Space-efficient blockchains

Georg Fuchsbauer

Sustainability in Computer Science, TUW, 1 Dec '25

SPyCoDe FWF
Der Wissenschaftsfonds.
WWTF

Bitcoin

What

- digital currency
  (most successful ever)
- decentralized
  (no bank)
- hard-coded inflation
- pseudonymous

How

- maintains public history
  of all transactions
- guaranteed consistency
- distributed consensus

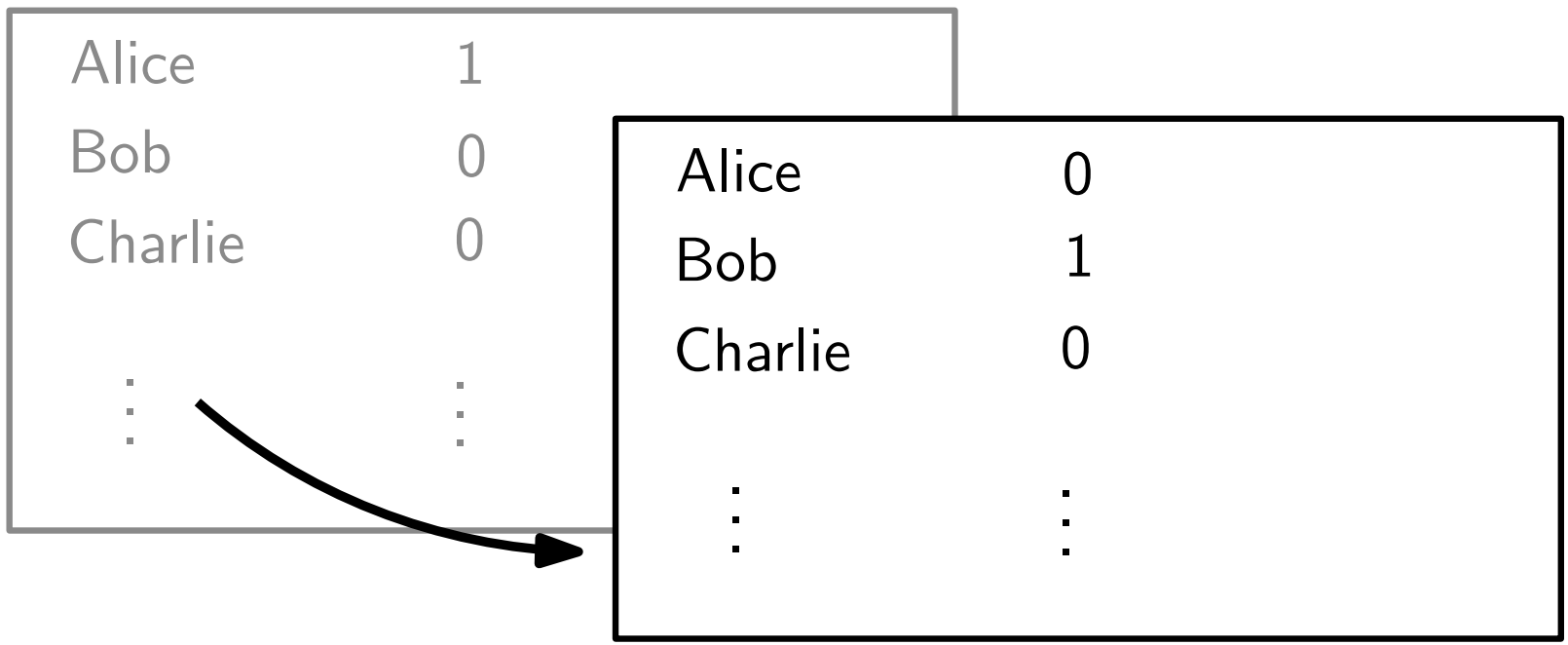
Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending

Ledger

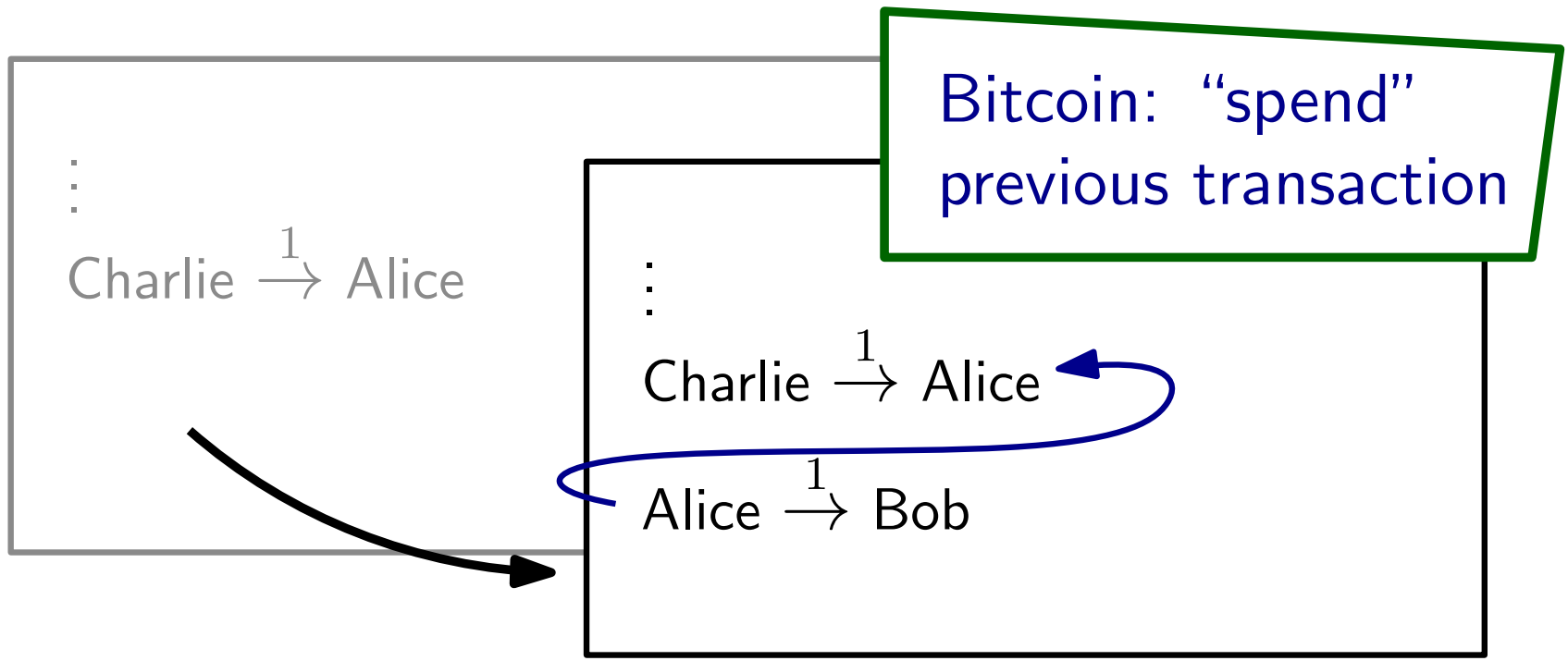
Public ledger (maintained by authority)



Alice: transfer 1 → Bob

Ledger

Public ledger (records all transactions)



Bitcoin:

– no notion of account

– only transactions

how to identify?

Transactions

- Identify a coin  with signature public key pk 
-  owns $\boxed{pk_A}$ i.e. it's in the ledger
-  creates $\boxed{pk_B}$
-  **signs** $\boxed{pk_A \rightarrow pk_B}$ and adds to ledger

transaction

SPyCoDe FWF Der Wissenschaftsfonds. WWTF TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Transactions

-  **signs** $pk_A \rightarrow pk_B$ 

  signs $pk_A \rightarrow pk_C$ 

Ledger only accepts if:

- exists transaction $* \rightarrow pk_A$!
- no transaction ~~$pk_A \rightarrow *$~~

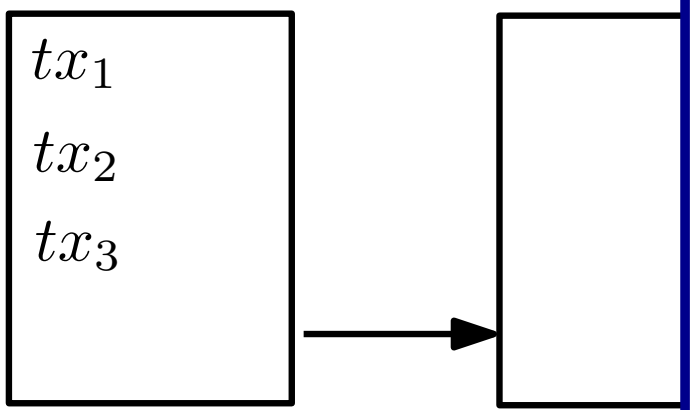
SPyCoDe FWF Der Wissenschaftsfonds. WWTF TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Decentralization

But: how do we **eliminate authority** that

- checks validity of tx's
- publishes new tx's in ledger?

The Blockchain



- Mining: *pay* maintainers ...
- Consensus ...

$\Rightarrow$ Krzysztof Pietrzak: *Sustainable Blockchains* (3 Nov 2025)

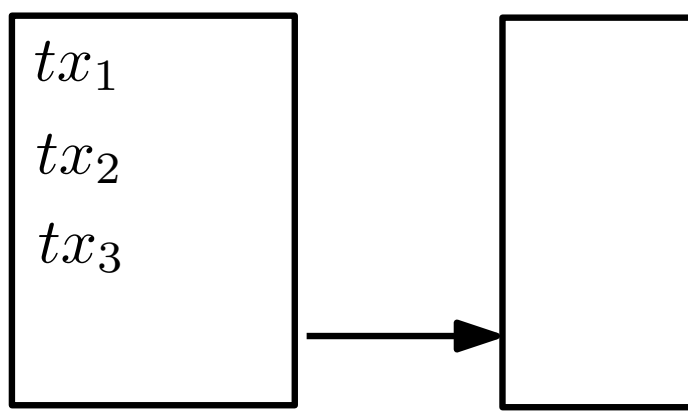
SPyCoDe FWF Der Wissenschaftsfonds. WWTF TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

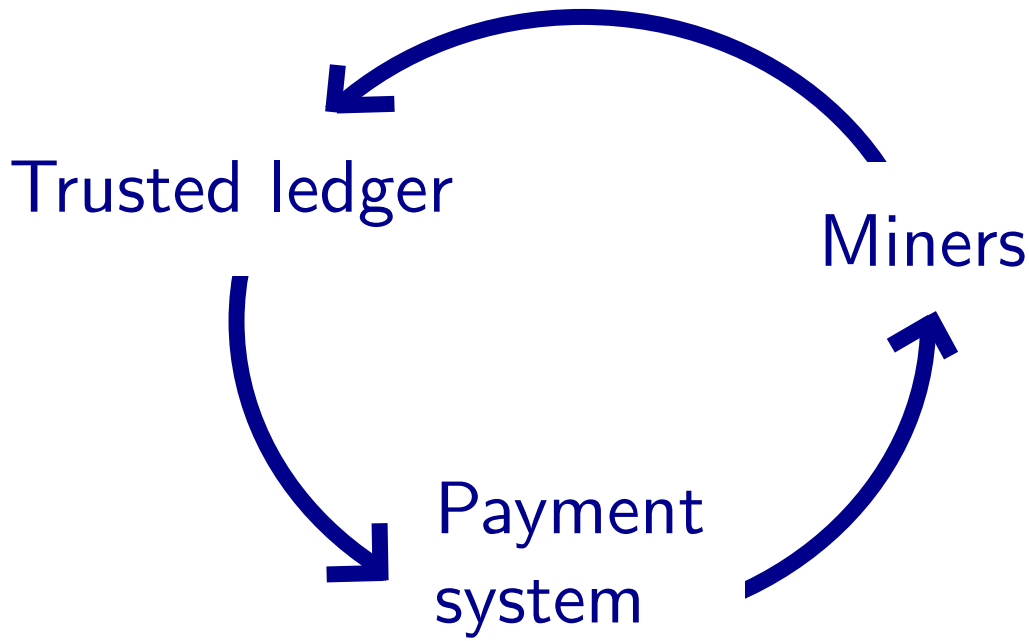
Decentralization

But: how do we **eliminate authority** that

- checks validity of tx's
- publishes new tx's in ledger?

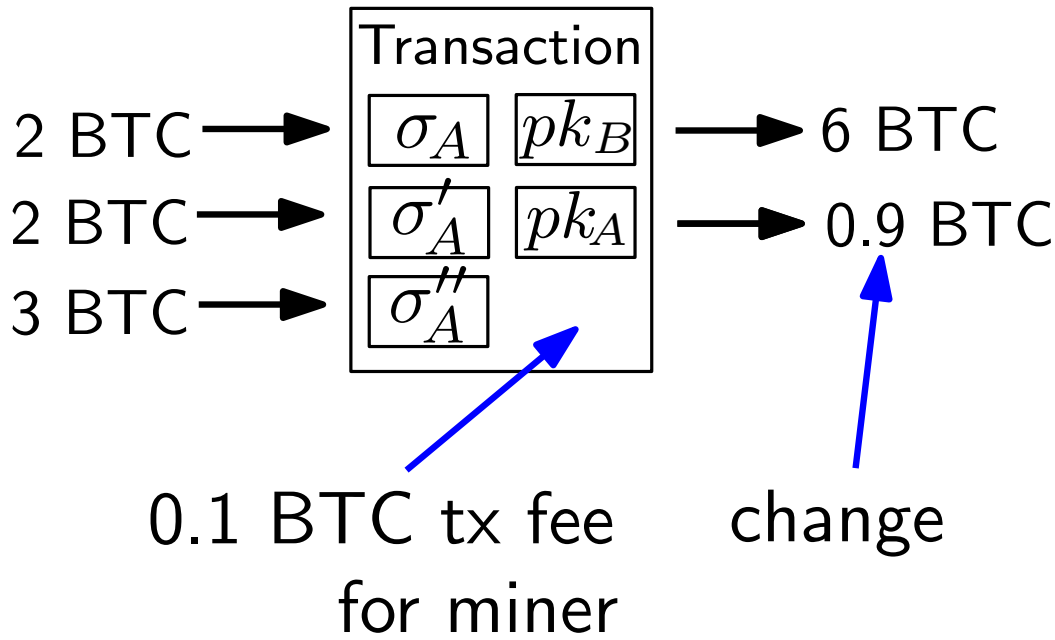
The Blockchain



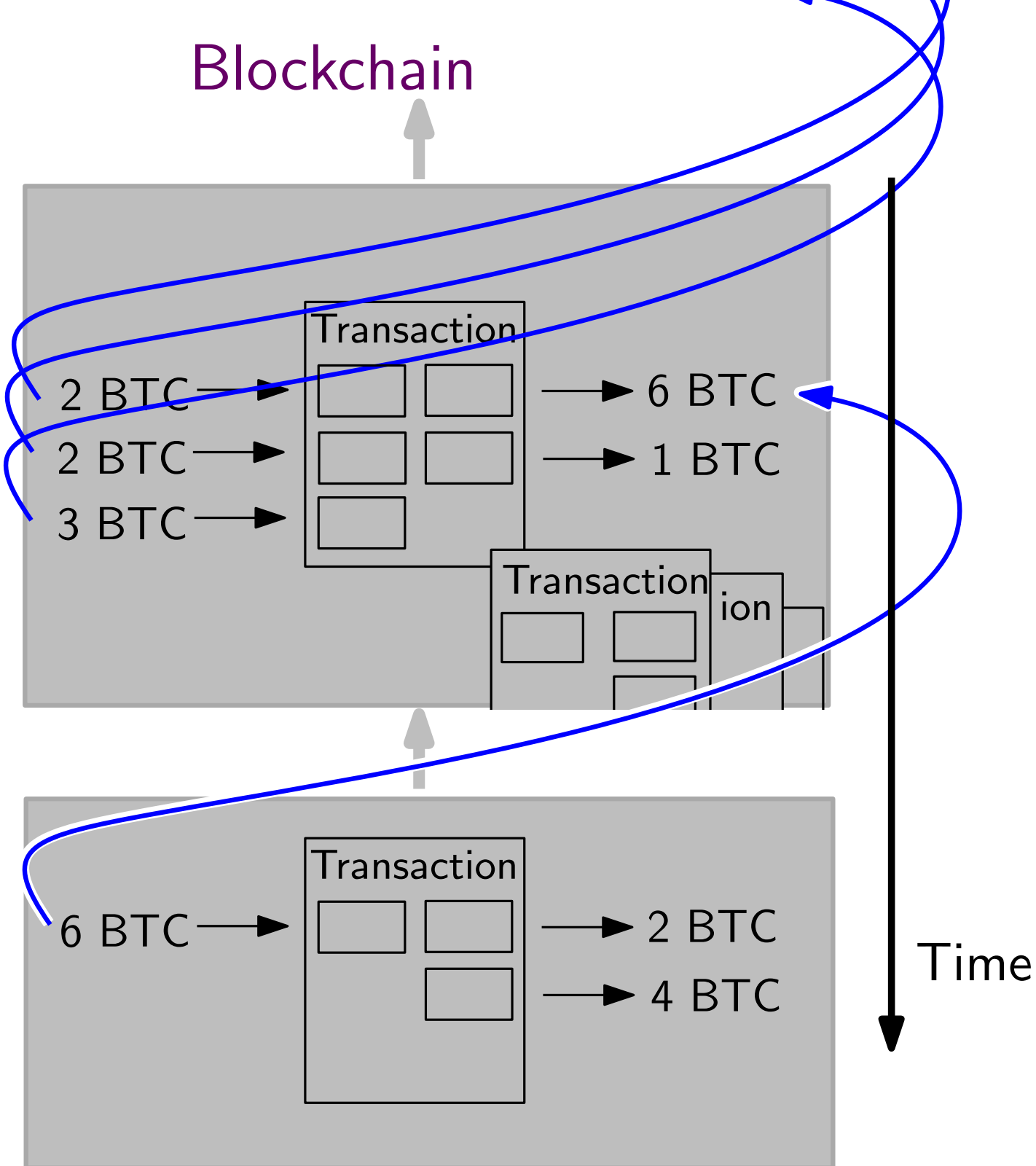


Transaction details

- **Transactions**

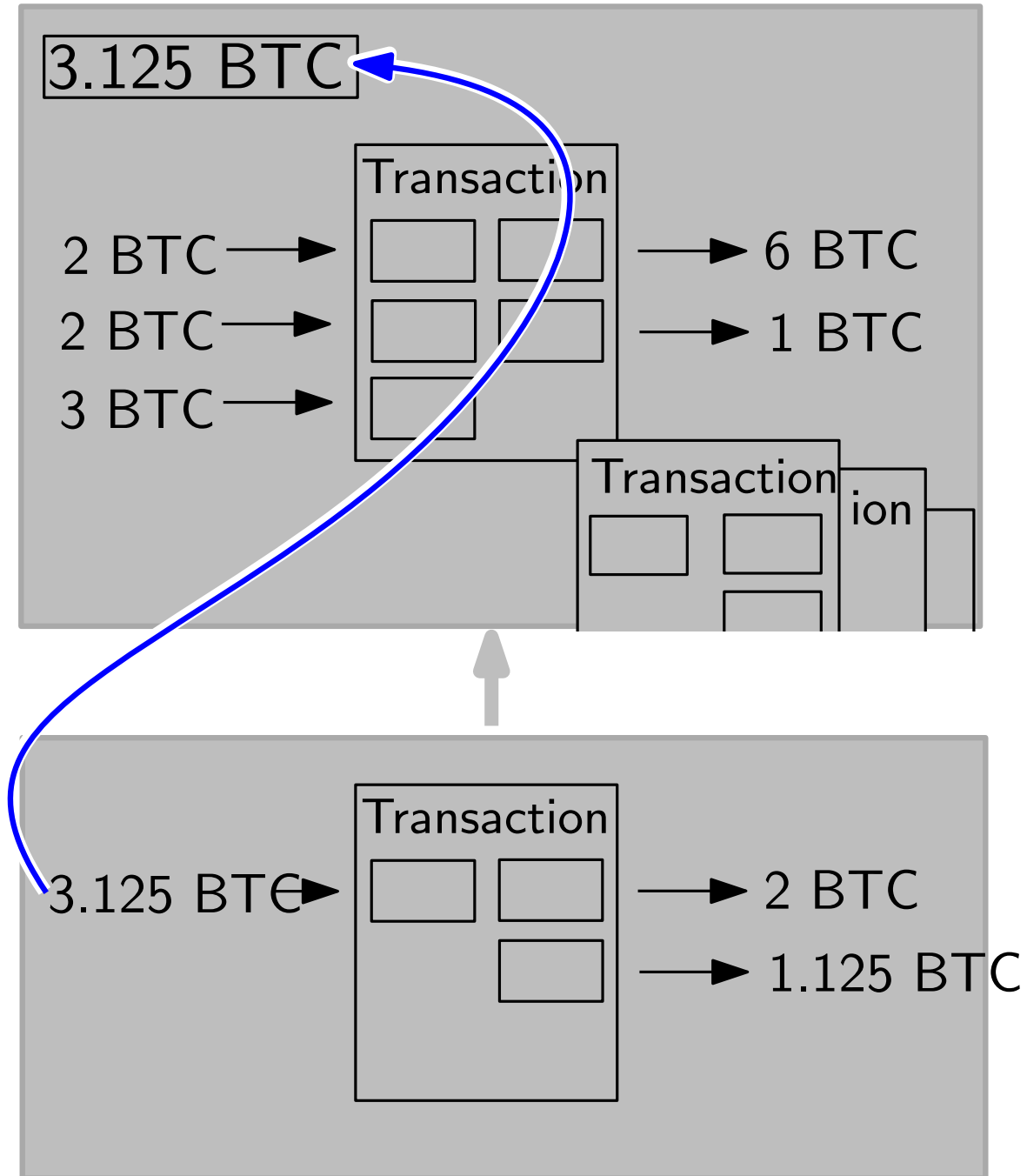


- **Block**
- Reference to previous output
- **Blockchain**

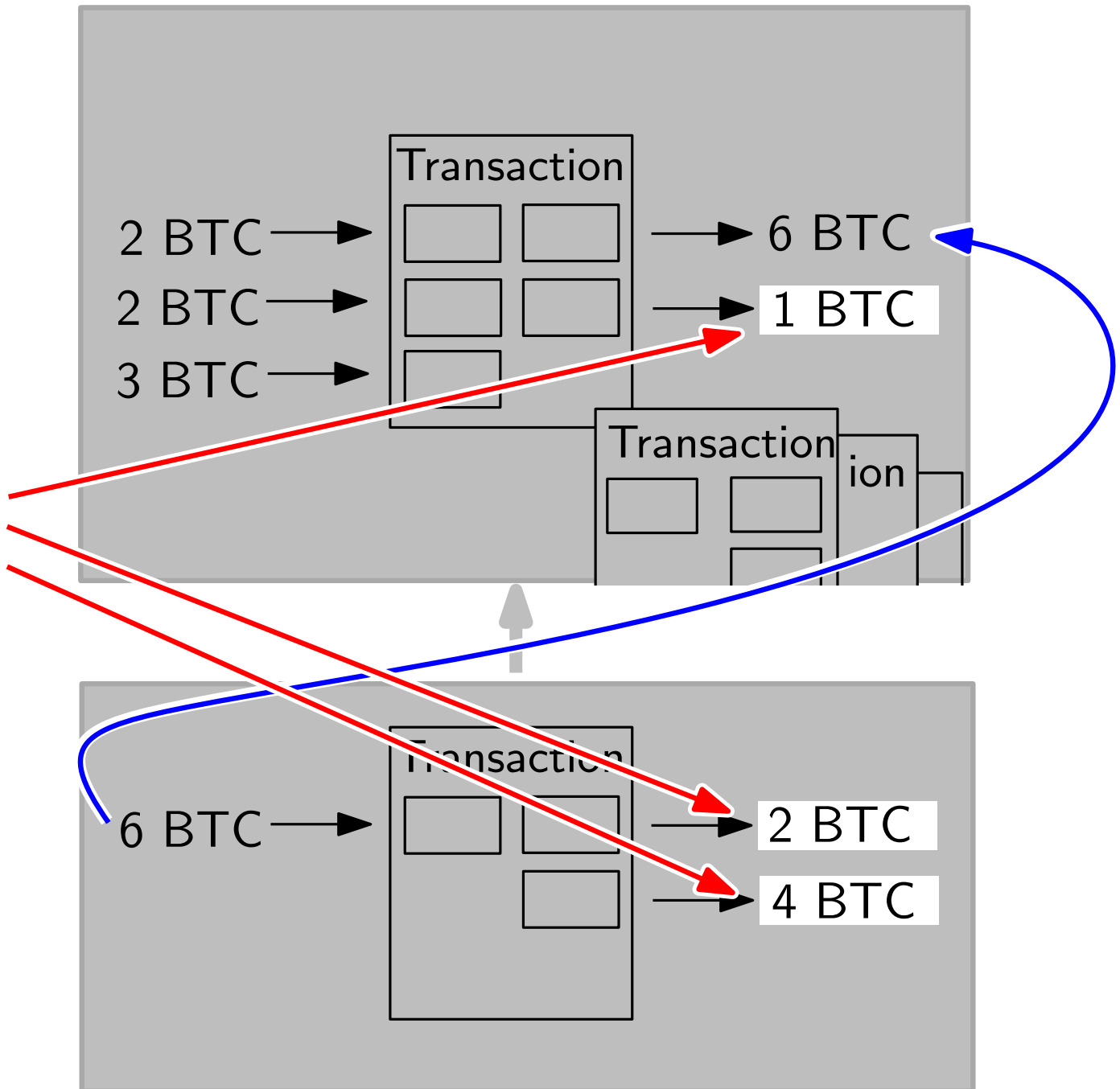


Blockchain

- **Coinbase transaction**



Blockchain

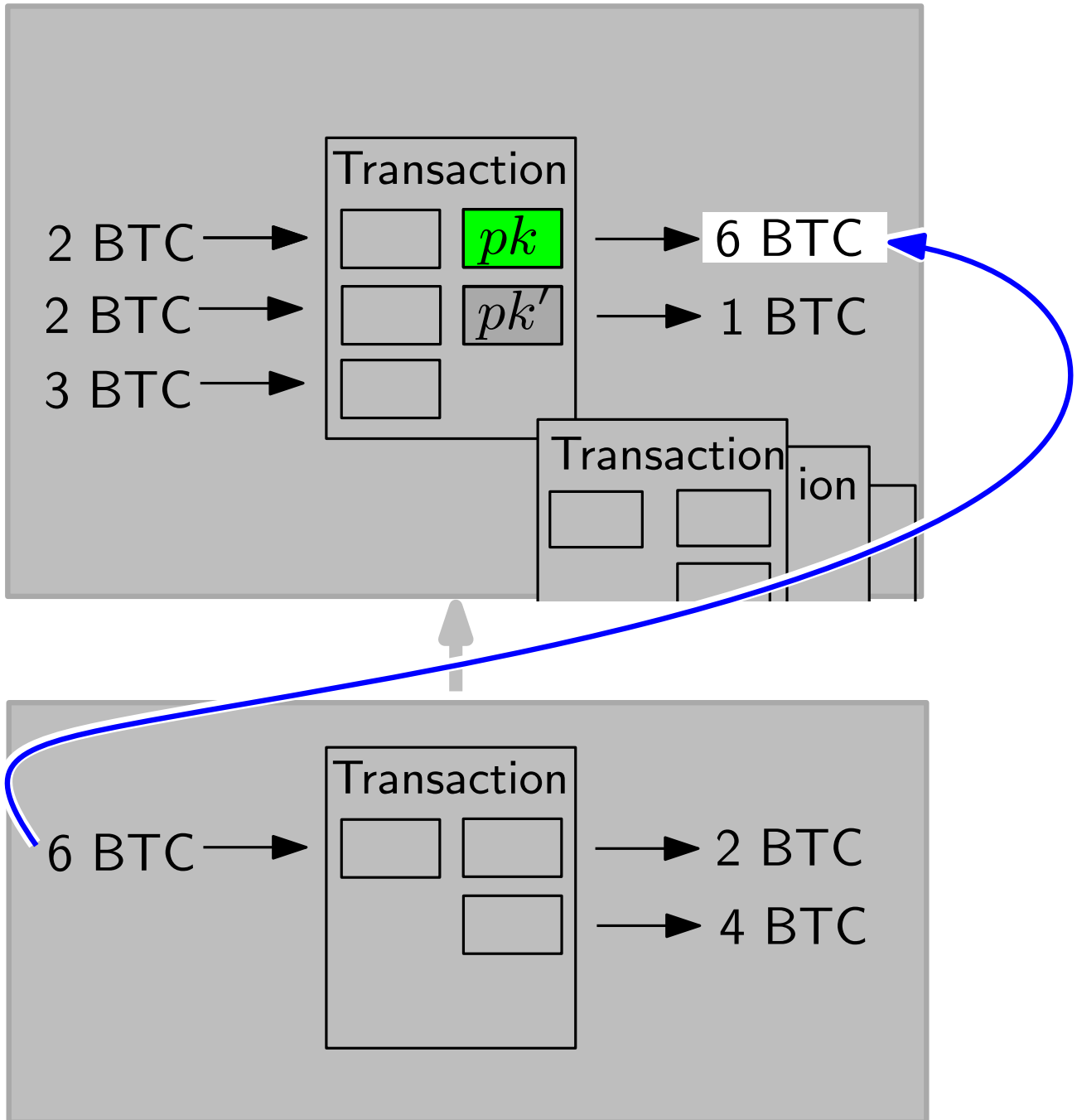


Unspent transaction outputs (UTXO's)

= existing money in system

Bitcoin

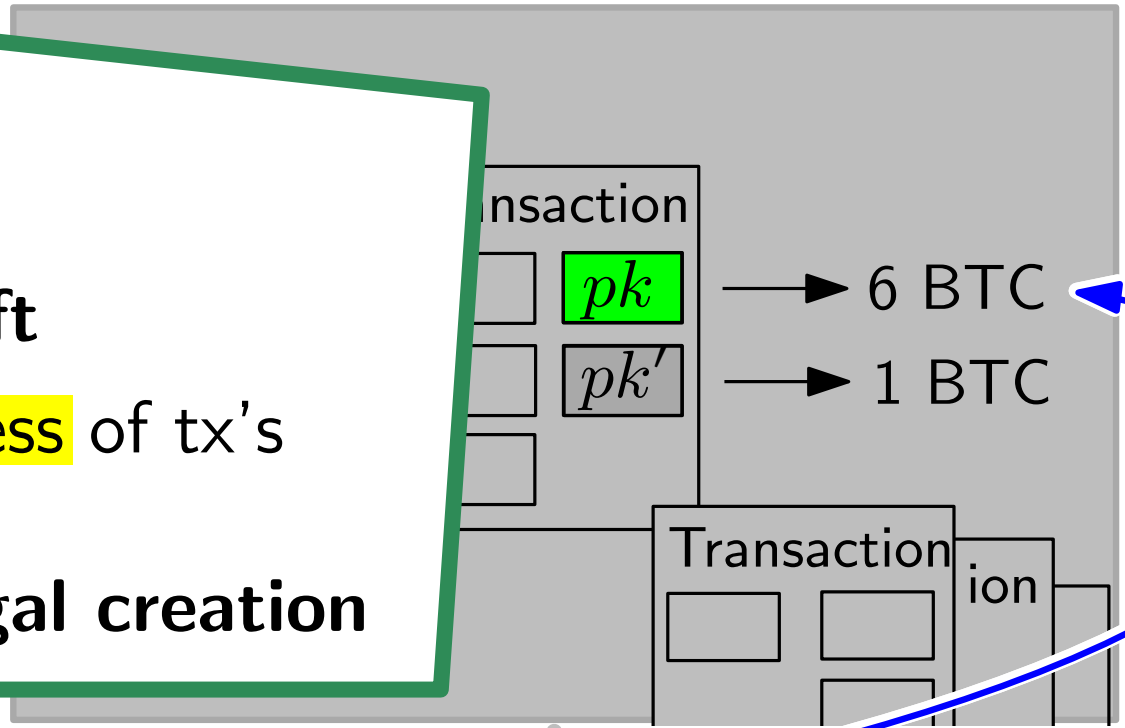
- **Owning**
  an output



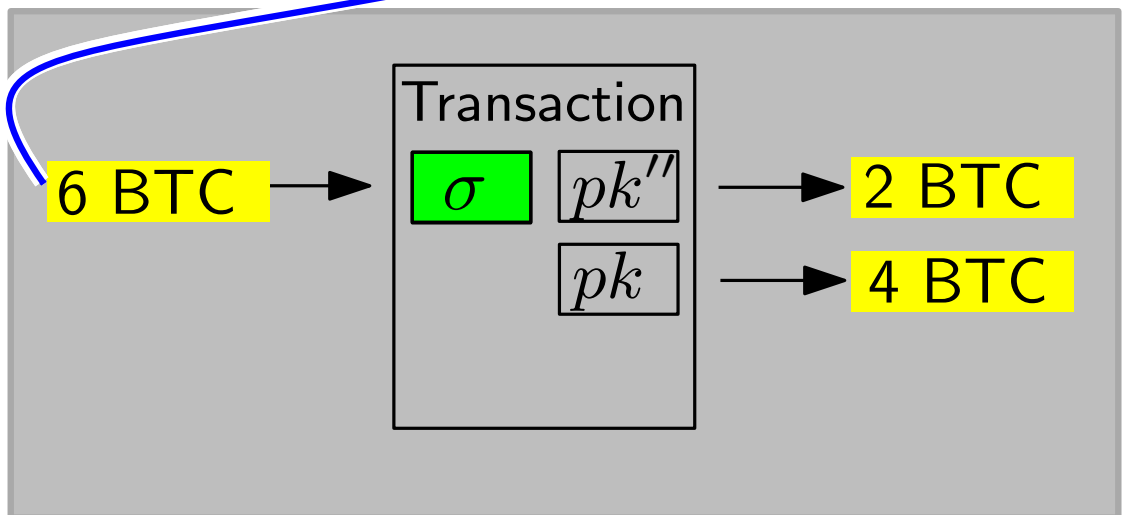
Bitcoin

Security

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



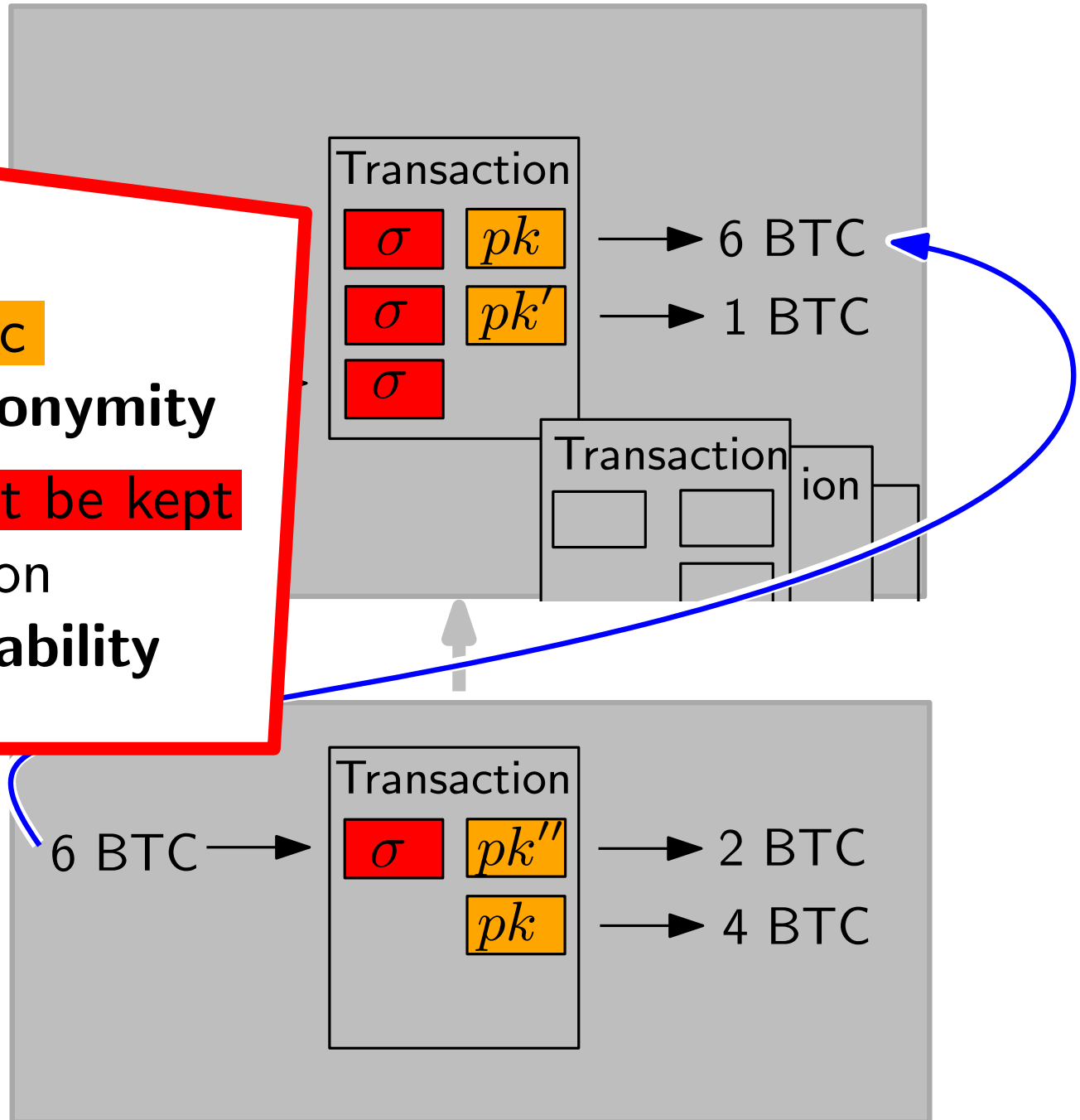
σ is **signature** under pk on tx



Bitcoin

Drawbacks

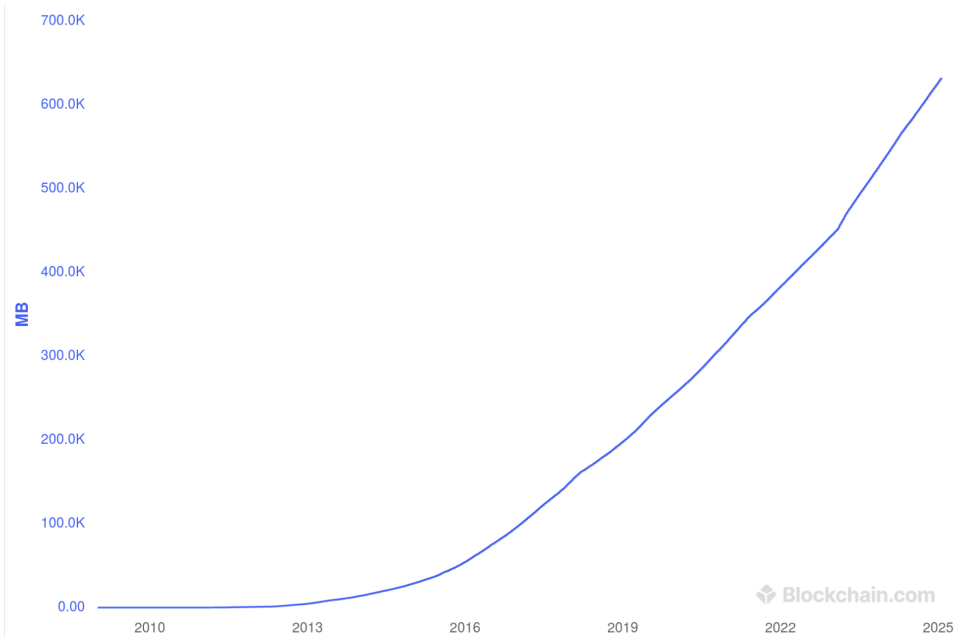
- all tx's public
  $\Rightarrow$ **weak anonymity**
- all data must be kept for verification
  $\Rightarrow$ **bad scalability**



Scalability

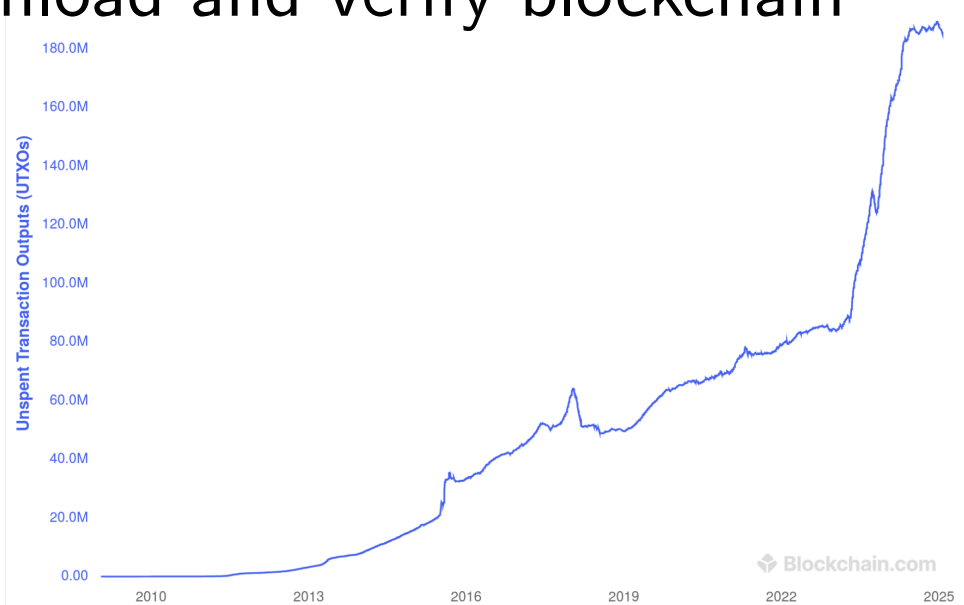
Blockchain size:
> 600 GB

ethereum:
> 1.4 TB



When starting full node: download and verify blockchain

Size of UTXO set:
11 GB

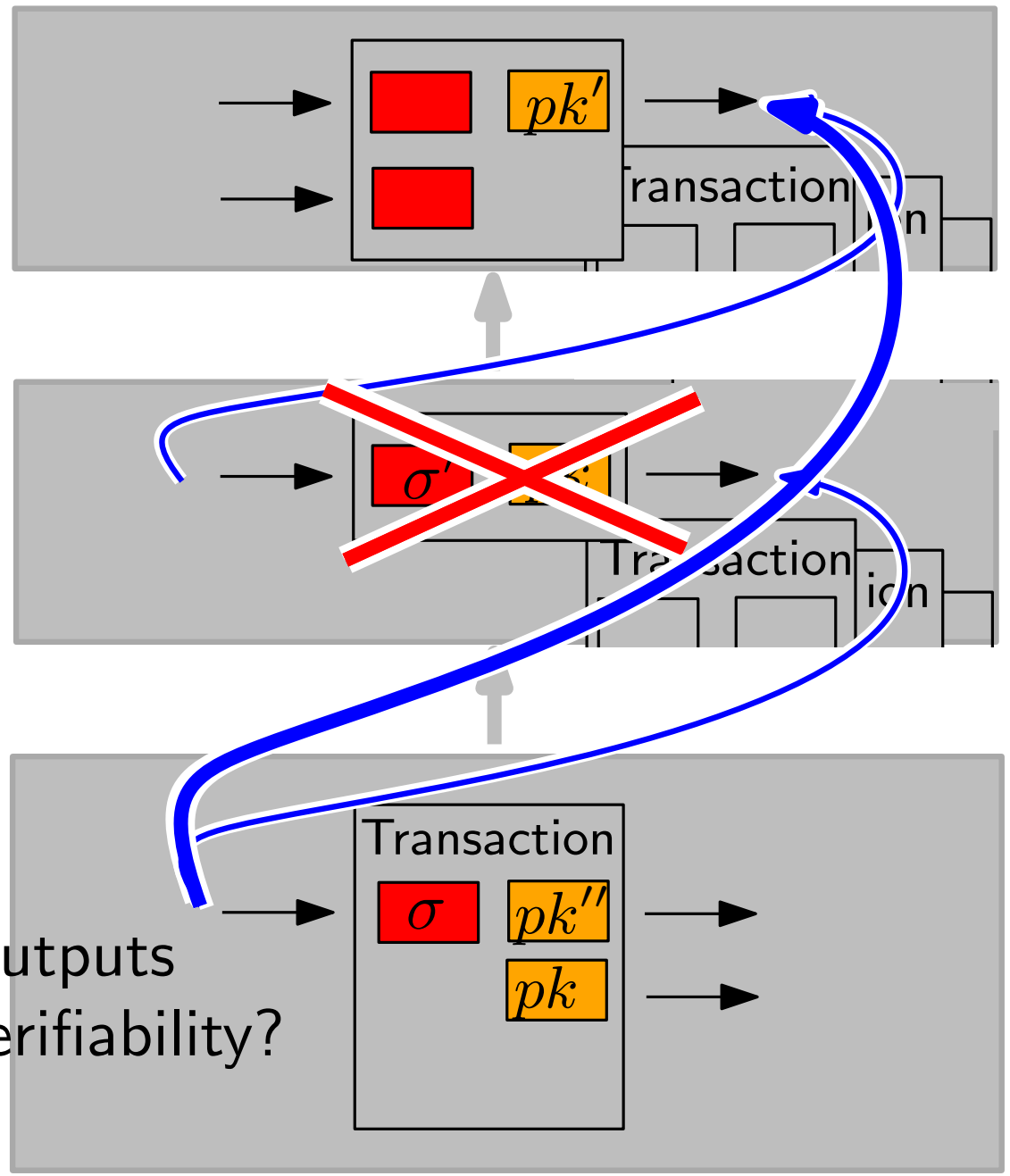


Scalability

“cut-through”

not possible
in Bitcoin:

σ' is needed
to verify validity



- only keep unspent outputs
- while maintaining verifiability?

Mimblewimble

- **Cryptocurrency scheme**

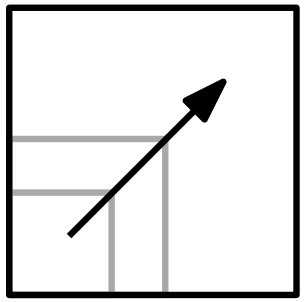
```
MIMBLEWIMBLE
Tom Elvis Jedusor
19 July, 2016

\****/
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 accompl                t by storing all transactions in a public
database called "th                and someone who genuinely wishes to check
this state must dow                e thing and basically replay each transacti
check each one as t                ile, most of these transactions have not
```

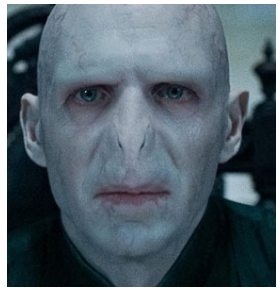
- proposed by “Tom Elvis Jedusor” in 2016

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

Mimblewimble

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





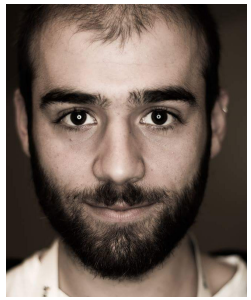
```
MIMBLEWIMBLE
Tom Elvis Jedusor
19 July, 2016

\****/
```

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.

Mimblewimble

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

formally analyzed in [FOS'19]

with Michele Orrù  and Yannick Seurin 

Aggregate Cash Systems:
A Cryptographic Investigation of Mimblewimble

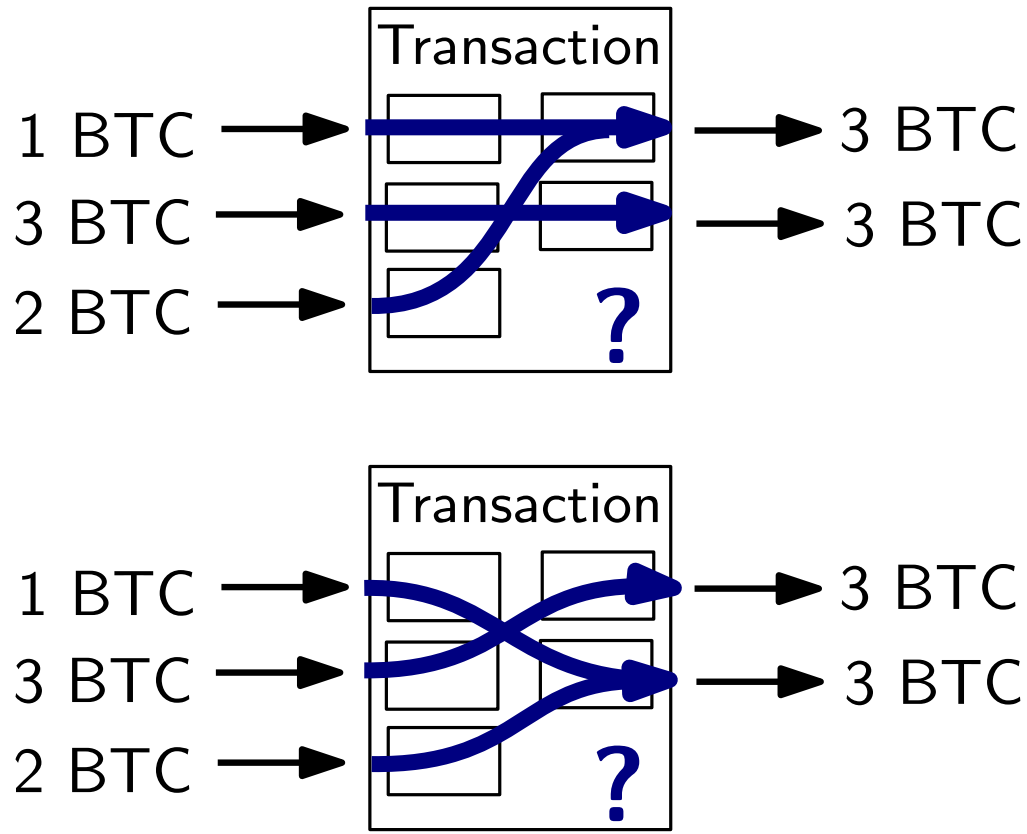
Georg Fuchsbauer[1,2], Michele Orrù[2,1], and Yannick Seurin[3]

[1] Inria
[2] École normale supérieure, CNRS, PSL University, Paris, France
[3] ANSSI, Paris, France
{georg.fuchsbauer, michele.orru}@ens.fr
yannick.seurin@m4x.org

Abstract. Mimblewimble is an electronic cash system proposed by an anonymous author in 2016. It combines several privacy-enhancing techniques initially envisioned for Bitcoin, such as Confidential Transactions (Maxwell, 2015), non-interactive merging of transactions

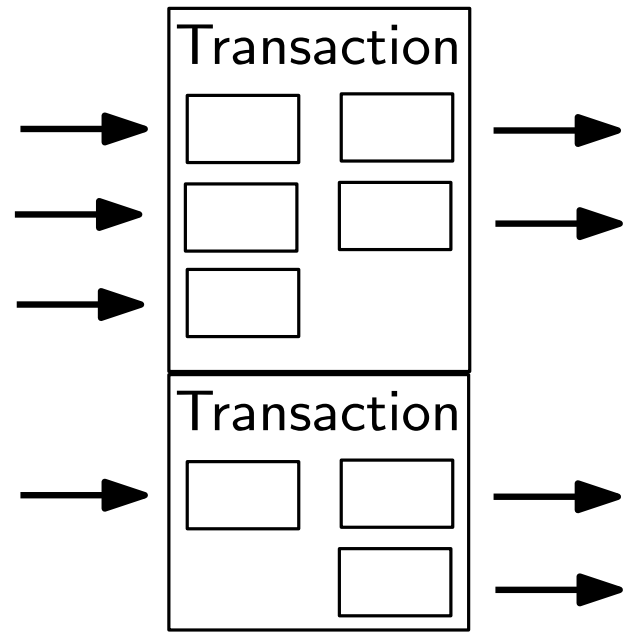
Anonymity

- Hiding path...



Anonymity

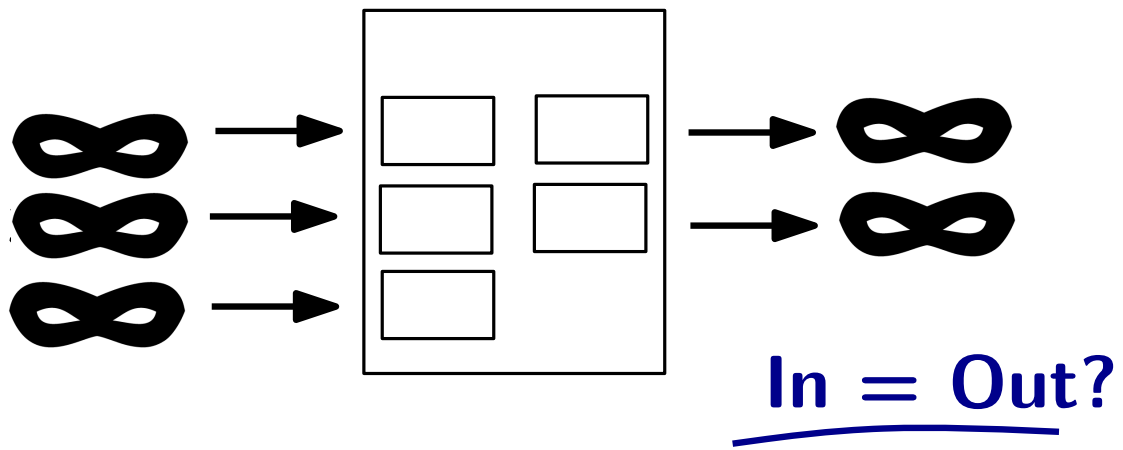
- Hiding path...



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

Anonymity

- Hiding amounts. . .

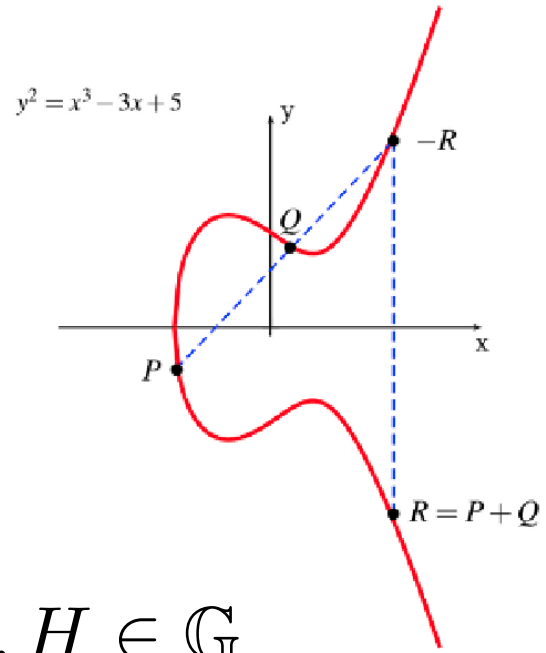


- **Confidential Transactions** [Maxwell]
  - hide the input and output *amounts*
  - not compatible with Bitcoin (used in MONERO)
  - balancedness verifiable?

Pedersen Commitments

Discrete-log-hard group $(\mathbb{G}, +)$

- generator G
- given $xG := \underbrace{G + \ldots + G}_{x \text{ times}}$, hard to find x



Pedersen Commitment with parameters $G, H \in \mathbb{G}$

- to *commit* to v, pick random r,
- $\mathbf{Com}(v; r) := vH + rG$

homomorphic:

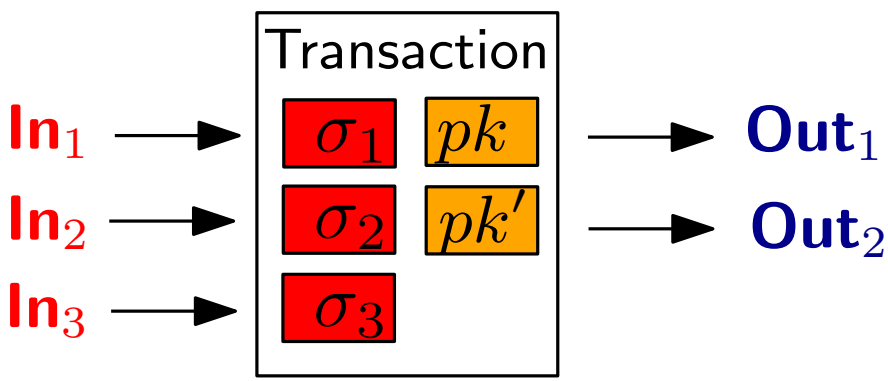
$$\begin{aligned}\mathbf{Com}(v_1; r_1) + \mathbf{Com}(v_2; r_2) &= (v_1 H + r_1 G) + (v_2 H + r_2 G) \\ &= (v_1 + v_2)H + (r_1 + r_2)G \\ &= \mathbf{Com}(v_1 + v_2; r_1 + r_2)\end{aligned}$$

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

Confidential Transactions

[Back, Maxwell '13 – '15]

- use *commitments* to amount **v**alues
- ensure that transactions do not create money?



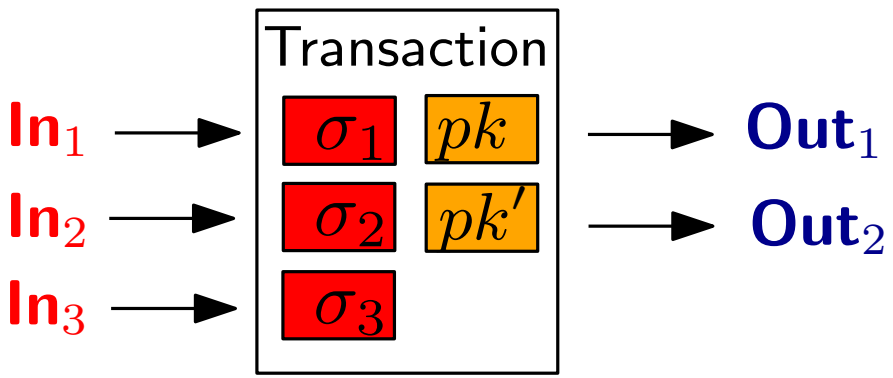
$$C = vH + rG$$

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

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

Confidential Transactions

Confidential transaction 



$$C = vH + rG$$

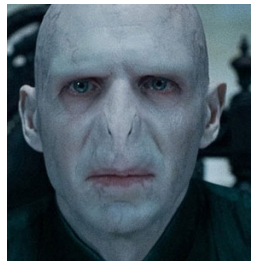
$$\sum \mathbf{Out} - \sum \mathbf{In} = \mathbf{0}$$

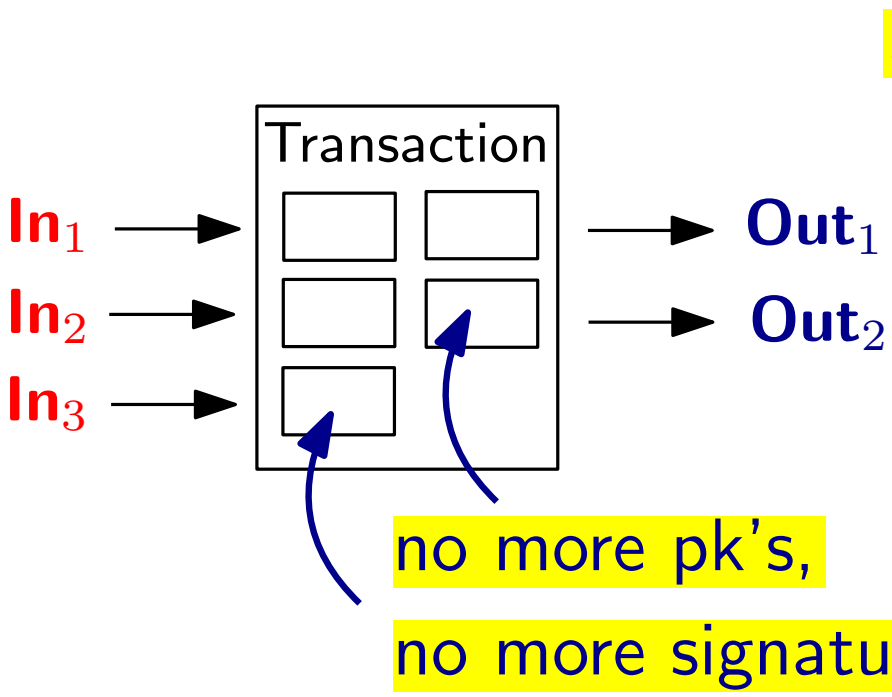
Signatures $\Rightarrow$

- no non-interactive CoinJoin
- no Cut-Through

Mimblewimble

[Jedusor '16]



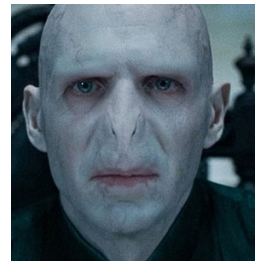


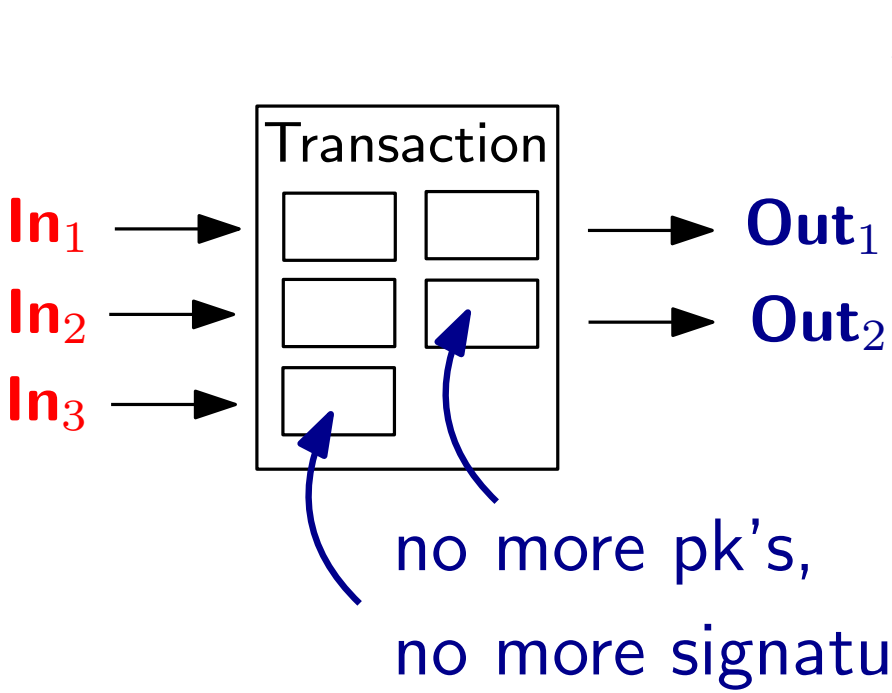
But: sender knows sum of output r's

$\Rightarrow$ users choose independent keys

Mimblewimble

[Jedusor '16]





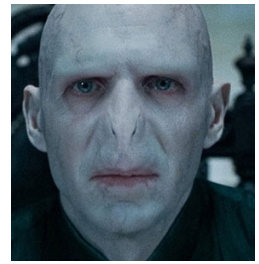
$$C = vH + rG$$

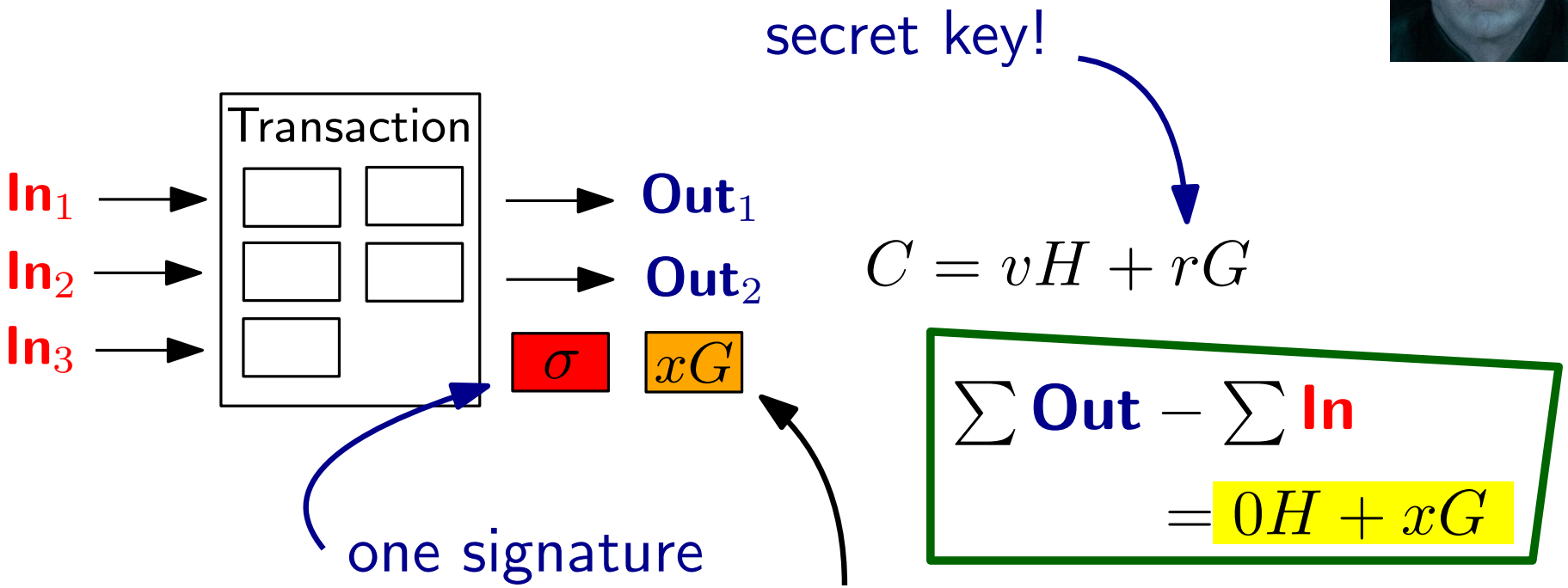
$$\sum \textbf{Out} - \sum \textbf{In} = 0H + xG$$

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

Mimblewimble

[Jedusor '16]





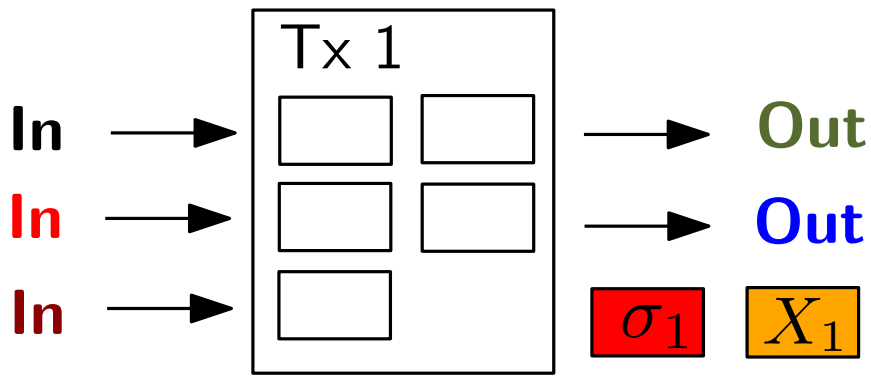
$$C = vH + rG$$

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

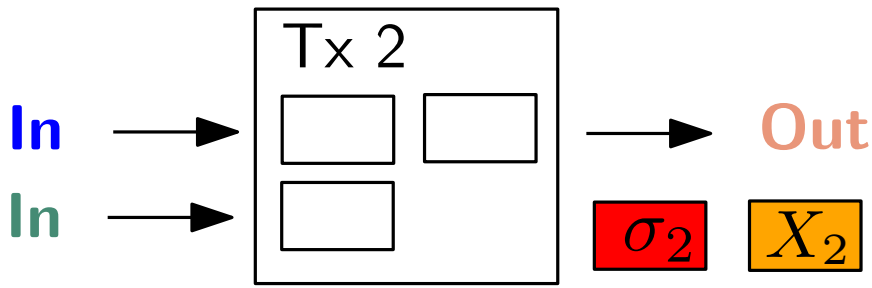
"proves" that $\overbrace{\sum \mathbf{Out} - \sum \mathbf{In}}$ is commitment to 0

σ not only proves balancedness, but also prevents theft of coins

Mimblewimble



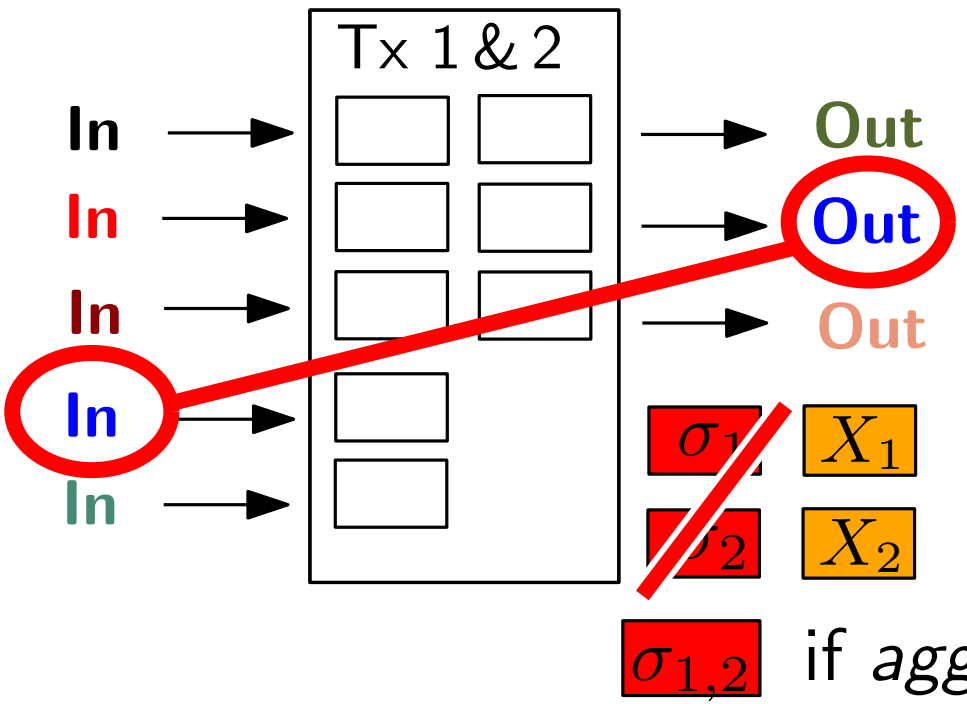
- $\sum \mathbf{Out}_1 - \sum \mathbf{In}_1 = X_1$
- σ_1 valid for X_1



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

Mimblewimble

Post-confirmation Cut-Through

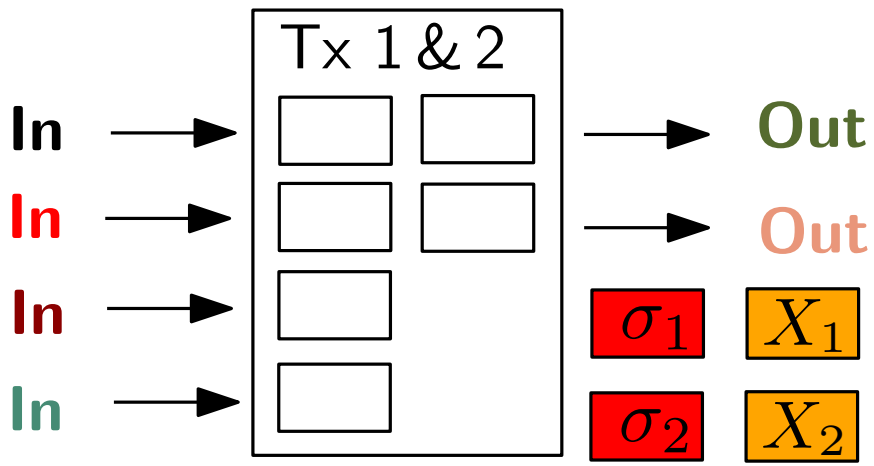


- $\sum$ **Out** $-\sum$ **In** $= X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

if *aggregate signature* scheme (BLS)

Mimblewimble

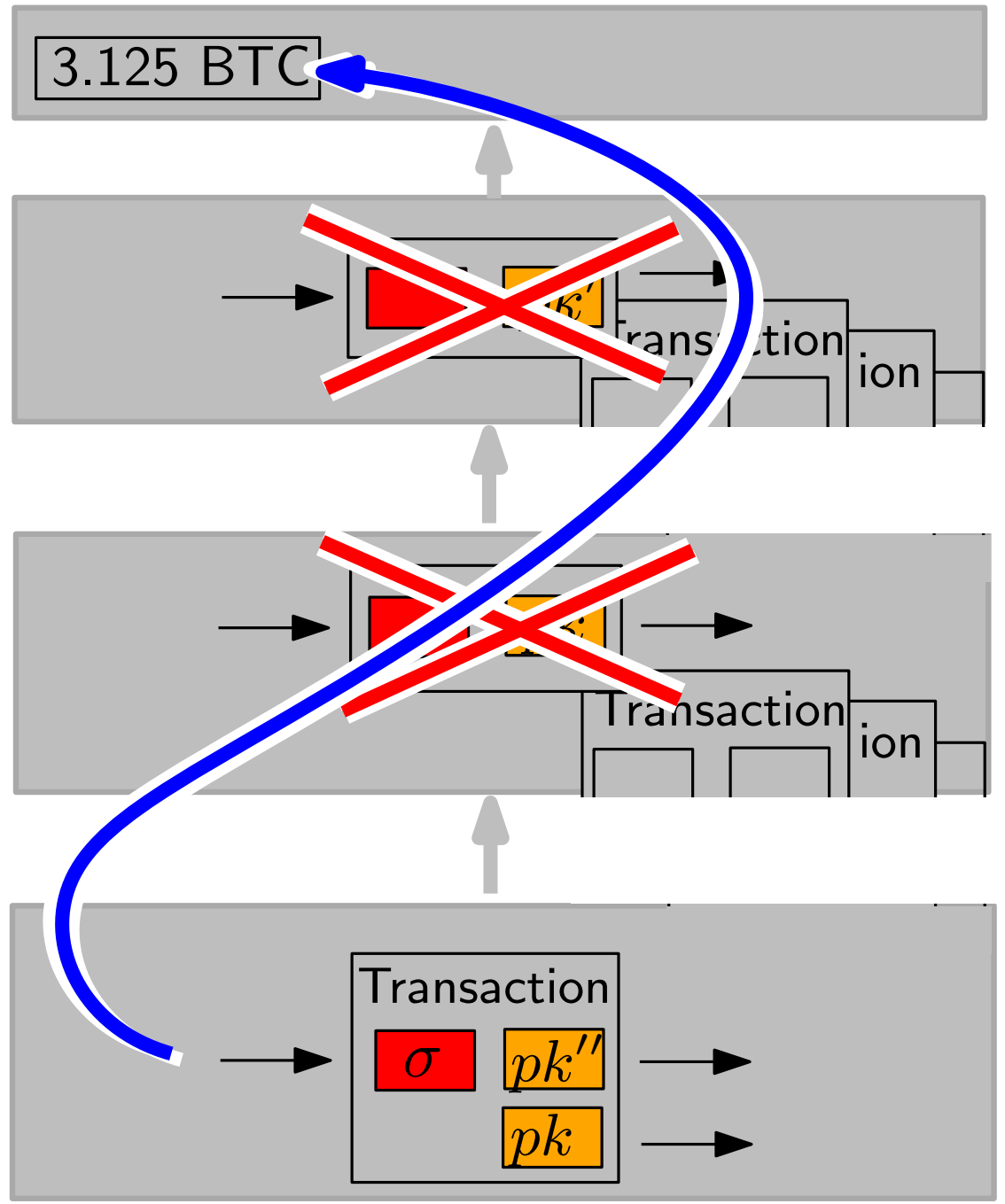
Post-confirmation Cut-Through



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

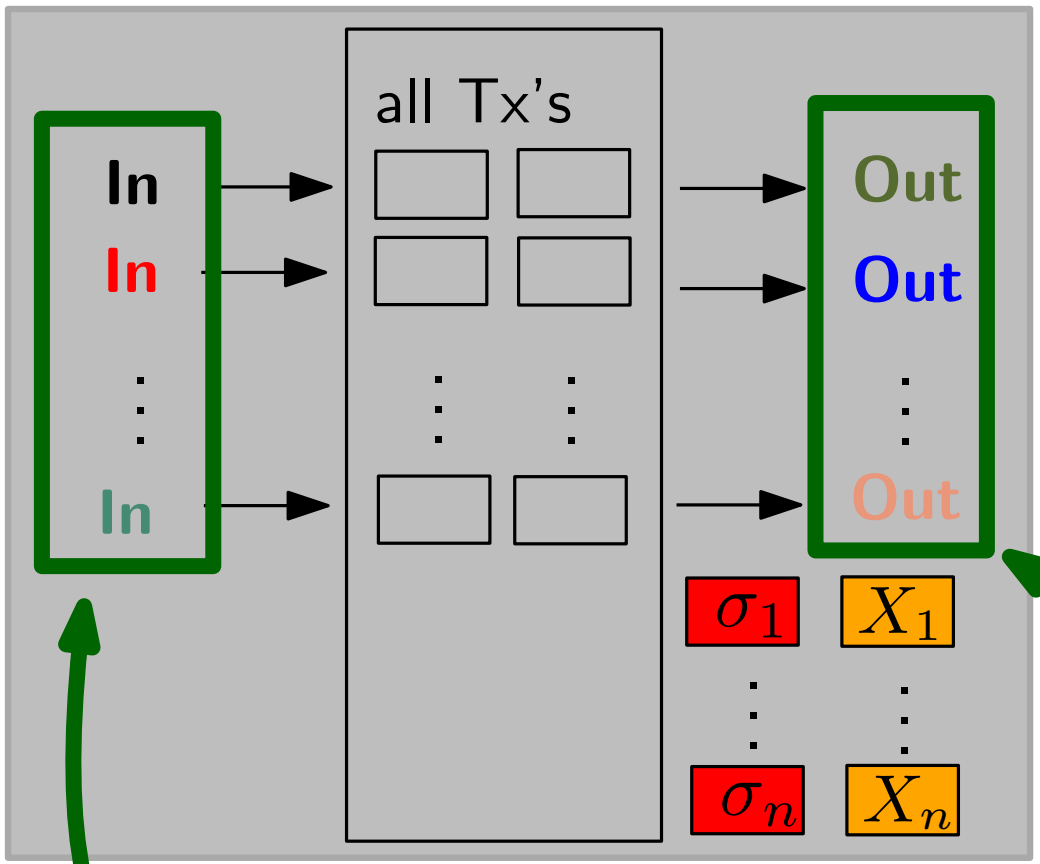
Scalability

“cut-through”



Mimblewimble

Cut through all transactions in blockchain



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

UTXO set

Only coinbase transactions

Applications

Implemented by several cryptocurrencies (since 2019):

#	Name	Price	1h %	24h %	7d %	Market Cap
1	Bitcoin BTC	$91,196.89	▲0.20%	▲0.60%	▲5.95%	$1,819,897,064,095
2	Ethereum ETH	$3,006.72	▼0.01%	▼0.06%	▲7.03%	$362,898,536,167
⋮						
1158	Grin GRIN	$0.03562	▲0.01%	▼1.89%	▼11.82%	$7,705,882
⋮						
1319	Beam BEAM	$0.03488	▲1.05%	▼7.56%	▲5.78%	$5,258,781

Applications

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

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

2021: fixed by Burkett, F, Orrù

analyzed by F, Orrù [FO'22]

Non-interactive Mimblewimble transactions, revisited

Georg Fuchsbauer[1] and Michele Orrù[2]

[1] TU Wien, Austria
[2] UC Berkeley, USA
first.last@{tuwien.ac.at,berkeley.edu}

Abstract. Mimblewimble is a cryptocurrency protocol that promises to overcome notorious blockchain scalability issues and provides user privacy. For a long time its wider adoption has been hindered by the lack of non-interactive transactions, that is, payments for which

Applications

2022: implemented in **Litecoin** as "MW extension blocks"

#	Name	Price	1h %	24h %	7d %	Market Cap
1	Bitcoin BTC	$91,196.89	▲0.20%	▲0.60%	▲5.95%	$1,819,897,064,095
2	Ethereum ETH	$3,006.72	▼0.01%	▼0.06%	▲7.03%	$362,898,536,167
⋮						
18	Ethena USDe USDe	$0.9995	▲0.00%	▲0.01%	▲0.08%	$7,195,807,649
19	Litecoin LTC	$84.27	▲0.11%	▲0.08%	▲1.13%	$6,453,544,744
20	Avalanche AVAX	$14.18	▲0.40%	▼3.68%	▲6.51%	$6,086,744,872

Mina

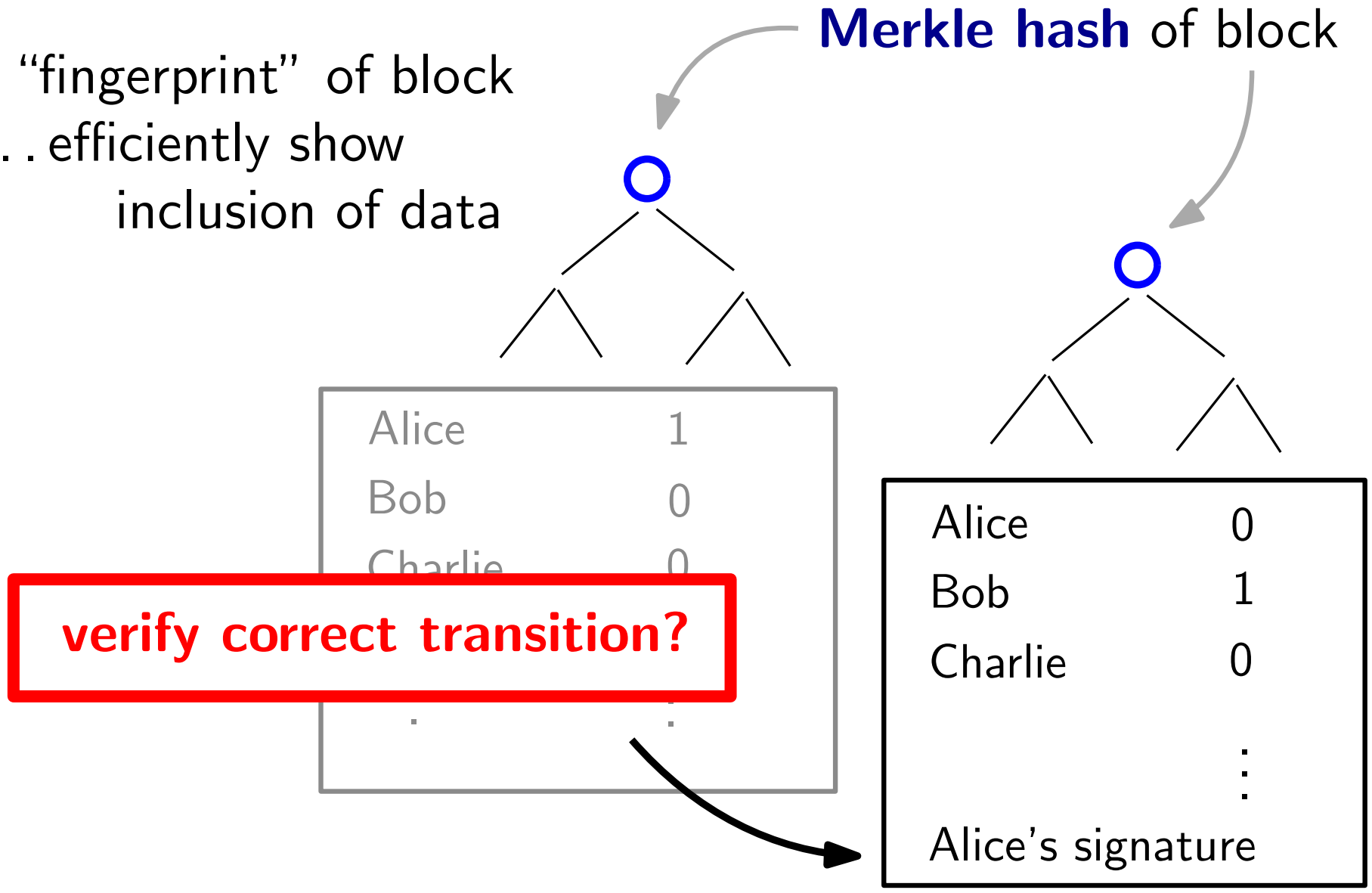
. . . constant-size blockchain

#	Name	Price	1h %	24h %	7d %	Market Cap
1	Bitcoin BTC	$91,196.89	▲0.20%	▲0.60%	▲5.95%	$1,819,897,064,095
2	Ethereum ETH	$3,006.72	▼0.01%	▼0.06%	▲7.03%	$362,898,536,167
⋮						
18	Ethena USDe USDe	$0.9995	▲0.00%	▲0.01%	▲0.08%	$7,195,807,649
19	Litecoin LTC	$84.27	▲0.11%	▲0.08%	▲1.13%	$6,453,544,744
20	Avalanche AVAX	$14.18	▲0.40%	▼3.68%	▲6.51%	$6,086,744,872
236	Mina MINA	$0.09780	▼0.47%	▼7.23%	▼8.84%	$123,639,196

Mina

. . . "fingerprint" of block

. . . efficiently show inclusion of data



Mina

SNARK . . . succinct proof for any NP statement

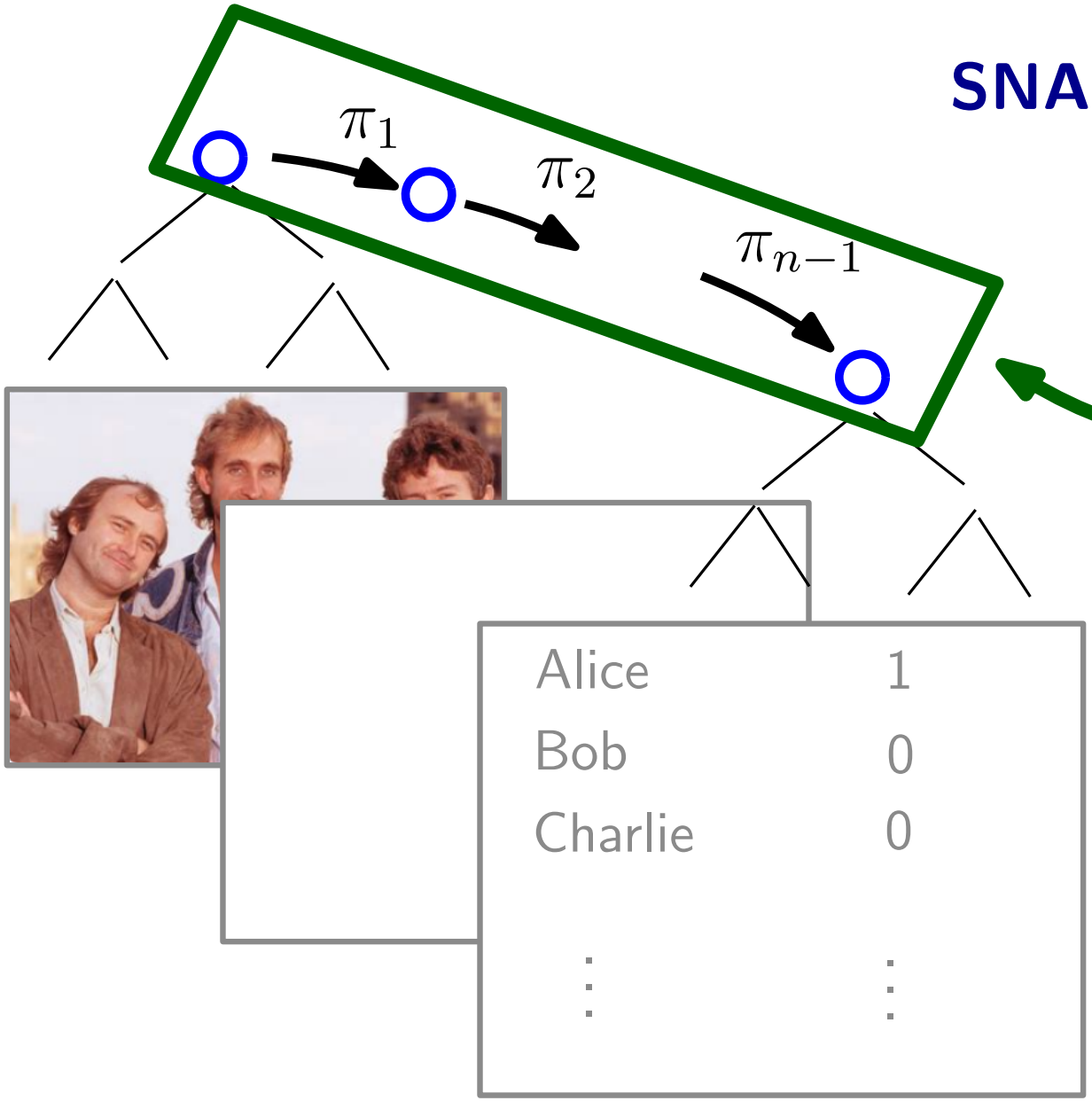
kilobytes. . .

π . . . proof that block correct

Alice	1
Bob	0
Charlie	0
⋮	⋮

Alice	0
Bob	1
Charlie	0
	⋮
Alice's signature	

Mina

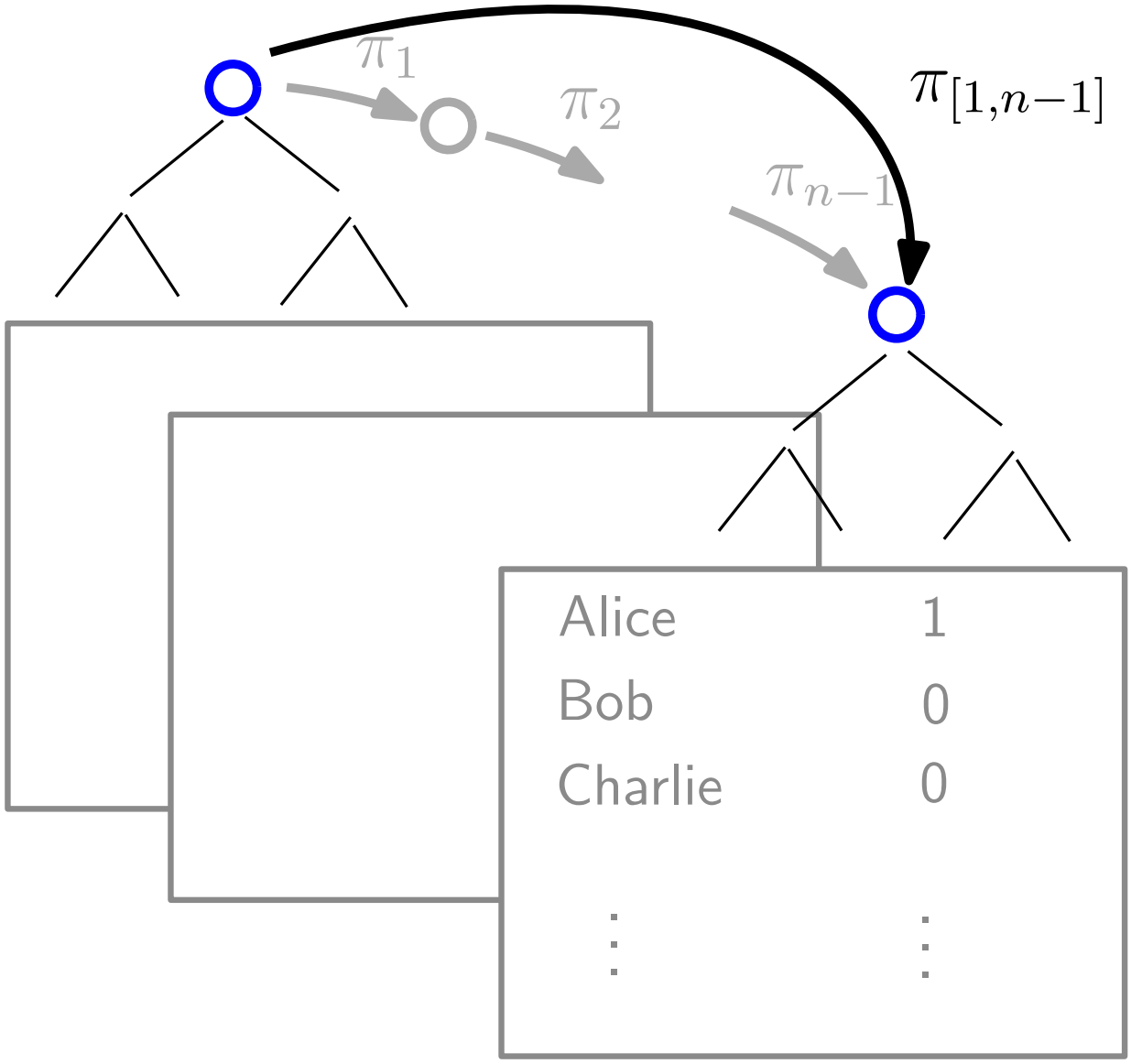


SNARK . . . succinct proof for any NP statement

Blockchain

constant size?

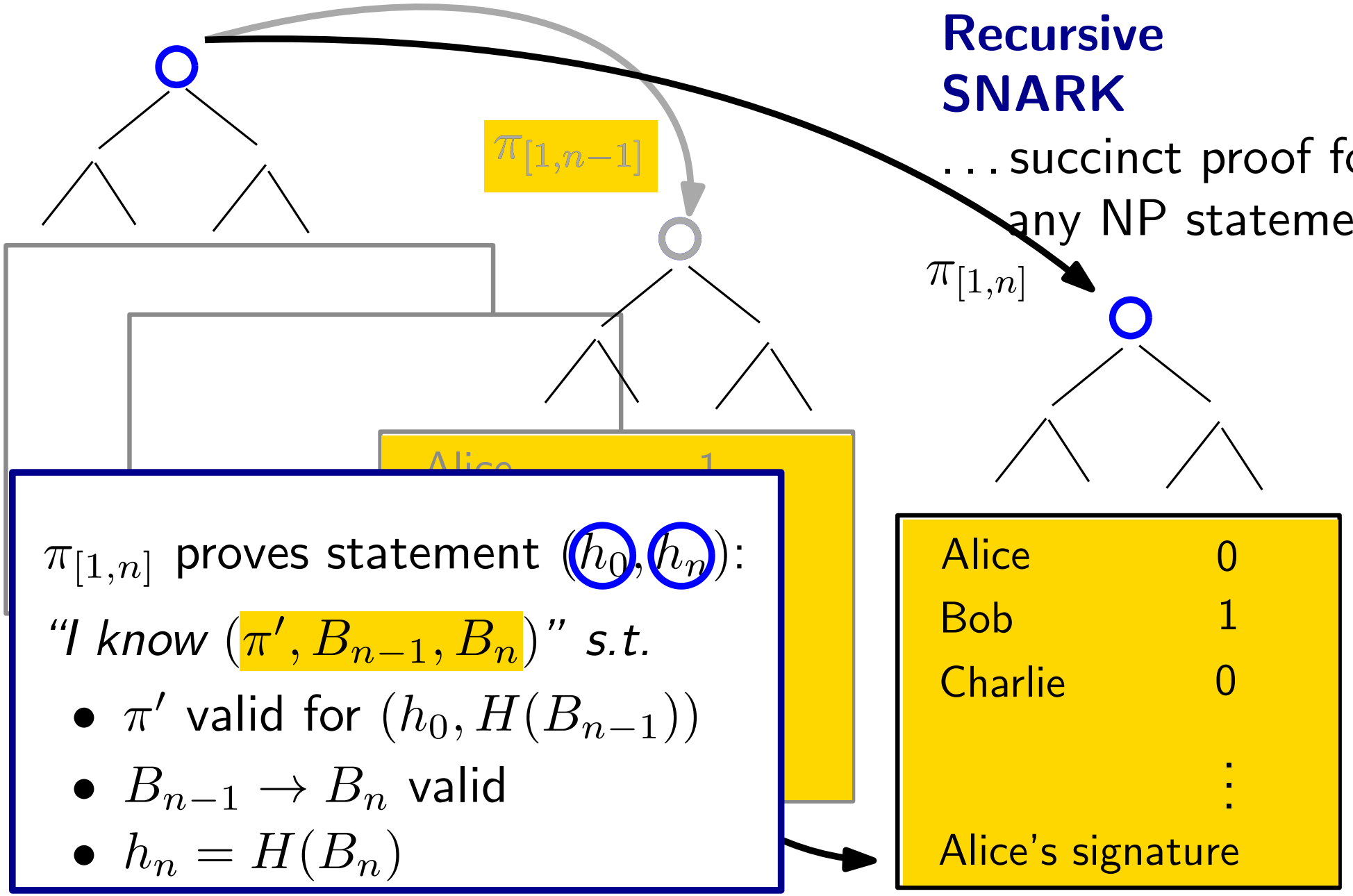
Mina



SNARK

. . . succinct proof for any NP statement

Mina



Recursive SNARK

. . . succinct proof for any NP statement

$\pi_{[1,n]}$ proves statement (h_0, h_n):

"I know (π', B_{n-1}, B_n)*" s.t.*

- π' valid for $(h_0, H(B_{n-1}))$
- $B_{n-1} \to B_n$ valid
- $h_n = H(B_n)$

Recursive SNARKs

- Proved statement includes "Verify(x', π')"

Random oracle model

- Idealization used to prove security of SNARKs analysis pretends hash function is *random function*
- e.g. *Plonk* (Mina uses variant)

Problem: "Verify(x', π')" uses actual hash function

$\Rightarrow$ security of π'??

On the Security of Plonk Without Random Oracles

Georg Fuchsbauer and Marek Sefranek

TU Wien, Vienna, Austria
firstname.lastname@tuwien.ac.at

Abstract. Plonk is one of the most influential and widely used zk-SNARKs, with proofs of constant size (0.5 kB) and sublinear verification time. Its setup is circuit-independent supporting proofs of arbitrary