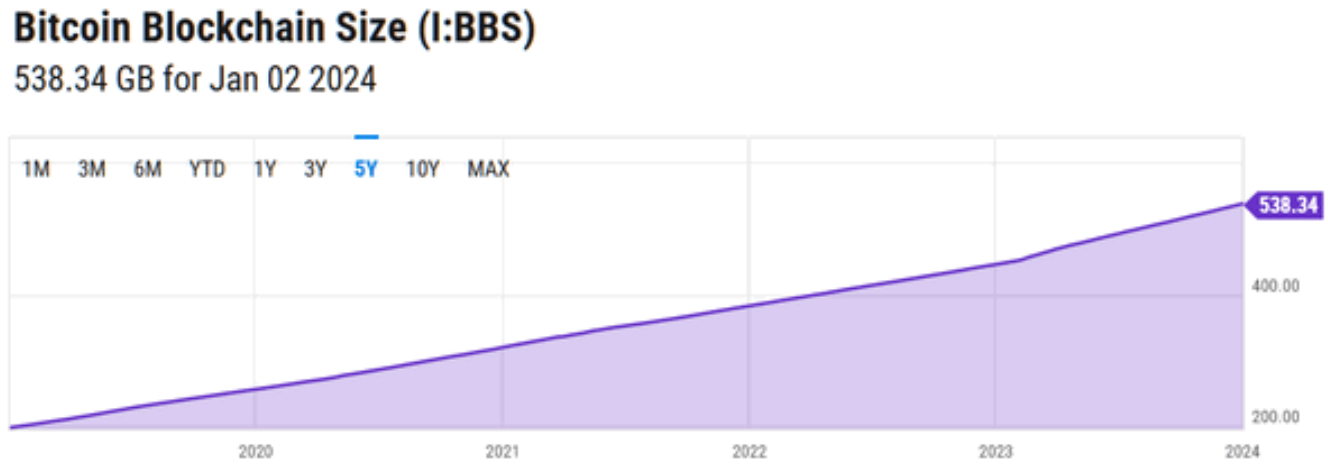
Bitcoin


Drawbacks
• all tx's public
⇒ weak anonymity
• all data must be kept for verification
⇒ bad scalability
Transaction
σ
pk
σ
pk′
σ
6 BTC
1 BTC
Transaction
ion
6 BTC
Transaction
σ
pk″
pk
2 BTC
4 BTC

Scalability

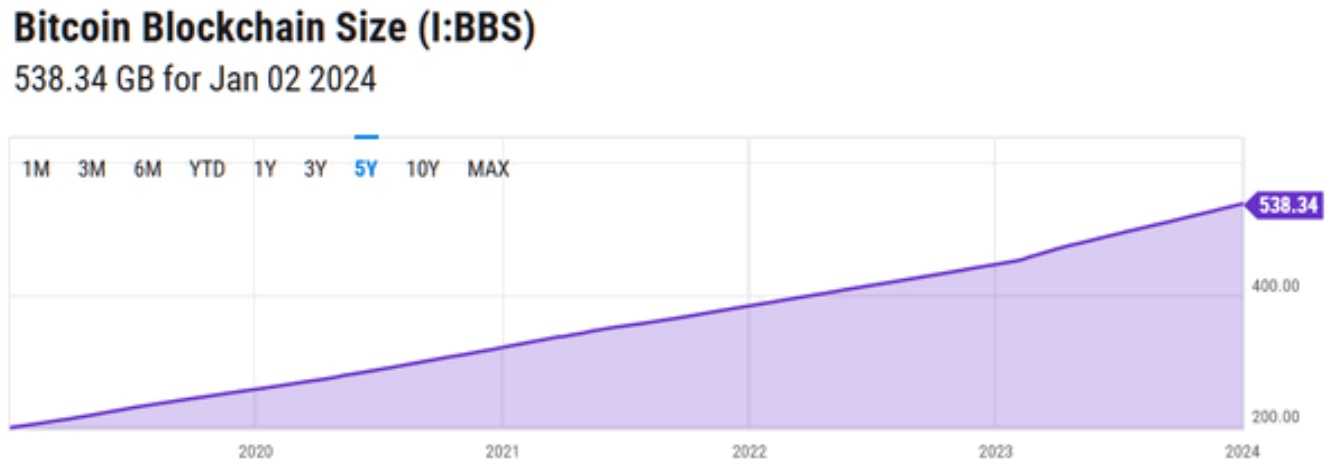
Blockchain size:

$> 500\,\text{GB}$

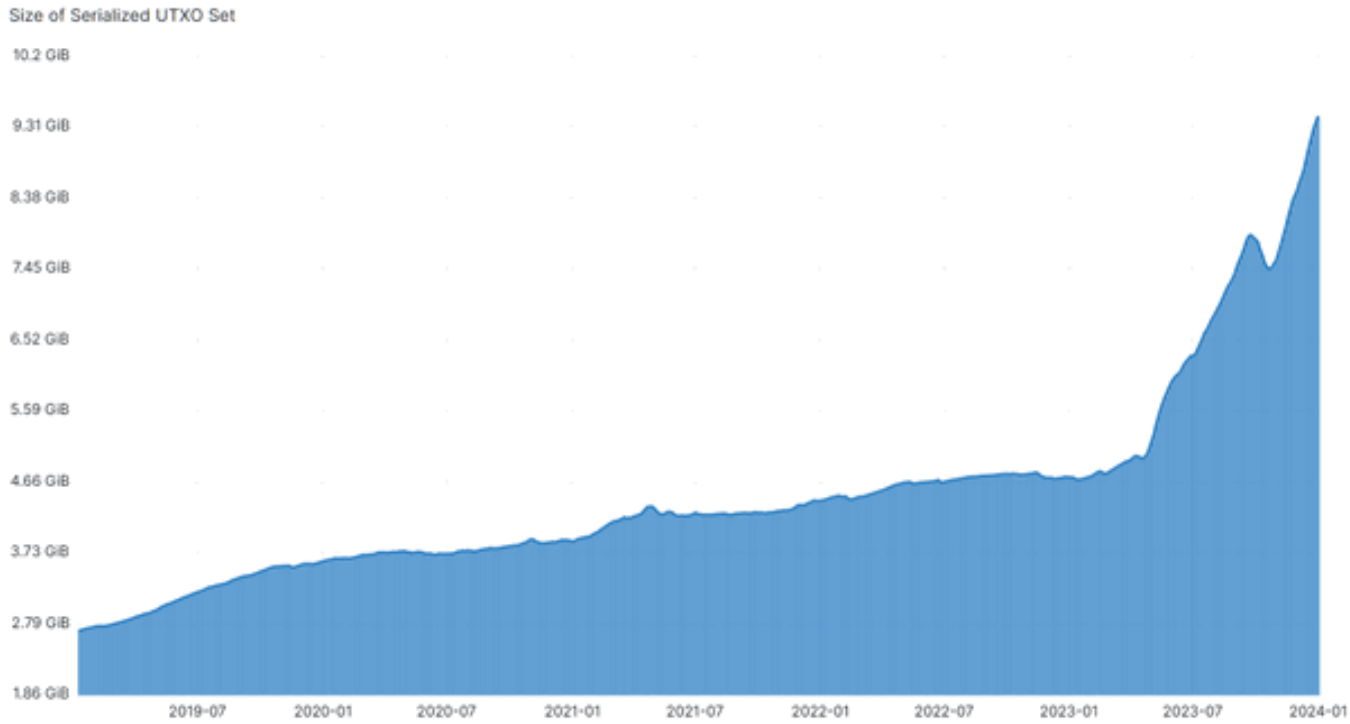


Scalability

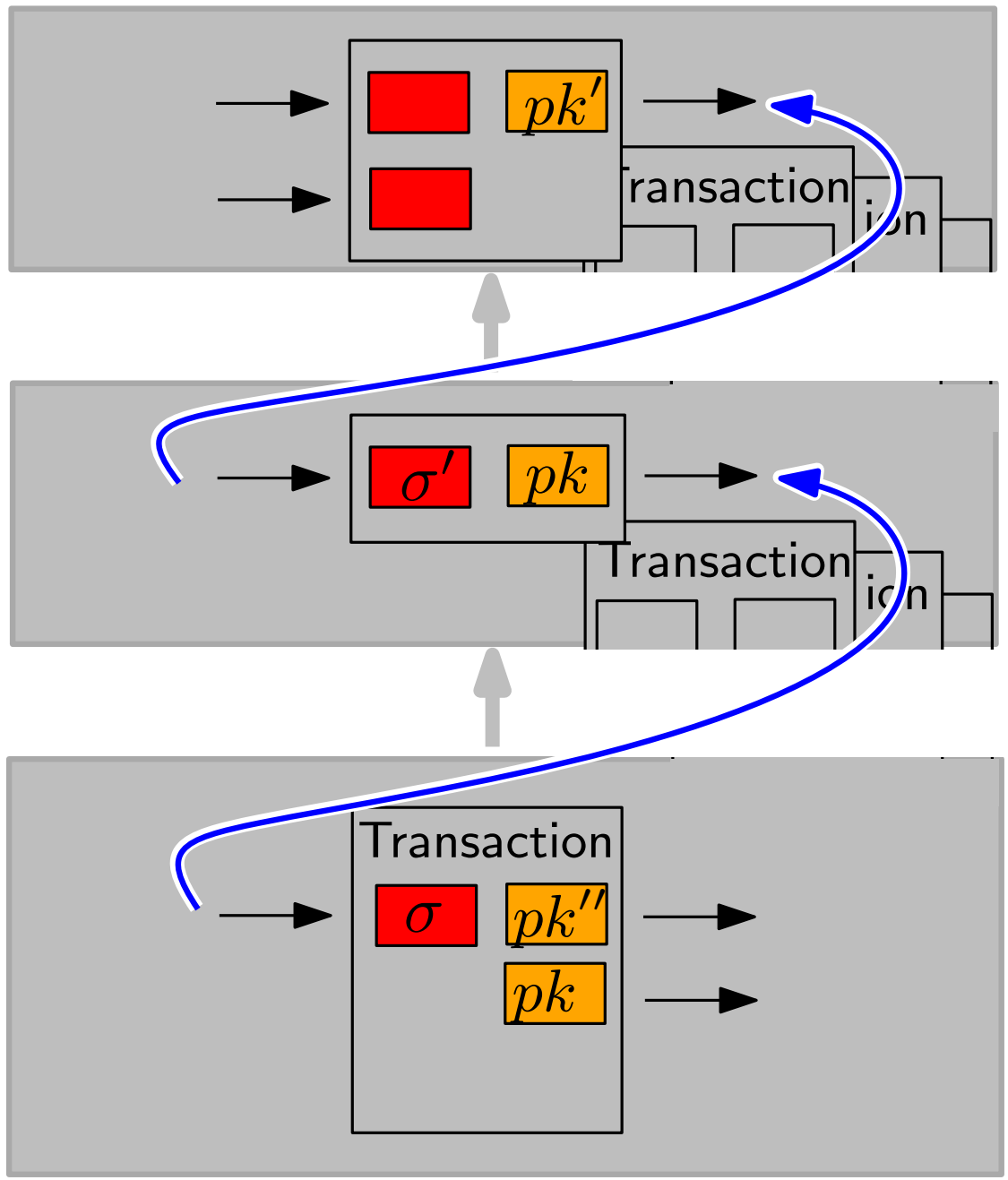
Blockchain size:
$> 500\,\text{GB}$



Size of UTXO set:
$< 10\,\text{GB}$

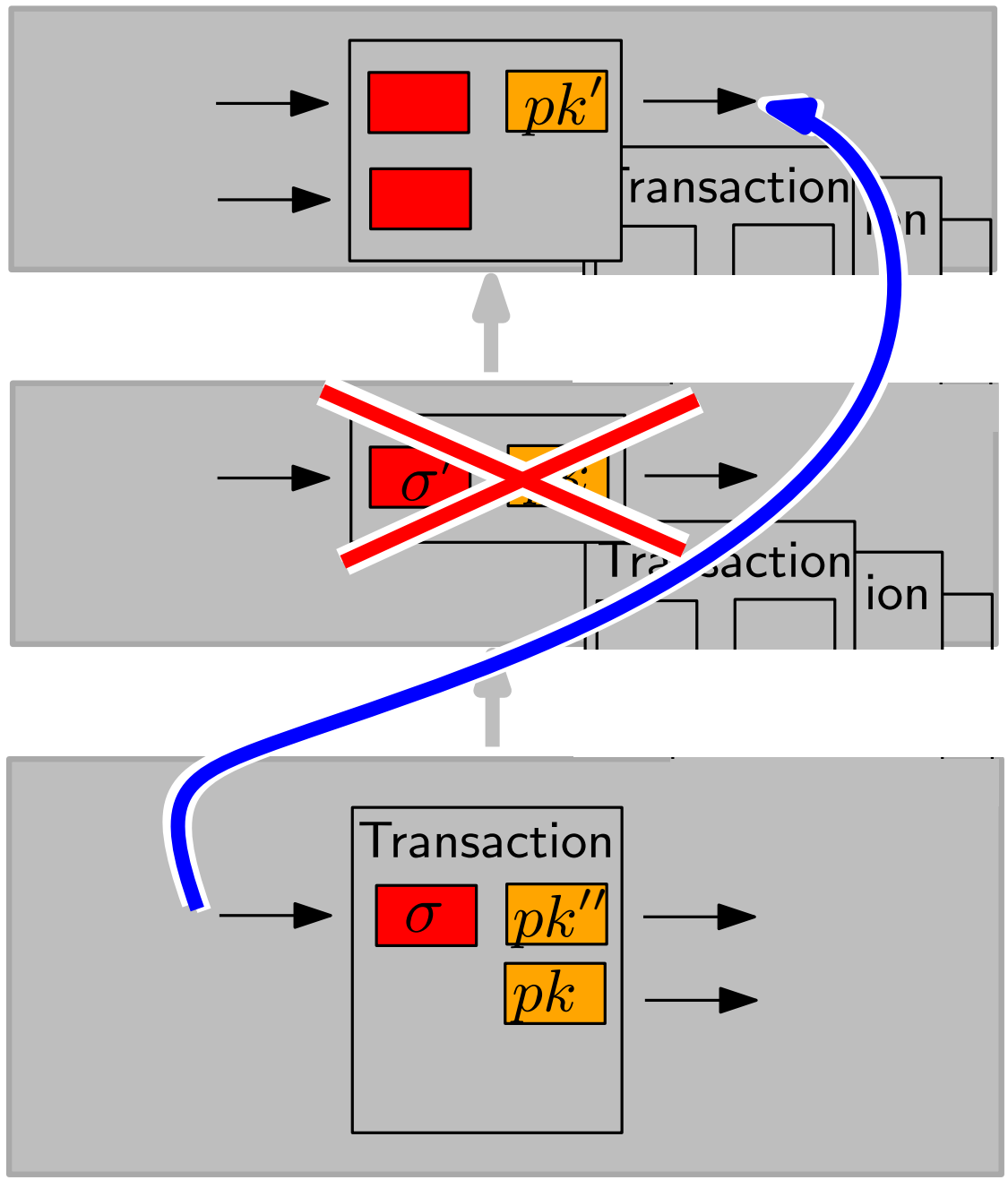


Scalability



Scalability

“cut-through”

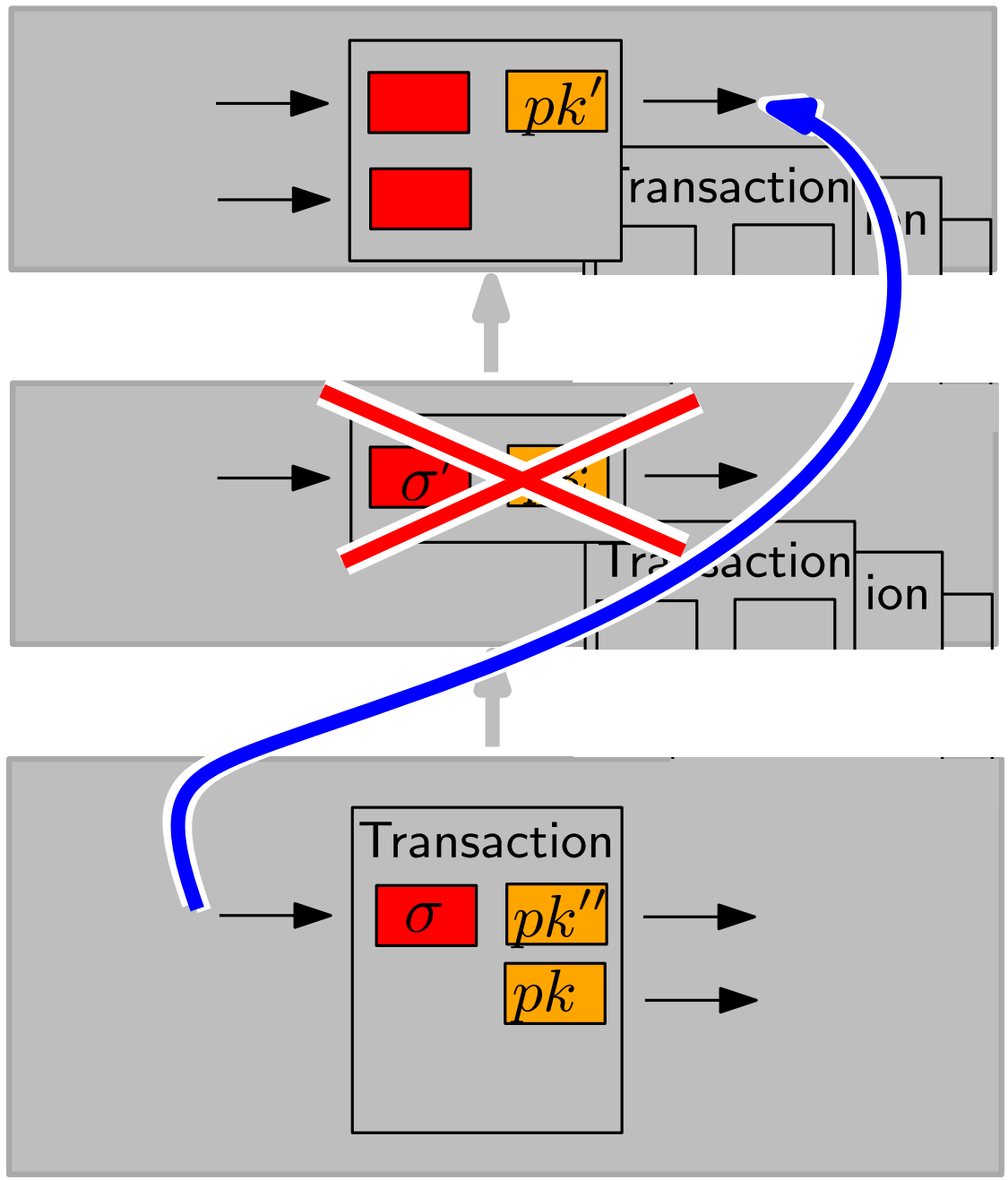


Scalability

“cut-through”

not possible
in Bitcoin:

σ' is needed
to verify validity



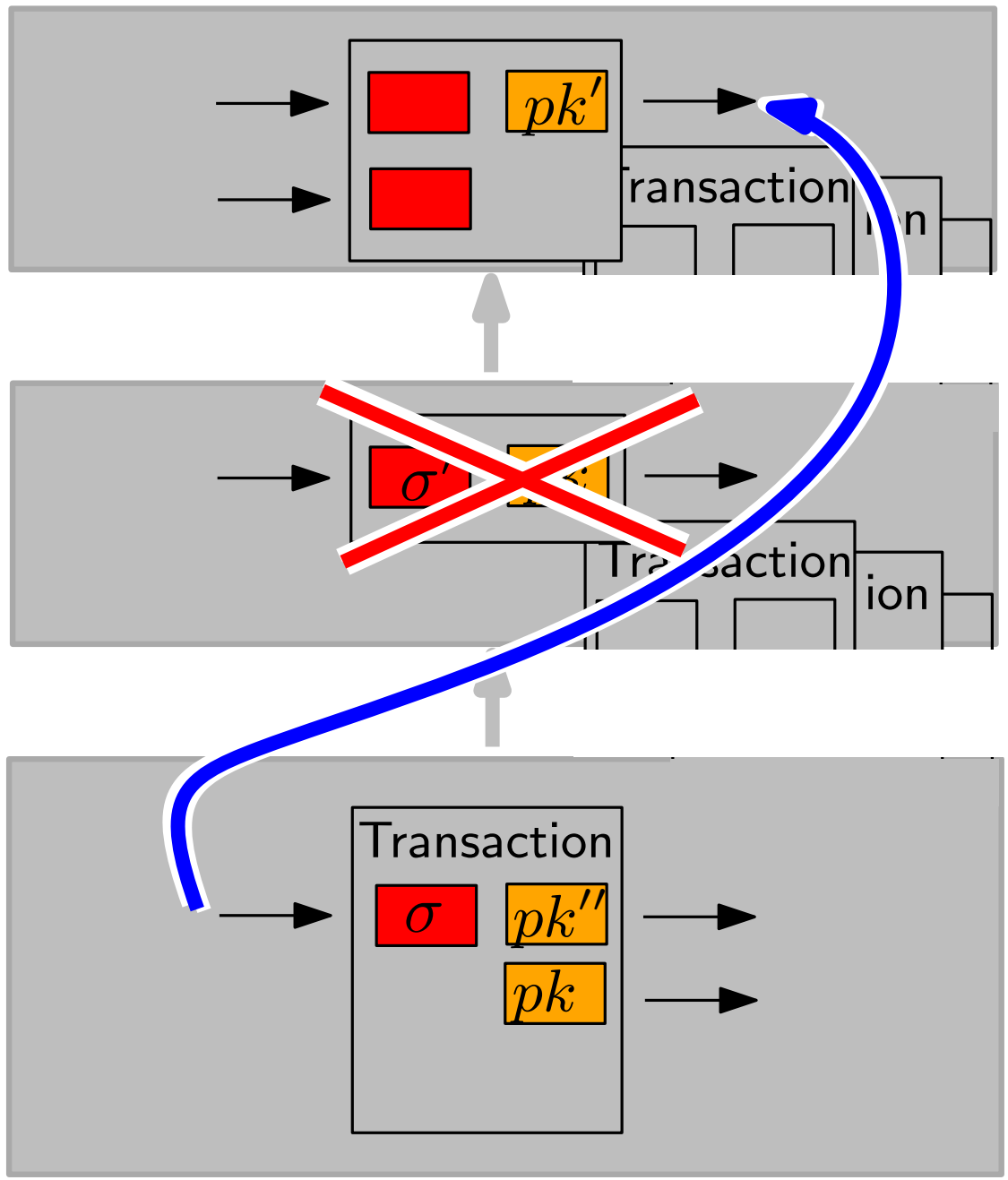
Scalability

“cut-through”

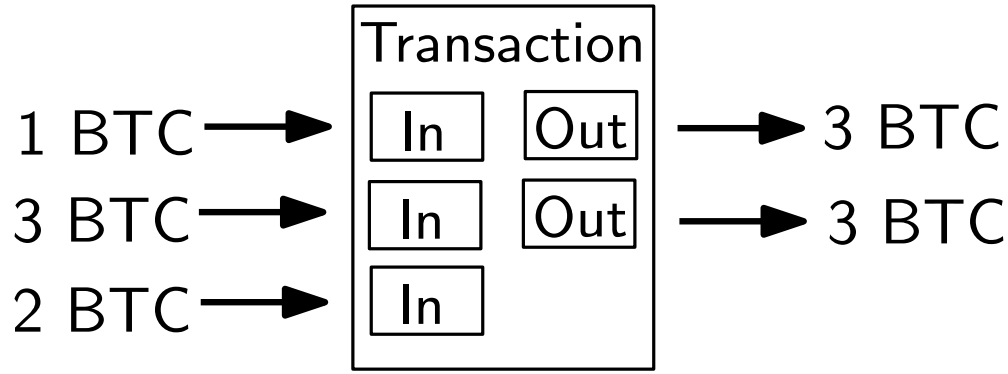
not possible
in Bitcoin:

σ' is needed
to verify validity

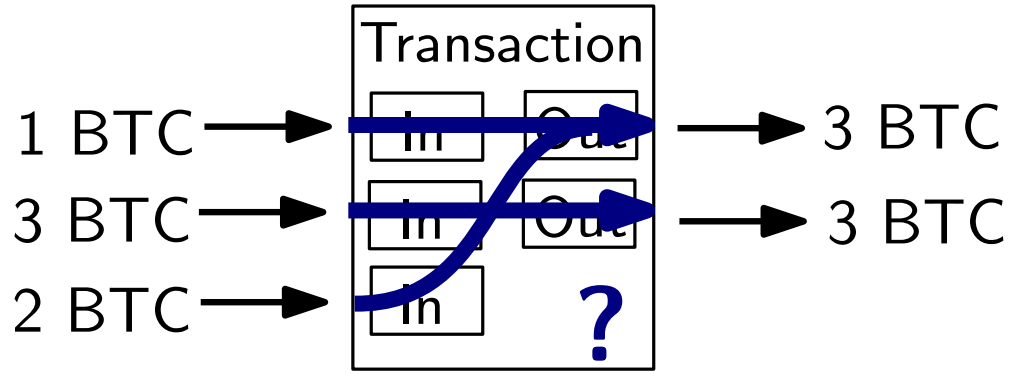
$\Rightarrow$ **Mimblewimble**



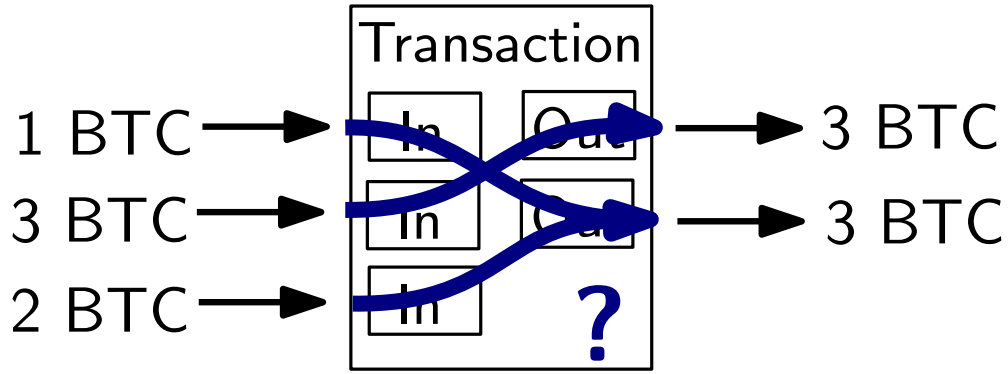
Anonymity



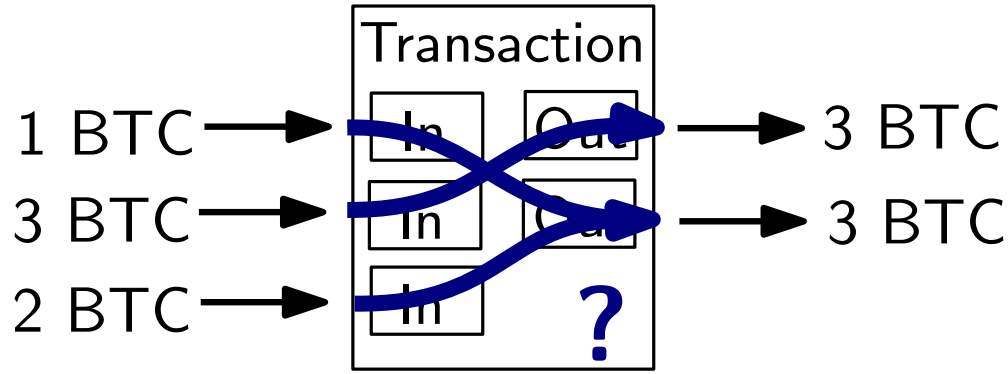
Anonymity



Anonymity

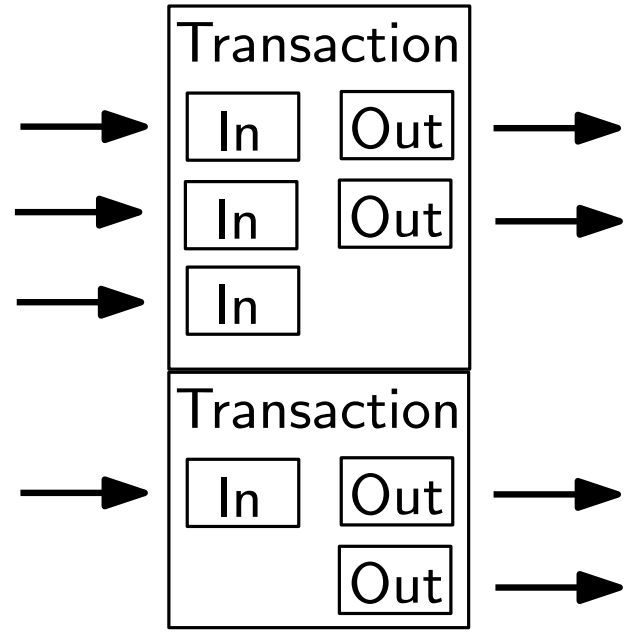


Anonymity



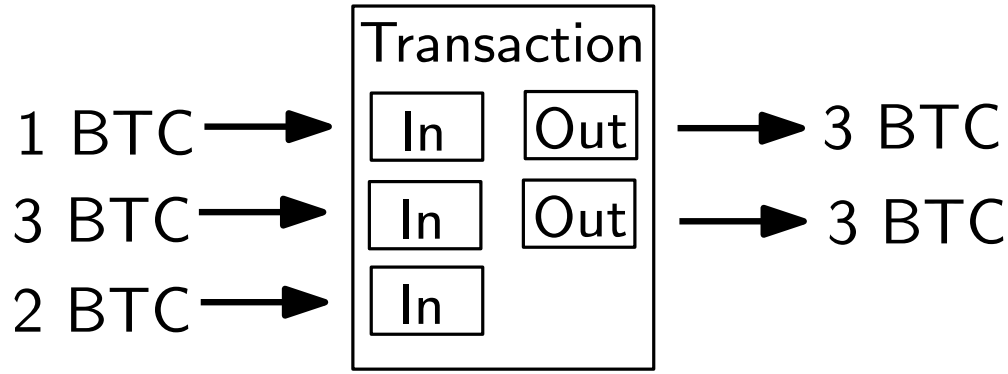
- **CoinJoin** [Maxwell'13]
  - no *link* between inputs and outputs

Anonymity

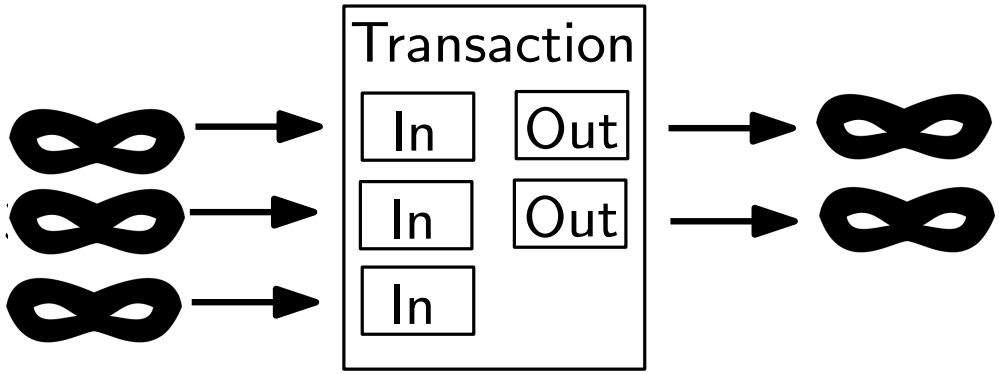


- **CoinJoin** [Maxwell'13]
  - no *link* between inputs and outputs
  - join many transactions?
  - in Bitcoin: only interactively, since all inputs must sign tx

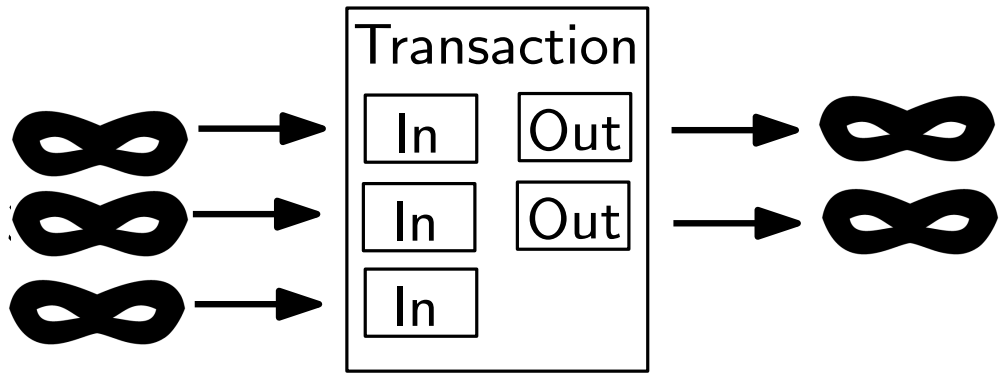
Anonymity



Anonymity

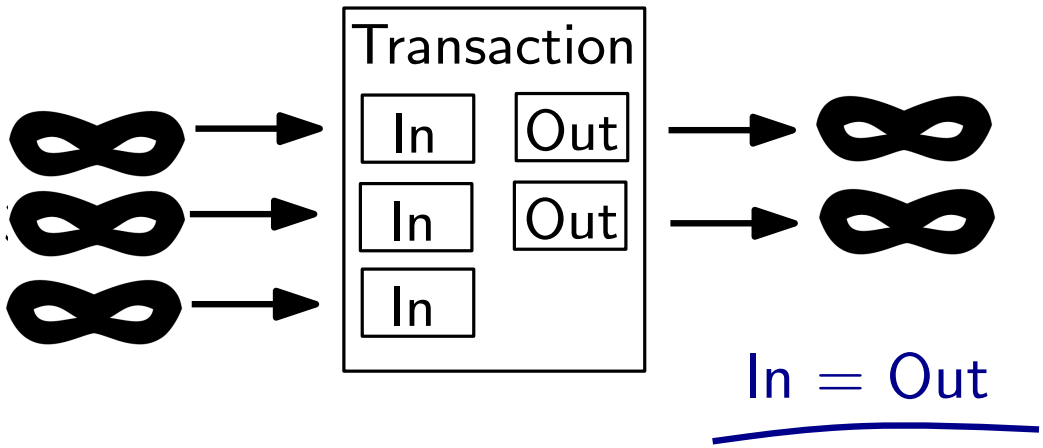


Anonymity



- **Confidential Transactions** [Maxwell]
  - hide the input and output *amounts*
  - not compatible with Bitcoin
  - balancedness verifiable?

Anonymity



- **Confidential Transactions** [Maxwell]
  - hide the input and output *amounts*
  - not compatible with Bitcoin
  - balancedness verifiable?

(by default in MONERO)

Anonymity

How can we get

- **Confidential transactions** (check balancedness)
- **Coin-join** (non-interactively)
- **Cut-through** (post-confirmation)

while **maintaining verifiability?**

- **Confider**
  - hide th
  - not co
  - balanc

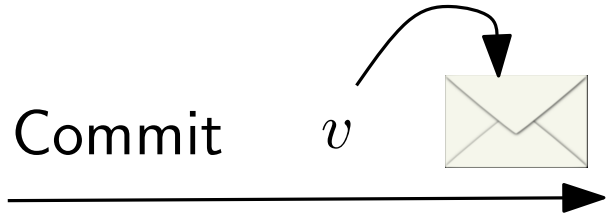
Anonymity

- **Confider**
  - hide th
  - not co
  - balanc

Pedersen commitment

Commitment

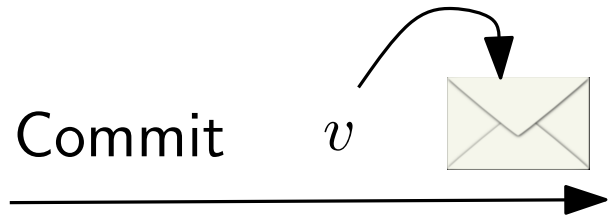
- "digital envelope"



Pedersen commitment

Commitment

- "digital envelope"

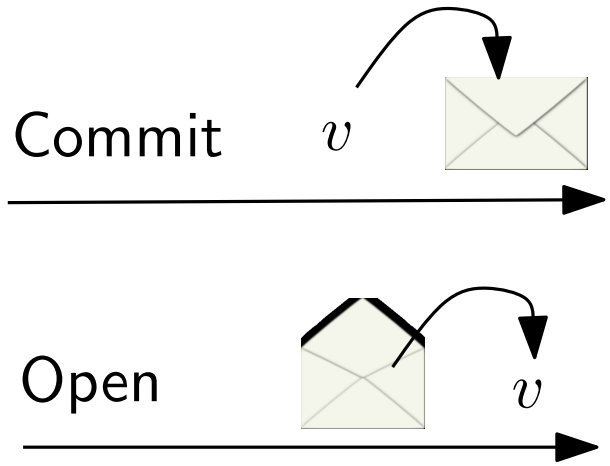


- **hiding:** commitment hides v

Pedersen commitment

Commitment

- "digital envelope"

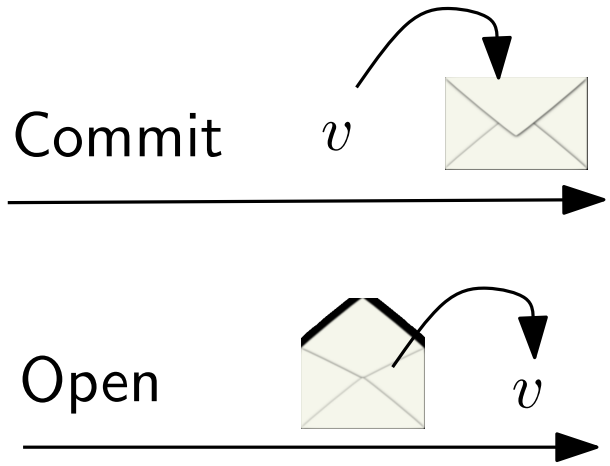


- **hiding:** commitment hides v

Pedersen commitment

Commitment

- "digital envelope"



- **hiding:** commitment hides v
- **binding:** Alice can open commitment only to one value

Pedersen commitment

Commitment

- "digital envelope"

Commit v

Open

Pedersen

$G, H \in \mathbb{G}$

pick random r

$C := vH + rG$

reveal v and r

Pedersen commitment

Commitment

- "digital envelope"

Commit v

Open

Pedersen

$G, H \in \mathbb{G}$

pick random r

$C := vH + rG$

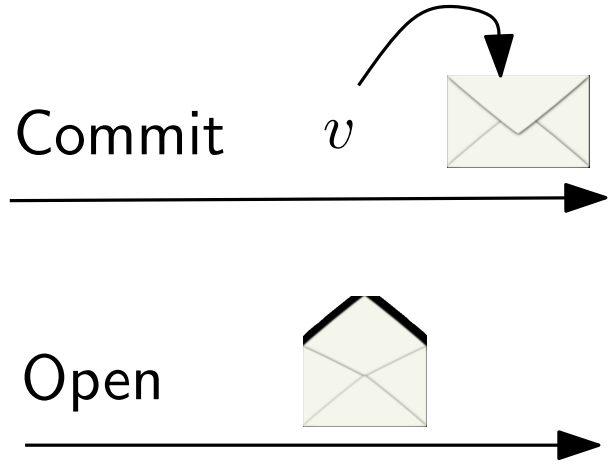
reveal v and r

- **hiding:** for any v exists r so that C commits v

Pedersen commitment

Commitment

- "digital envelope"



Pedersen

$G, H \in \mathbb{G}$

pick random r

$C := vH + rG$

reveal v and r

- **binding:** from $v \neq v', r, r'$ with

$$vH + rG = C = v'H + r'G$$

$$\Rightarrow \text{ compute } \log_G H$$

Pedersen commitment

Commitment

- "digital envelope"

Commit v

Open

Pedersen

$G, H \in \mathbb{G}$

pick random r

Com$(v; r) := vH + rG$

reveal v and r

- commitments are **homomorphic**:

$$\textbf{Com}(v_1; r_1) + \textbf{Com}(v_2; r_2) = (v_1 H + r_1 G) + (v_2 H + r_2 G)$$

Pedersen commitment

Commitment

- "digital envelope"

Commit v

Open

Pedersen

$G, H \in \mathbb{G}$

pick random r

Com$(v; r) := vH + rG$

reveal v and r

- commitments are **homomorphic**:

$$\begin{aligned}\textbf{Com}(v_1; r_1) + \textbf{Com}(v_2; r_2) &= (v_1 H + r_1 G) + (v_2 H + r_2 G) \\ &= (v_1 + v_2)H + (r_1 + r_2)G \\ &= \textbf{Com}(v_1 + v_2; r_1 + r_2)\end{aligned}$$

Pedersen commitment

Commitment

- "digital envelope"

Commit v

Open

Pedersen

$G, H \in \mathbb{G}$

pick random r

Com$(v; r) := vH + rG$

reveal v and r

- commitments are **homomorphic**:

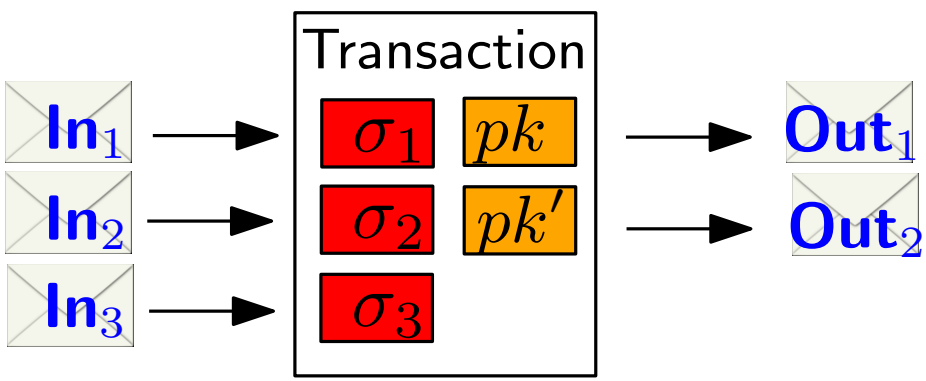
$$\begin{aligned}\textbf{Com}(v_1; r_1) + \textbf{Com}(v_2; r_2) &= (v_1 H + r_1 G) + (v_2 H + r_2 G) \\ &= (v_1 + v_2)H + (r_1 + r_2)G \\ &= \textbf{Com}(v_1 + v_2; r_1 + r_2)\end{aligned}$$

e.g.: $\textbf{Com}(1; 5) + \textbf{Com}(1; 10) - \textbf{Com}(2; 15) = \mathbf{0}$

Confidential Transactions

[Back, Maxwell '13 – '15]

- use *commitments* to amounts

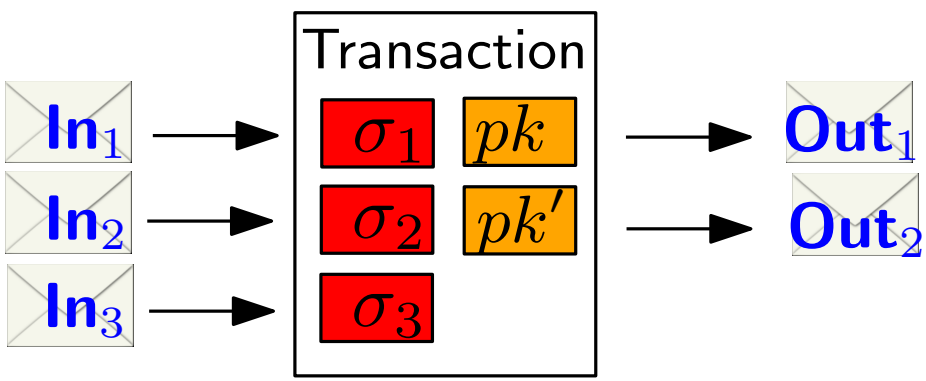


$$C = vH + rG$$

Confidential Transactions

[Back, Maxwell '13 – '15]

- use *commitments* to amounts
- ensure that transactions do not create money?



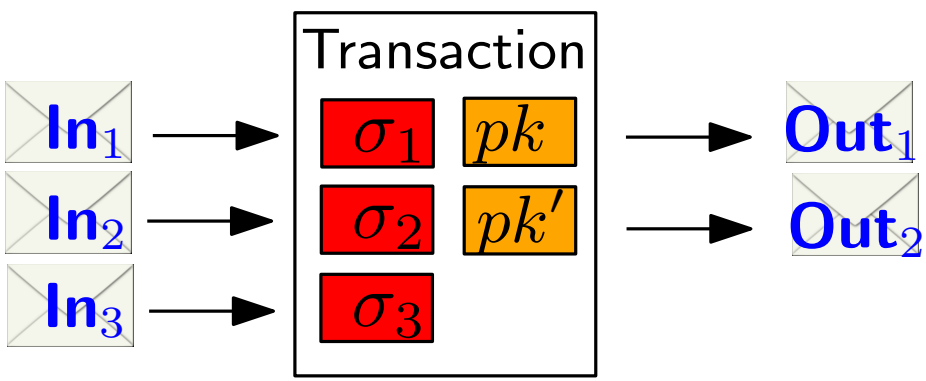
$$C = vH + rG$$

$$\mathbf{Out}_1 + \ldots + \mathbf{Out}_n - \mathbf{In}_1 - \ldots - \mathbf{In}_\ell = 0$$

Confidential Transactions

[Back, Maxwell '13 – '15]

- use *commitments* to amounts
- ensure that transactions do not create money?



$$C = vH + rG$$

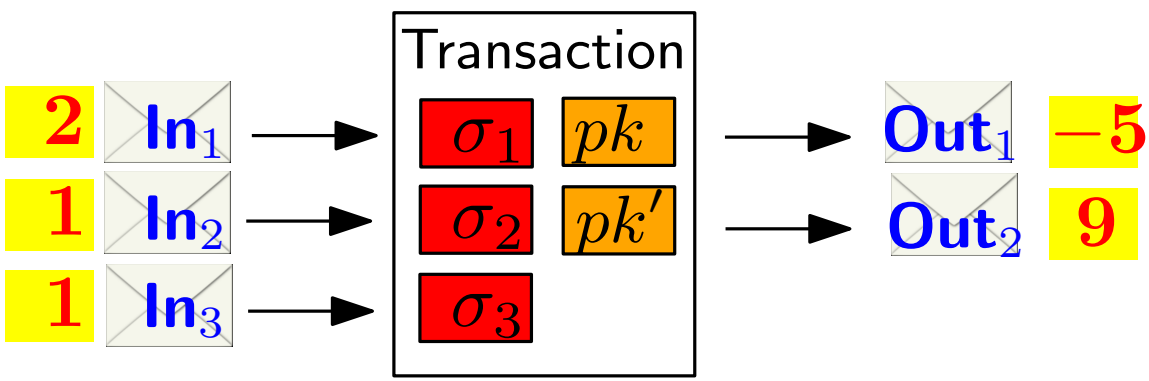
$$\sum \mathbf{Out} - \sum \mathbf{In} = 0$$

$$\begin{aligned}
&\sum C_i^{\text{out}} - \sum C_i^{\text{in}} \\
&= \sum (v_i^{\text{out}} H + r_i^{\text{out}} G) - \sum (v_i^{\text{in}} H + r_i^{\text{in}} G) \\
&= (\underbrace{\sum v_i^{\text{out}} - \sum v_i^{\text{in}}}_{\stackrel{!}{=}0}) H + (\underbrace{\sum r_i^{\text{out}} - \sum r_i^{\text{in}}}_{\stackrel{!}{=}0}) G
\end{aligned}$$

Confidential Transactions

[Back, Maxwell '13 – '15]

- use *commitments* to amounts
- ensure that transactions do not create money?



$$C = vH + rG$$

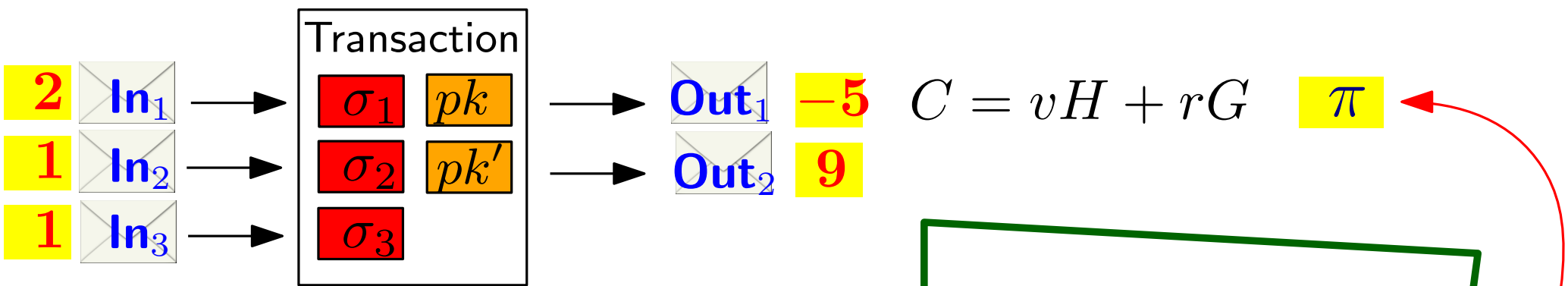
$$\sum \textbf{Out} - \sum \textbf{In} = 0$$

- negative amounts!

Confidential Transactions

[Back, Maxwell '13 – '15]

- use *commitments* to amounts
- ensure that transactions do not create money?



- negative amounts!

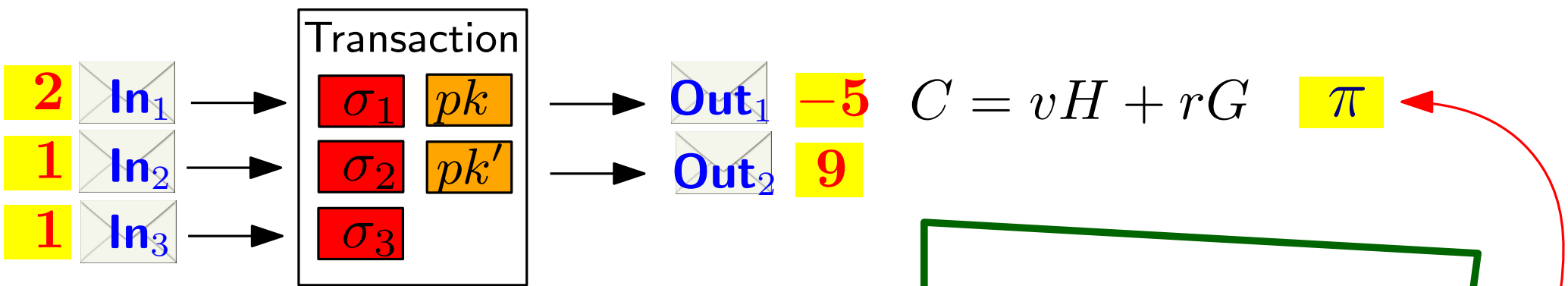
Range proofs

– add proofs that committed values are in $\in [0, 2^{64}]$

Confidential Transactions

[Back, Maxwell '13 – '15]

- use *commitments* to amounts
- ensure that transactions do not create money?



$$\sum \mathbf{Out} - \sum \mathbf{In} = 0$$

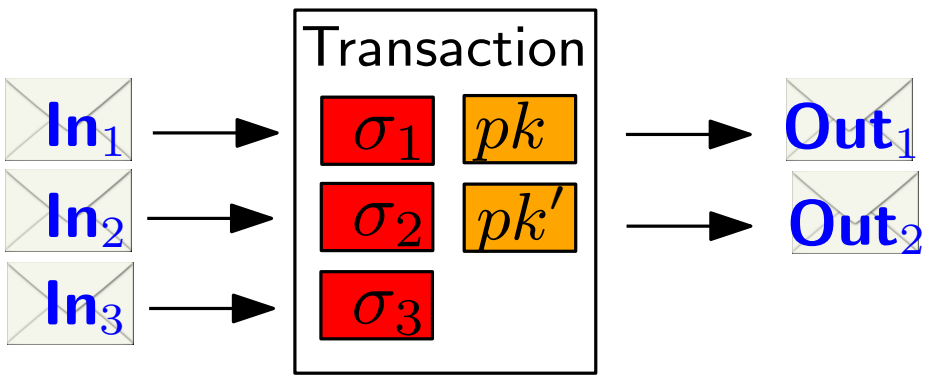
- negative amounts!

Range proofs

– add proofs that committed values are in $\in [0, 2^{64}]$

Confidential Transactions

Confidential transaction

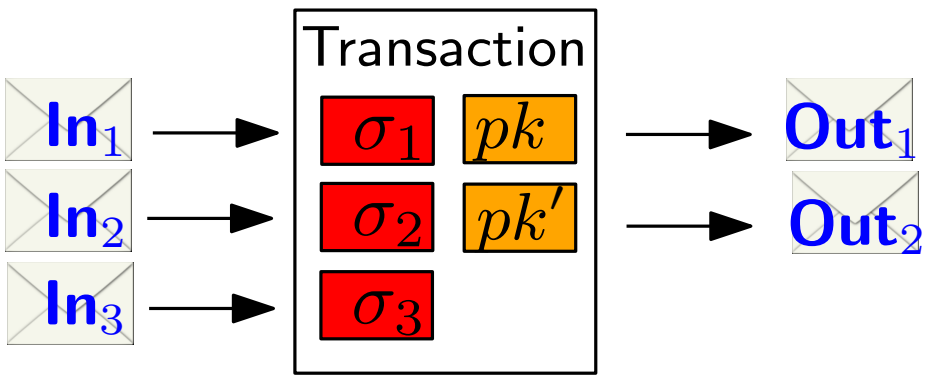


$$C = vH + rG, \quad \pi$$

$$\sum \mathbf{Out} - \sum \mathbf{In} = 0$$

Confidential Transactions

Confidential transaction



$$C = vH + rG, \quad \pi$$

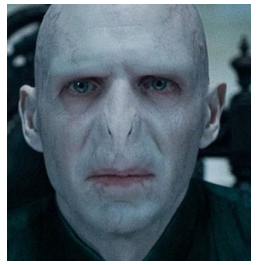
$$\sum \mathbf{Out} - \sum \mathbf{In} = 0$$

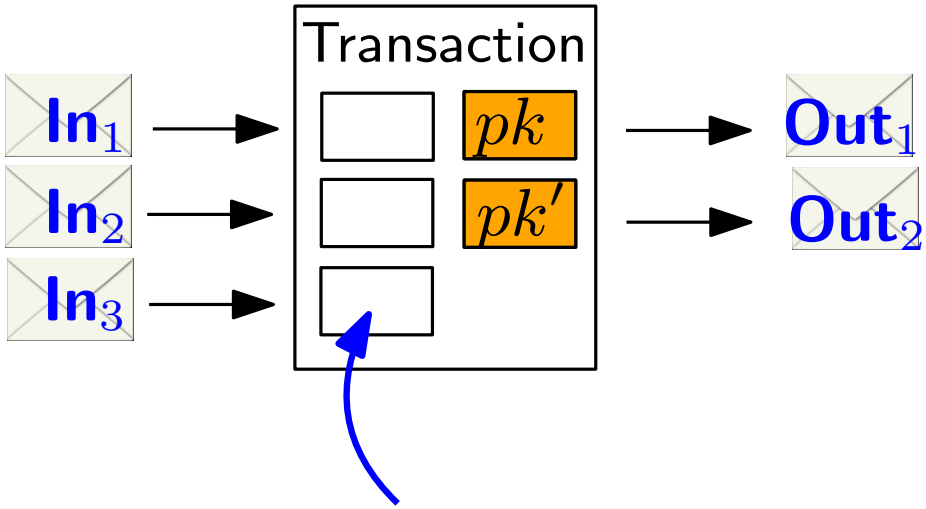
Signatures $\Rightarrow$

- no non-interactive CoinJoin
- no Cut-Through

Mimblewimble

[Jedusor '16]





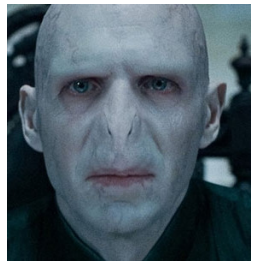
$$C = vH + rG, \quad \pi$$

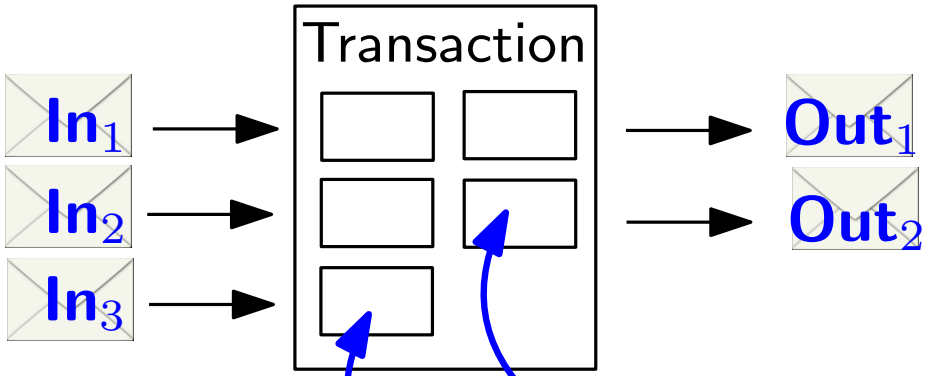
$$\sum \mathbf{Out} - \sum \mathbf{In} = 0$$

no more signatures!

Mimblewimble

[Jedusor '16]





$$C = vH + rG, \quad \pi$$

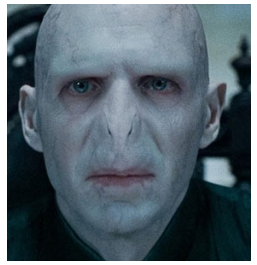
$$\sum \textbf{Out} - \sum \textbf{In} = 0$$

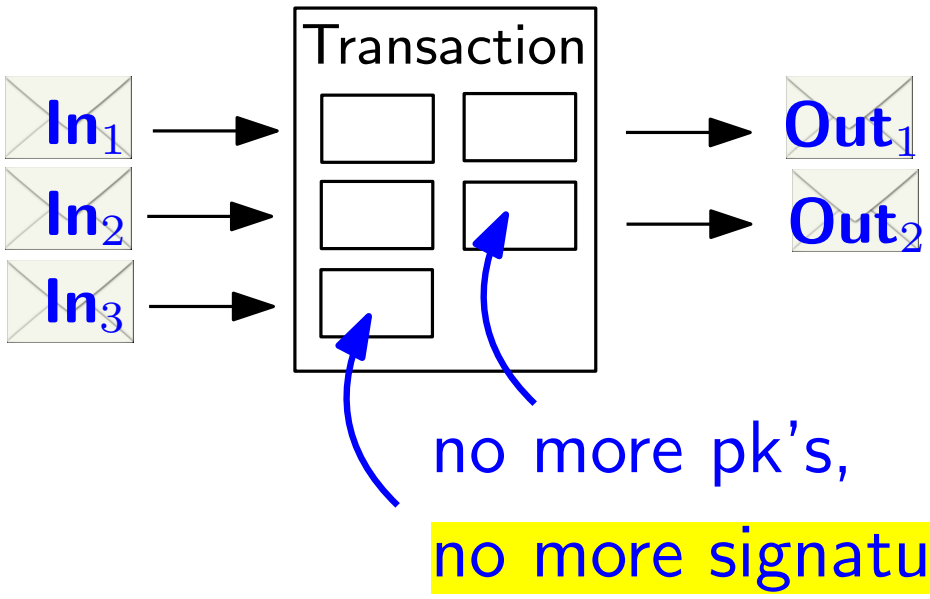
no more pk's,

no more signatures!

Mimblewimble

[Jedusor '16]





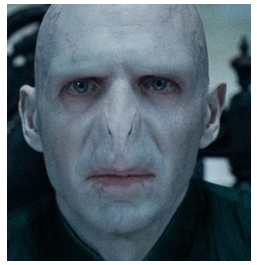
secret key!

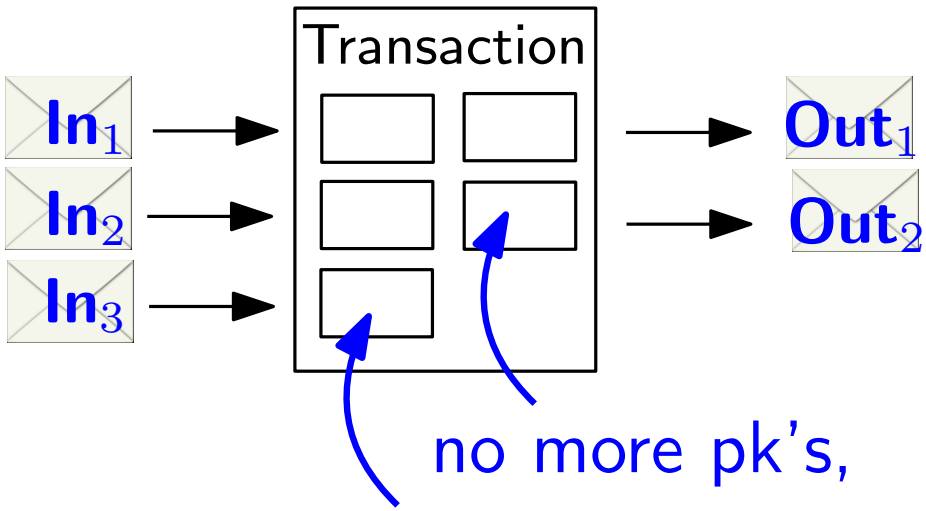
$$C = vH + rG, \quad \pi$$

$$\sum \mathbf{Out} - \sum \mathbf{In} = 0$$

Mimblewimble

[Jedusor '16]





secret key!

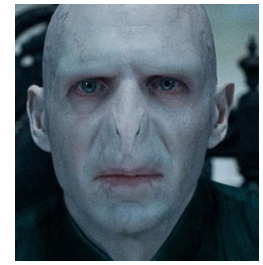
$$C = vH + rG, \quad \pi$$

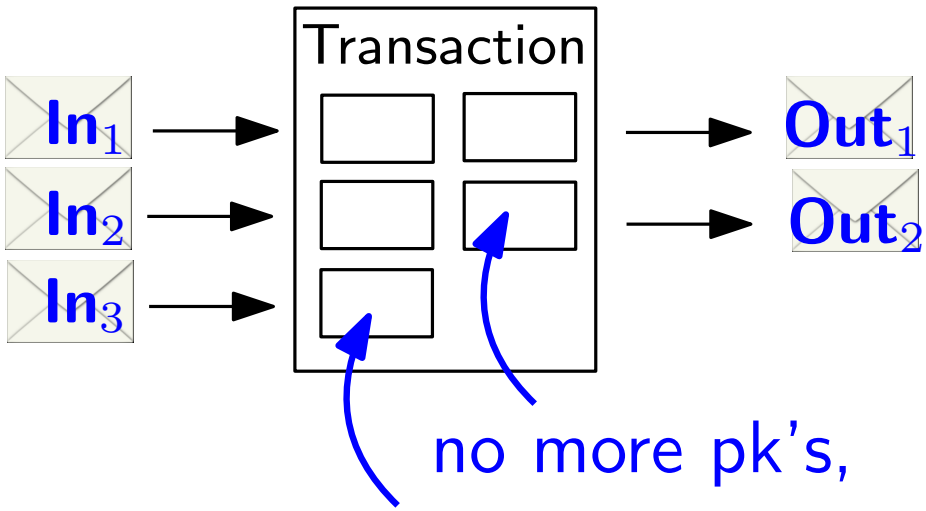
$$\sum \mathbf{Out} - \sum \mathbf{In} = \mathbf{0}$$

But: sender knows sum of output r's

Mimblewimble

[Jedusor '16]





secret key!

$$C = vH + rG, \quad \pi$$

$$\sum \mathbf{Out} - \sum \mathbf{In} = 0$$

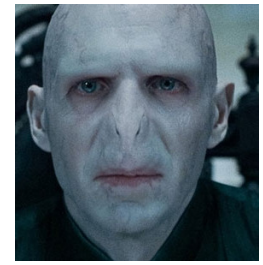
no more pk's,

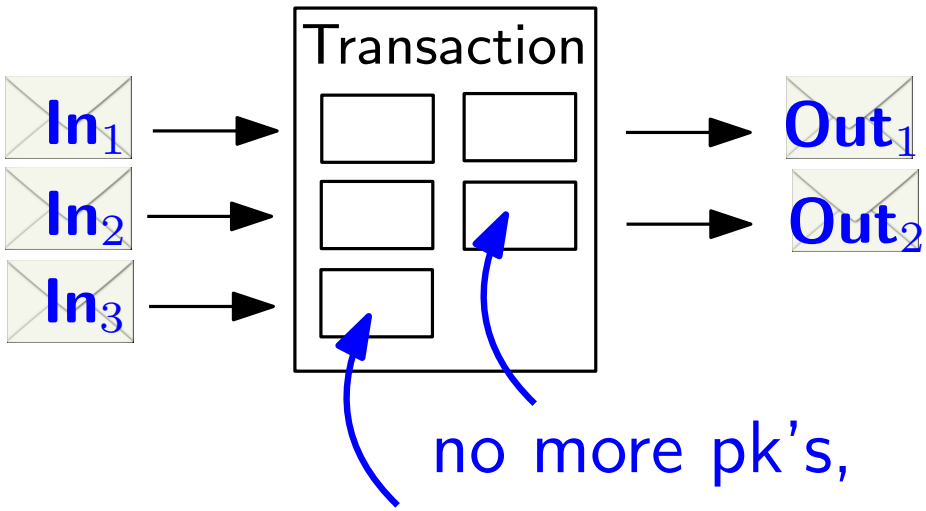
no more signatures!

$$\begin{aligned}
&\sum C_i^{\text{out}} - \sum C_i^{\text{in}} \\
&= \sum (v_i^{\text{out}} H + r_i^{\text{out}} G) - \sum (v_i^{\text{in}} H + r_i^{\text{in}} G) \\
&= (\underbrace{\sum v_i^{\text{out}} - \sum v_i^{\text{in}}}_{\stackrel{!}{=}0}) H + (\underbrace{\sum r_i^{\text{out}} - \sum r_i^{\text{in}}}_{=:x}) G
\end{aligned}$$

Mimblewimble

[Jedusor '16]





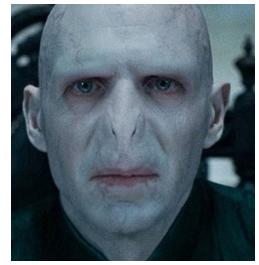
$$C = vH + rG, \quad \pi$$

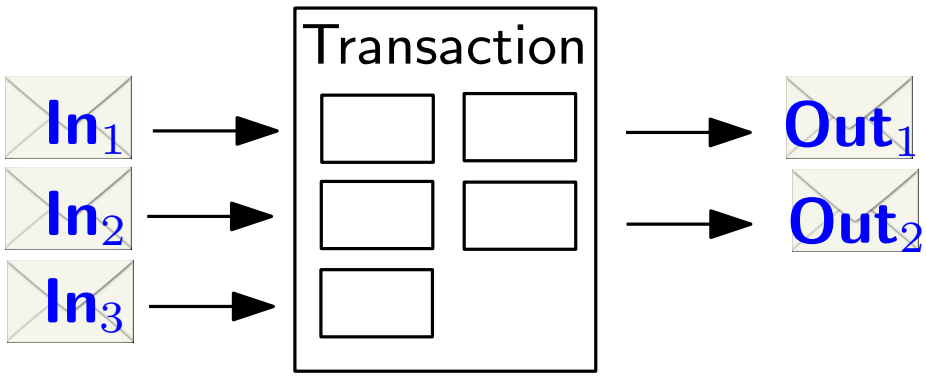
$$\sum \textbf{Out} - \sum \textbf{In} = 0H + xG$$

$$\begin{aligned}
&\sum C_i^{\text{out}} - \sum C_i^{\text{in}} \\
&= \sum (v_i^{\text{out}} H + r_i^{\text{out}} G) - \sum (v_i^{\text{in}} H + r_i^{\text{in}} G) \\
&= (\underbrace{\sum v_i^{\text{out}} - \sum v_i^{\text{in}}}_{\stackrel{!}{=}0}) H + (\underbrace{\sum r_i^{\text{out}} - \sum r_i^{\text{in}}}_{=:x}) G
\end{aligned}$$

Mimblewimble

[Jedusor '16]





secret key!

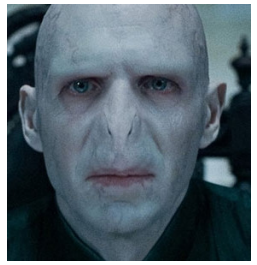
$$C = vH + rG, \quad \pi$$

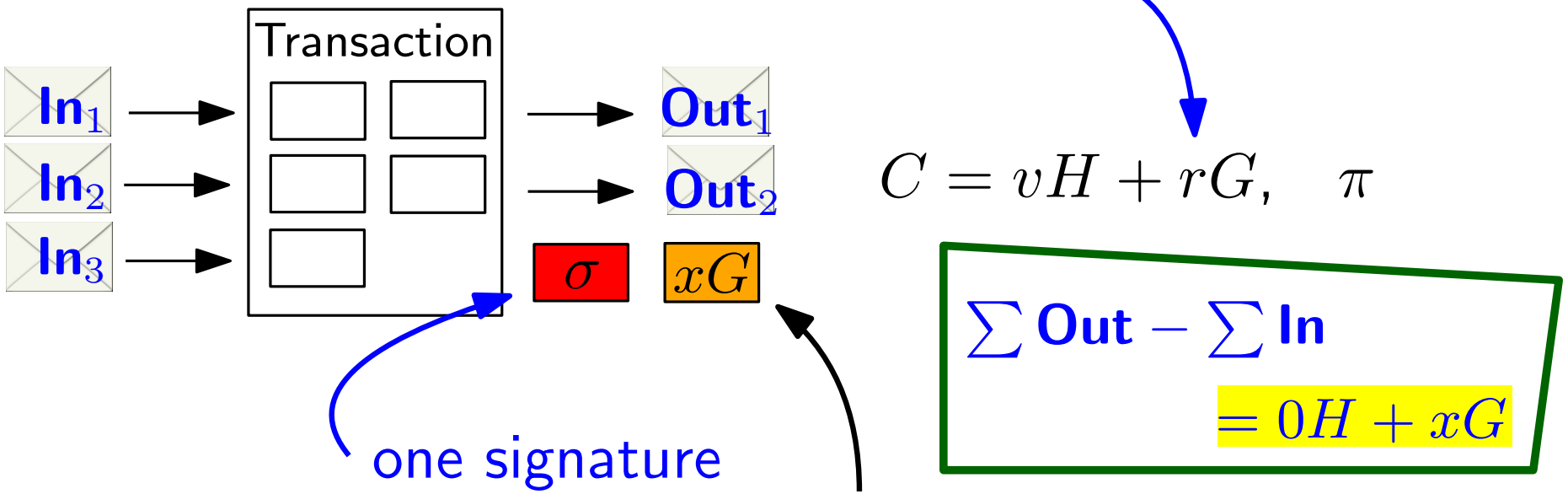
$$\sum \textbf{Out} - \sum \textbf{In} = 0H + xG$$

$\Rightarrow$ prove that $\sum \text{Out} - \sum \text{In}$ is commitment to 0

Mimblewimble

[Jedusor '16]



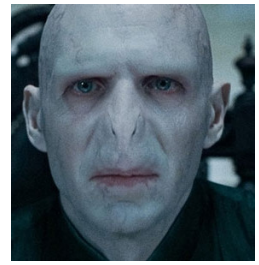


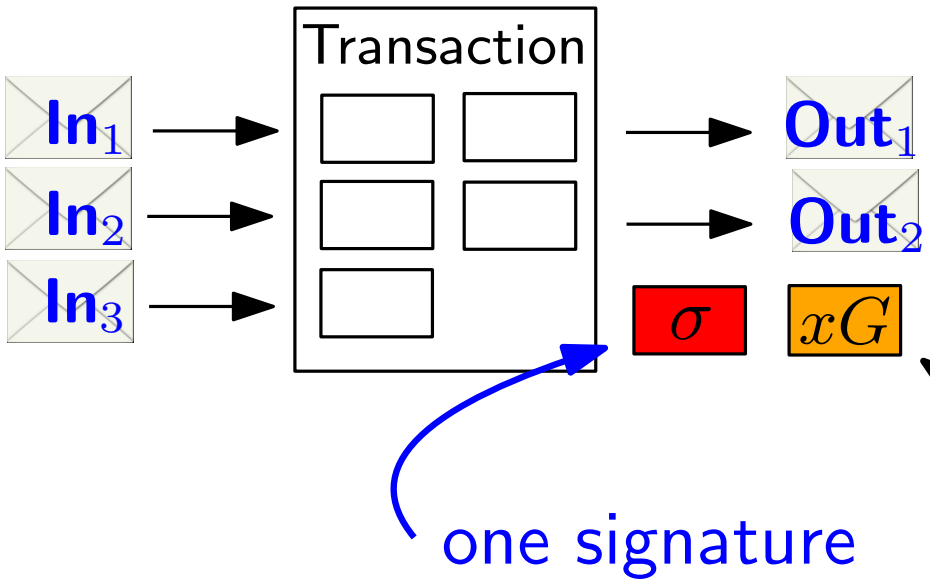
"proves" that $\overbrace{\sum \text{Out} - \sum \text{In}}$

is commitment to 0

Mimblewimble

[Jedusor '16]





secret key!

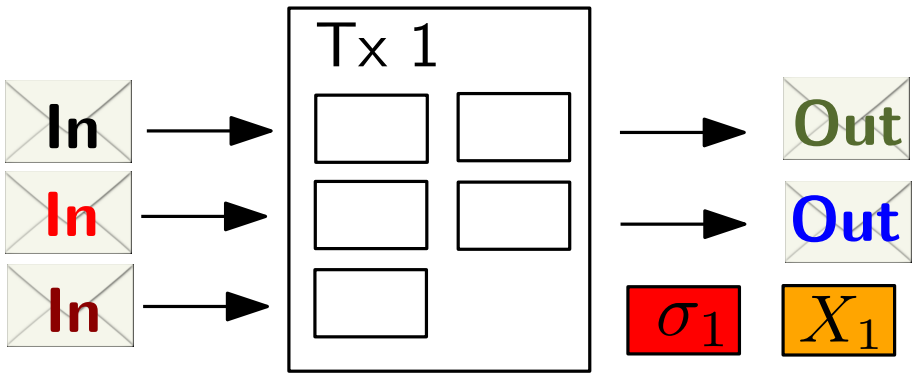
$$C = vH + rG, \quad \pi$$

$$\sum \textbf{Out} - \sum \textbf{In} = 0H + xG$$

one signature

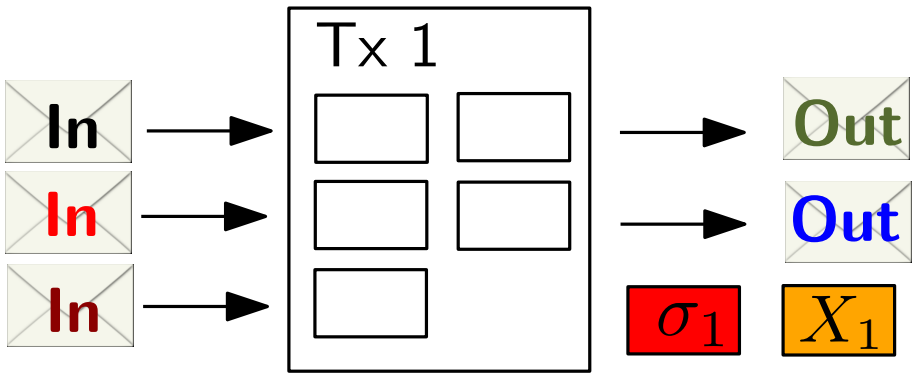
"proves" that $\overbrace{\sum \text{Out} - \sum \text{In}}$ is commitment to 0

Mimblewimble

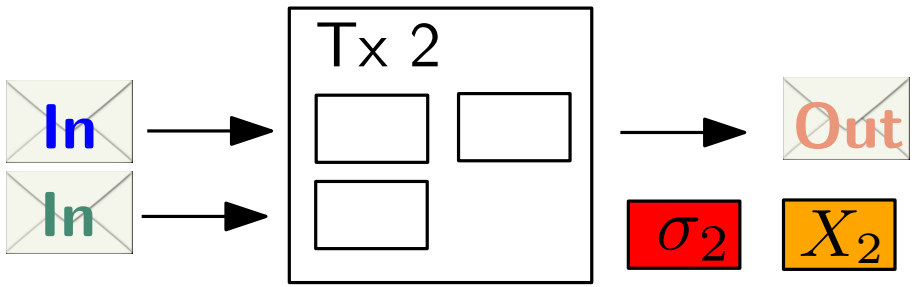


- $\sum \mathbf{Out}_1 - \sum \mathbf{In}_1 = X_1$
- σ_1 valid for X_1

Mimblewimble



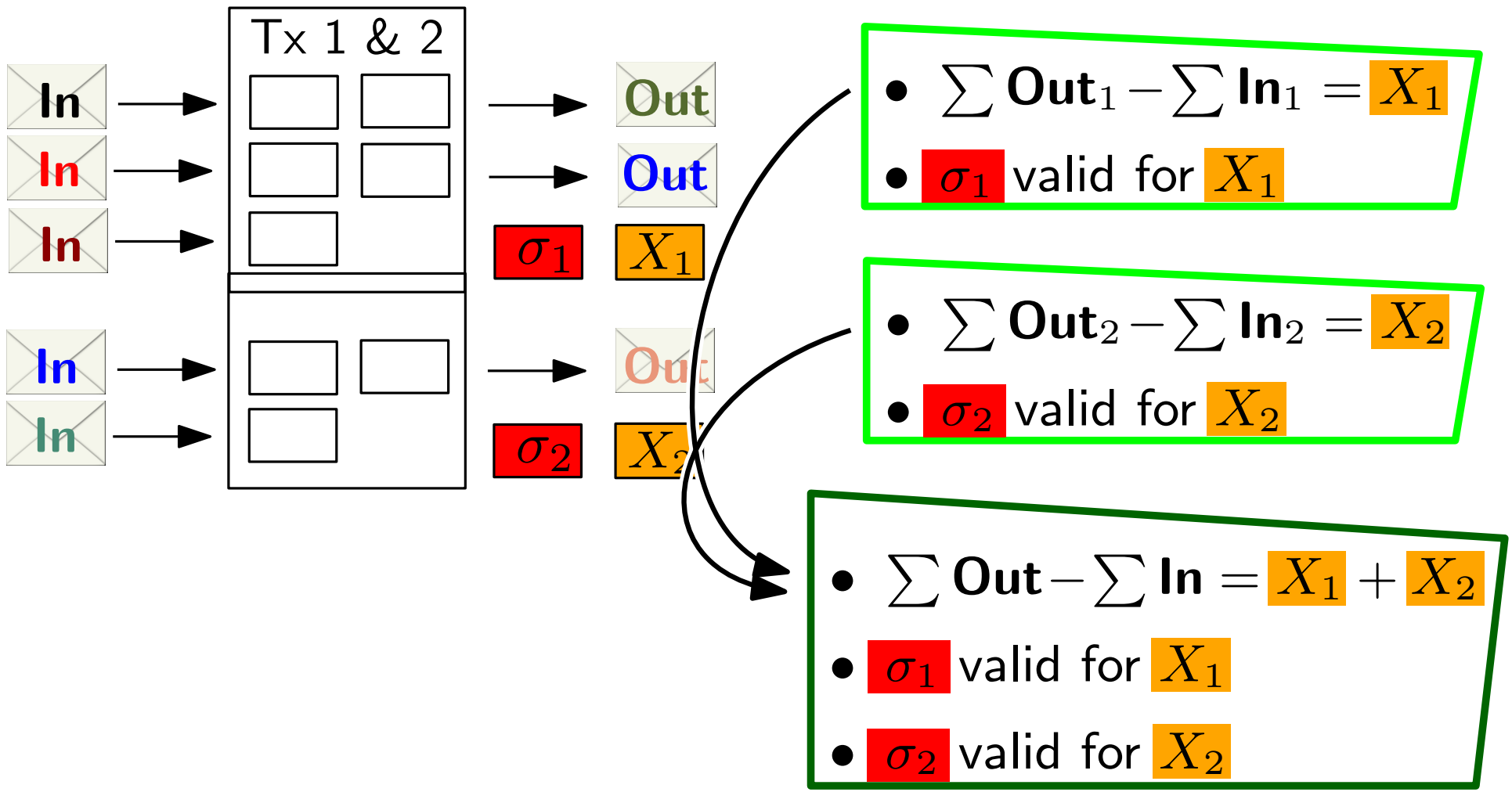
- $\sum \mathbf{Out}_1 - \sum \mathbf{In}_1 = X_1$
- σ_1 valid for X_1



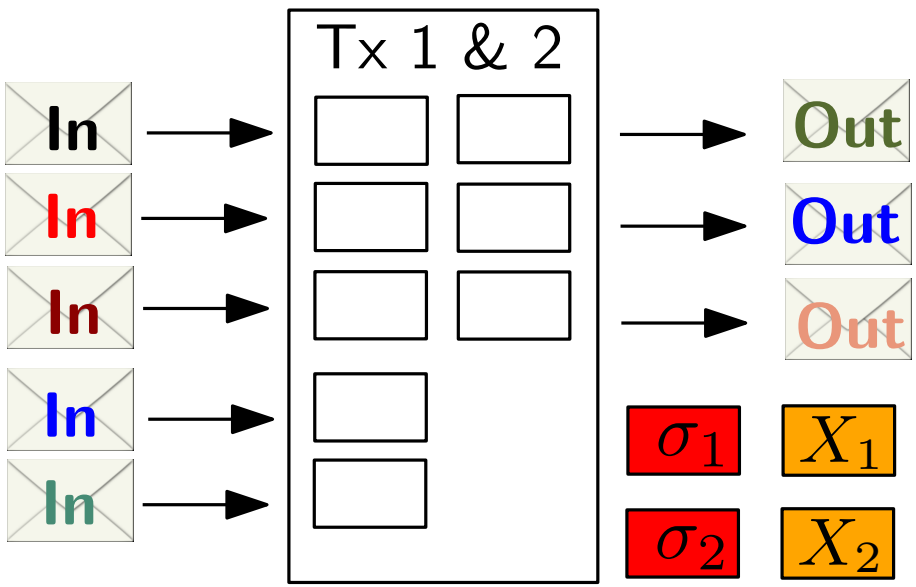
- $\sum \mathbf{Out}_2 - \sum \mathbf{In}_2 = X_2$
- σ_2 valid for X_2

Mimblewimble

Non-interactive CoinJoin

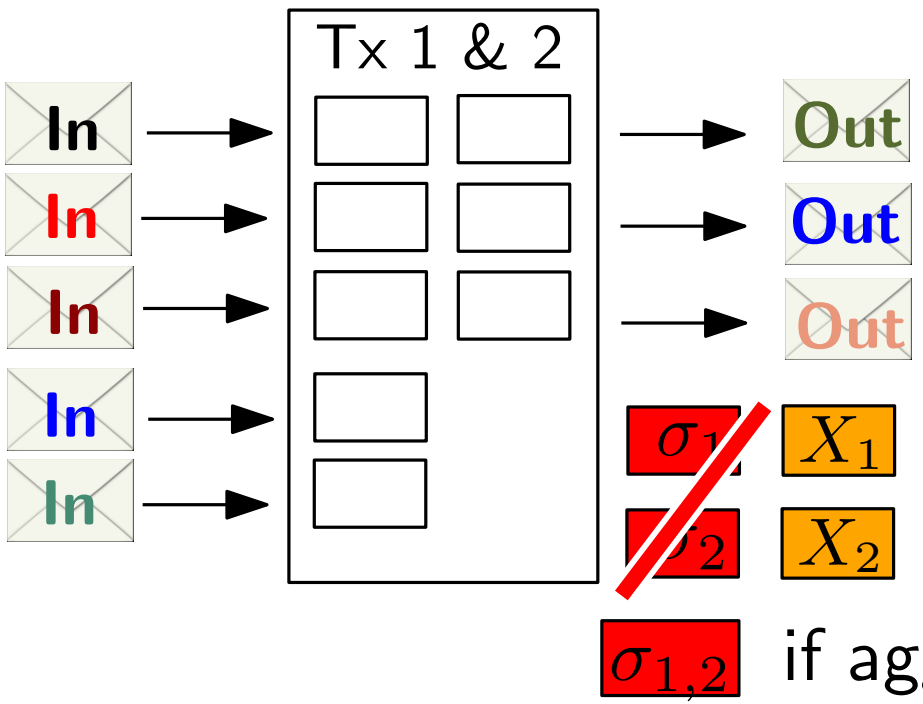


Mimblewimble



- $\sum$ **Out** $-\sum$ **In** $= X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

Mimblewimble

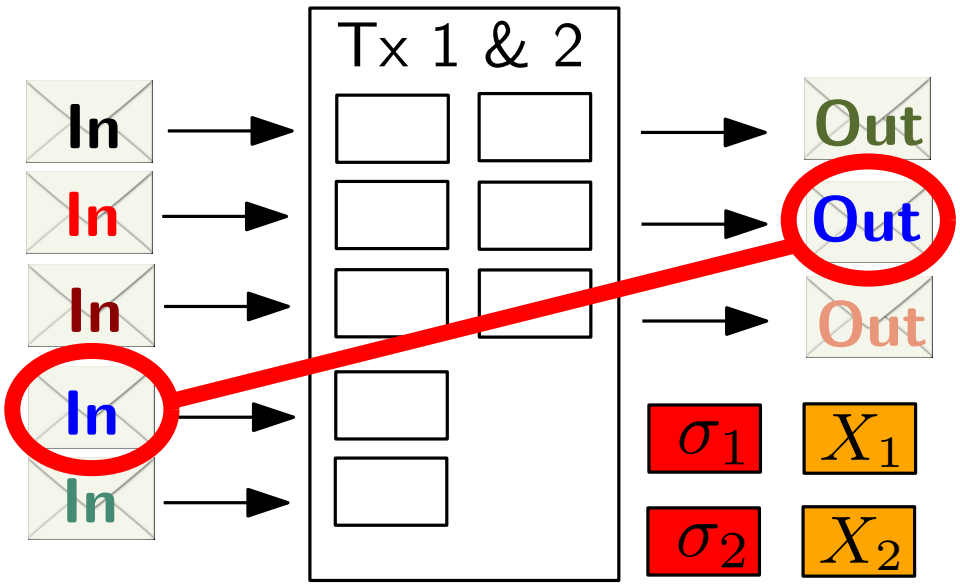


- $\sum \textbf{Out} - \sum \textbf{In} = X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

$\sigma_{1,2}$ if aggregate signature scheme (BLS)

Mimblewimble

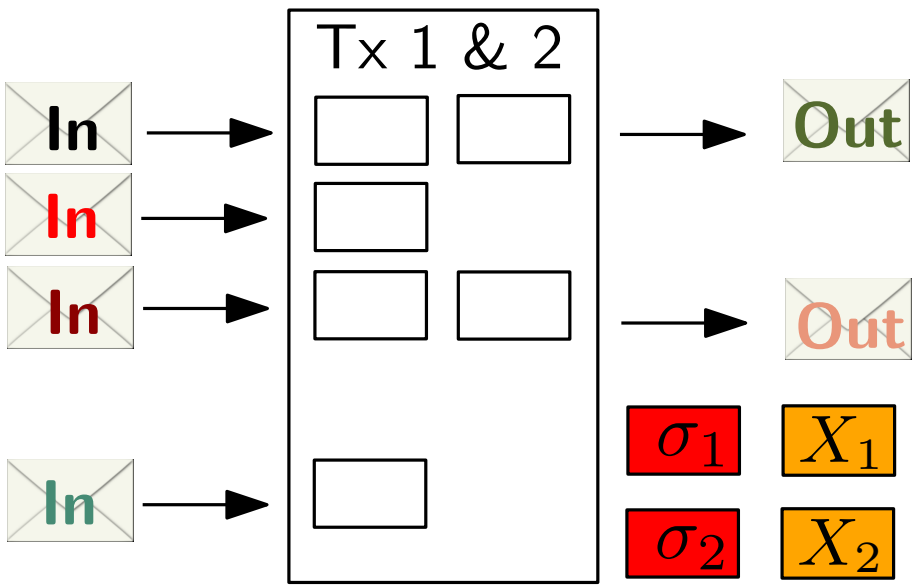
Post-confirmation Cut-Through



- $\sum$ **Out** $-\sum$ **In** $= X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

Mimblewimble

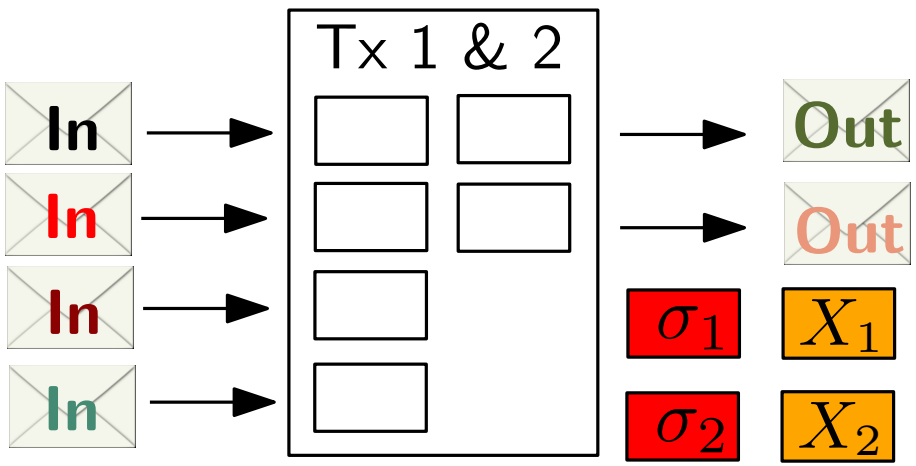
Post-confirmation Cut-Through



- $\sum$ **Out** $-\sum$ **In** $= X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

Mimblewimble

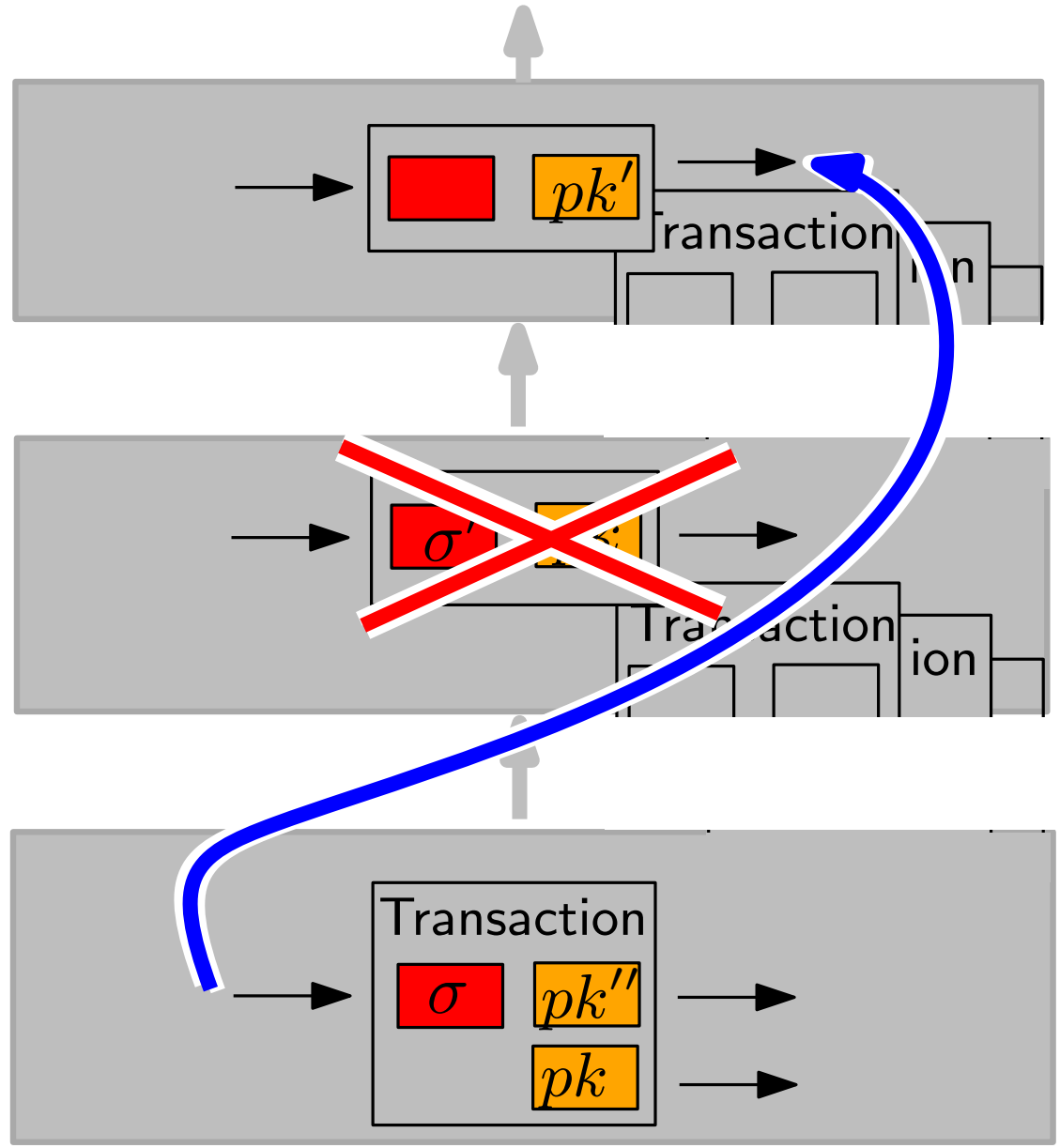
Post-confirmation Cut-Through



- $\sum$ **Out** $-\sum$ **In** $= X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

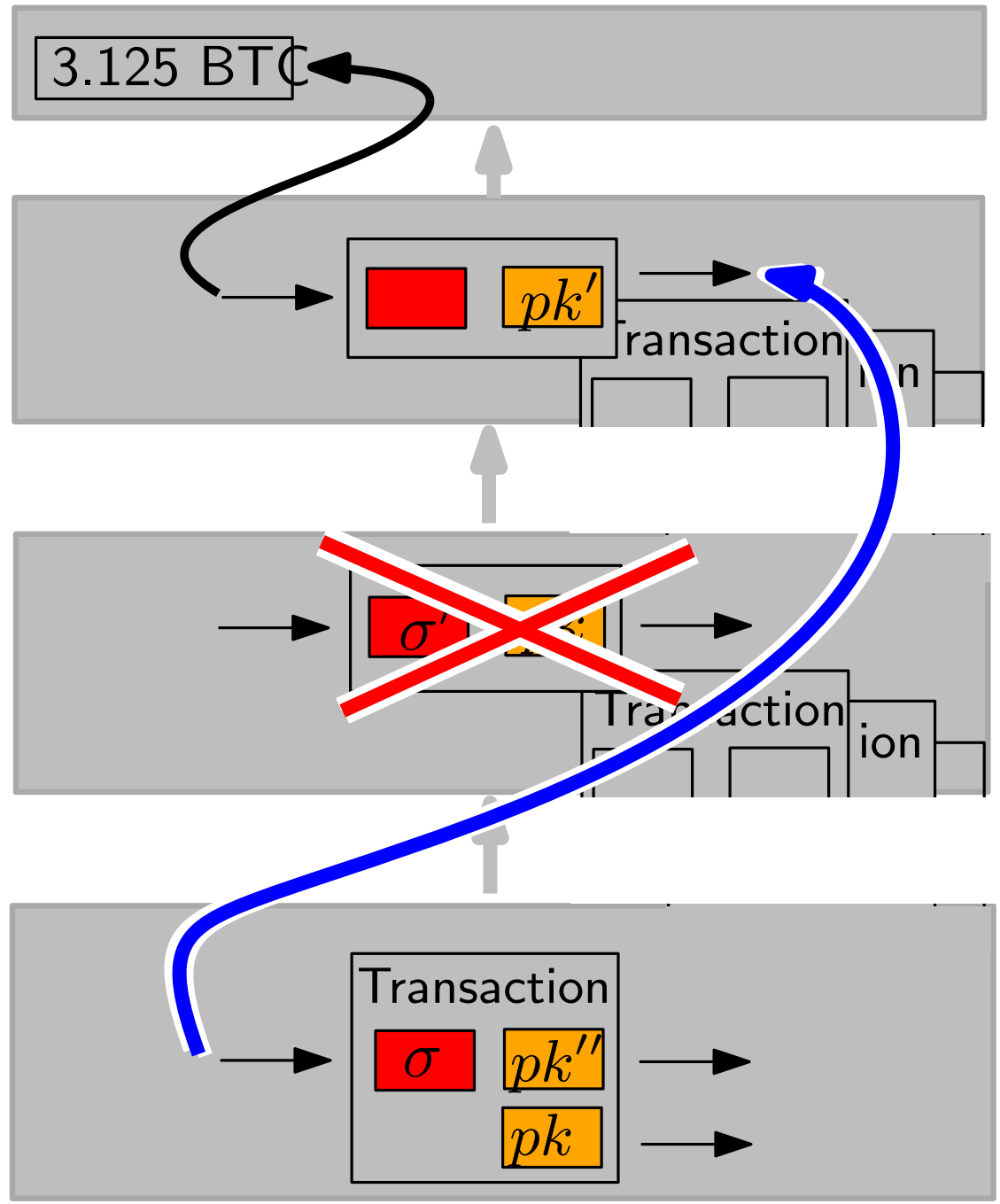
Scalability

“cut-through”



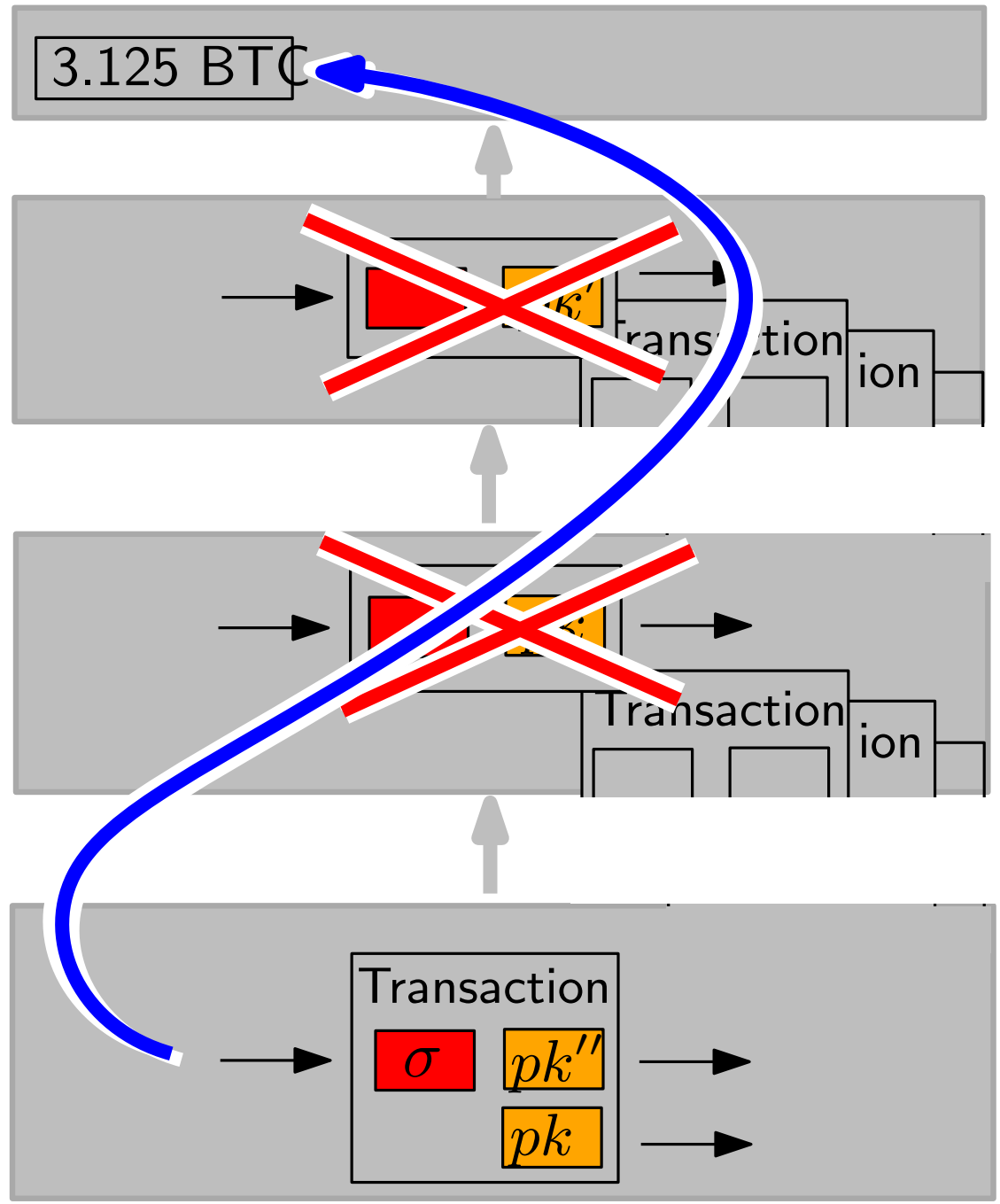
Scalability

“cut-through”



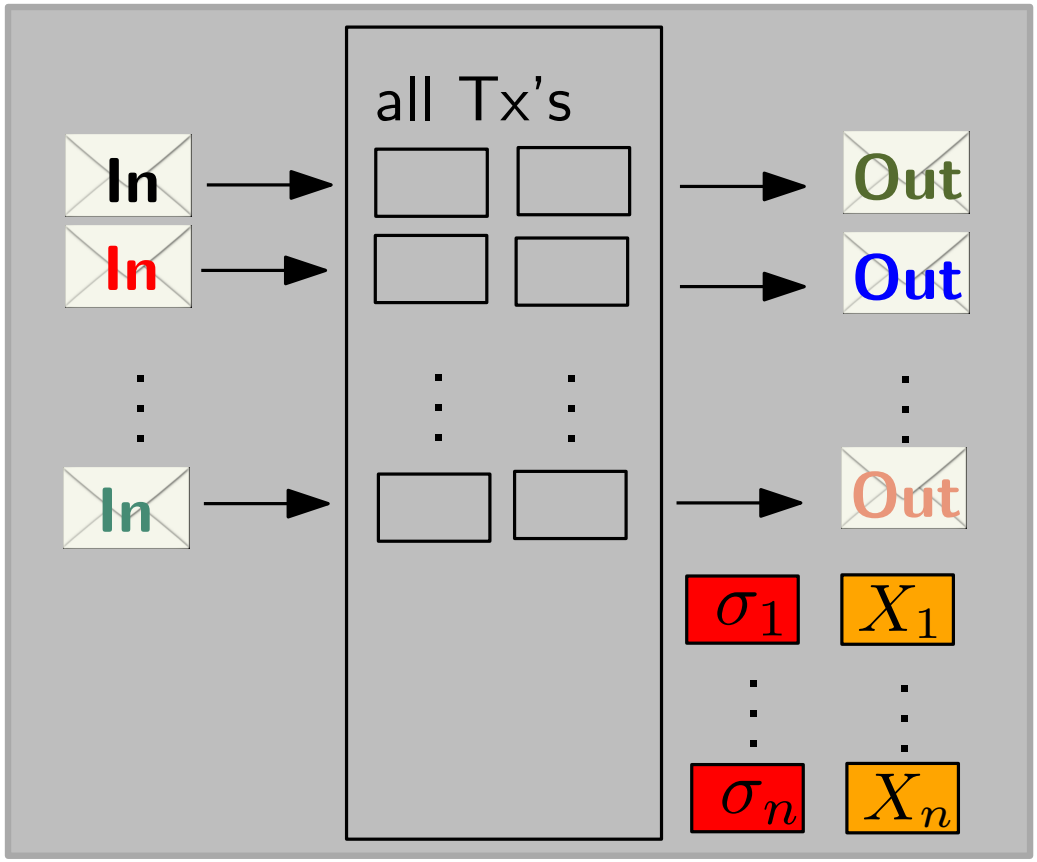
Scalability

“cut-through”



Mimblewimble

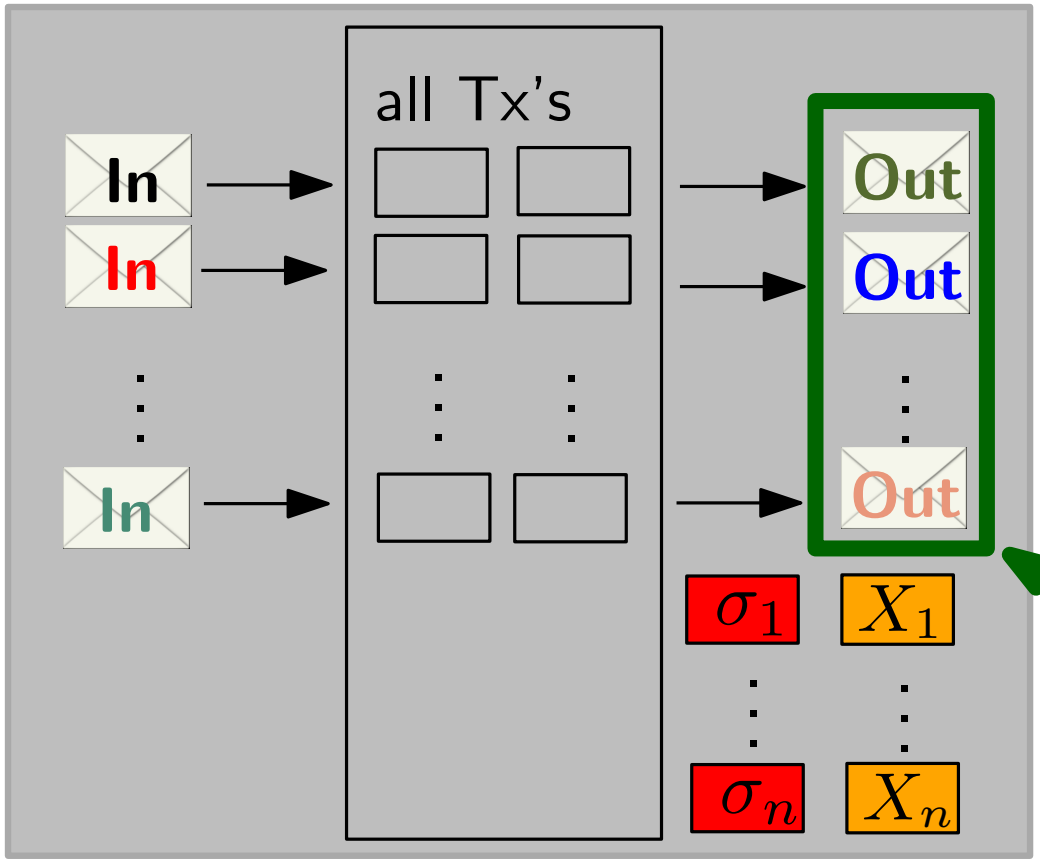
Cut through all transactions in blockchain



- $\sum$ **Out** $-\sum$ **In** $= \sum X_i$
- $\forall i$: σ_i valid for X_i

Mimblewimble

Cut through all transactions in blockchain

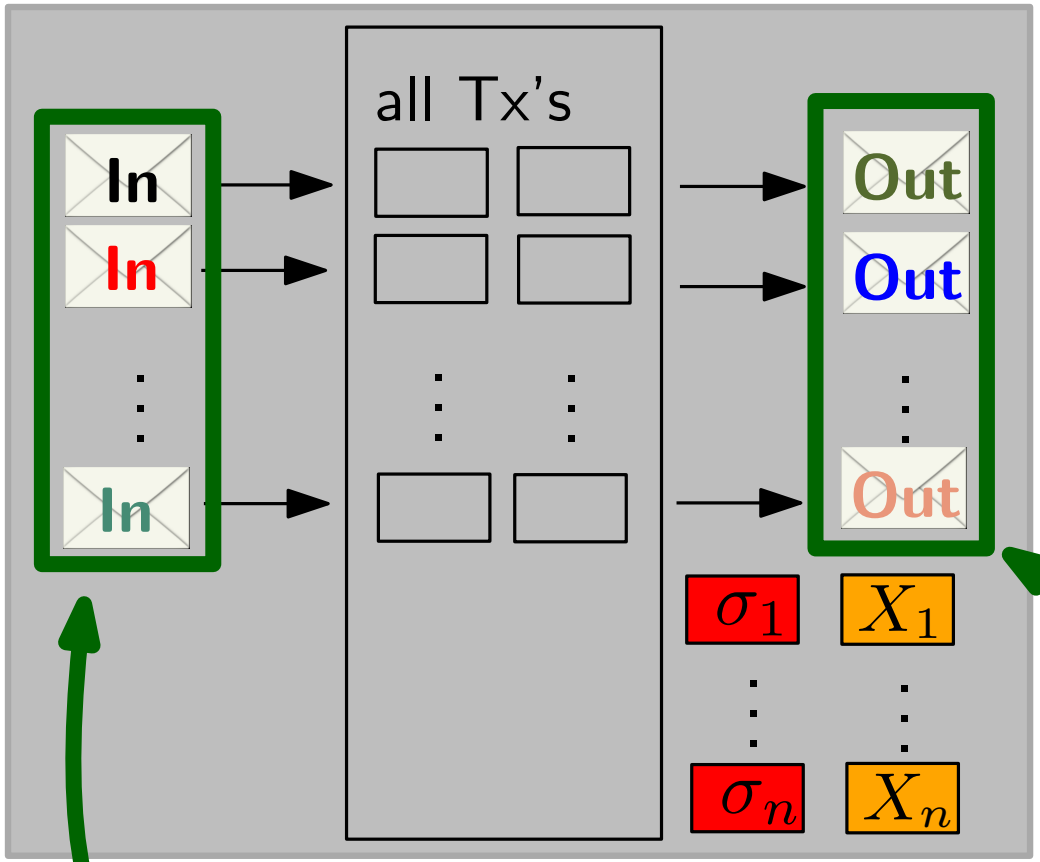


- $\sum$ **Out** $-\sum$ **In** $= \sum X_i$
- $\forall i$: σ_i valid for X_i

UTXO set

Mimblewimble

Cut through all transactions in blockchain



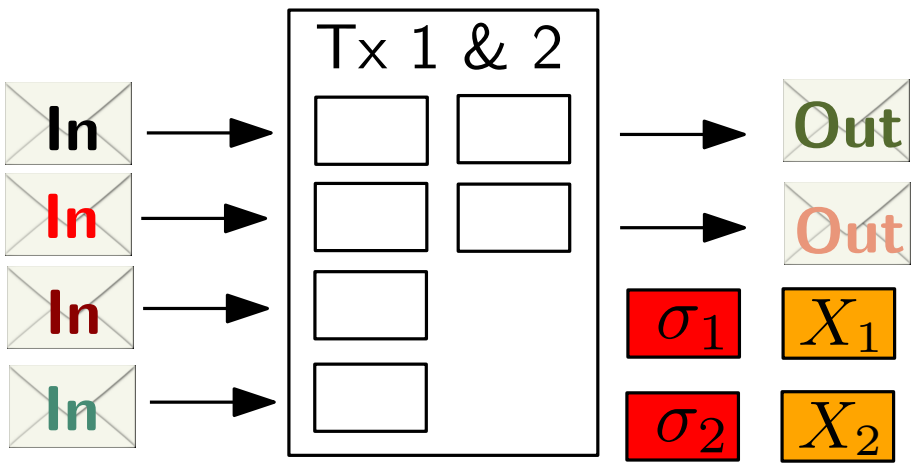
- $\sum$ **Out** $-\sum$ **In** $= \sum X_i$
- $\forall i$: σ_i valid for X_i

UTXO set

Only coinbase transactions

Mimblewimble

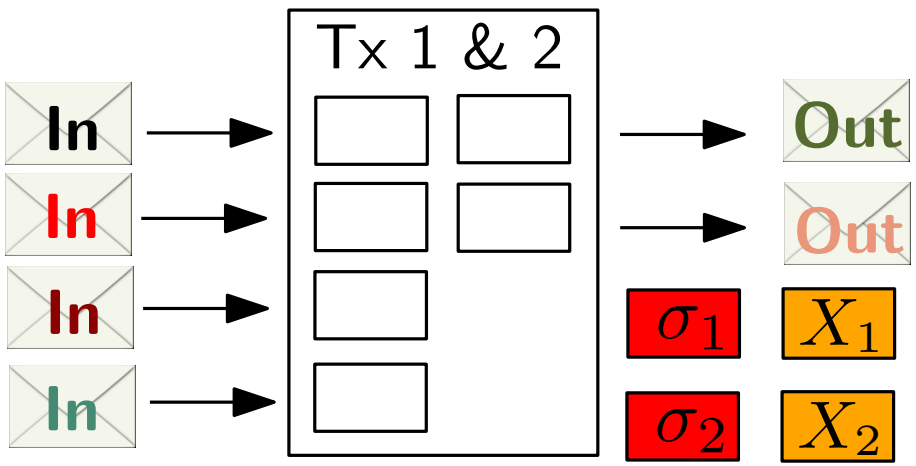
Privacy?



- $\sum$ **Out** $-\sum$ **In** $= X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

Mimblewimble

Privacy?

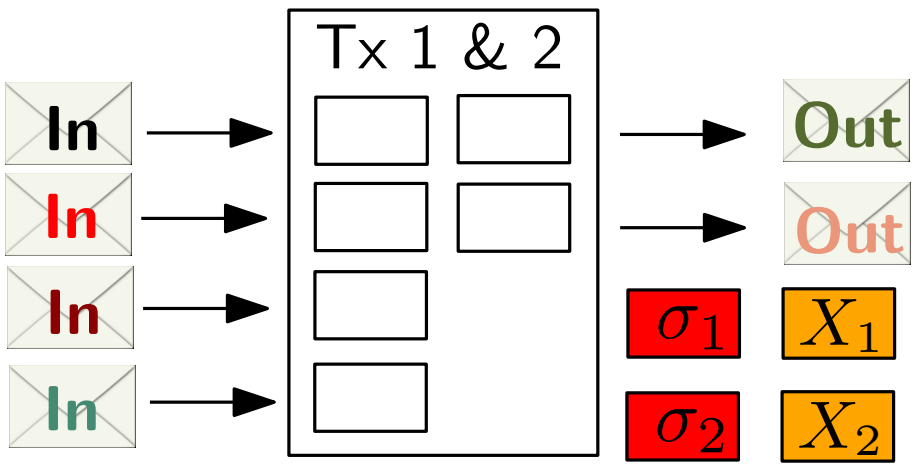


- $\sum \textbf{Out} - \sum \textbf{In} = X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

- Shuffle inputs and outputs

Mimblewimble

Privacy?

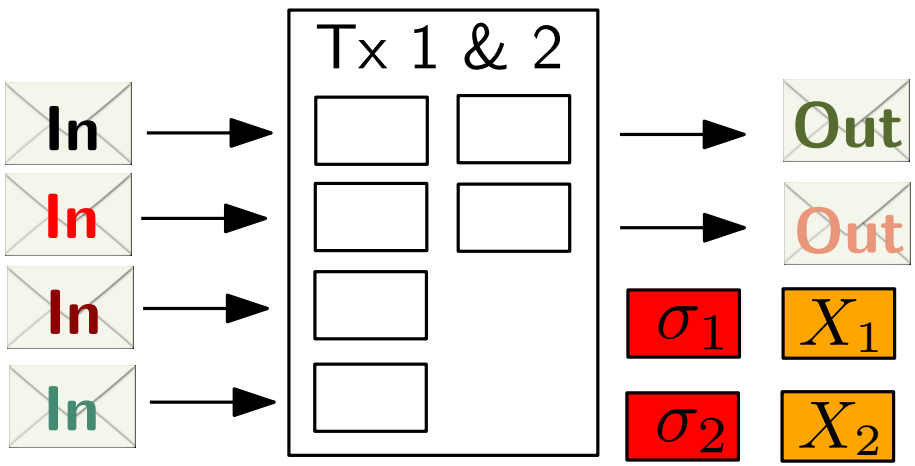


- $\sum$ **Out** $-\sum$ **In** $= X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

- Shuffle inputs and outputs
- Hides in/out relation?

Mimblewimble

Privacy?

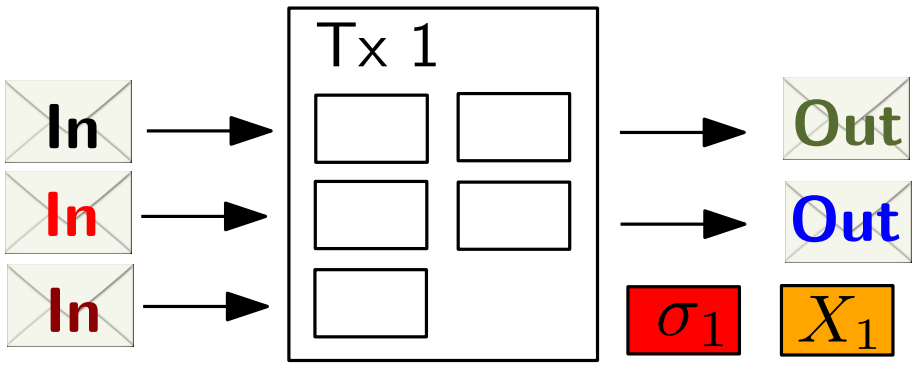


- $\sum \mathbf{Out} - \sum \mathbf{In} = X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

- Shuffle inputs and outputs
- Hides in/out relation?
- No! We have $\sum \mathbf{Out}_i - \sum \mathbf{In}_i = X_i \Rightarrow$ solve subset-sum

Mimblewimble

Privacy?



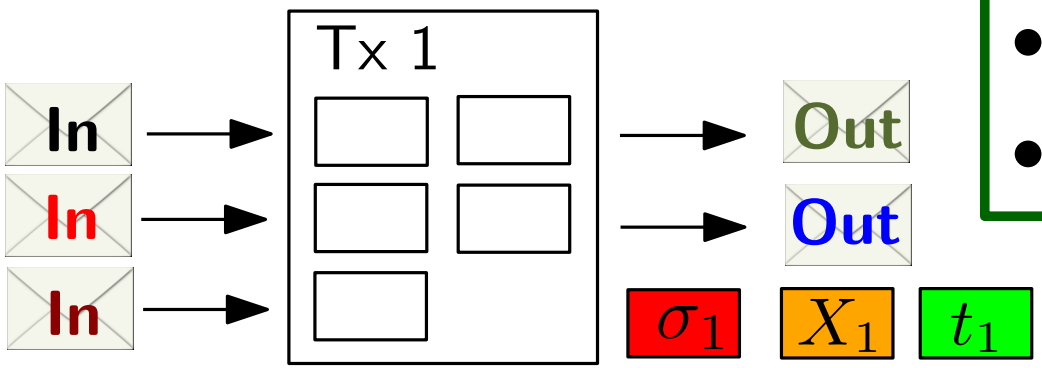
- $\sum \mathbf{Out}_1 - \sum \mathbf{In}_1 = X_1$
- σ_1 valid for X_1

Kernel offset:

- Choose random t_i, set $X_i := \sum \mathbf{Out}_i - \sum \mathbf{In}_i - t_i G$

Mimblewimble

Privacy?



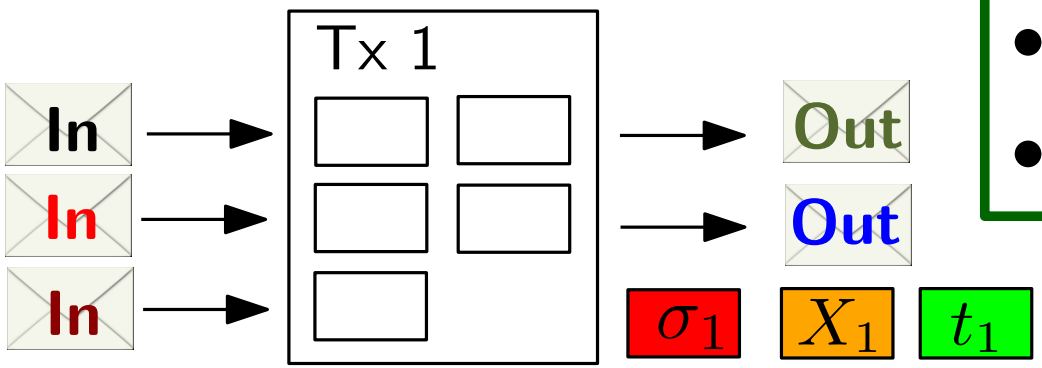
- $\sum \mathbf{Out}_1 - \sum \mathbf{In}_1 = X_1 + t_1 G$
- σ_1 valid for X_1

Kernel offset:

- Choose random t_i, set $X_i := \sum \mathbf{Out}_i - \sum \mathbf{In}_i - t_i G$

Mimblewimble

Privacy?



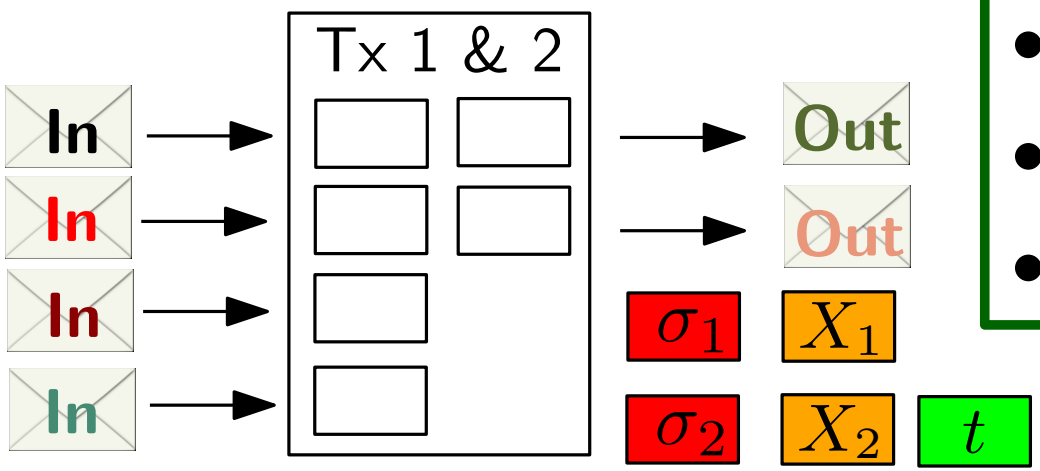
- $\sum \mathbf{Out}_1 - \sum \mathbf{In}_1 = X_1 + t_1 G$
- σ_1 valid for X_1

Kernel offset:

- Choose random t_i, set $X_i := \sum \mathbf{Out}_i - \sum \mathbf{In}_i - t_i G$
- When merging tx_1 and tx_2, set $t := t_1 + t_2$

Mimblewimble

Privacy?



- $\sum \mathbf{Out} - \sum \mathbf{In} = X_1 + X_2 + tG$
- σ_1 valid for X_1
- σ_2 valid for X_2

Kernel offset:

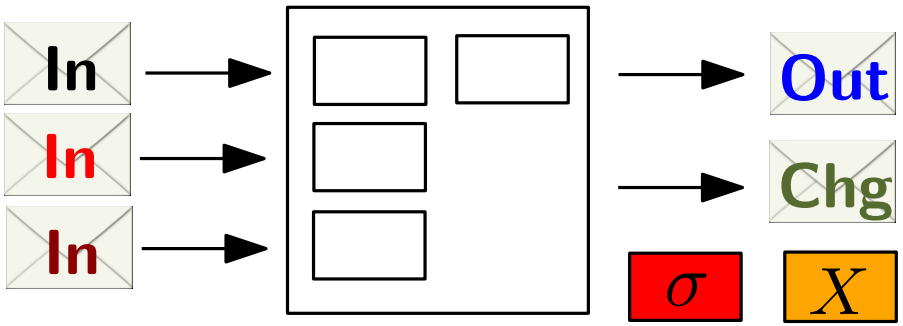
- For tx_i, choose random t_i, set $X_i := \sum \mathbf{Out}_i - \sum \mathbf{In}_i - t_i G$
- When merging tx_1 and tx_2, set $t := t_1 + t_2$

Mimblewimble

How are transactions actually created?

Mimblewimble

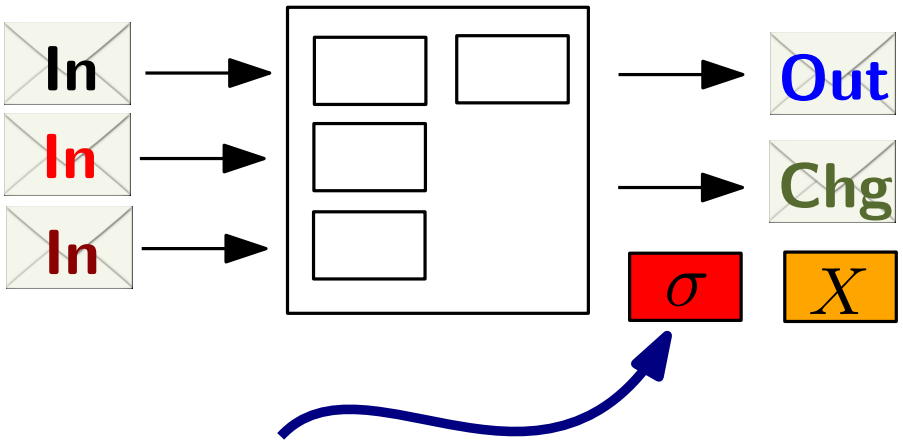
How are transactions actually created?



- $\sum \textbf{Out} - \sum \textbf{In} = X$
- σ valid for X

Mimblewimble

How are transactions actually created?

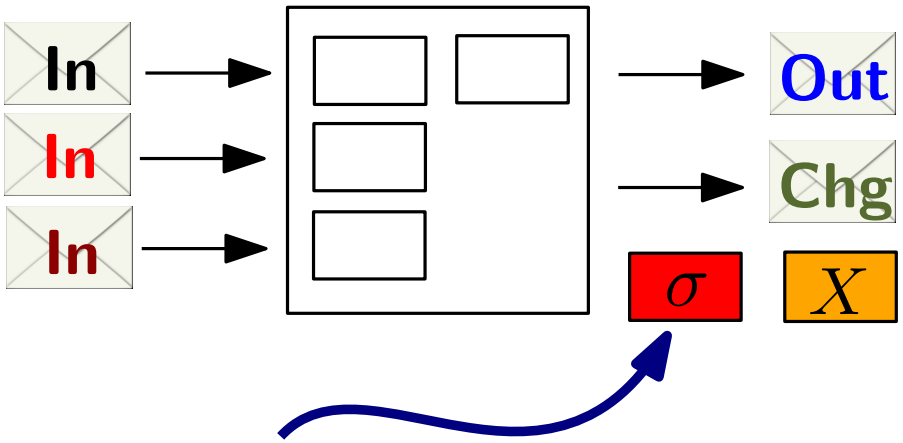


- $\sum \textbf{Out} - \sum \textbf{In} = X$
- σ valid for X

signature under key $r_{\text{Out}} + r_{\text{Chg}} - \sum r_{In}$

Mimblewimble

How are transactions actually created?



- $\sum$ **Out** $-\sum$ **In** $= X$
- σ valid for X

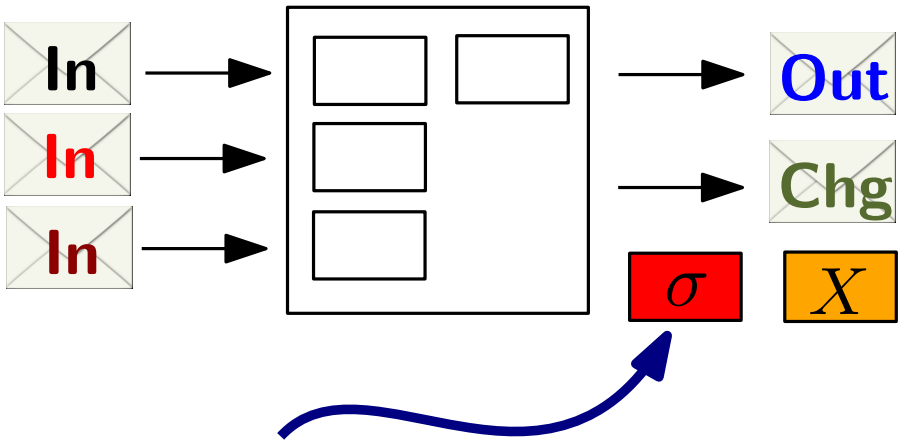
signature under key $r_{\text{Out}} + r_{\text{Chg}} - \sum r_{In}$

known by sender

known by receiver

Mimblewimble

How are transactions actually created?



- $\sum$ **Out** $-\sum$ **In** $= X$
- σ valid for X

signature under key $r_{\text{Out}} + r_{\text{Chg}} - \sum r_{In}$

known by sender

known by receiver

Threshold-signing for key $r_{\text{Out}}G + (r_{\text{Chg}} - \sum r_{In})G$

Mimblewimble

[FOS19]

- **Formal security models:**
  - inflation-resistance
  - coin-theft-resistance
  - privacy

Mimblewimble

[FOS19]

- **Formal security models:**
  - inflation-resistance
  - coin-theft-resistance
  - privacy
- **Abstraction of Mimblewimble** from:
  - homomorphic commitments
  - compatible signatures
  - simulation-extractable NIZK range proofs

Mimblewimble

[FOS19]

- **Formal security models:**
  - inflation-resistance
  - coin-theft-resistance
  - privacy
- **Abstraction of Mimblewimble** from:
  - homomorphic commitments
  - compatible signatures

  ] . . . satisfying joint security
  - simulation-extractable NIZK range proofs

Mimblewimble

[FOS19]

- **Formal security models:**
  - inflation-resistance
  - coin-theft-resistance
  - privacy
- **Abstraction of Mimblewimble** from:
  - homomorphic commitments
  - compatible signatures

  ] . . . satisfying joint security
  - simulation-extractable NIZK range proofs
- **Proof** that abstraction satisfies model

Mimblewimble

[FOS19]

- **Formal security models:**
  - inflation-resistance
  - coin-theft-resistance
  - privacy
- **Abstraction of Mimblewimble** from:
  - homomorphic commitments
  - compatible signatures

  ] ...satisfying joint security
  - simulation-extractable NIZK range proofs
- **Proof** that abstraction satisfies model
- **Instantiations:** proof that
  - Pedersen + Schnorr
  - Pedersen + (aggregate) BLS

  ] ...satisfy joint security