Space-efficient blockchains

Georg Fuchsbauer

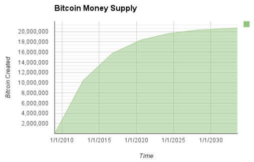
Sustainability in Computer Science, TUW, 27 Jan ‘25

Bitcoin

What

- digital currency
  (most successful ever)
- decentralized
  (no bank)

- hard-coded inflation



- pseudonymous

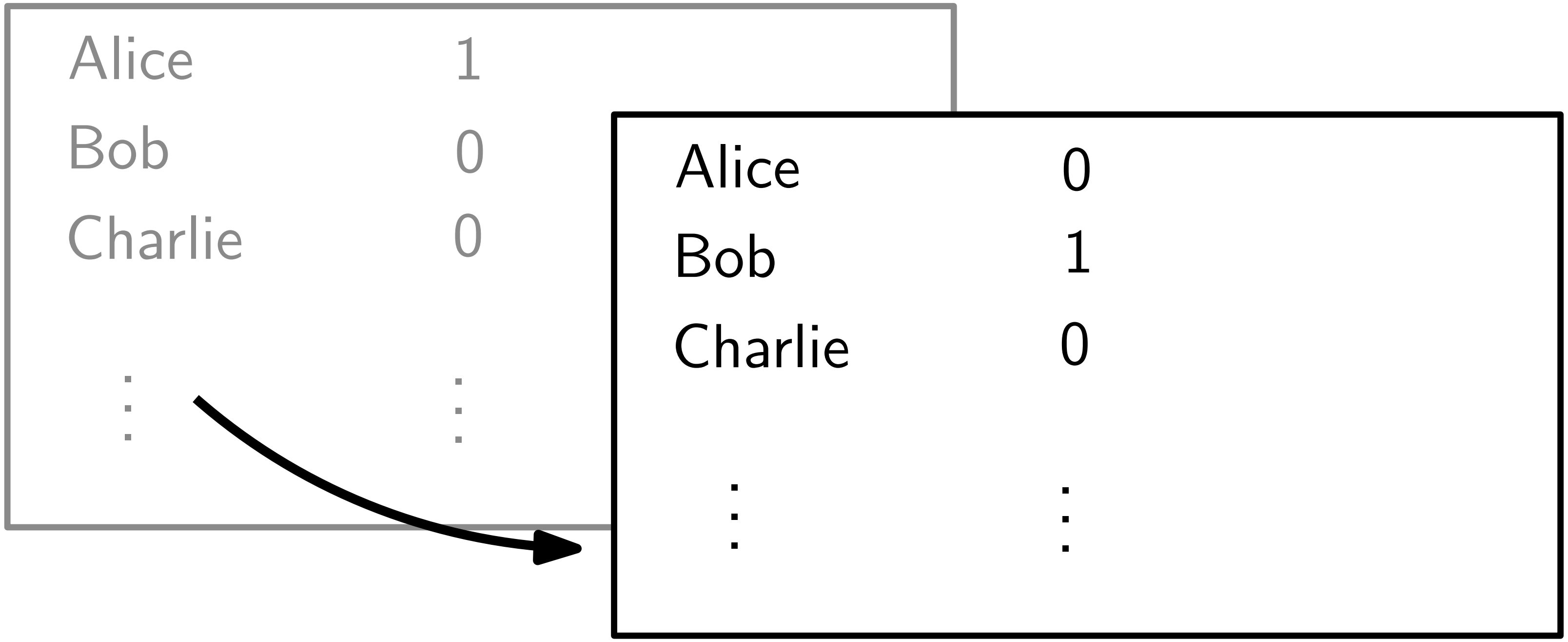
How

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

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

Ledger

Public ledger (maintained by authority)



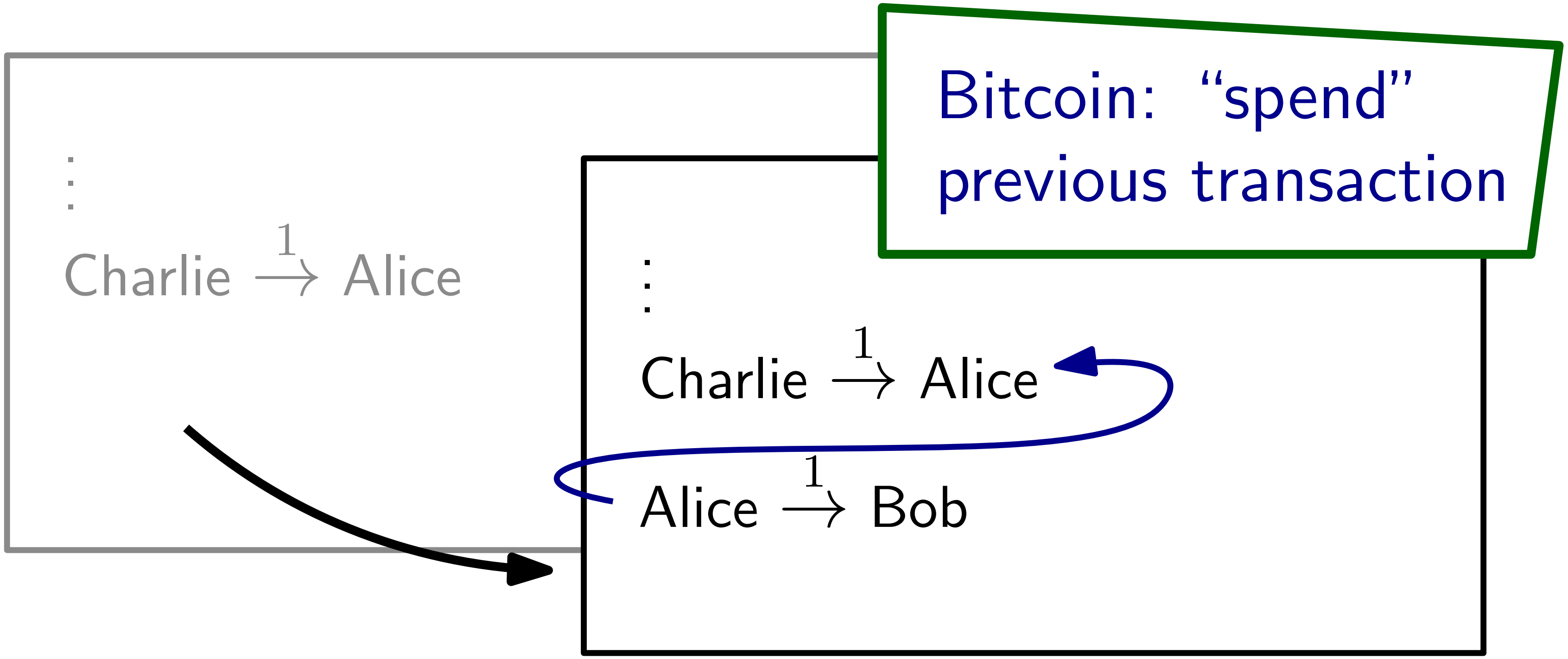
Alice	1
Bob	0
Charlie	0
⋮	⋮

Alice	0
Bob	1
Charlie	0
⋮	⋮

Alice: transfer 1 → Bob

Ledger

Public ledger (records all transactions)



⋮

Charlie $\xrightarrow{1}$ Alice

⋮

Charlie $\xrightarrow{1}$ Alice

Alice $\xrightarrow{1}$ Bob

Bitcoin: “spend” previous transaction

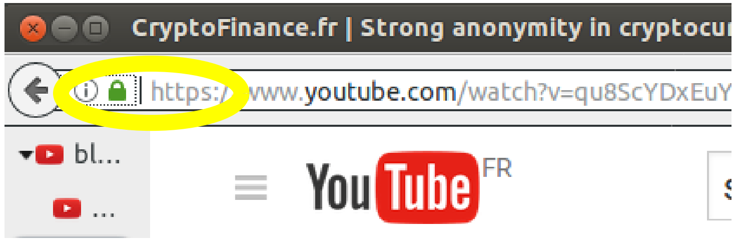
Bitcoin:

– no notion of account

– only transactions

how to identify?

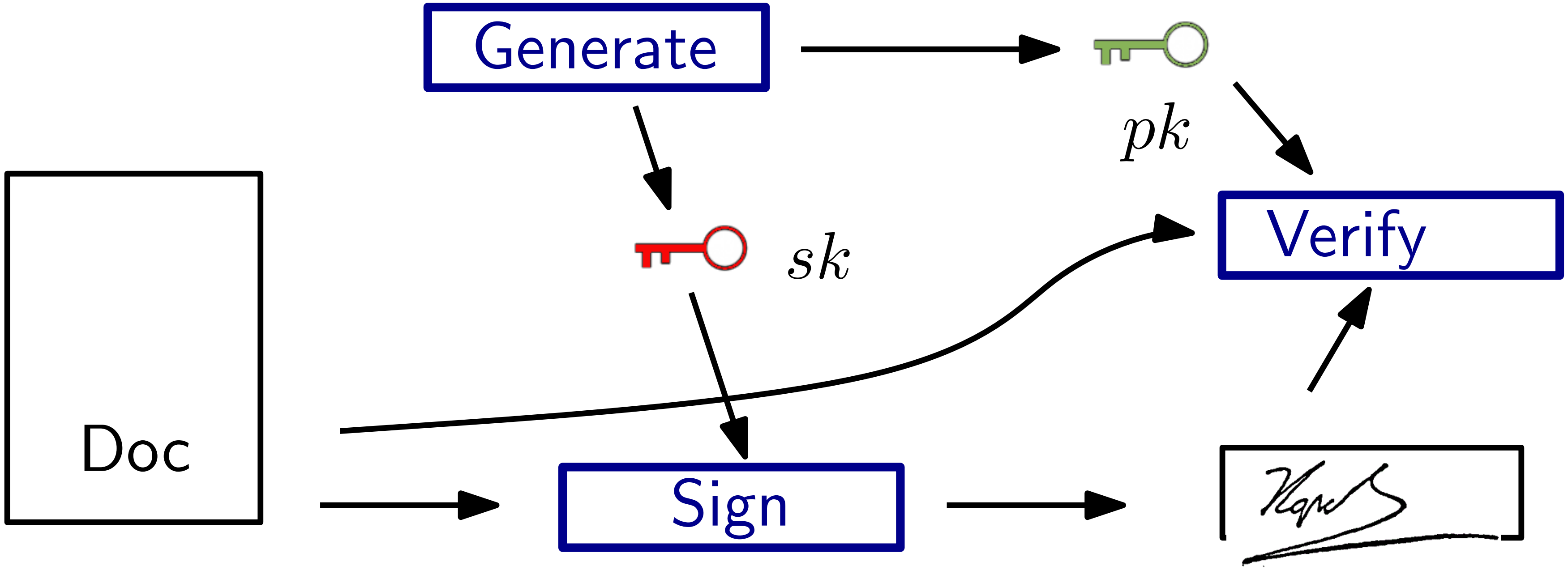
Digital signatures



CryptoFinance.fr | Strong anonymity in cryptocu

https://www.youtube.com/watch?v=qu8ScYDxEuY

- Alice can create a **key pair**
  - **secret key** used to sign messages
  - **public key** lets anyone verify signatures



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

Digital signatures

- Alice can create a **key pair**
  - **secret key** used to sign messages
  - **public key** lets anyone verify signatures

Simplification:

- Public key ↔ coin
- Secret key: enables spending of coin

Transactions

- owns $\boxed{pk_A}$ i.e. it's in the ledger
- creates $\boxed{pk_B}$
- signs $\boxed{pk_A \to pk_B}$ and adds to ledger

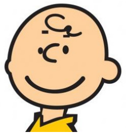
transaction

Double-spending

hard to create

physical coin $\neq$ digital coin

easy to copy!

- signs $\boxed{pk_A \to pk_B}$
- signs $\boxed{pk_A \to pk_C}$ 

Ledger only accepts if

- exists transaction $\boxed{\;*\; \to pk_A}$!
- no transaction $\cancel{\boxed{pk_A \to \;*\;}}$

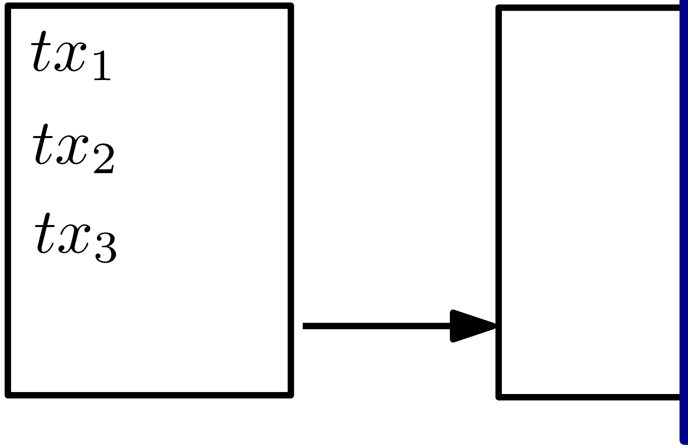
SPyCoDe FWF Der Wissenschaftsfonds.
W|W|T|F
TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

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 . . .

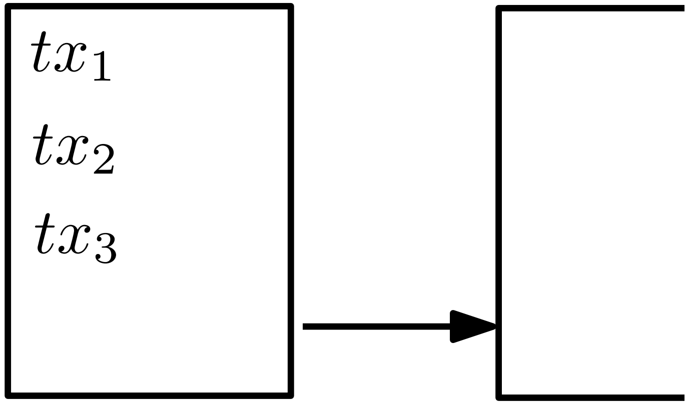
$\Rightarrow$ Krzysztof Pietrzak:
Sustainable Blockchains
(25 Nov 2024)

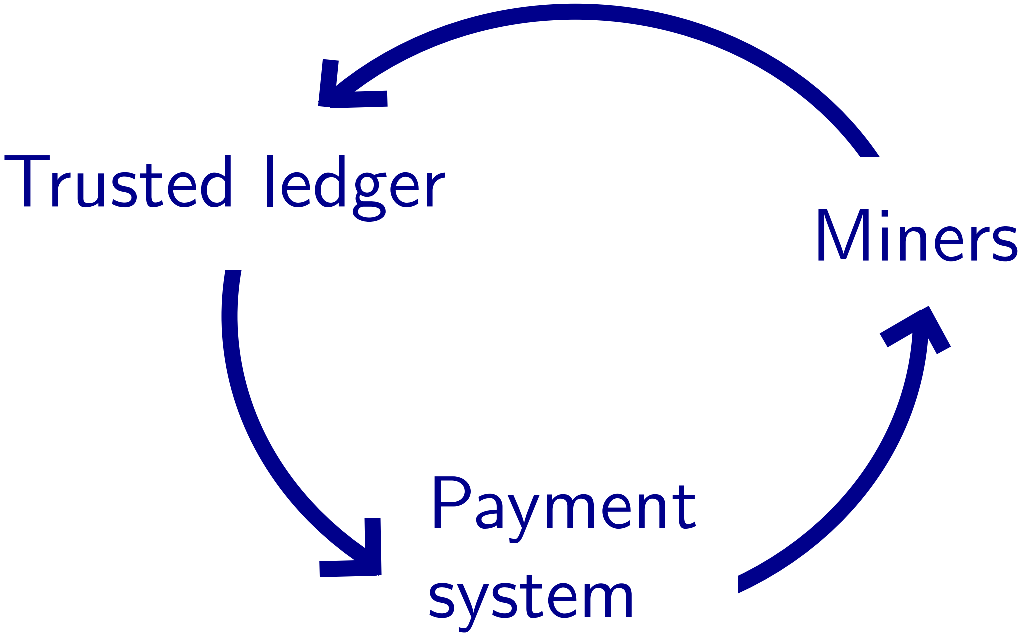
Decentralization

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

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

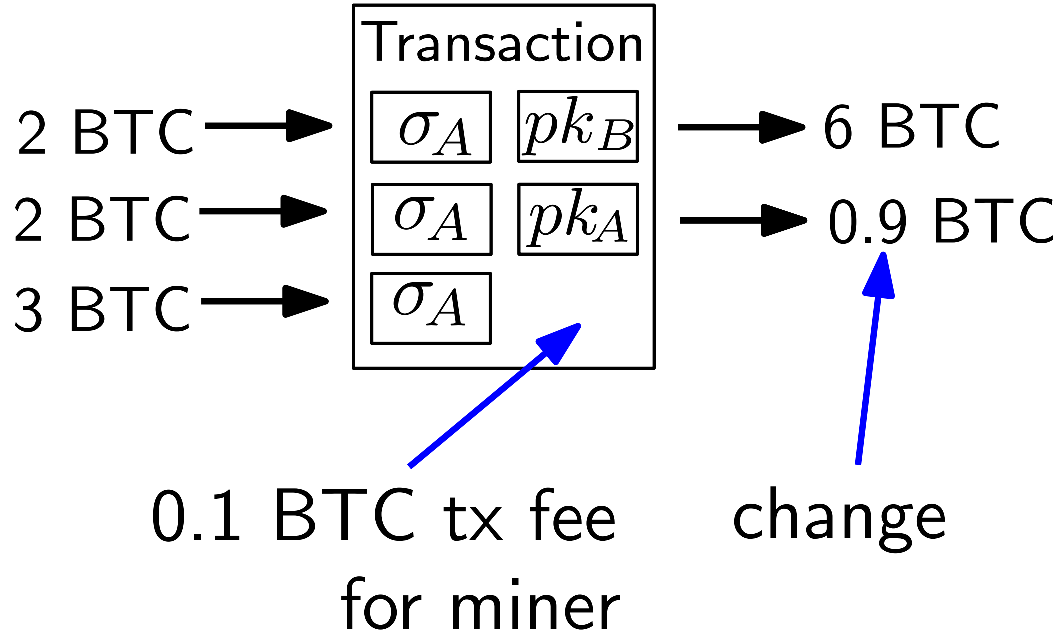
The Blockchain





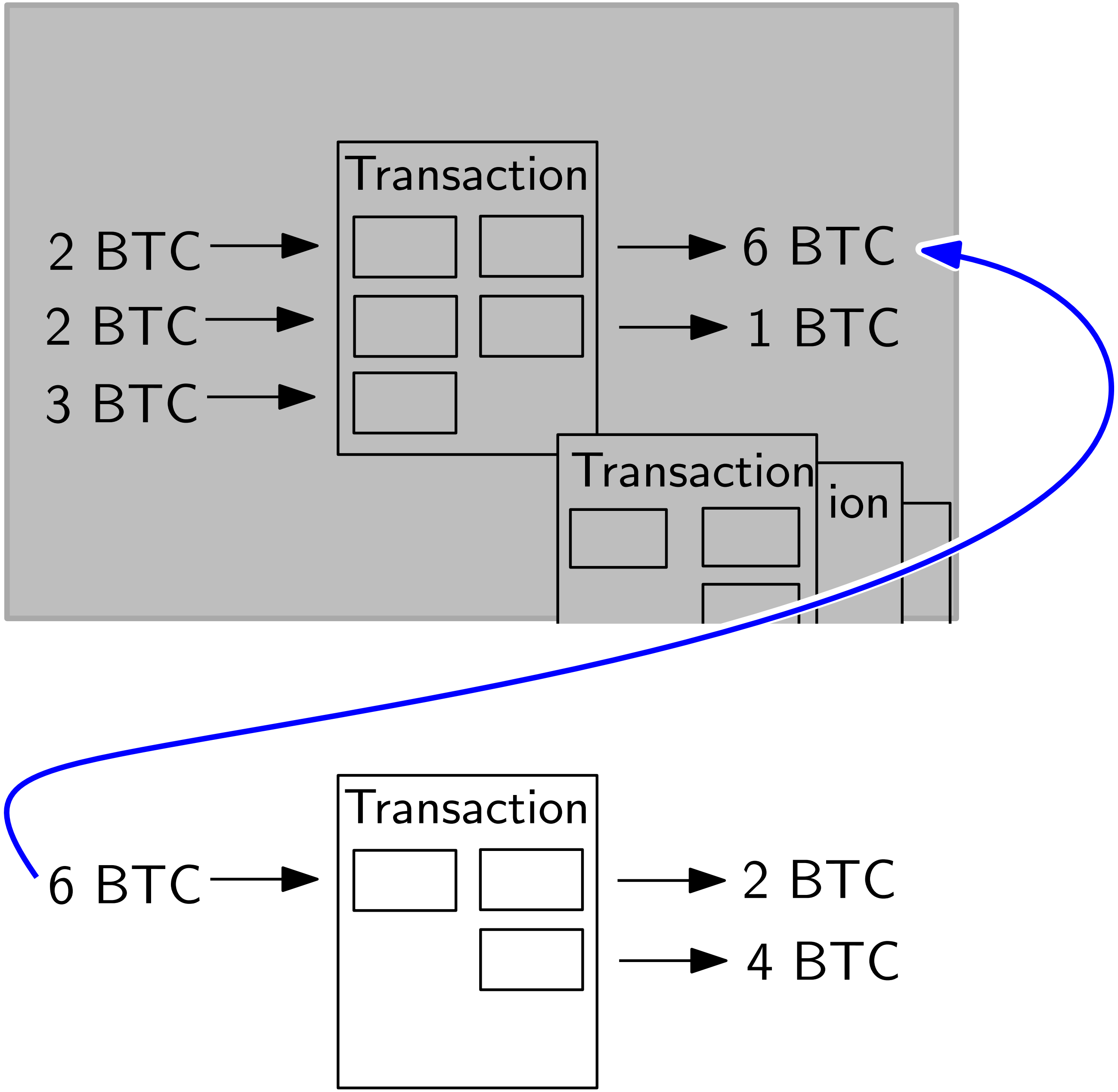
Transaction details

- **Transactions**

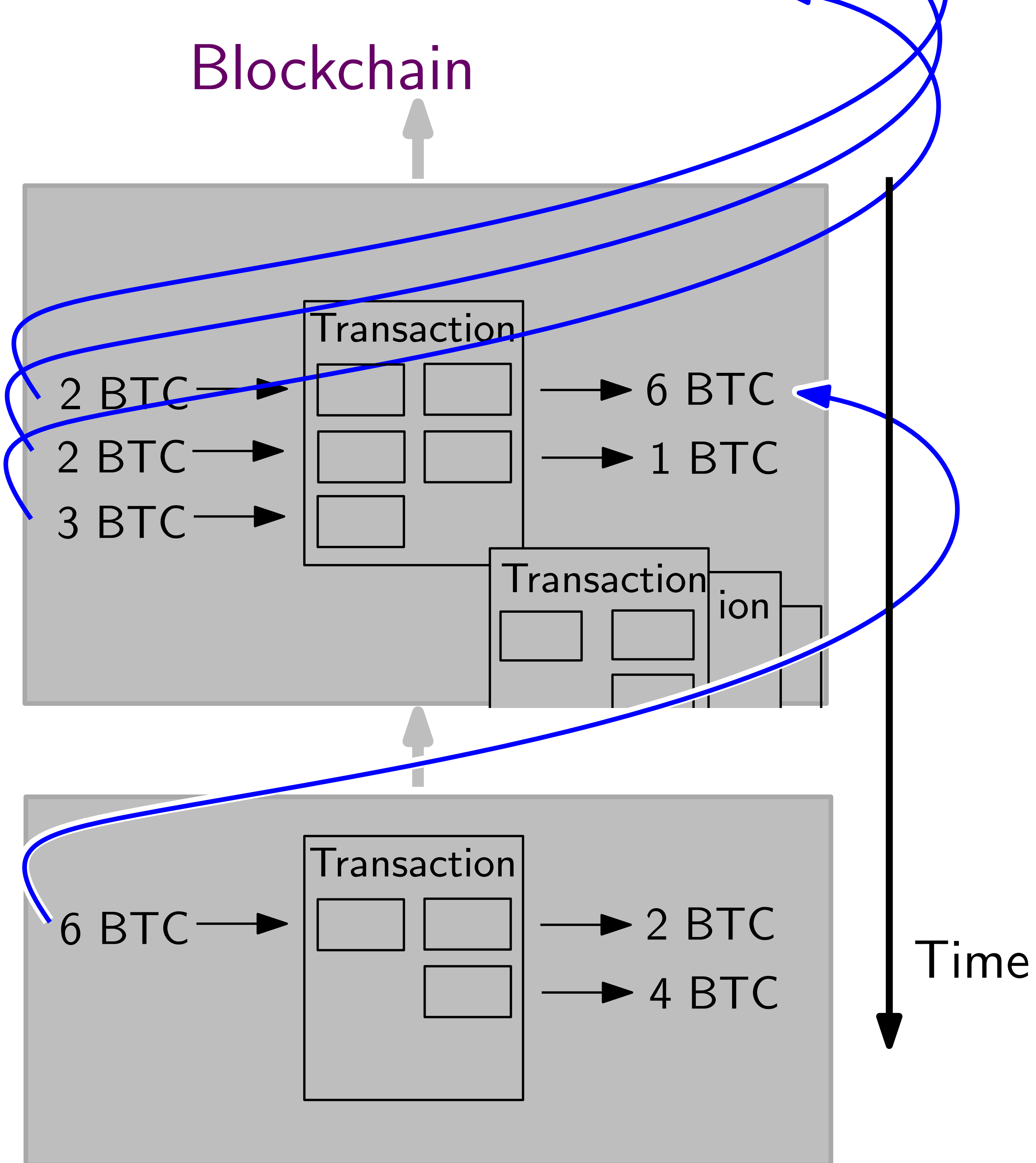


Blockchain

- **Block**



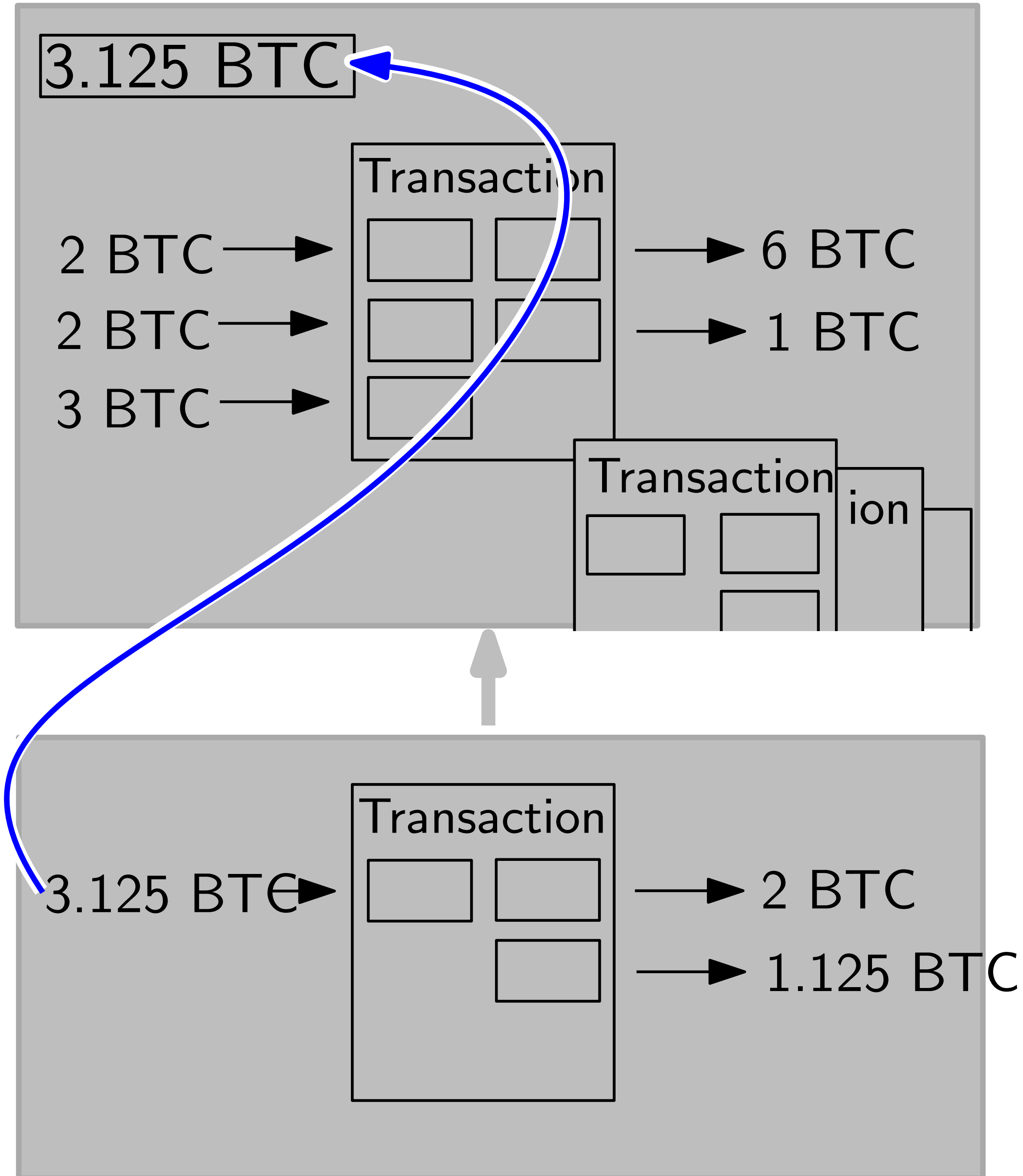
- Reference to previous output


Blockchain
Transaction
2 BTC
2 BTC
3 BTC
6 BTC
1 BTC
Transaction
ion
Transaction
6 BTC
2 BTC
4 BTC
Time

- **Blockchain**

Blockchain

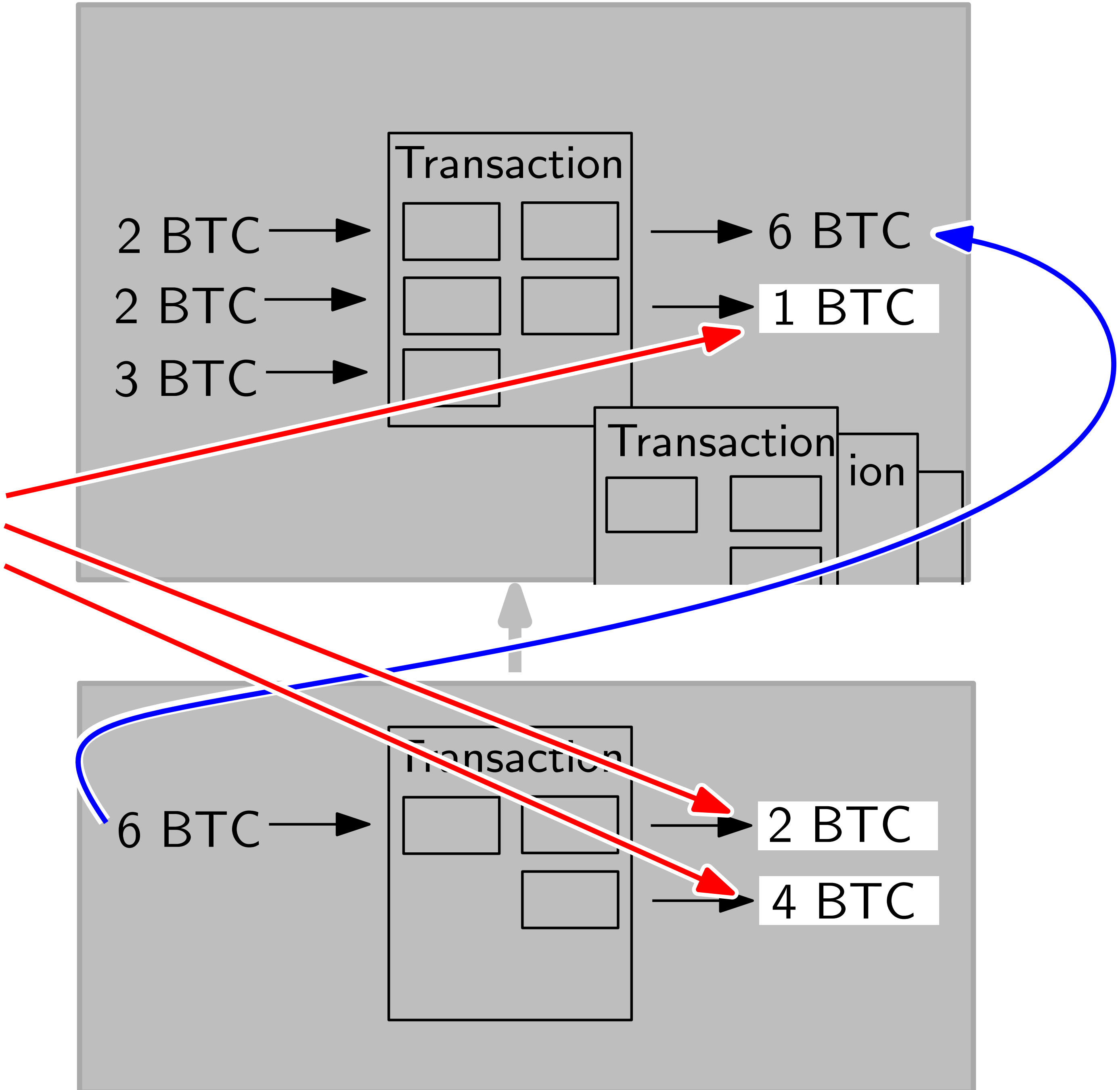
- **Coinbase transaction**



Blockchain

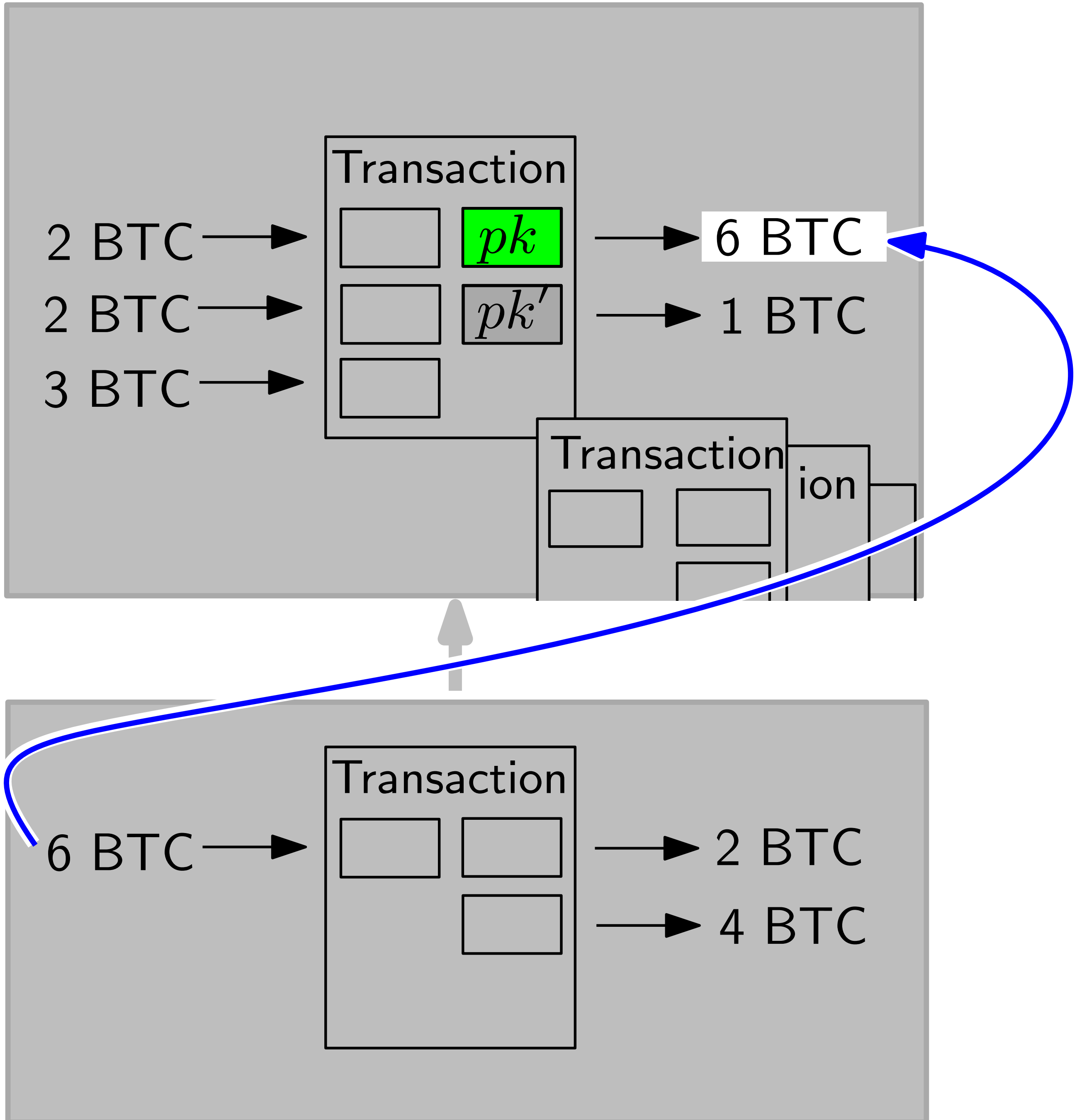
Unspent transaction outputs (UTXO's)

= existing money in system



Bitcoin

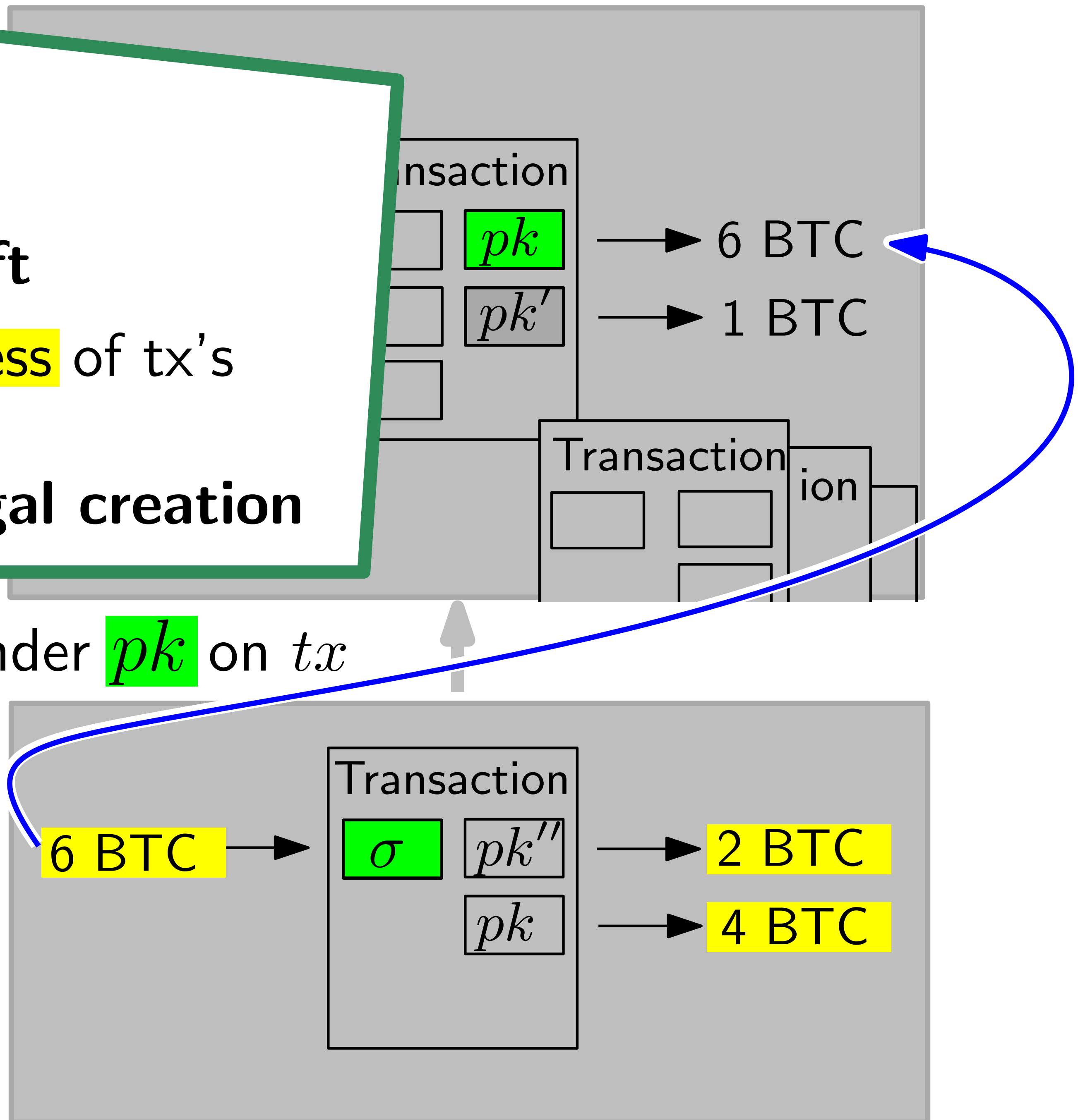
- **Owning**
  an output



Bitcoin

Security

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

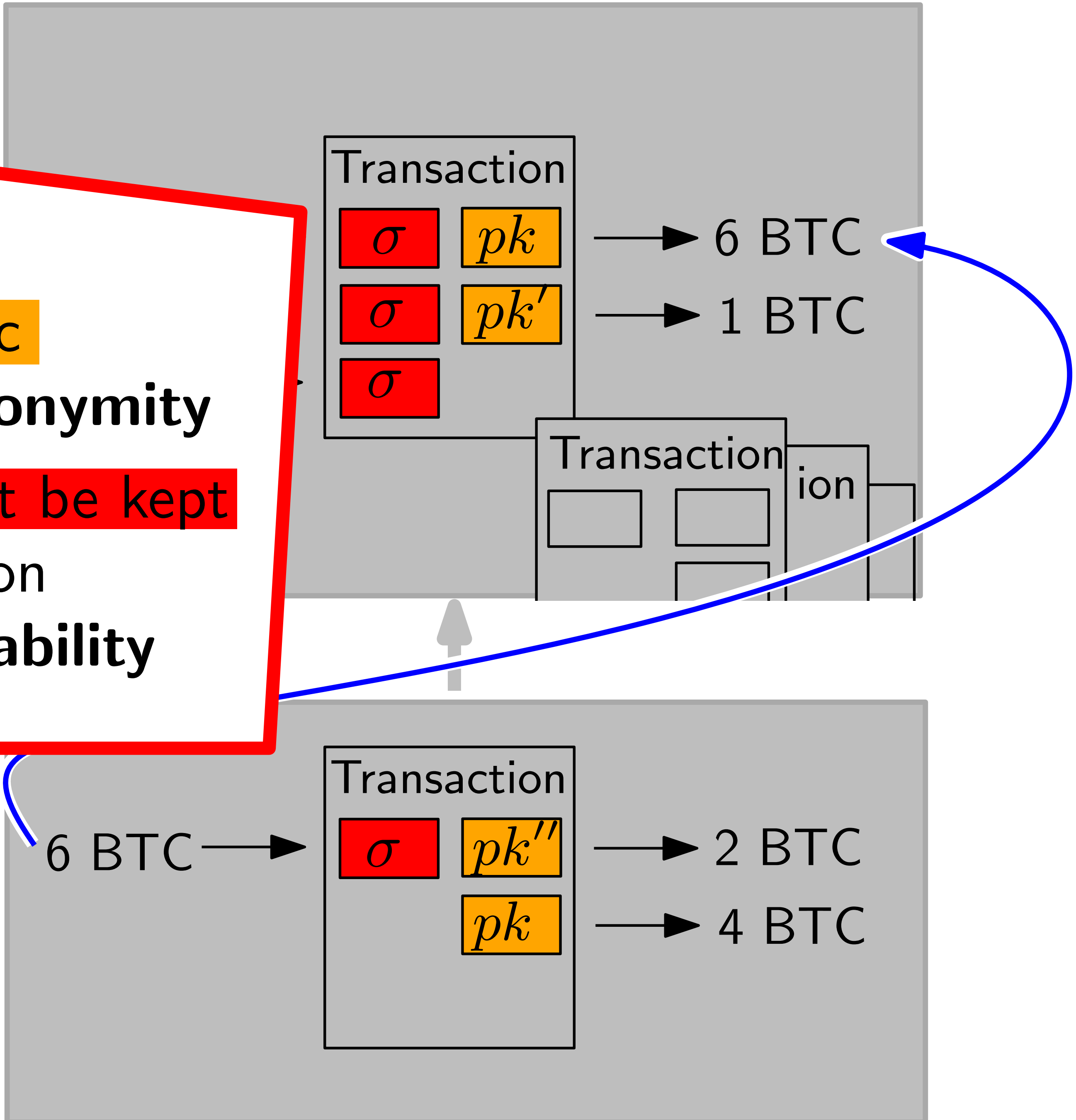


σ is **signature** under pk on tx

Bitcoin

Drawbacks

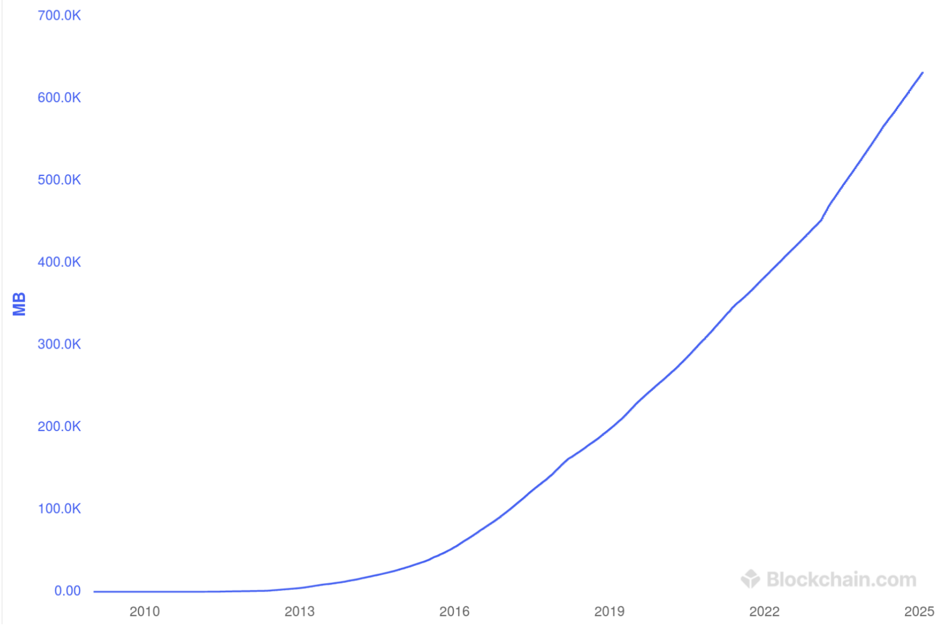
- all tx's public
  ⇒ **weak anonymity**
- all data must be kept for verification
  ⇒ **bad scalability**



Scalability

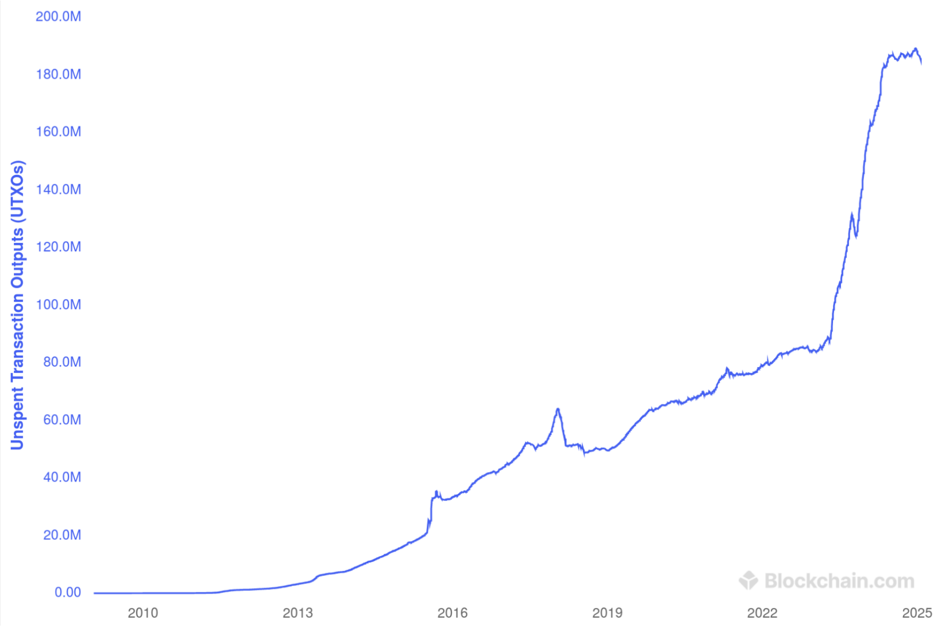
Blockchain size:
> 600 GB

ethereum:
> 1.2 TB



When starting full node: download and verify blockchain

Size of UTXO set:
< 10 GB

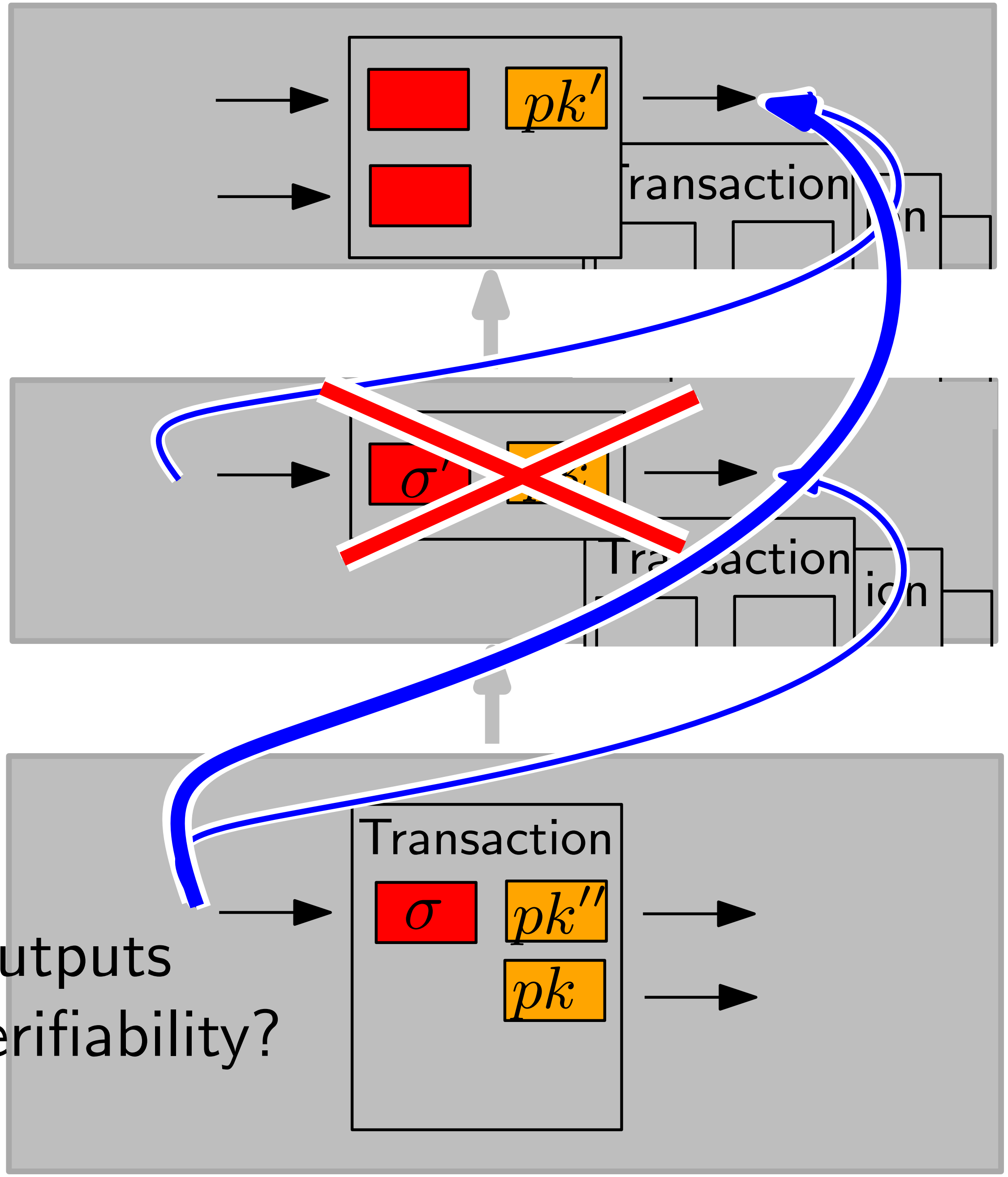


Scalability

"cut-through"

not possible
in Bitcoin:

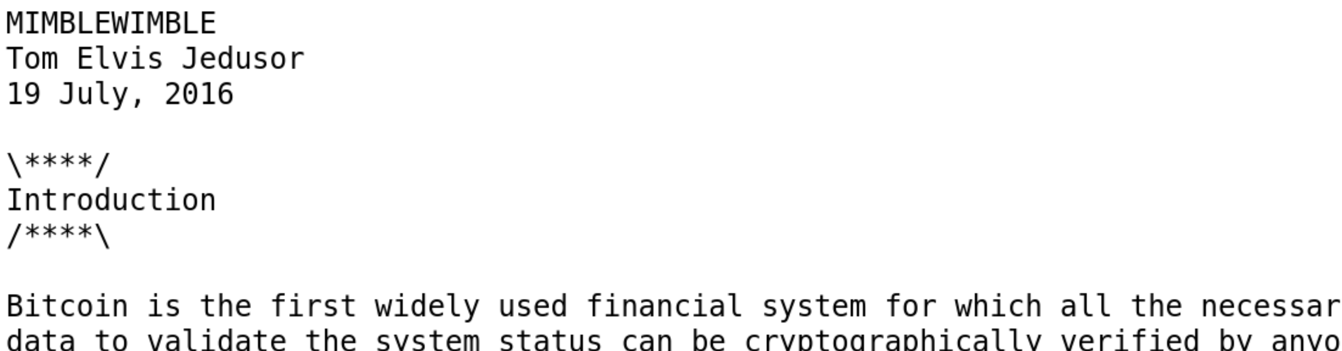
σ' is needed
to verify validity



- only keep unspent outputs
- while maintaining verifiability?

Mimblewimble

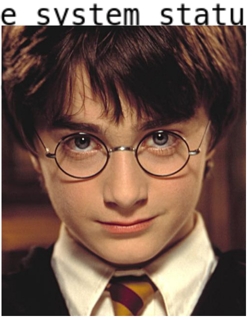
- **Cryptocurrency scheme**



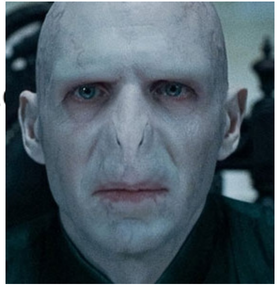
```
MIMBLEWIMBLE
Tom Elvis Jedusor
19 July, 2016

\****/
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
```



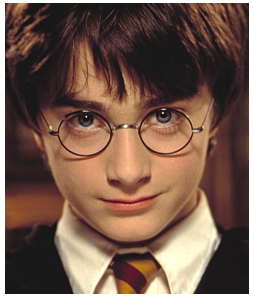
- proposed by “Tom Elvis Jedusor” in 2016

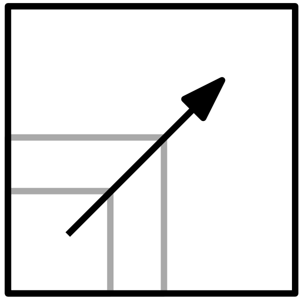


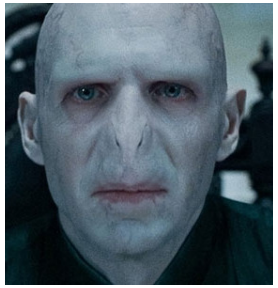
- uses ideas from Gregory Maxwell
- further developed by Andrew Poelstra

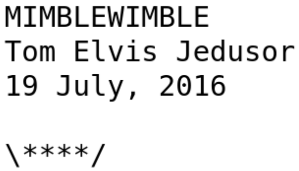
Mimblewimble

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



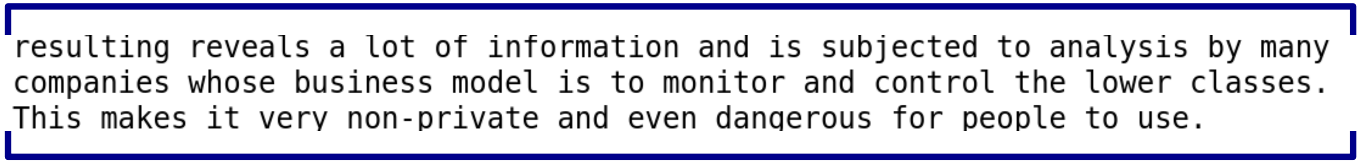






```
MIMBLEWIMBLE
Tom Elvis Jedusor
19 July, 2016

\****/
```



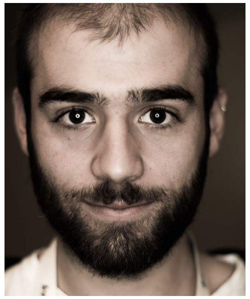
```
resulting reveals a lot of information and is subjected to analysis by many
companies whose business model is to monitor and control the lower classes.
This makes it very non-private and even dangerous for people to use.
```

Mimblewimble

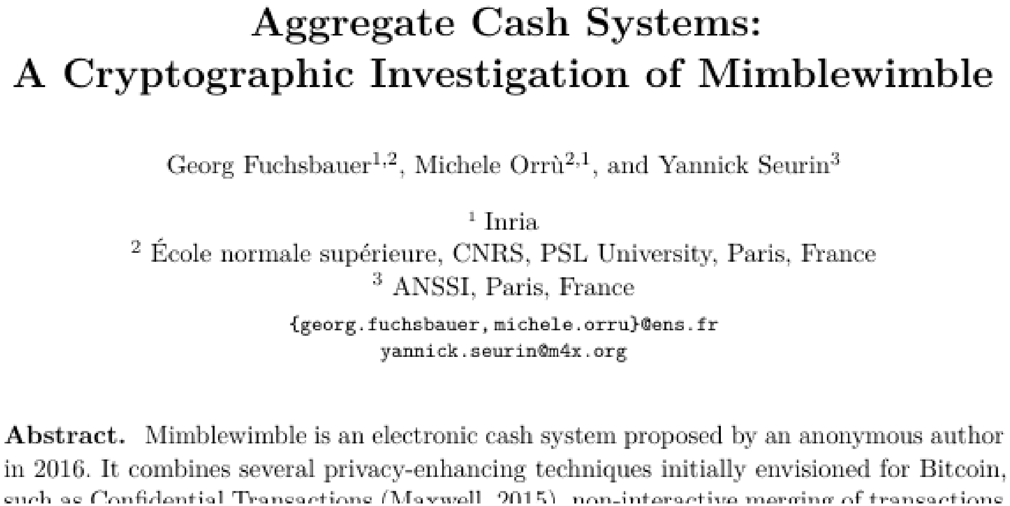
- **Cryptocurrency scheme**
  - **Privacy** (all amounts hidden; input/output relation blurred)
  - **Scalability** (forget about spent tx's)

formally analyzed in [FOS'19]

with Michele Orrù



and Yannick Seurin



Aggregate Cash Systems:
A Cryptographic Investigation of Mimblewimble

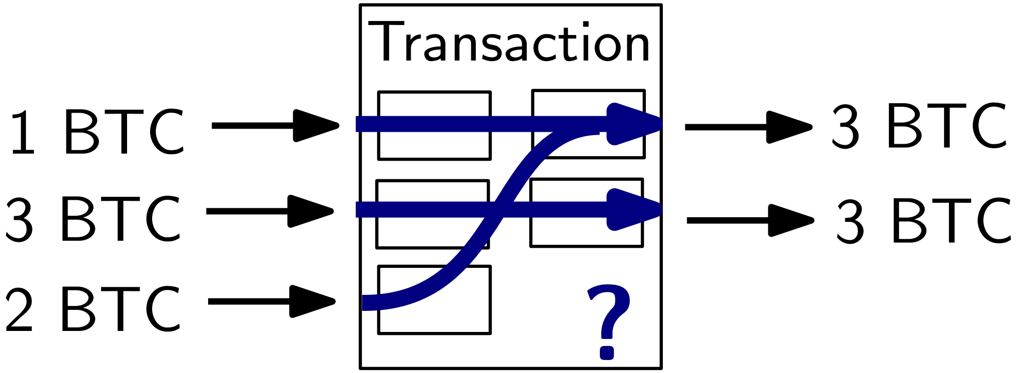
Georg Fuchsbauer[1,2], Michele Orrù[2,1], and Yannick Seurin[3]

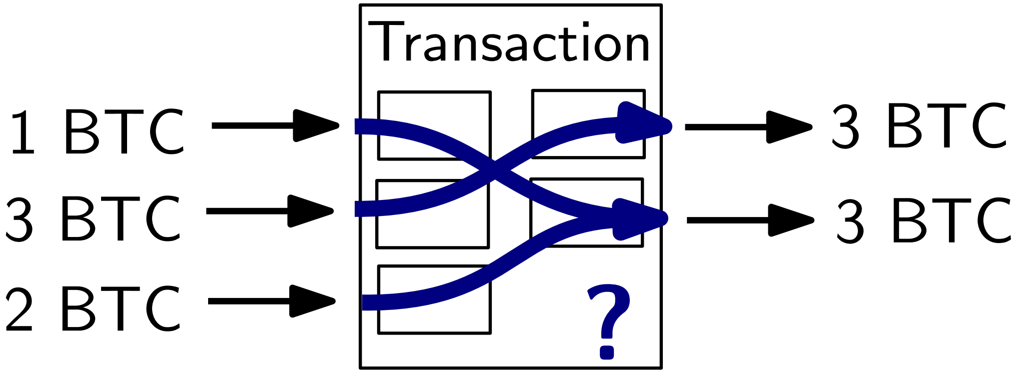
[1] Inria
[2] École normale supérieure, CNRS, PSL University, Paris, France
[3] ANSSI, Paris, France
{georg.fuchsbauer,michele.orru}@ens.fr
yannick.seurin@m4x.org

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

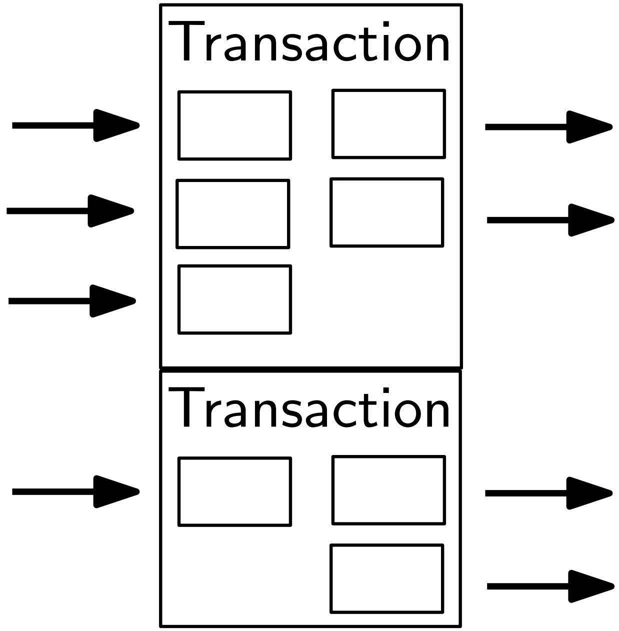
- Hiding path. . .





Anonymity

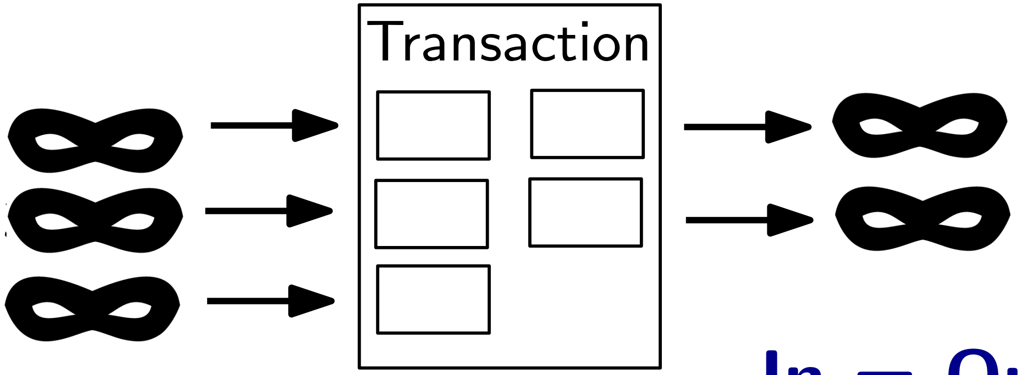
- Hiding path. . .



- **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. . .



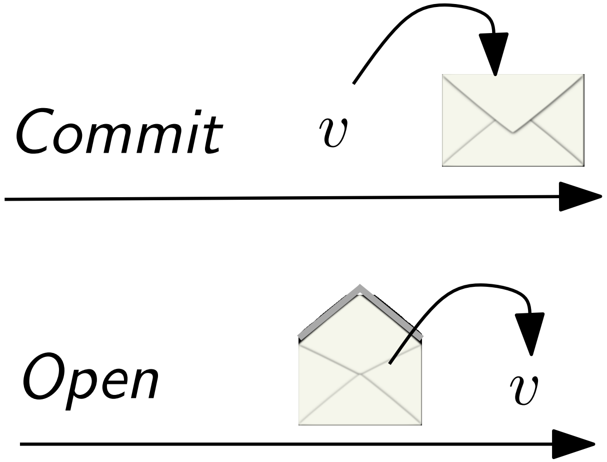
In = Out?

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

Pedersen commitment

Commitment

- “digital envelope”

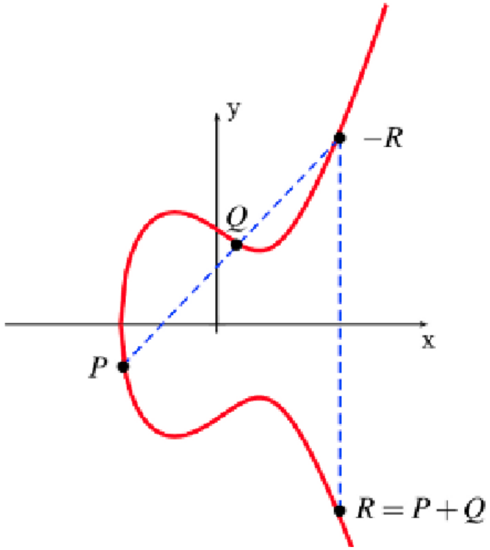


- **hiding:** commitment hides v
- **binding:** Alice can open commitment only to one value

Pedersen commitment

Commitment

- "digital envelope"



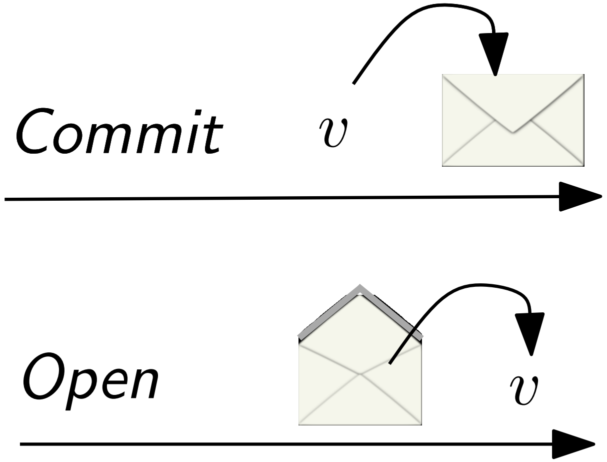
Discrete-log-hard group $(\mathbb{G}, +)$

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

Pedersen commitment

Commitment

- “digital envelope”



Commit v

Open v

Pedersen

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

pick random r

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

reveal v and r

- Commitments are **homomorphic**:

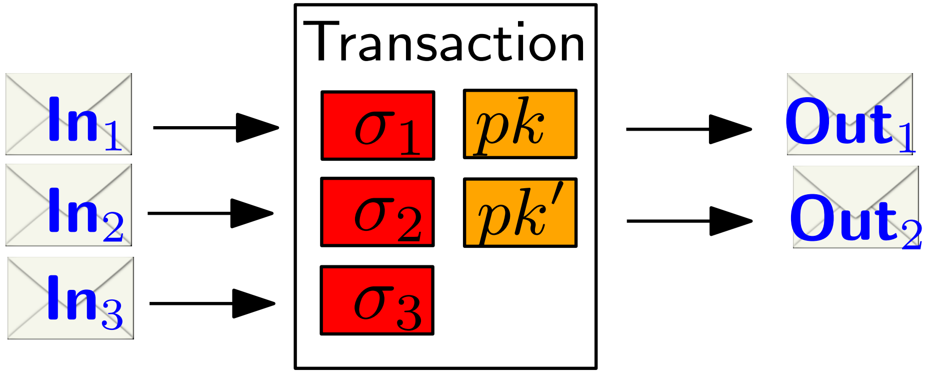
$$
\begin{aligned}
\textbf{Com}(v_1; r_1) + \textbf{Com}(v_2; r_2) &= (v_1 H + r_1 G) \; + \; (v_2 H + r_2 G) \\
&= (v_1 + v_2)H + (r_1 + r_2)G \\
&= \textbf{Com}(v_1 + v_2; r_1 + r_2)
\end{aligned}
$$

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



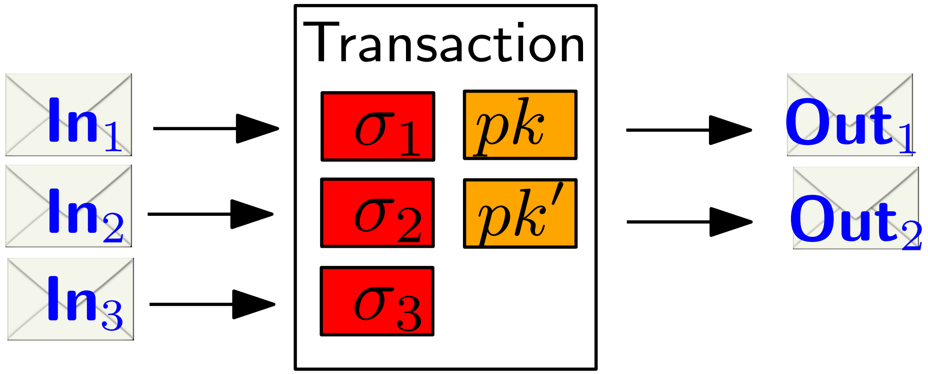
$$C = vH + rG$$

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

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

Confidential Transactions

Confidential transaction



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

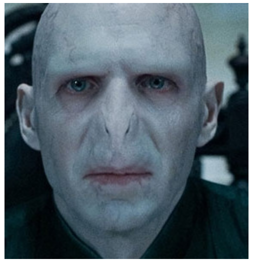
$$\sum \mathbf{Out} - \sum \mathbf{In} = \mathbf{0}$$

Signatures ⇒

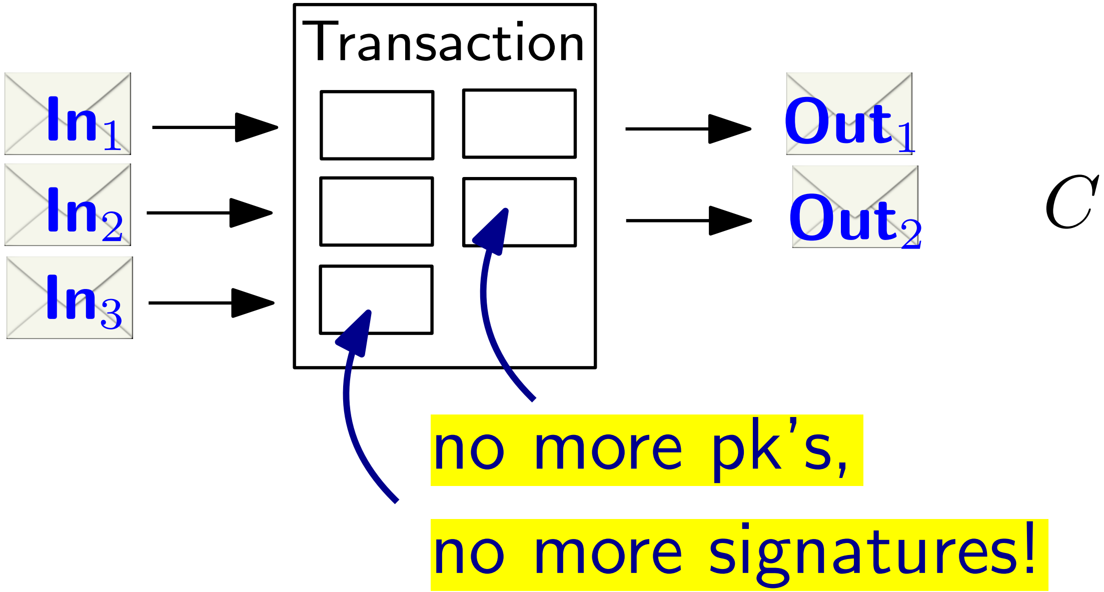
- no non-interactive CoinJoin
- no Cut-Through

Mimblewimble

[Jedusor '16]



secret key!



no more pk's,

no more signatures!

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

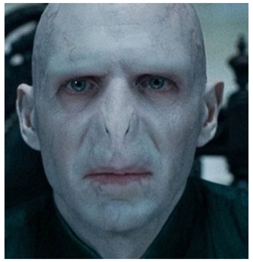
$$\boxed{\sum \textbf{Out} - \sum \textbf{In} = 0}$$

But: sender knows sum of output r's

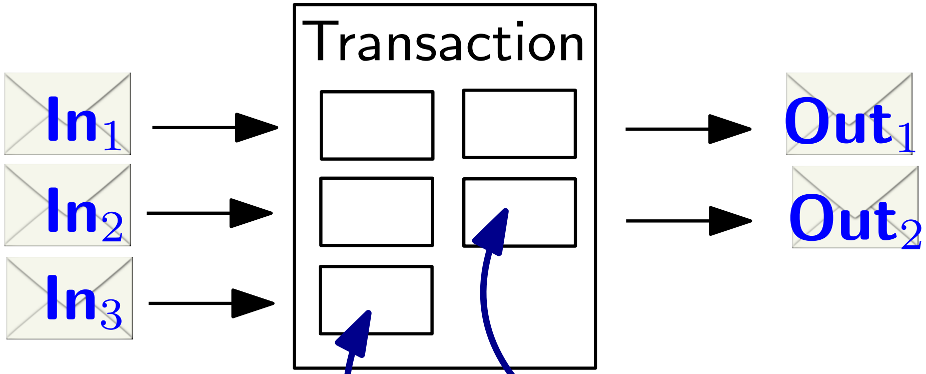
⇒ users choose independent keys

Mimblewimble

[Jedusor '16]



secret key!



no more pk's,

no more signatures!

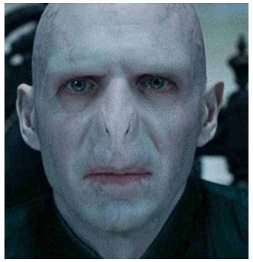
$$C = vH + rG, \quad \pi$$

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

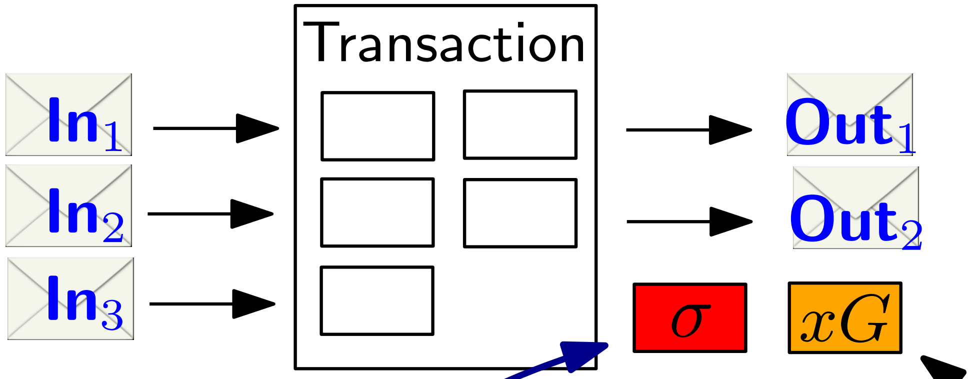
$$\begin{aligned}
&\sum C_i^{\text{out}} - \sum C_i^{\text{in}} \\
&= \sum (v_i^{\text{out}} H + r_i^{\text{out}} G) - \sum (v_i^{\text{in}} H + r_i^{\text{in}} G) \\
&= \underbrace{(\sum v_i^{\text{out}} - \sum v_i^{\text{in}})}_{\overset{!}{=}0} H + \underbrace{(\sum r_i^{\text{out}} - \sum r_i^{\text{in}})}_{=:x}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

Mimblewimble

[Jedusor '16]



secret key!



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

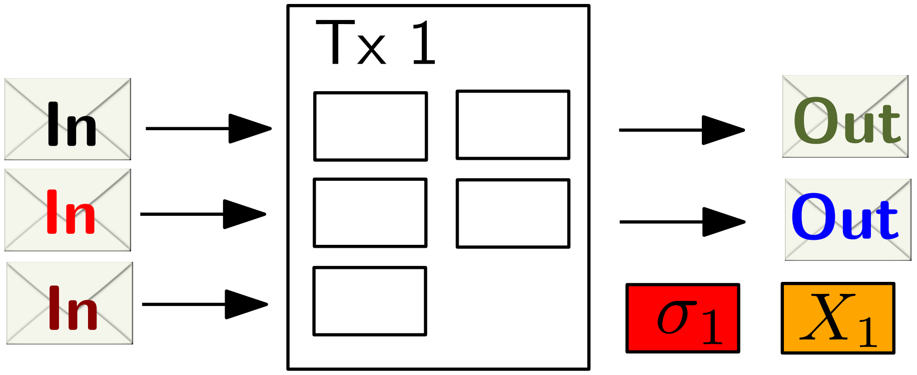
$$\boxed{\sum \textbf{Out} - \sum \textbf{In} = 0H + xG}$$

one signature

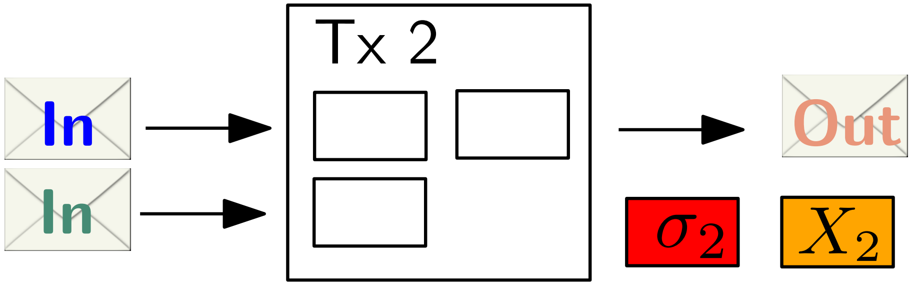
“proves” that $\overbrace{\sum \text{Out} - \sum \text{In}}$ is commitment to 0

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

Mimblewimble

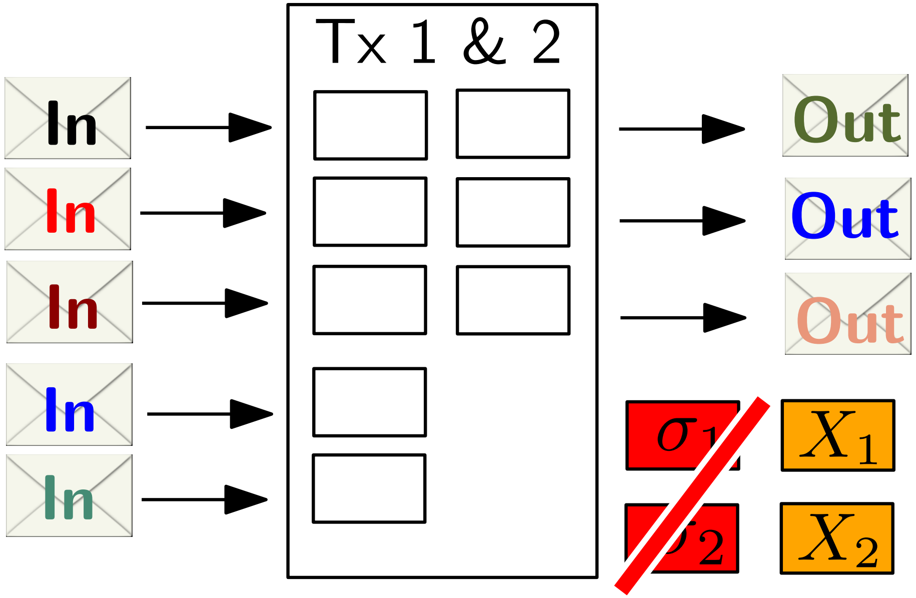


- $\sum \textbf{Out}_1 - \sum \textbf{In}_1 = X_1$
- σ_1 valid for X_1



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

Mimblewimble

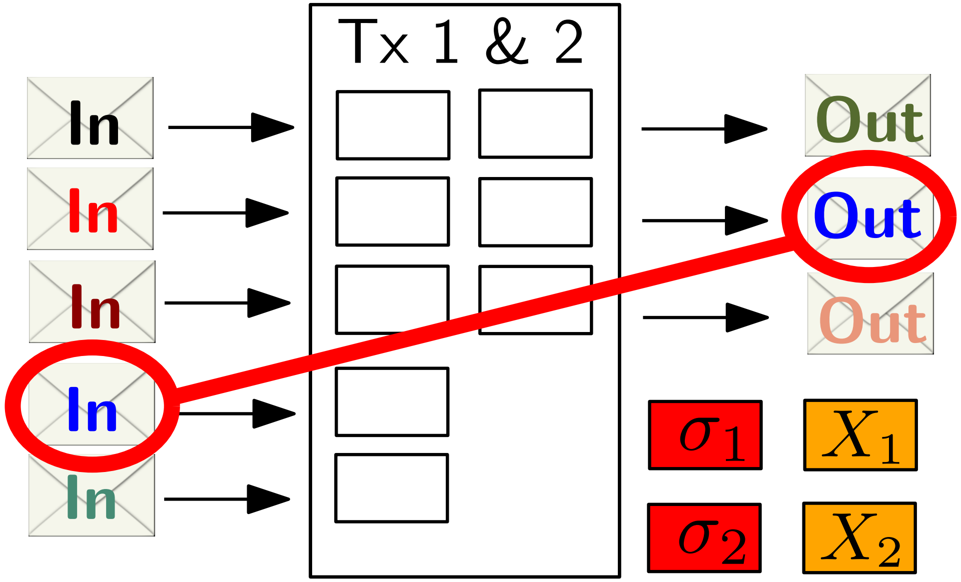


- $\sum \textbf{Out} - \sum \textbf{In} = X_1 + X_2$
- σ_1 valid for X_1
- σ_2 valid for X_2

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

Mimblewimble

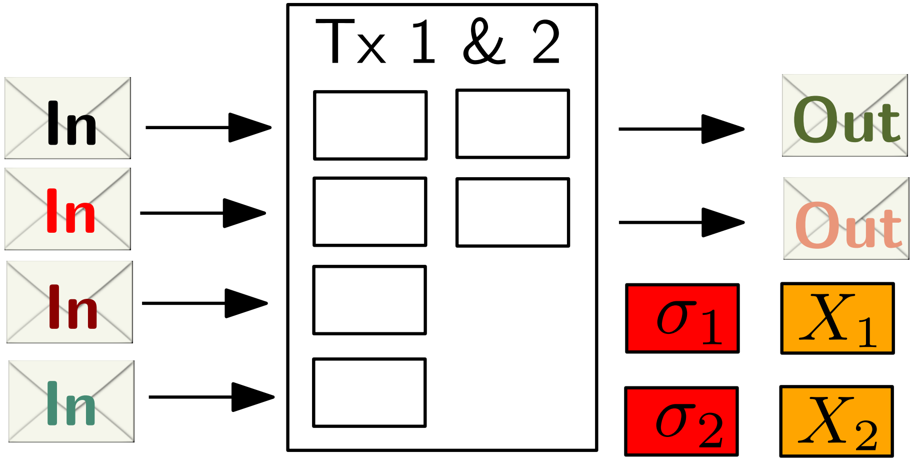
Post-confirmation Cut-Through



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

Mimblewimble

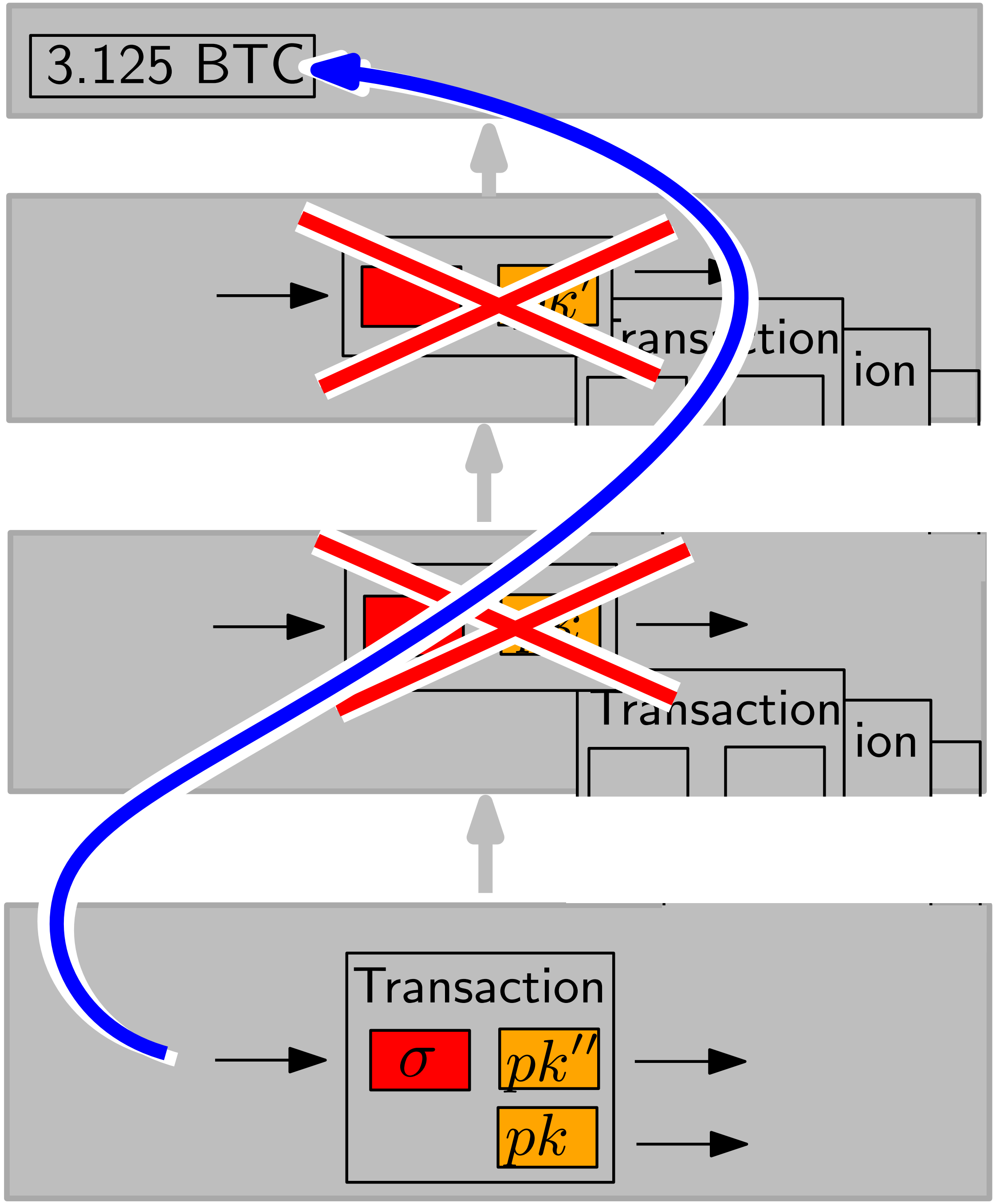
Post-confirmation Cut-Through



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

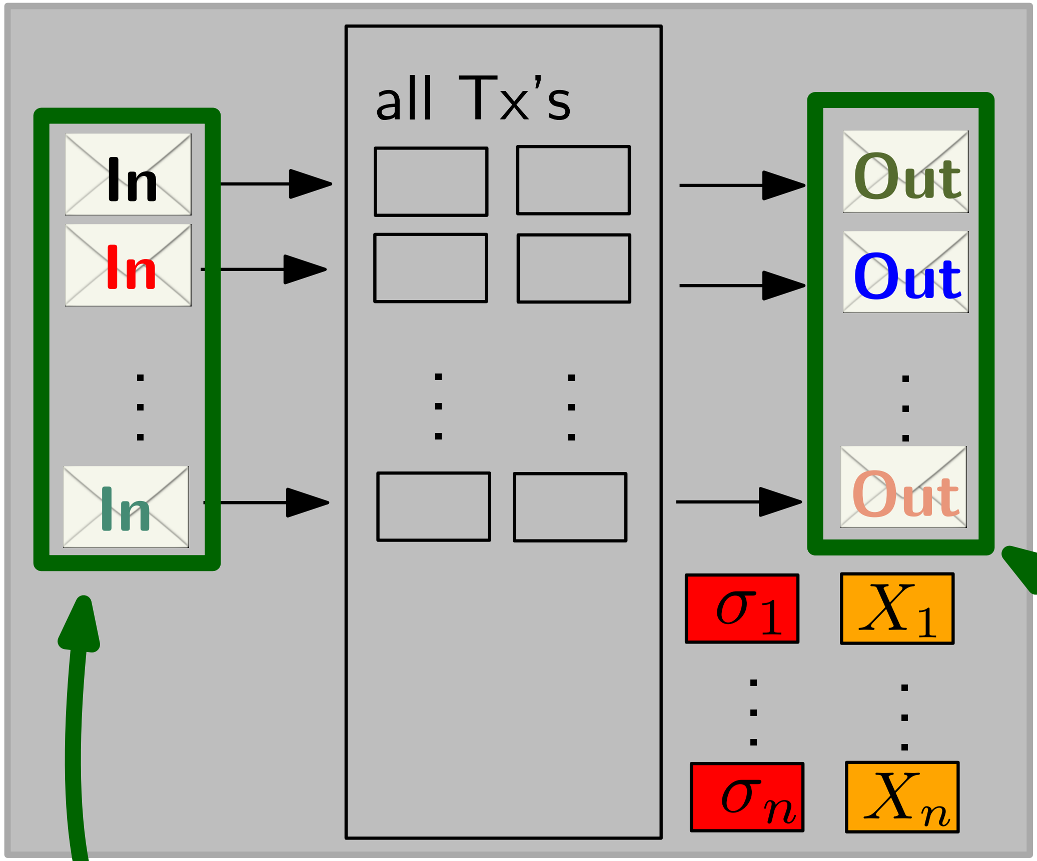
Scalability

“cut-through”



Mimblewimble

Cut through all transactions in blockchain



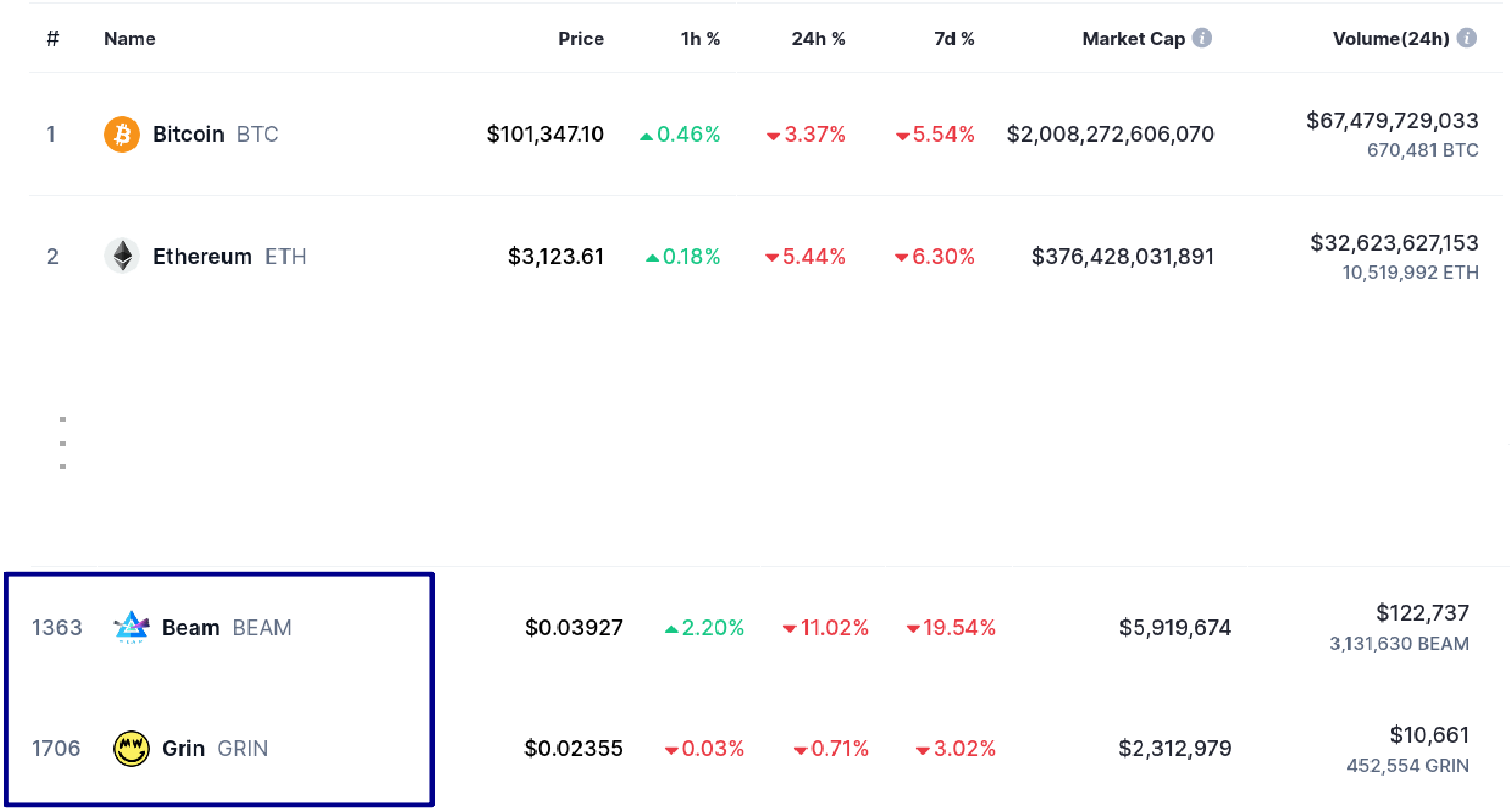
- $\sum$ **Out** $-\sum$ **In** $= \sum X_i$
- $\forall i : \sigma_i$ valid for X_i

UTXO set

Only coinbase transactions

Applications

Implemented by several cryptocurrencies (since 2019):



#	Name	Price	1h %	24h %	7d %	Market Cap	Volume(24h)
1	Bitcoin BTC	$101,347.10	▲0.46%	▼3.37%	▼5.54%	$2,008,272,606,070	$67,479,729,033 670,481 BTC
2	Ethereum ETH	$3,123.61	▲0.18%	▼5.44%	▼6.30%	$376,428,031,891	$32,623,627,153 10,519,992 ETH
⋮							
1363	Beam BEAM	$0.03927	▲2.20%	▼11.02%	▼19.54%	$5,919,674	$122,737 3,131,630 BEAM
1706	Grin GRIN	$0.02355	▼0.03%	▼0.71%	▼3.02%	$2,312,979	$10,661 452,554 GRIN

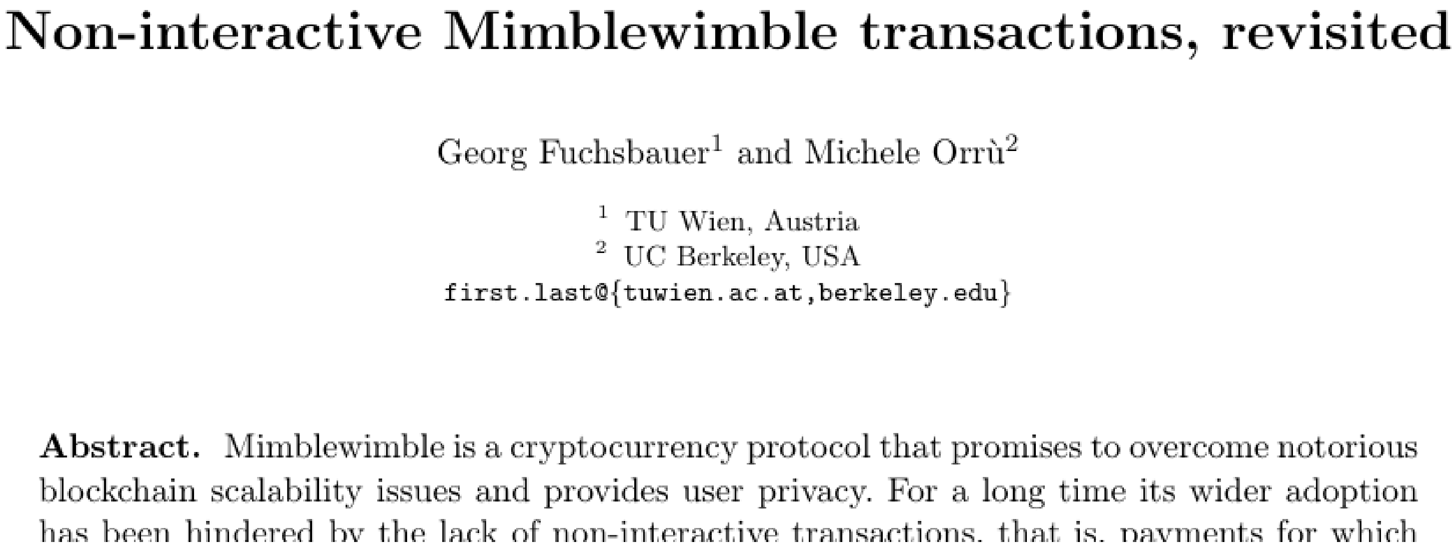
Applications

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

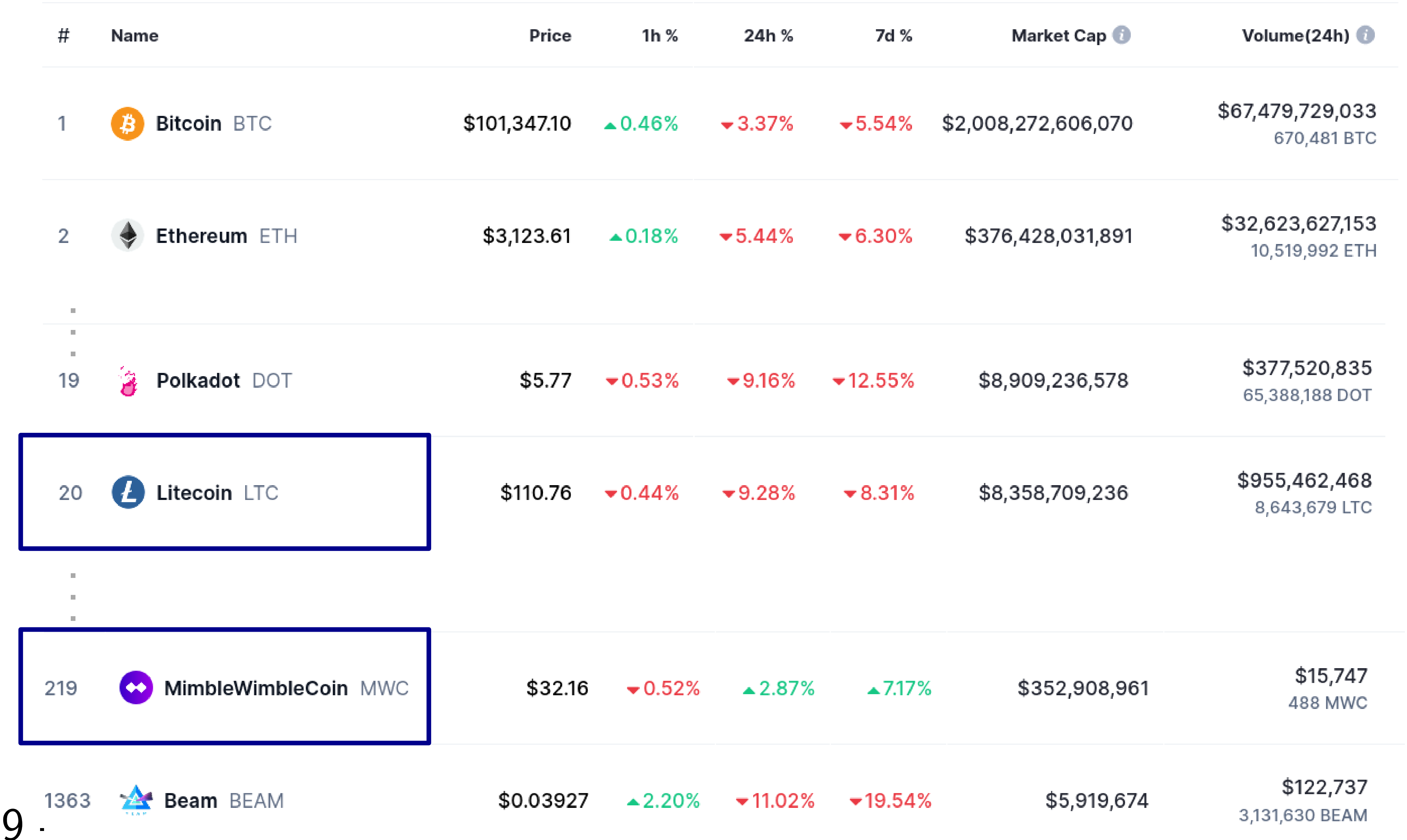
Georg Fuchsbauer[1] and Michele Orrù[2]

[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”

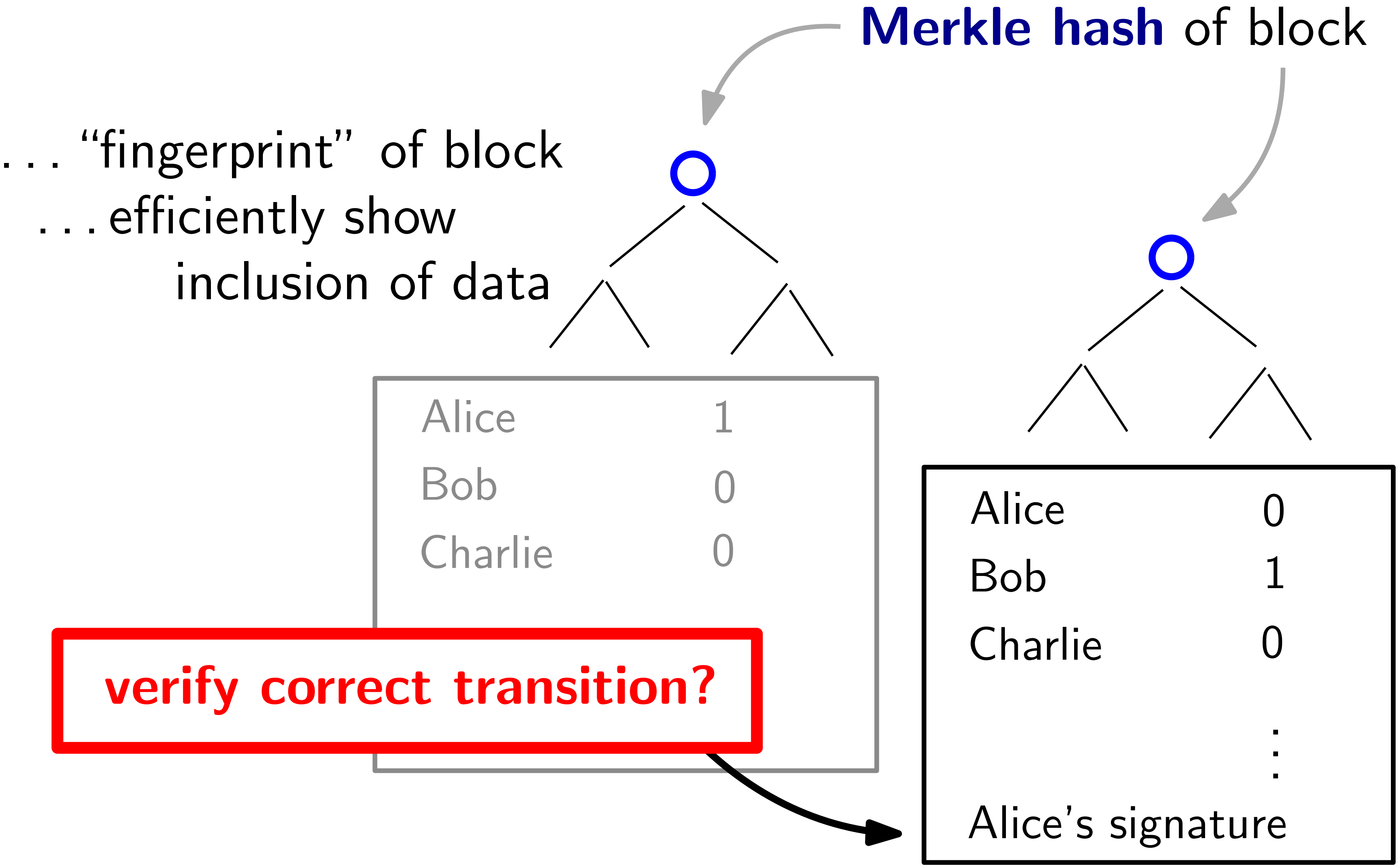


#	Name	Price	1h %	24h %	7d %	Market Cap ⓘ	Volume(24h) ⓘ
1	Bitcoin BTC	$101,347.10	▲0.46%	▼3.37%	▼5.54%	$2,008,272,606,070	$67,479,729,033 670,481 BTC
2	Ethereum ETH	$3,123.61	▲0.18%	▼5.44%	▼6.30%	$376,428,031,891	$32,623,627,153 10,519,992 ETH
⋮							
19	Polkadot DOT	$5.77	▼0.53%	▼9.16%	▼12.55%	$8,909,236,578	$377,520,835 65,388,188 DOT
20	Litecoin LTC	$110.76	▼0.44%	▼9.28%	▼8.31%	$8,358,709,236	$955,462,468 8,643,679 LTC
⋮							
219	MimbleWimbleCoin MWC	$32.16	▼0.52%	▲2.87%	▲7.17%	$352,908,961	$15,747 488 MWC
1363	Beam BEAM	$0.03927	▲2.20%	▼11.02%	▼19.54%	$5,919,674	$122,737 3,131,630 BEAM

Coda / Mina

. . . constant-size blockchain

Coda / Mina



Merkle hash of block

. . . "fingerprint" of block

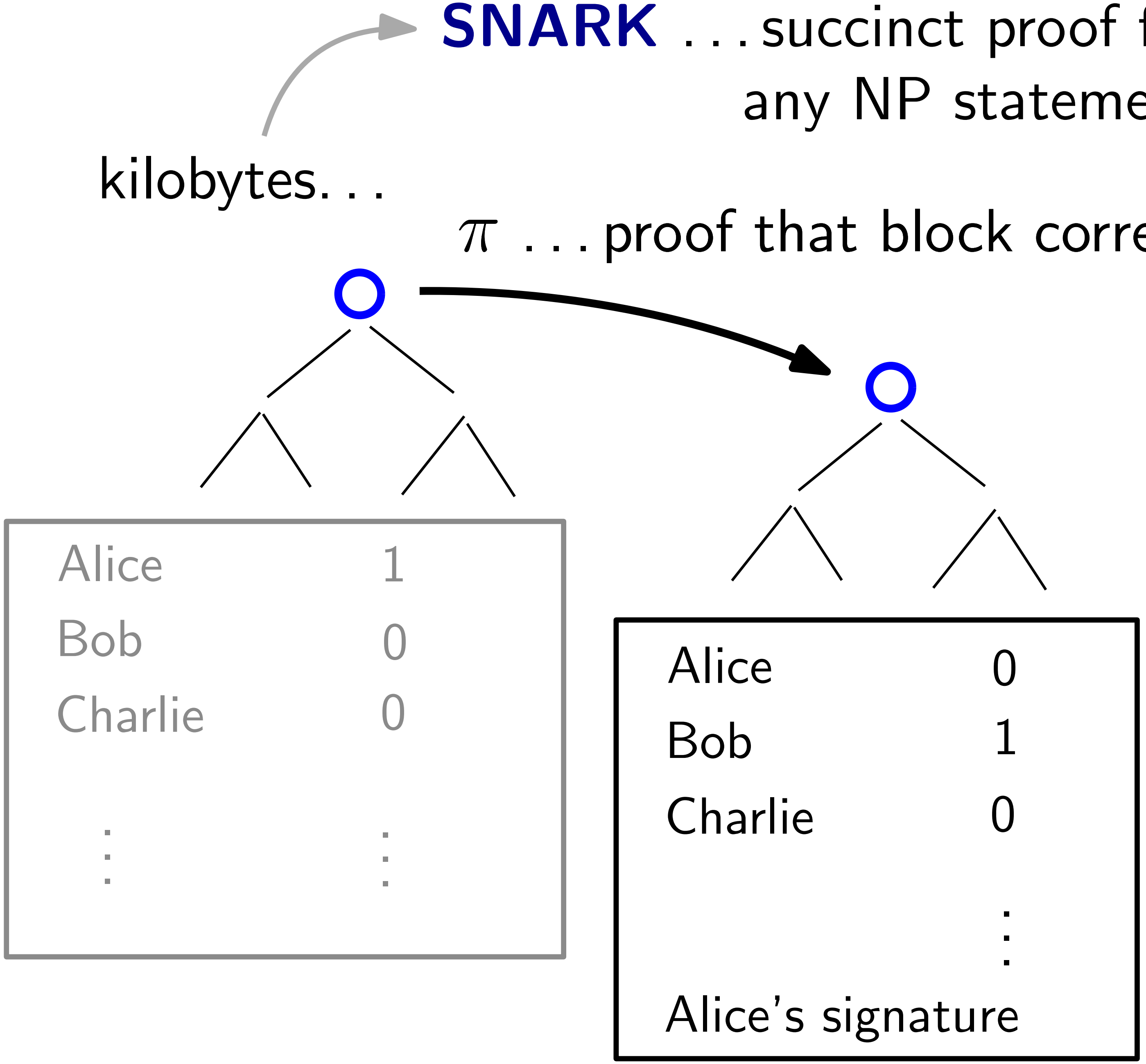
. . . efficiently show inclusion of data

Alice	1
Bob	0
Charlie	0

Alice	0
Bob	1
Charlie	0
	⋮
Alice's signature	

verify correct transition?

Coda / Mina



SNARK . . . succinct proof for any NP statement

kilobytes. . .

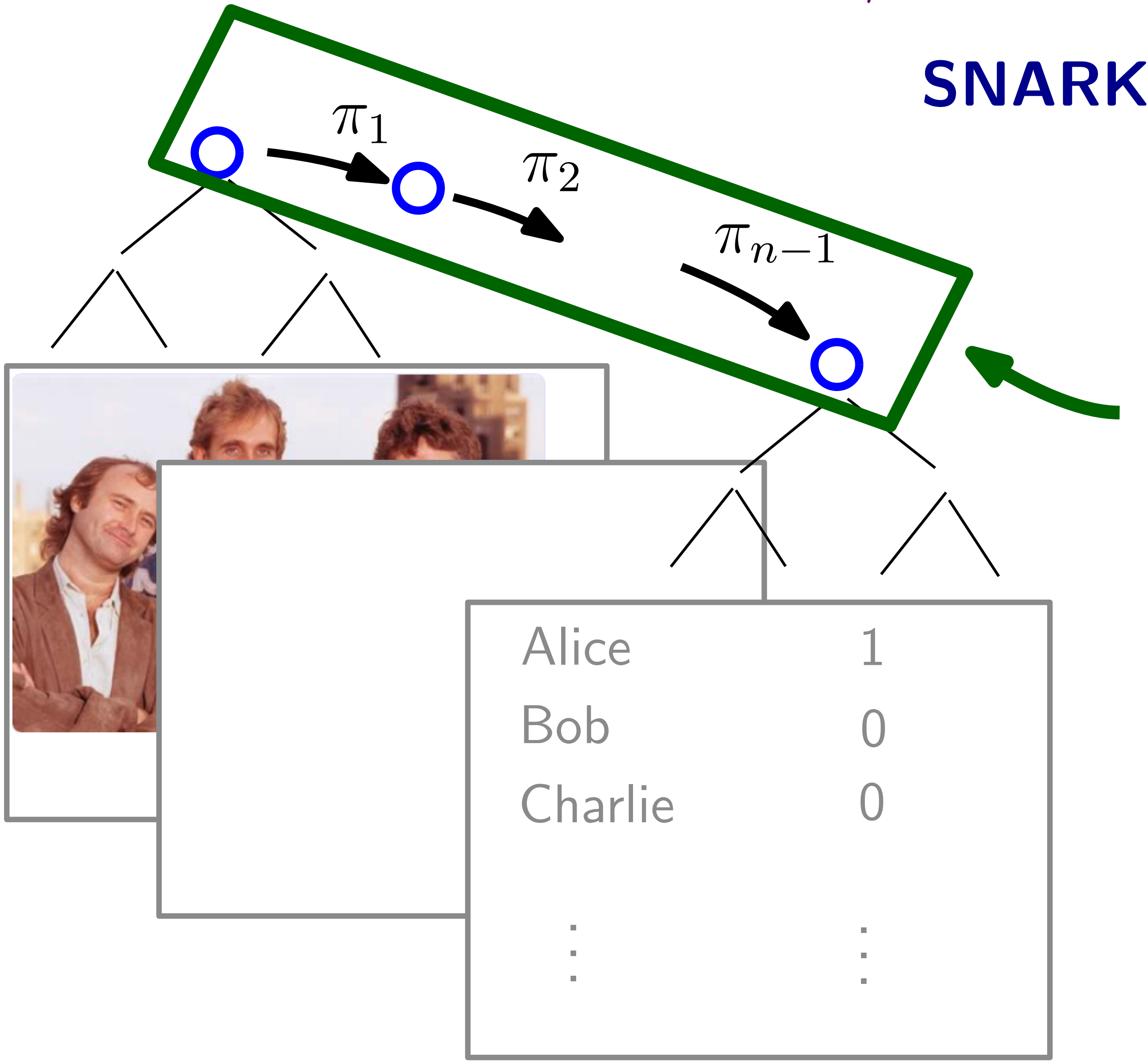
π . . . proof that block correct

Alice	1
Bob	0
Charlie	0
⋮	⋮

Alice	0
Bob	1
Charlie	0
	⋮
Alice's signature	

Coda / Mina

SNARK . . . succinct proof for any NP statement

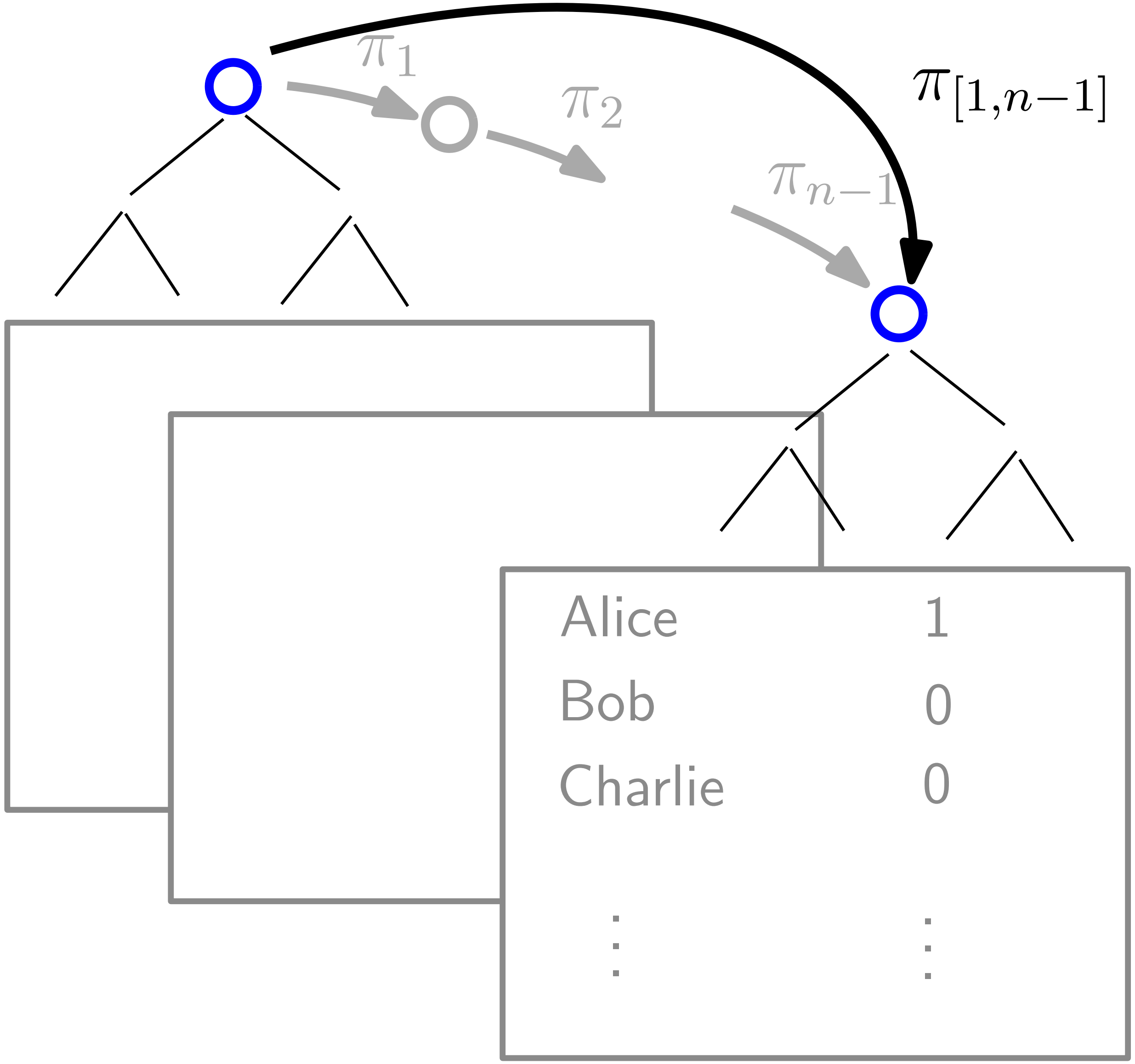


Alice	1
Bob	0
Charlie	0
⋮	⋮

Blockchain?

constant size?

Coda / Mina

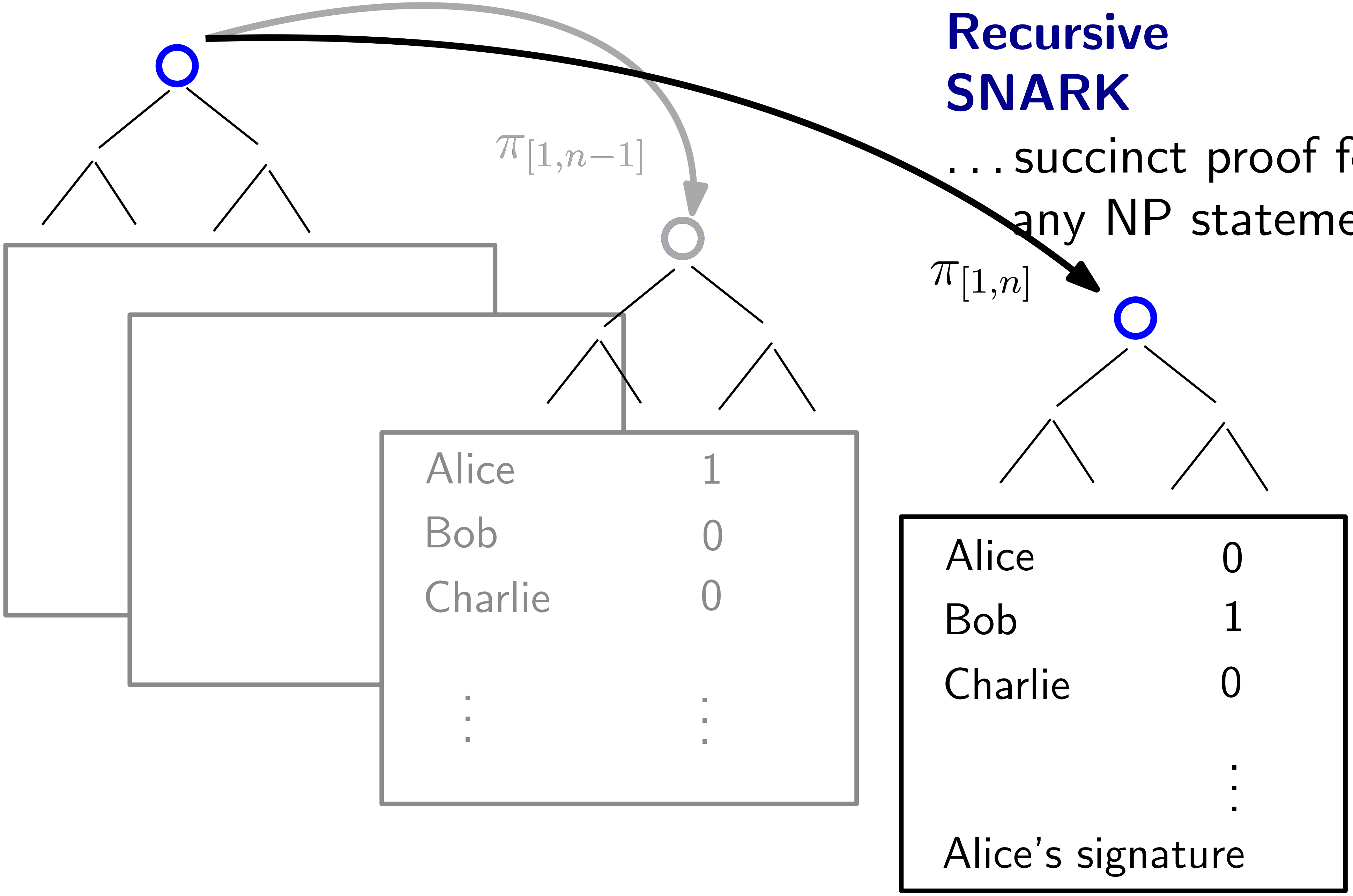


Alice	1
Bob	0
Charlie	0
⋮	⋮

SNARK

. . . succinct proof for any NP statement

Coda / Mina



Alice	1
Bob	0
Charlie	0
⋮	⋮

Alice	0
Bob	1
Charlie	0
	⋮
Alice's signature	

Recursive SNARK

…succinct proof for any NP statement