Zerocash:

addressing Bitcoin’s privacy problem

Madars Virza

based on joint works with

Eli Ben-Sasson

Alessandro Chiesa

Christina Garman

Matthew Green

Ian Miers

Eran Tromer

Outline

Outline

1. Bitcoin’s privacy problem

Outline

1. Bitcoin's privacy problem

2. Zerocash: privacy-preserving decentralized currency

Outline

1. Bitcoin’s privacy problem

2. Zerocash: privacy-preserving decentralized currency

3. Zcash: deploying Zerocash in practice…

“Would you like a new credit card?
You will pay **almost no fees**!”

“Would you like a new credit card?
You will pay **almost no fees**!”

“Sweet! What about the fine print?”

“Would you like a new credit card?
You will pay **almost no fees**!”

“Sweet! What about the fine print?”

“Every payment you make,

“Would you like a new credit card?
You will pay **almost no fees**!”

“Sweet! What about the fine print?”

“Every payment you make,
we will publicly broadcast it.”

Sender	Recipient	Amount	Time	Location

“Would you like a new credit card?
You will pay **almost no fees**!”

“Sweet! What about the fine print?”

“Every payment you make,
we will publicly broadcast it.”

Sender	Recipient	Amount	Time	Location
Madars	Starbucks	$10	March 24, 10:05 am	Cambridge, MA

“Would you like a new credit card?
You will pay **almost no fees**!”

“Sweet! What about the fine print?”

“Every payment you make,
we will publicly broadcast it.”

Sender	Recipient	Amount	Time	Location
Madars	Starbucks	$10	March 24, 10:05 am	Cambridge, MA
Madars	Whole Foods	$50	March 25, 1:23 pm	Boston, MA

“Would you like a new credit card?
You will pay **almost no fees**!”

“Sweet! What about the fine print?”

“Every payment you make,
we will publicly broadcast it.”

Sender	Recipient	Amount	Time	Location
Madars	Starbucks	$10	March 24, 10:05 am	Cambridge, MA
Madars	Whole Foods	$50	March 25, 1:23 pm	Boston, MA
…	…	…	…	…

“Would you like a new credit card?
You will pay **almost no fees**!”

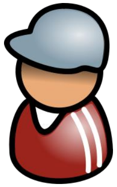
“Sweet! What about the fine print?”

“Every payment you make,
we will publicly broadcast it.
No big deal, right?”

Sender	Recipient	Amount	Time	Location
Madars	Starbucks	$10	March 24, 10:05 am	Cambridge, MA
Madars	Whole Foods	$50	March 25, 1:23 pm	Boston, MA
…	…	…	…	…

“No big deal.”

“No big deal.” “Very invasive deal!”



“No big deal.” “Very invasive deal!”

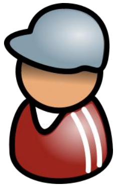
Personal medical information
(e.g. therapist choices, prescriptions)

“No big deal.” “Very invasive deal!”

Personal medical information
(e.g. therapist choices, prescriptions)

→ **denied coverage** in future?

“No big deal.” “Very invasive deal!”



Personal medical information
(e.g. therapist choices, prescriptions)

→ **denied coverage** in future?

Merchant cash flow

“No big deal.” “Very invasive deal!”

Personal medical information
(e.g. therapist choices, prescriptions)

→ **denied coverage** in future?

Merchant cash flow

→ exposed to **competitors**

“No big deal.” “Very invasive deal!”

Personal medical information
(e.g. therapist choices, prescriptions)

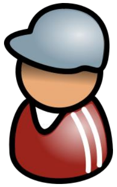
→ **denied coverage** in future?

Merchant cash flow

→ exposed to **competitors**

Current location and travel patterns

“No big deal.” “Very invasive deal!”



Personal medical information
(e.g. therapist choices, prescriptions)

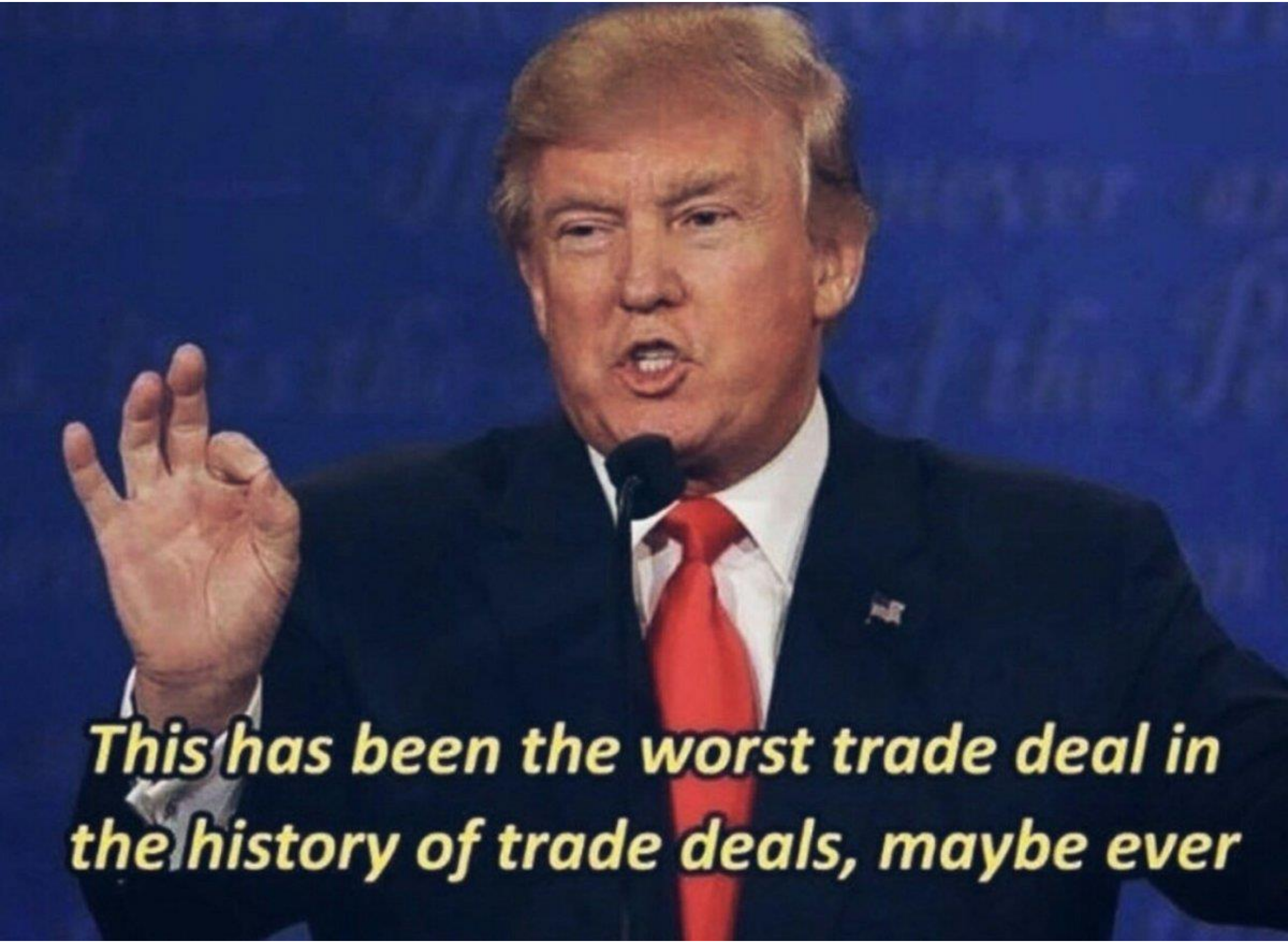
→ **denied coverage** in future?

Merchant cash flow

→ exposed to **competitors**

Current location and travel patterns

→ a gold mine for **stalkers**


This has been the worst trade deal in the history of trade deals, maybe ever

Your bank won’t give you this absurd deal

Your bank won’t give you this absurd deal

Federal privacy laws mandate opt-out from data sharing

Your bank won't give you this absurd deal

Federal privacy laws mandate opt-out from data sharing

Gramm-Leach-Bliley Act: up to $100k fine per violation

Your bank won’t give you this absurd deal

Federal privacy laws mandate opt-out from data sharing

Gramm-Leach-Bliley Act: up to $100k fine per violation

But what about Bitcoin?

Your bank won’t give you this absurd deal

Federal privacy laws mandate opt-out from data sharing

Gramm-Leach-Bliley Act: up to $100k fine per violation

But what about Bitcoin?

There is no opt-out on the blockchain:

Your bank won't give you this absurd deal

Federal privacy laws mandate opt-out from data sharing

Gramm-Leach-Bliley Act: up to $100k fine per violation

But what about Bitcoin?

There is no opt-out on the blockchain:

Sender
14…..
f7….
…

Your bank won’t give you this absurd deal

Federal privacy laws mandate opt-out from data sharing

Gramm-Leach-Bliley Act: up to $100k fine per violation

But what about Bitcoin?

There is no opt-out on the blockchain:

Sender	Recipient
14…..	1b….
f7….	38….
…	…

Your bank won’t give you this absurd deal

Federal privacy laws mandate opt-out from data sharing

Gramm-Leach-Bliley Act: up to $100k fine per violation

But what about Bitcoin?

There is no opt-out on the blockchain:

Sender	Recipient	Amount
14…..	1b….	0.1 ₿
f7….	38….	2 ₿
…	…	…

Your bank won’t give you this absurd deal

Federal privacy laws mandate opt-out from data sharing

Gramm-Leach-Bliley Act: up to $100k fine per violation

But what about Bitcoin?

There is no opt-out on the blockchain:

Sender	Recipient	Amount	Time
14…..	1b….	0.1 ₿	March 24, 10:05 am
f7….	38….	2 ₿	March 25, 1:23 pm
…	…	…	…

Your bank won’t give you this absurd deal

Federal privacy laws mandate opt-out from data sharing

Gramm-Leach-Bliley Act: up to $100k fine per violation

But what about Bitcoin?

There is no opt-out on the blockchain:

Sender	Recipient	Amount	Time
14…..	1b….	0.1 ₿	March 24, 10:05 am
f7….	38….	2 ₿	March 25, 1:23 pm
…	…	…	…

“This is not the same! Those are just addresses!”

“Those are just addresses”

“Those are just addresses”

… that are known by people you interact with

“Those are just addresses”

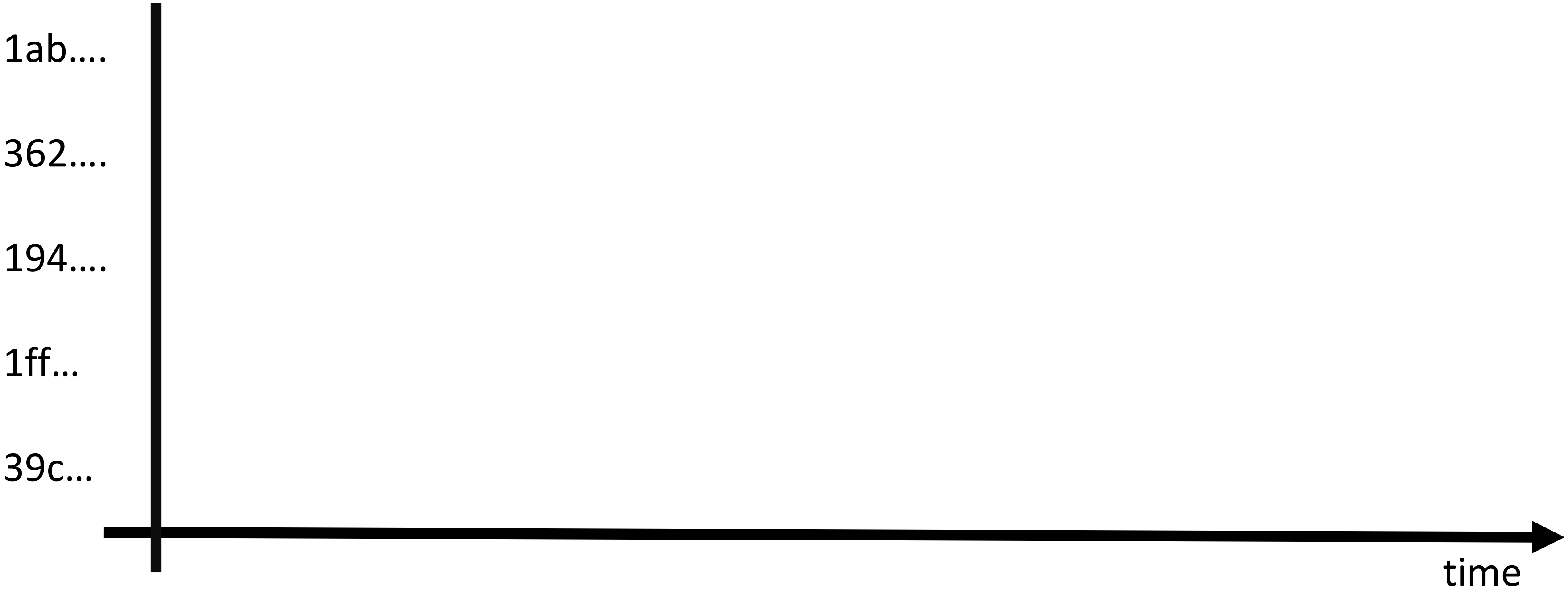
… that are known by people you interact with

... and **anyone else** can analyze the ledger

"Those are just addresses"

… that are known by people you interact with

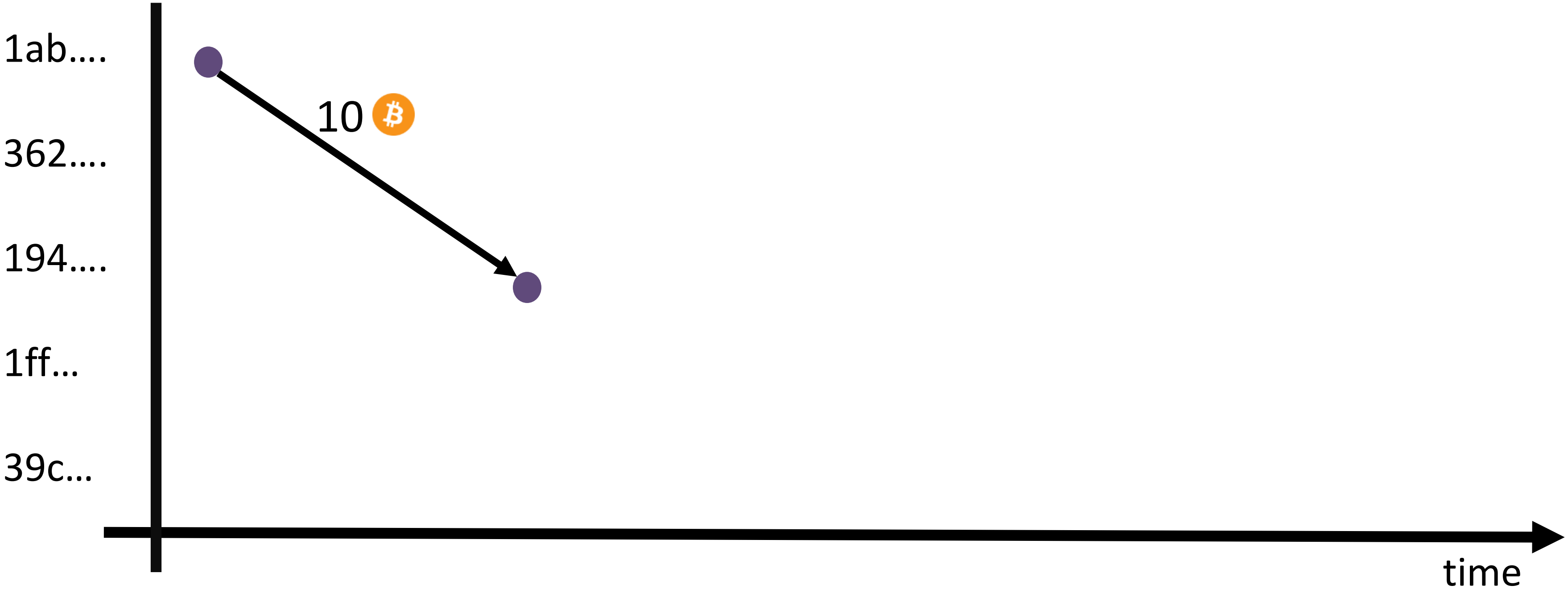
... and **anyone else** can analyze the ledger



“Those are just addresses”

… that are known by people you interact with

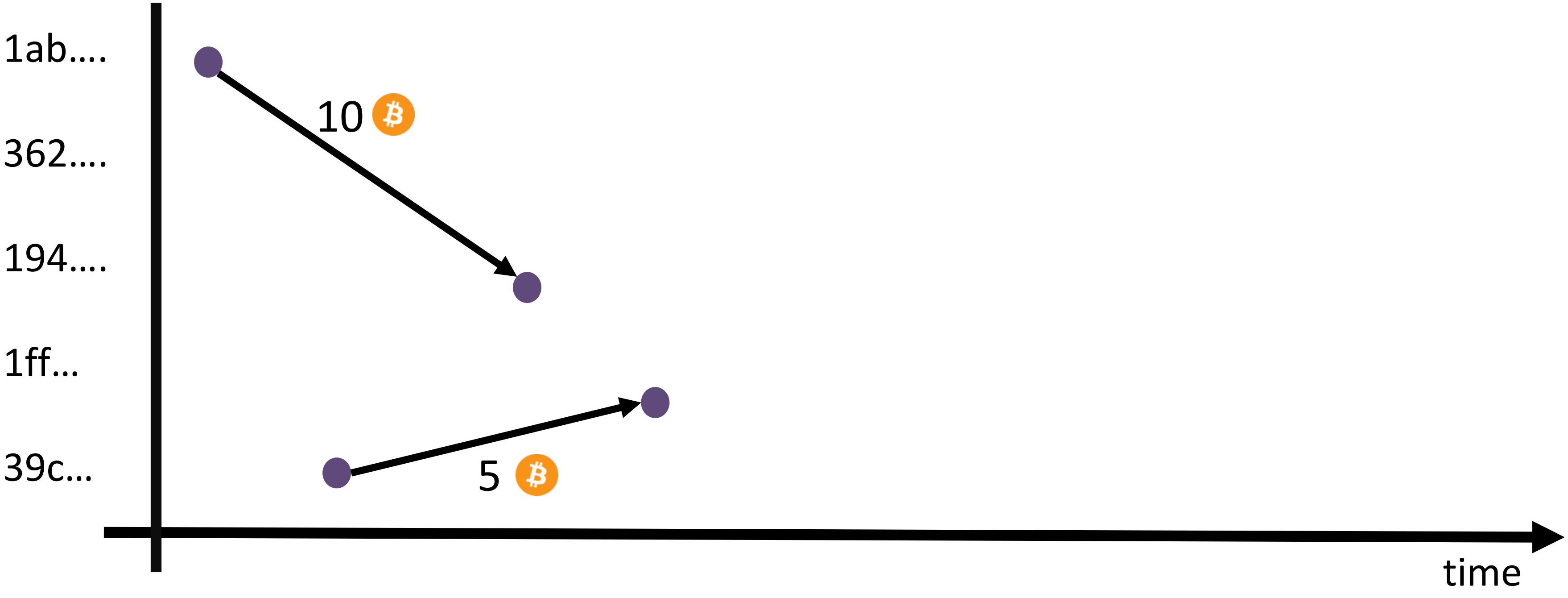
... and **anyone else** can analyze the ledger



“Those are just addresses”

... that are known by people you interact with

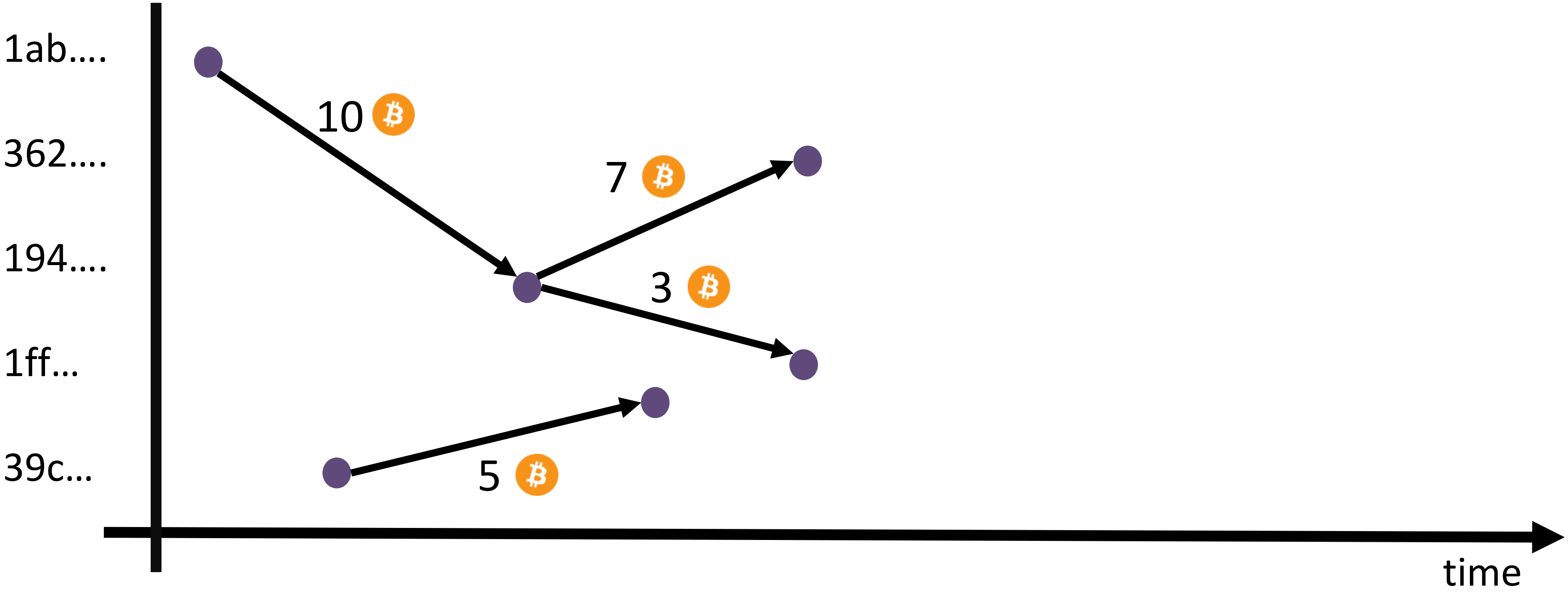
... and **anyone else** can analyze the ledger



“Those are just addresses”

... that are known by people you interact with

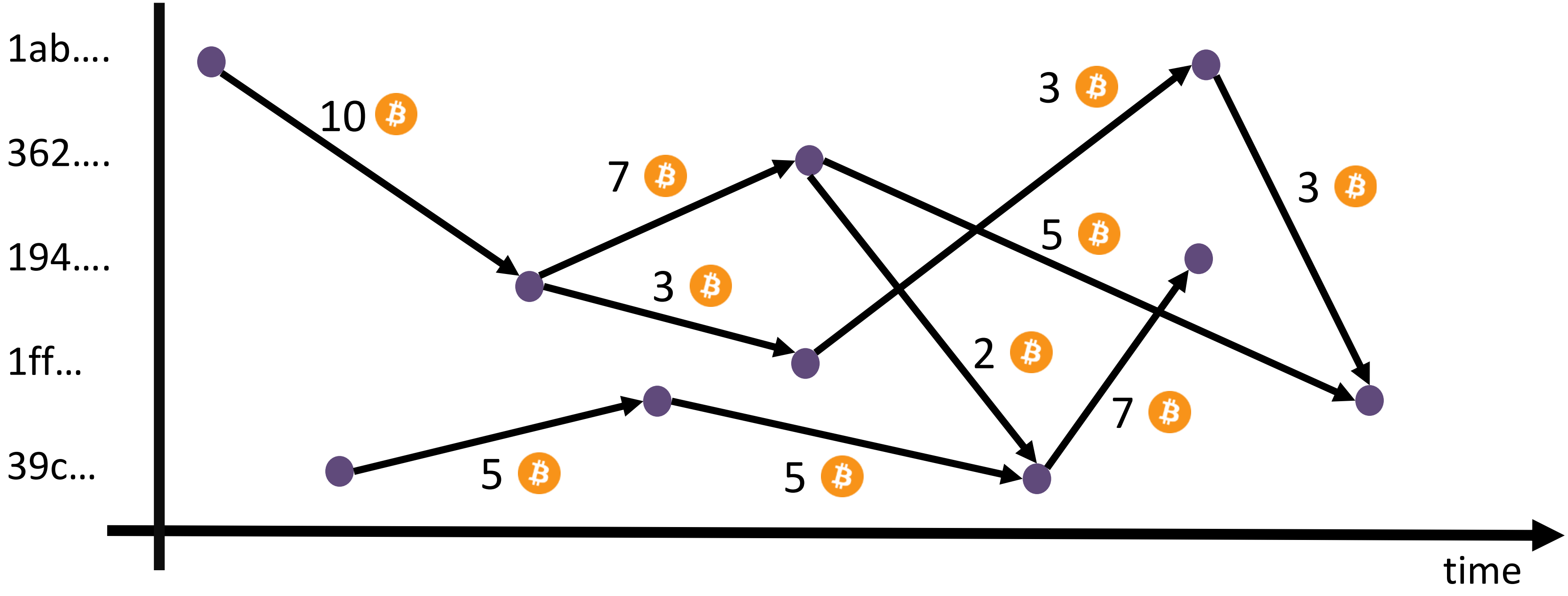
... and **anyone else** can analyze the ledger



“Those are just addresses”

… that are known by people you interact with

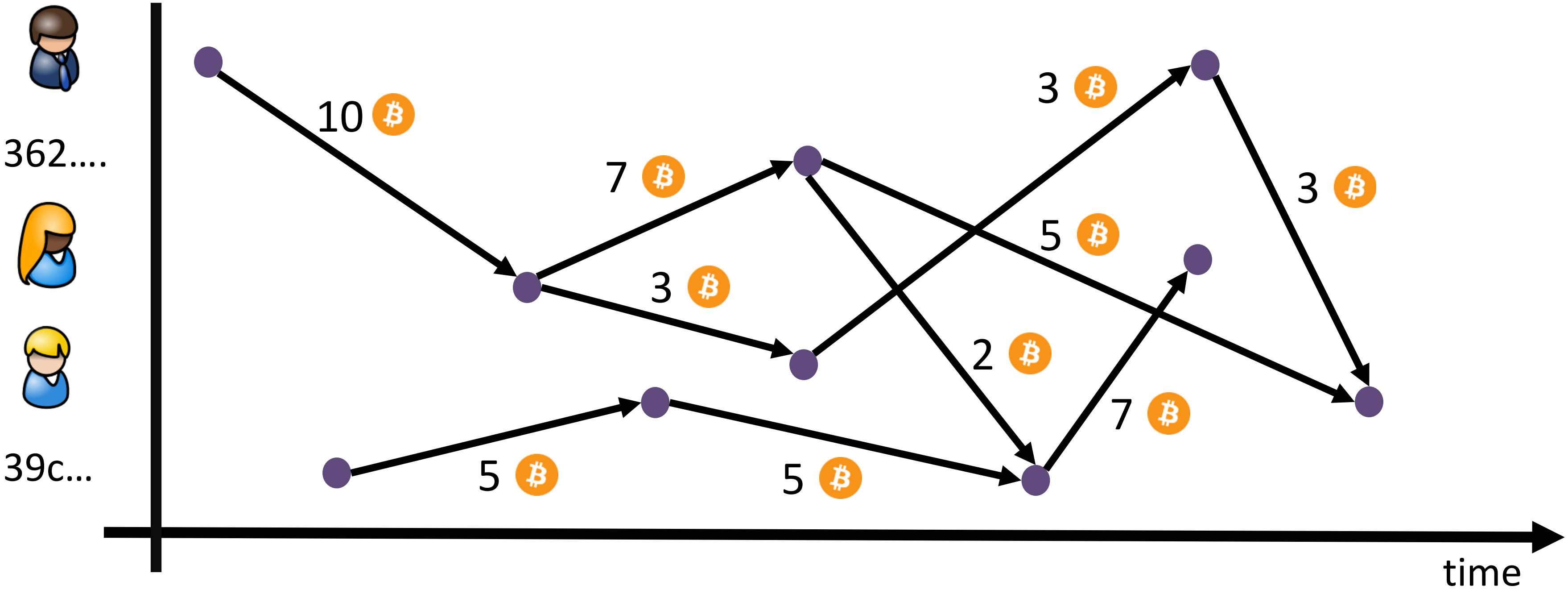
... and **anyone else** can analyze the ledger



“Those are just addresses”

... that are known by people you interact with

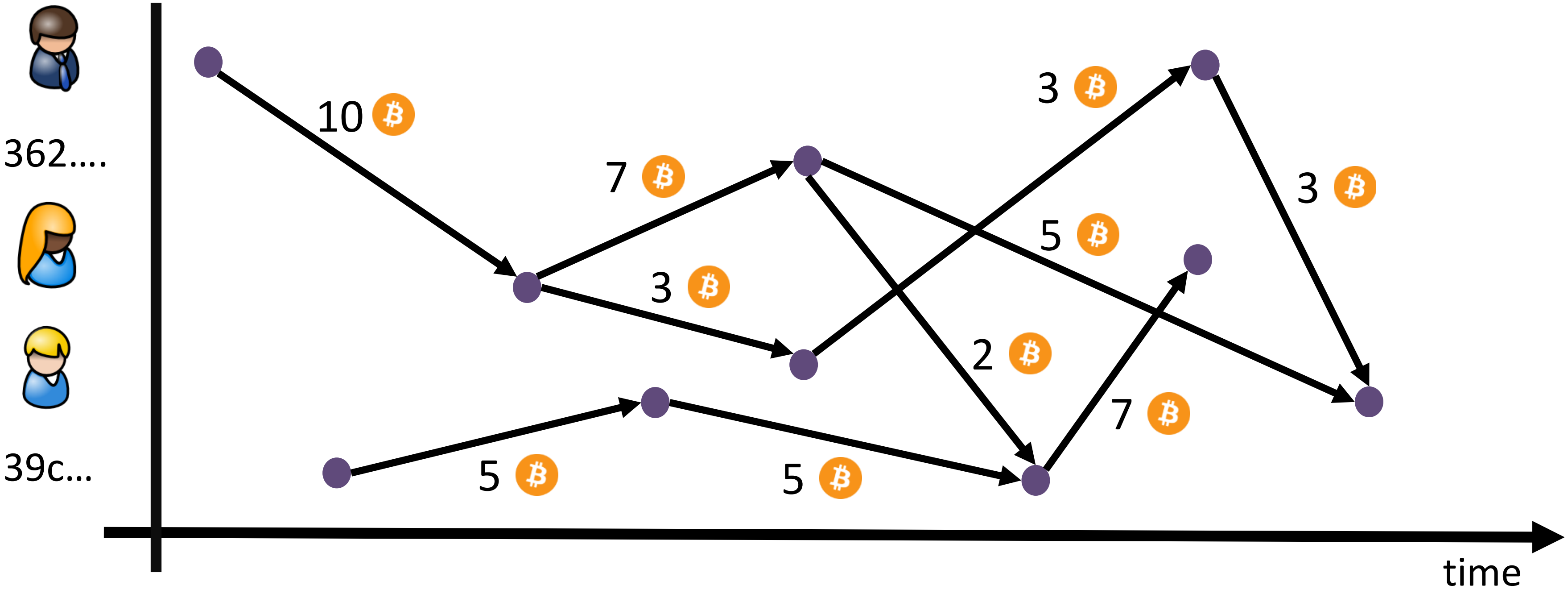
... and **anyone else** can analyze the ledger



“Those are just addresses”

... that are known by people you interact with

... and **anyone else** can analyze the ledger

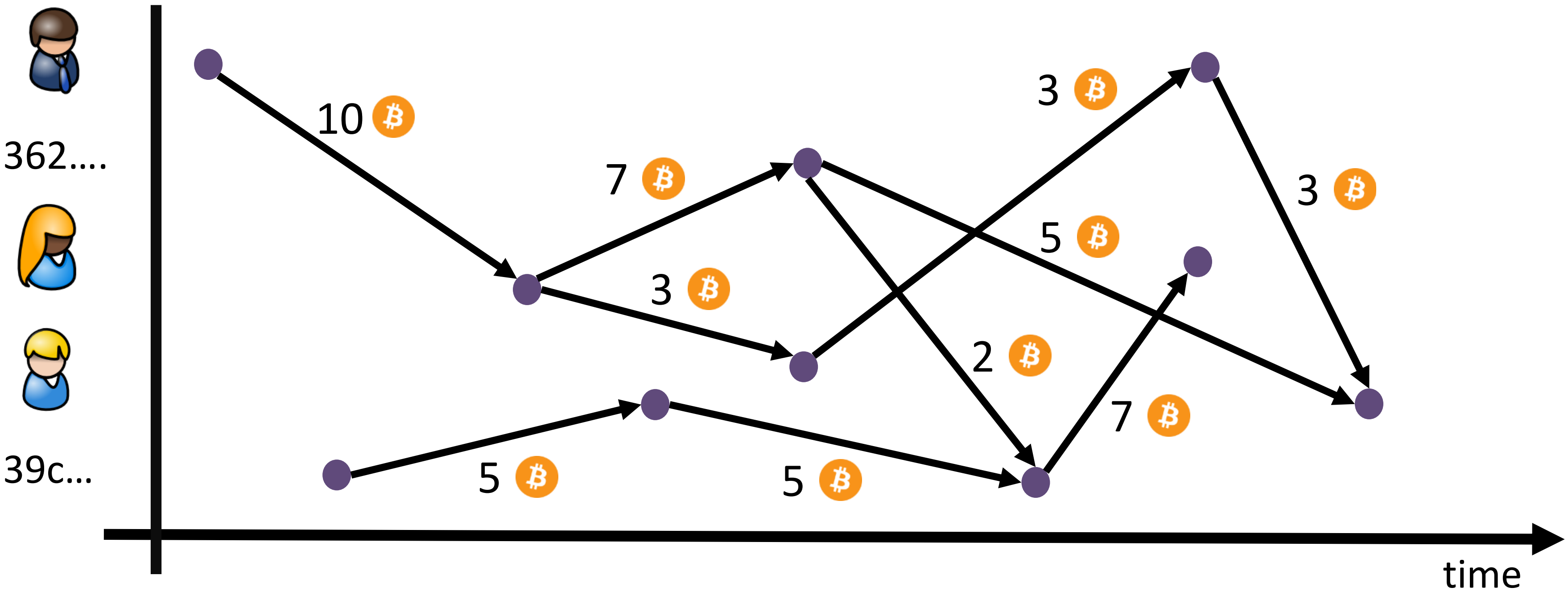


Fuse with **external databases**!

“Those are just addresses”

... that are known by people you interact with

... and **anyone else** can analyze the ledger

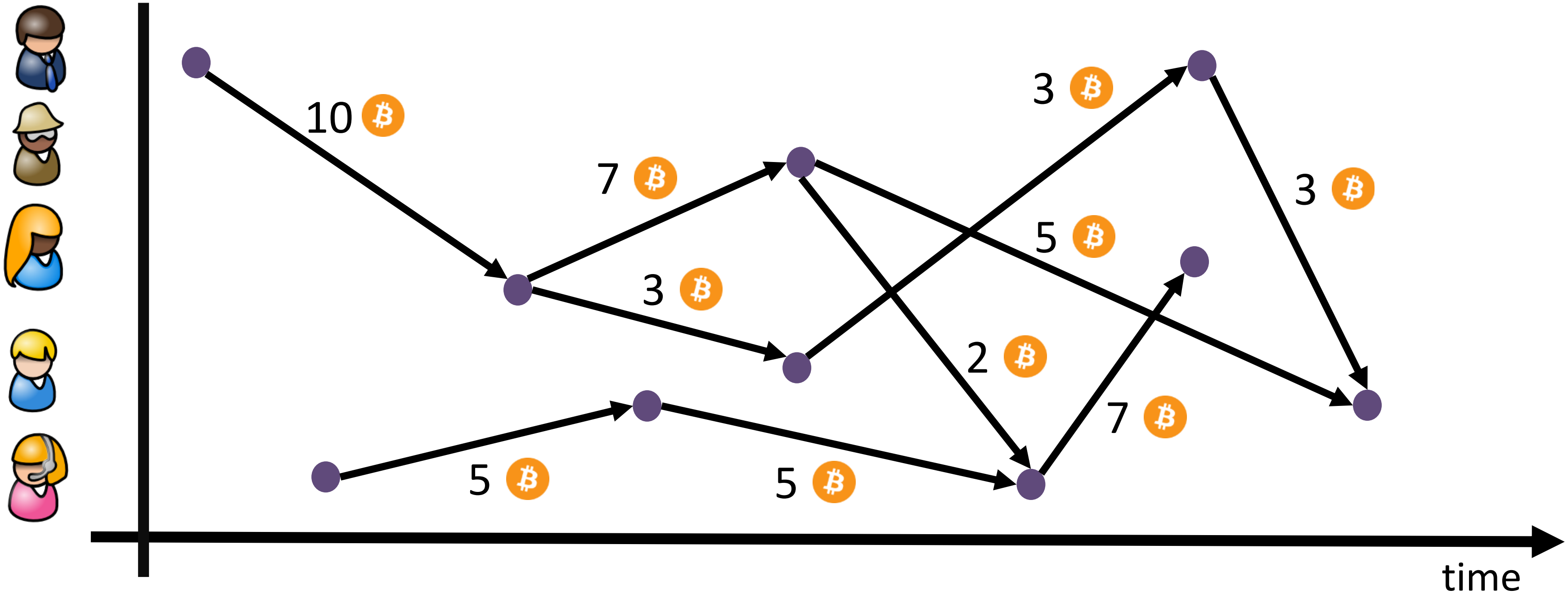


Fuse with **external databases**! 2016: IRS subpoenas Coinbase

“Those are just addresses”

… that are known by people you interact with

… and **anyone else** can analyze the ledger

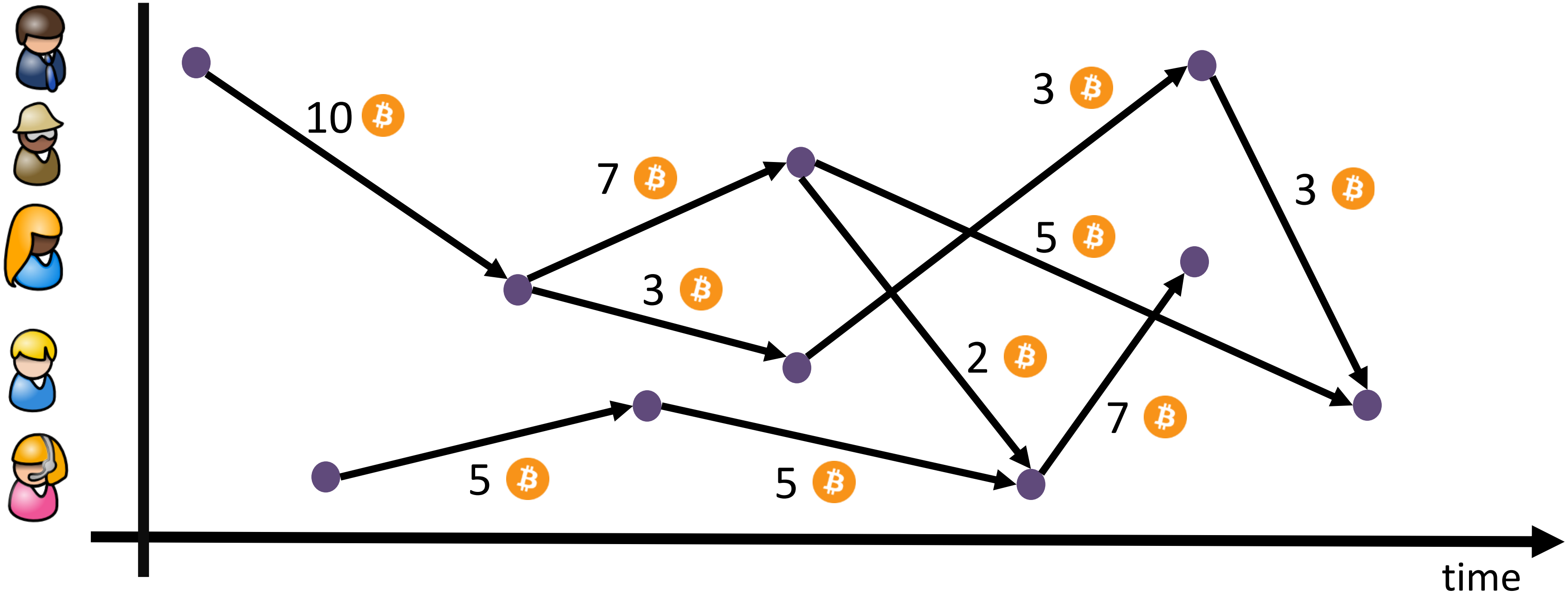


Fuse with **external databases**! 2016: IRS subpoenas Coinbase

“Those are just addresses”

… that are known by people you interact with

… and **anyone else** can analyze the ledger



Fuse with **external databases**! 2016: IRS subpoenas Coinbase

In practice: academics, FBI Silk Road investigations, …

Possible mitigations

Possible mitigations

- Use fresh addresses

Possible mitigations

- Use fresh addresses
- Mix coins together in a “tumbler”/“laundry”

Possible mitigations

- Use fresh addresses
- Mix coins together in a “tumbler”/“laundry”

1ab… ●

323… ●

1a3… ●

…

129… ●

Possible mitigations

- Use fresh addresses
- Mix coins together in a “tumbler”/“laundry”

1ab…

323…

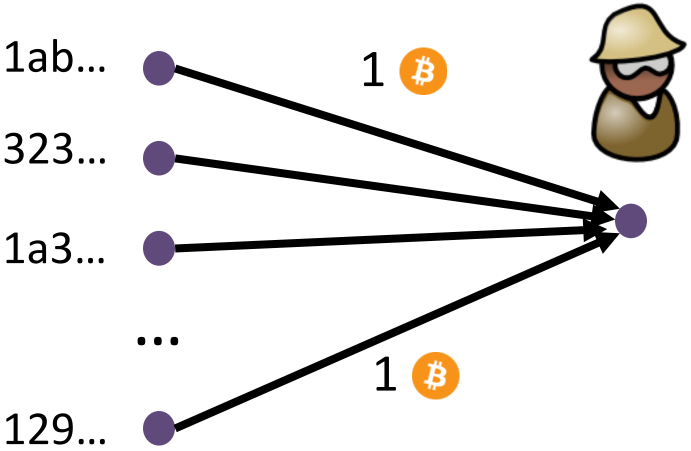
1a3…

…

129…

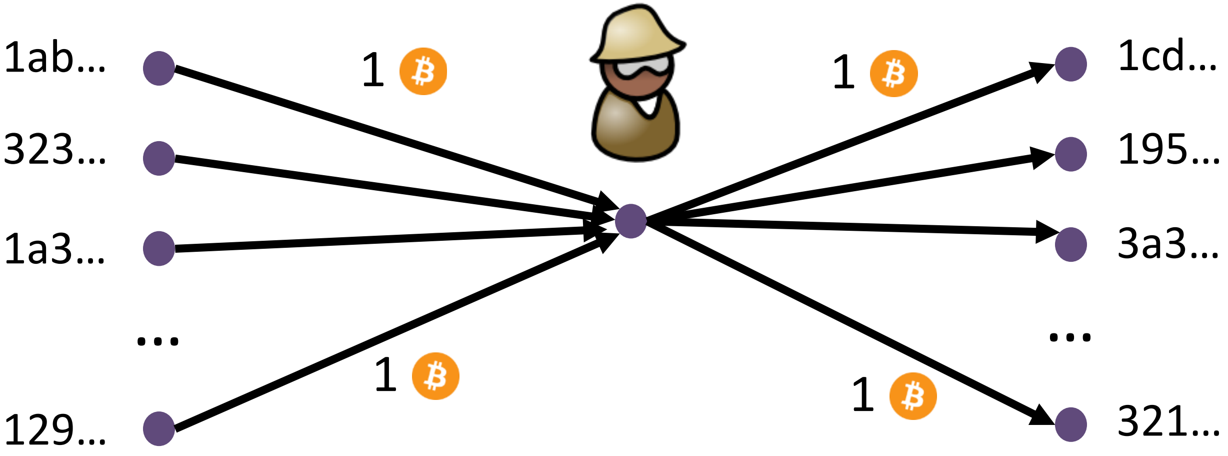
Possible mitigations

- Use fresh addresses
- Mix coins together in a “tumbler”/“laundry”



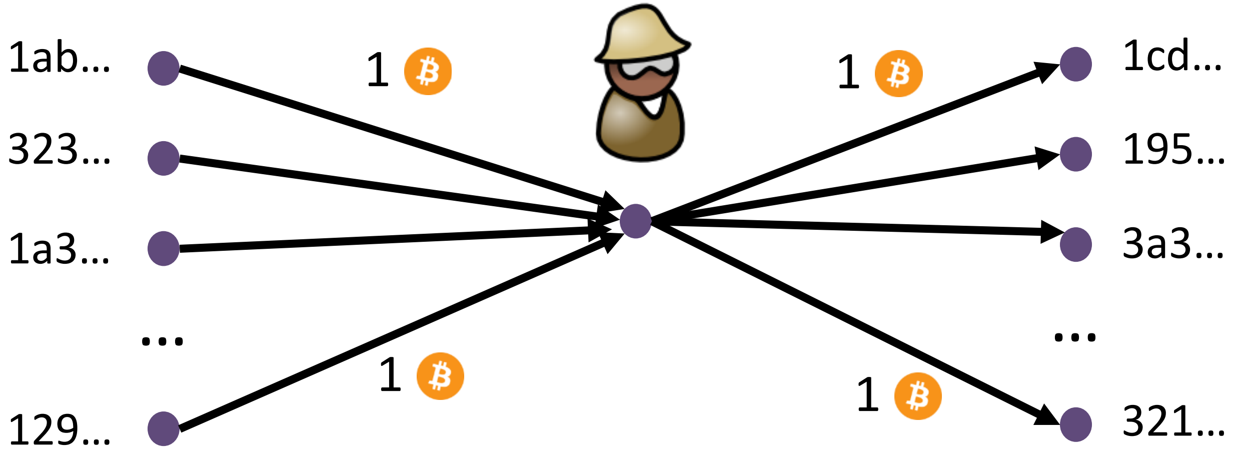
Possible mitigations

- Use fresh addresses
- Mix coins together in a “tumbler”/“laundry”



Possible mitigations

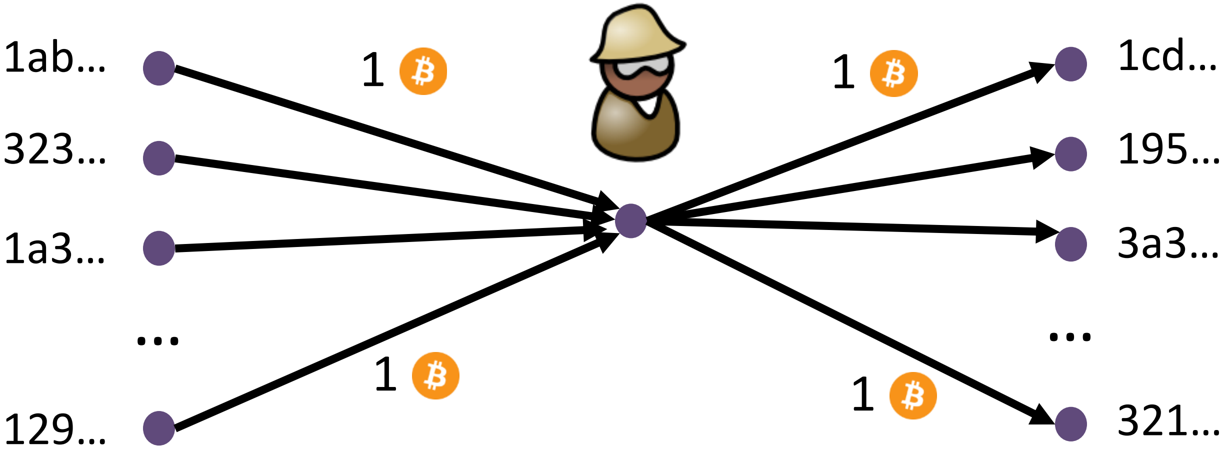
- Use fresh addresses
- Mix coins together in a “tumbler”/“laundry”



“Seems” harder to analyze, but tracks remain.

Possible mitigations

- Use fresh addresses
- Mix coins together in a “tumbler”/“laundry”

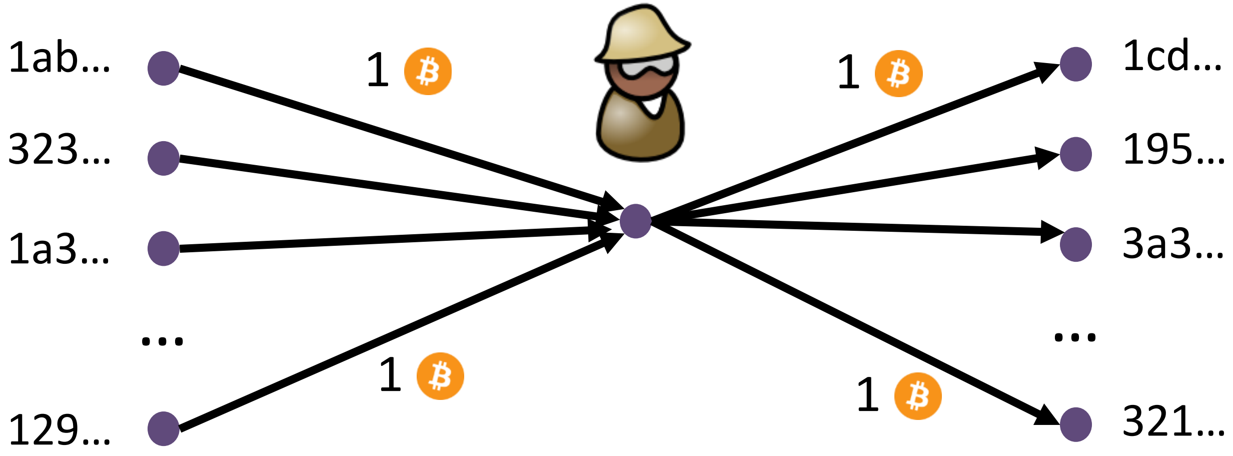


“Seems” harder to analyze, but tracks remain.

Methods of analysis only get **stronger**.

Possible mitigations

- Use fresh addresses
- Mix coins together in a “tumbler”/“laundry”



“Seems” harder to analyze, but tracks remain.

Methods of analysis only get **stronger**.

Your Bitcoin history is publicly saved **forever**.

Fungibility and trust in Bitcoin economy

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Ex.: Exchanges reject funds involved in DAO hack

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Ex.: Exchanges reject funds involved in DAO hack

Startups selling “fresh” coins at premium prices

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Ex.: Exchanges reject funds involved in DAO hack

Startups selling “fresh” coins at premium prices

Consequences:

Fungibility and trust in Bitcoin economy

Fungibility = "dollar is a dollar, no matter history"

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Ex.: Exchanges reject funds involved in DAO hack

Startups selling "fresh" coins at premium prices

Consequences:

- Recipients can **devalue your coins** when accepting

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Ex.: Exchanges reject funds involved in DAO hack

Startups selling “fresh” coins at premium prices

Consequences:

- Recipients can **devalue your coins** when accepting

Only way to know value: ask a **central party**?!

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Ex.: Exchanges reject funds involved in DAO hack

Startups selling “fresh” coins at premium prices

Consequences:

- Recipients can **devalue your coins** when accepting

  Only way to know value: ask a **central party**?!

- **Price discrimination**.

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Ex.: Exchanges reject funds involved in DAO hack

Startups selling “fresh” coins at premium prices

Consequences:

- Recipients can **devalue your coins** when accepting

  Only way to know value: ask a **central party**?!

- **Price discrimination**. Get a raise and get a rent hike?

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Ex.: Exchanges reject funds involved in DAO hack

Startups selling “fresh” coins at premium prices

Consequences:

- Recipients can **devalue your coins** when accepting

  Only way to know value: ask a **central party**?!

- **Price discrimination**. Get a raise and get a rent hike?

- **Censorship**.

Fungibility and trust in Bitcoin economy

Fungibility = “dollar is a dollar, no matter history”

(Crawfurd v. The Royal Bank, 1749)

Not so for cryptocurrencies! Bad coins **taint** good ones.

Ex.: Exchanges reject funds involved in DAO hack

Startups selling “fresh” coins at premium prices

Consequences:

- Recipients can **devalue your coins** when accepting

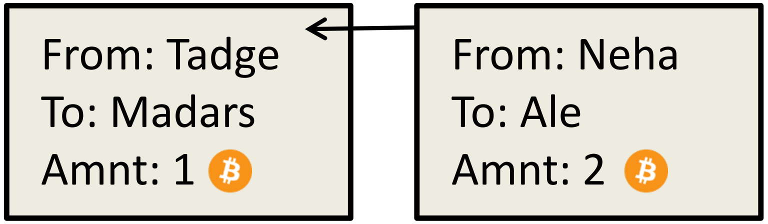
  Only way to know value: ask a **central party**?!

- **Price discrimination**. Get a raise and get a rent hike?

- **Censorship**. Miners could pick-and-choose transactions

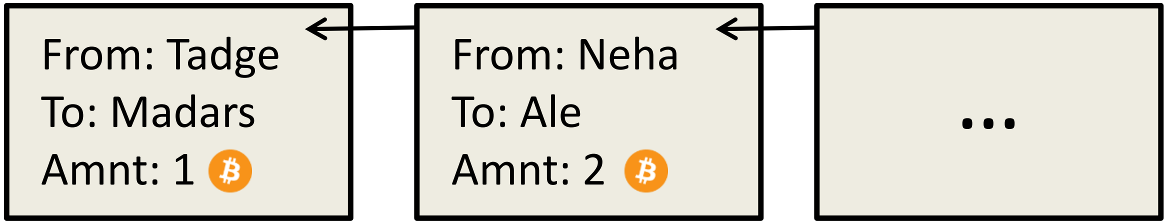
Q: If privacy is important, why isn’t Bitcoin private?

From: Tadge
To: Madars
Amnt: 1 ₿



From: Tadge
To: Madars
Amnt: 1 ₿

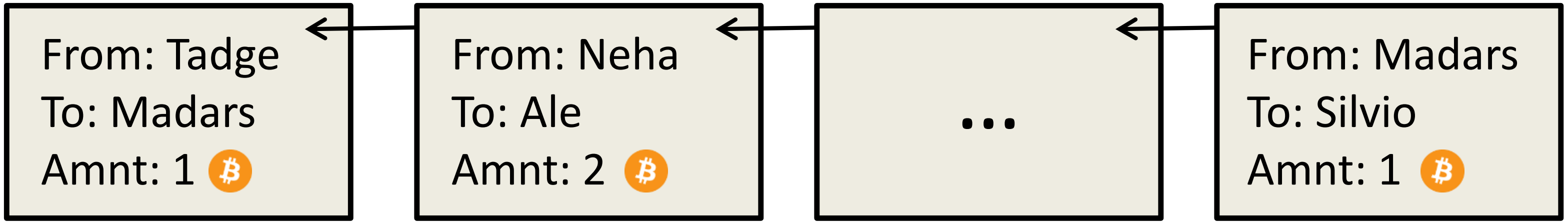
From: Neha
To: Ale
Amnt: 2 ₿



From: Tadge
To: Madars
Amnt: 1 ₿

From: Neha
To: Ale
Amnt: 2 ₿

...



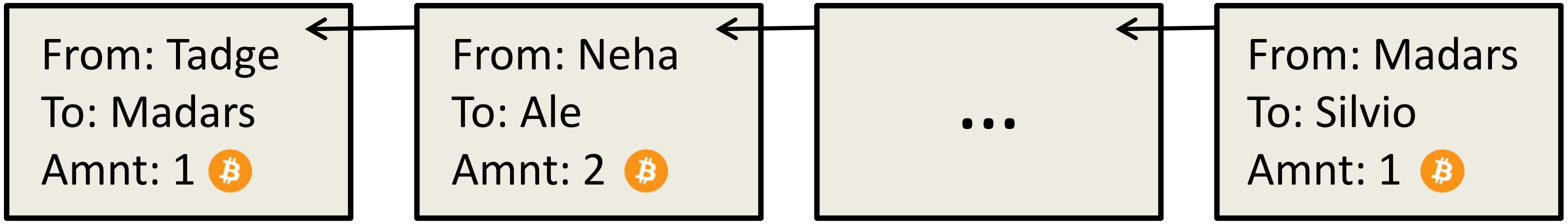
From: Tadge
To: Madars
Amnt: 1 ₿

From: Neha
To: Ale
Amnt: 2 ₿

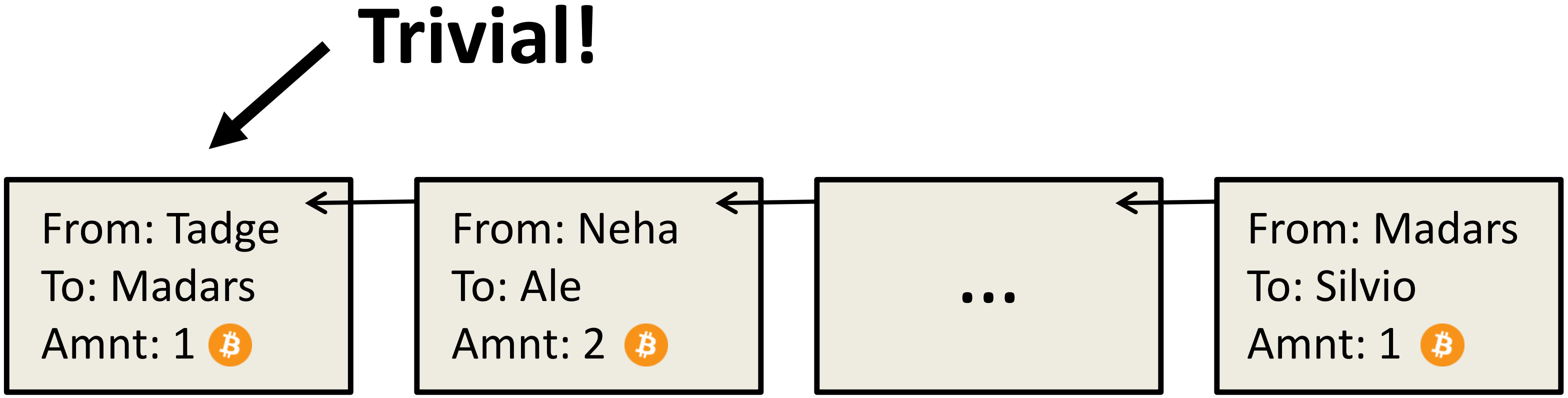
…

From: Madars
To: Silvio
Amnt: 1 ₿

How does Silvio know I have 1 Bitcoin to spend?



How does Silvio know I have 1 Bitcoin to spend?



Trivial!

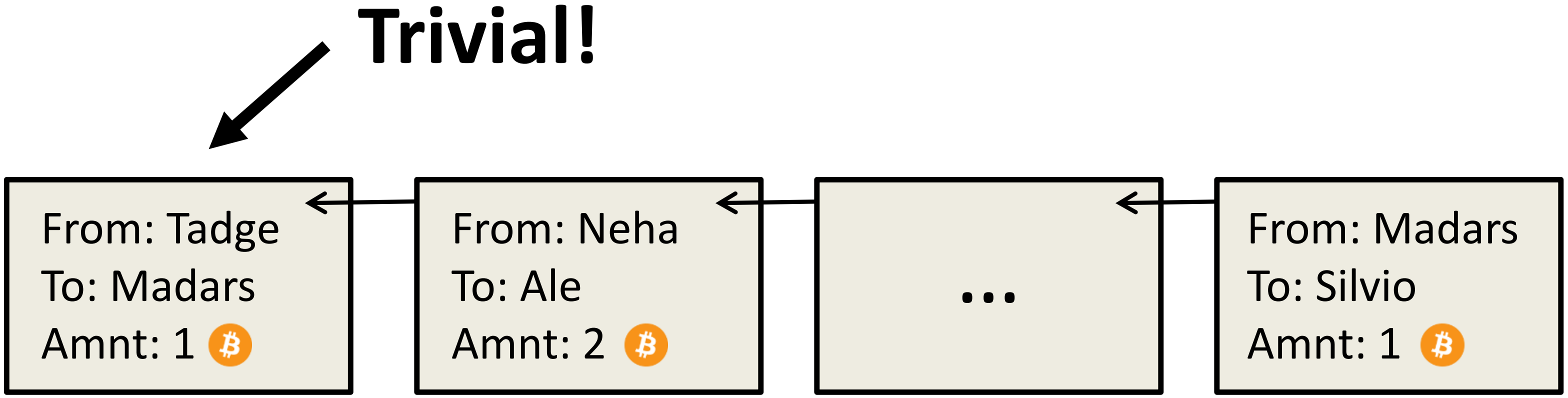
From: Tadge
To: Madars
Amnt: 1 ₿

From: Neha
To: Ale
Amnt: 2 ₿

…

From: Madars
To: Silvio
Amnt: 1 ₿

How does Silvio know I have 1 Bitcoin to spend?



Trivial!

From: Tadge
To: Madars
Amnt: 1 ₿

From: Neha
To: Ale
Amnt: 2 ₿

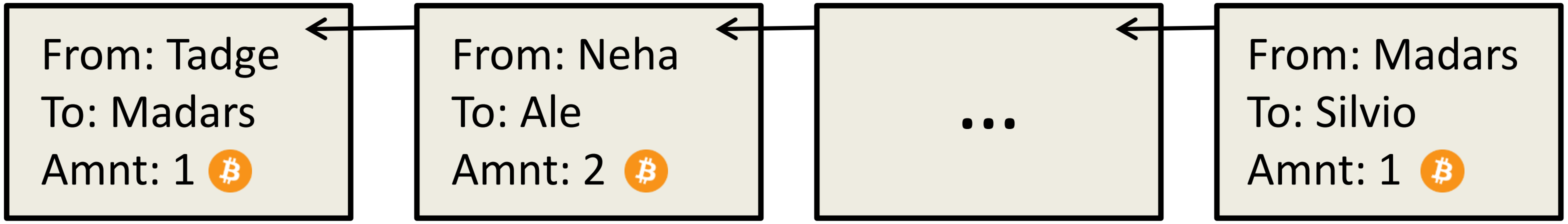
…

From: Madars
To: Silvio
Amnt: 1 ₿

What if we encrypted the blockchain?

How does Silvio know I have 1 Bitcoin to spend?

Trivial!



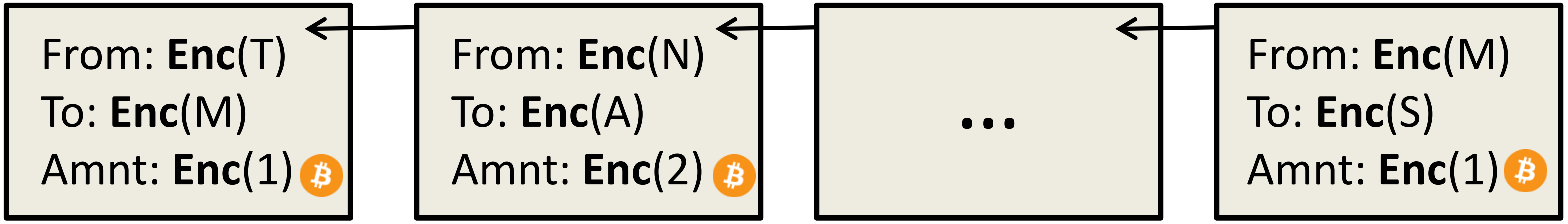
From: Tadge
To: Madars
Amnt: 1 ₿

From: Neha
To: Ale
Amnt: 2 ₿

…

From: Madars
To: Silvio
Amnt: 1 ₿

What if we encrypted the blockchain?



From: **Enc**(T)
To: **Enc**(M)
Amnt: **Enc**(1) ₿

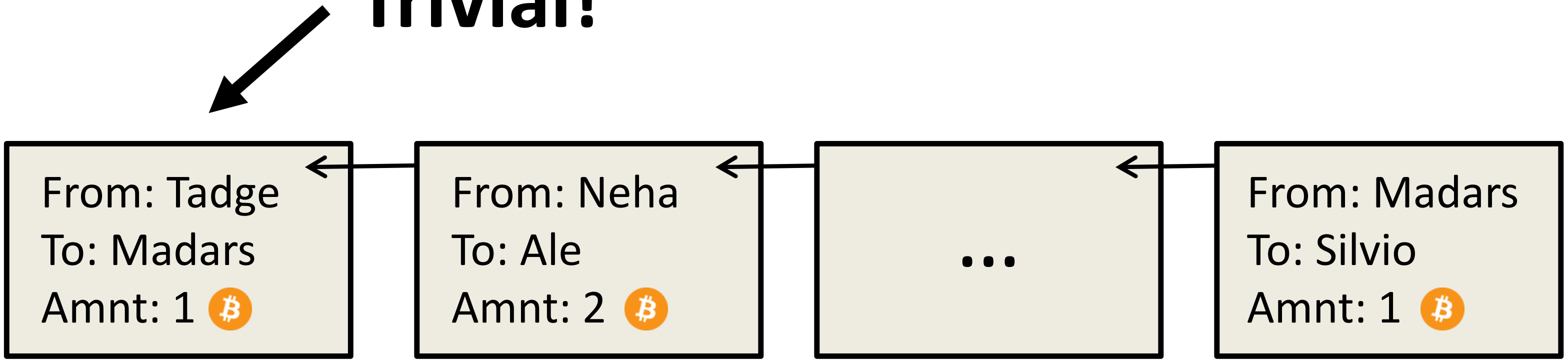
From: **Enc**(N)
To: **Enc**(A)
Amnt: **Enc**(2) ₿

…

From: **Enc**(M)
To: **Enc**(S)
Amnt: **Enc**(1) ₿

How does Silvio know I have 1 Bitcoin to spend?

Trivial!



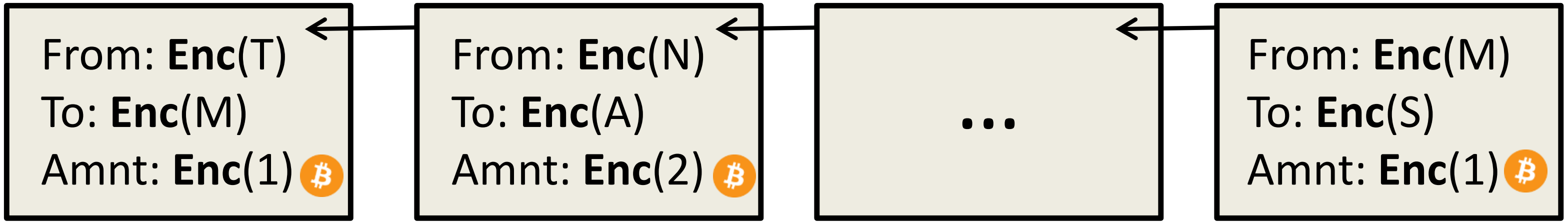
From: Tadge
To: Madars
Amnt: 1 ₿

From: Neha
To: Ale
Amnt: 2 ₿

…

From: Madars
To: Silvio
Amnt: 1 ₿

What if we encrypted the blockchain?



From: **Enc**(T)
To: **Enc**(M)
Amnt: **Enc**(1) ₿

From: **Enc**(N)
To: **Enc**(A)
Amnt: **Enc**(2) ₿

…

From: **Enc**(M)
To: **Enc**(S)
Amnt: **Enc**(1) ₿

Privacy is fundamentally at odds with correctness.

Outline

1. Bitcoin's privacy problem

2. Zerocash: privacy-preserving decentralized currency

3. Zcash: deploying Zerocash in practice…

Zerocash

A cryptocurrency with following properties:

Zerocash

A cryptocurrency with following properties:

Decentralized

Designed to sit on top of any ledger-based currency

Zerocash

A cryptocurrency with following properties:

Decentralized

Designed to sit on top of any ledger-based currency

Privacy-preserving

Provably hides tx origin, destination and amount

Zerocash

A cryptocurrency with following properties:

Decentralized

Designed to sit on top of any ledger-based currency

Privacy-preserving

Provably hides tx origin, destination and amount

Efficient

Tx: 1 KB in size, <1 min to produce, <6 ms to verify

New notion:

New notion:

Decentralized Anonymous Payments

New notion:

Decentralized Anonymous Payments

New notion:

Decentralized Anonymous Payments

Algorithms:

New notion:

Decentralized Anonymous Payments

Algorithms: `Setup` `CreateAddress` `Mint`

`Send` `VerifyTransaction` `Receive`

New notion:

Decentralized Anonymous Payments

Algorithms: `Setup` `CreateAddress` `Mint`

`Send` `VerifyTransaction` `Receive`

Security requirements:

New notion:

Decentralized Anonymous Payments

Algorithms: `Setup` `CreateAddress` `Mint`

`Send` `VerifyTransaction` `Receive`

Security requirements:

Ledger indistinguishability

Nothing revealed besides public information, even by chosen-transaction adversary.

New notion:

Decentralized Anonymous Payments

Algorithms: `Setup` `CreateAddress` `Mint`
`Send` `VerifyTransaction` `Receive`

Security requirements:

Ledger indistinguishability

Nothing revealed besides public information, even by chosen-transaction adversary.

Balance

Can’t spend more money than received or minted.

New notion:

Decentralized Anonymous Payments

Algorithms: `Setup` `CreateAddress` `Mint` `Send` `VerifyTransaction` `Receive`

Security requirements:

Ledger indistinguishability

Nothing revealed besides public information, even by chosen-transaction adversary.

Balance

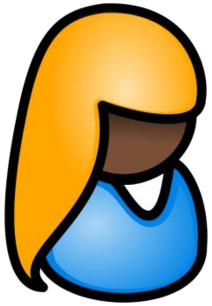
Can’t spend more money than received or minted.

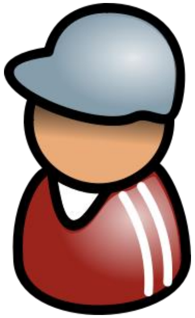
Transaction non-malleability

Cannot manipulate transactions en route to ledger.

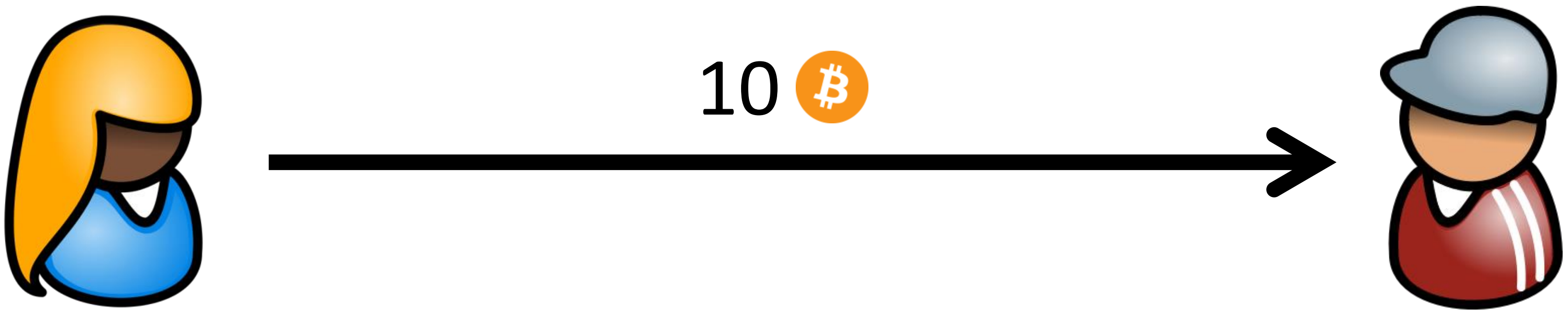
10k ft view: integrity when all is hidden

10k ft view: integrity when all is hidden

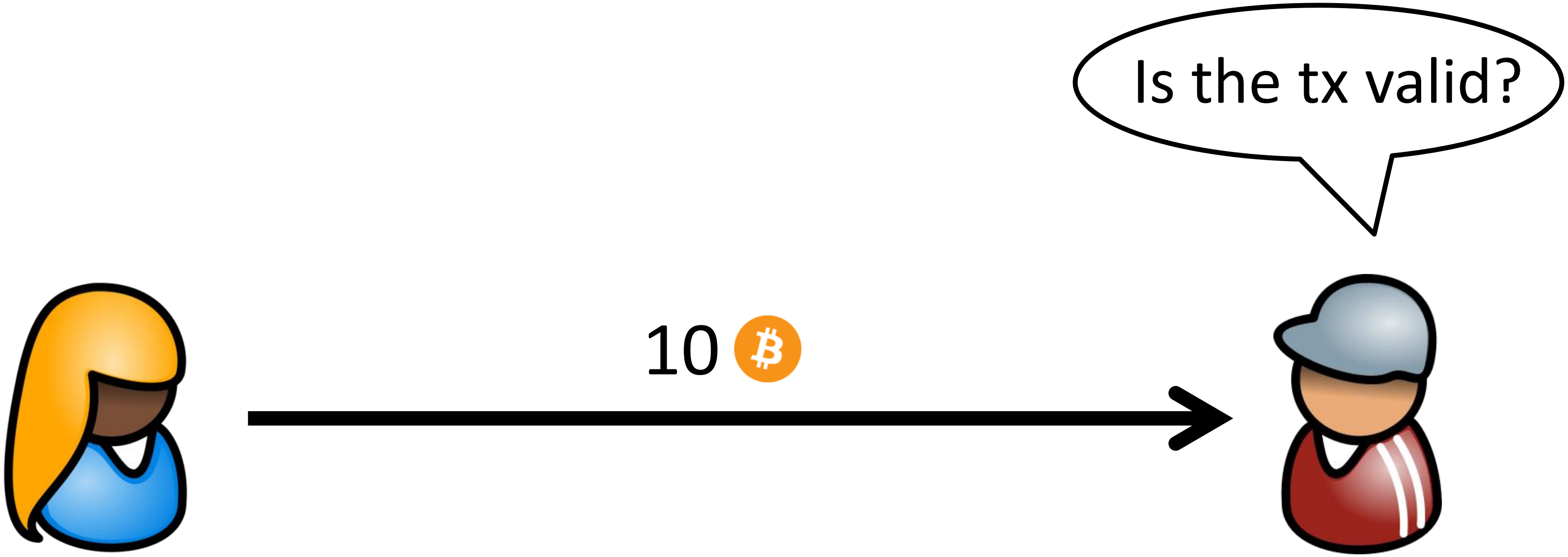




10k ft view: integrity when all is hidden



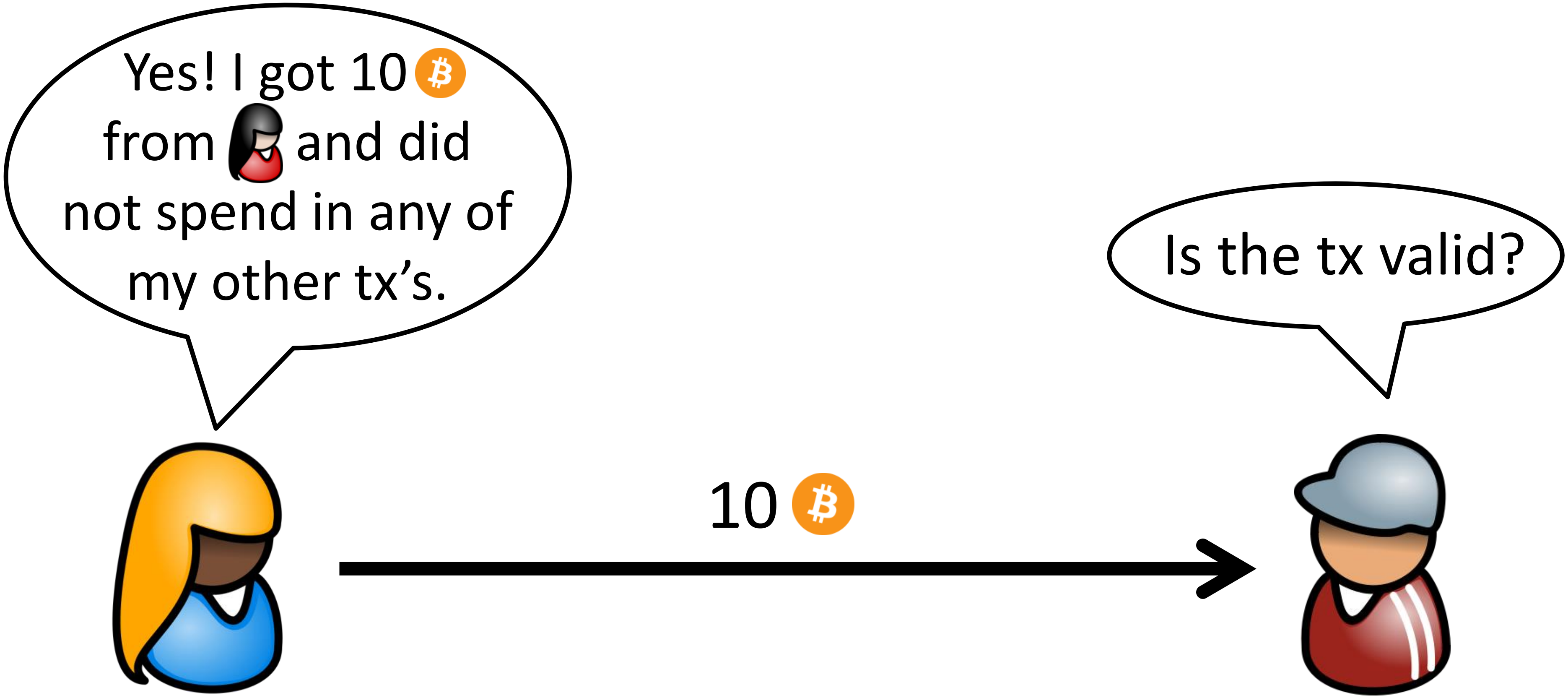
10k ft view: integrity when all is hidden



10 ₿

Is the tx valid?

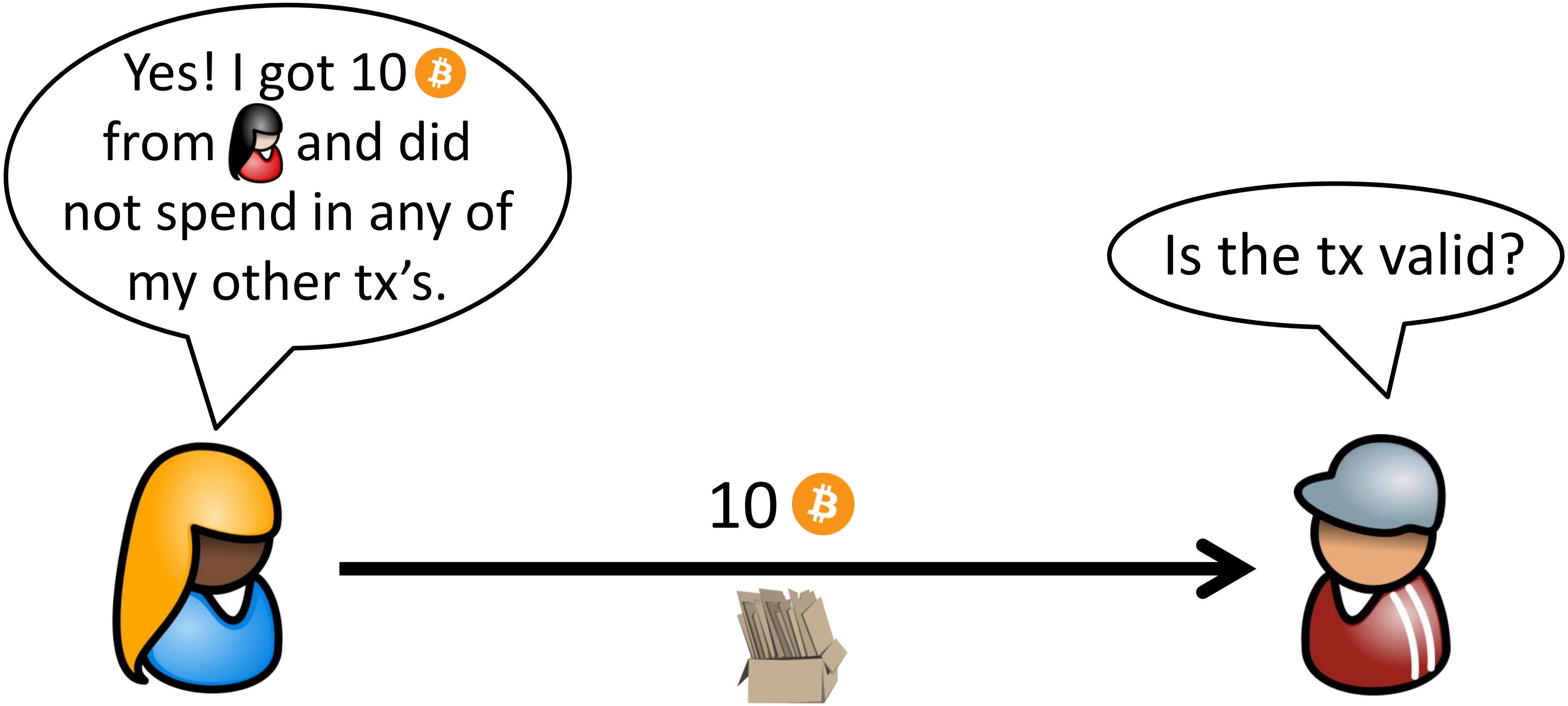
10k ft view: integrity when all is hidden



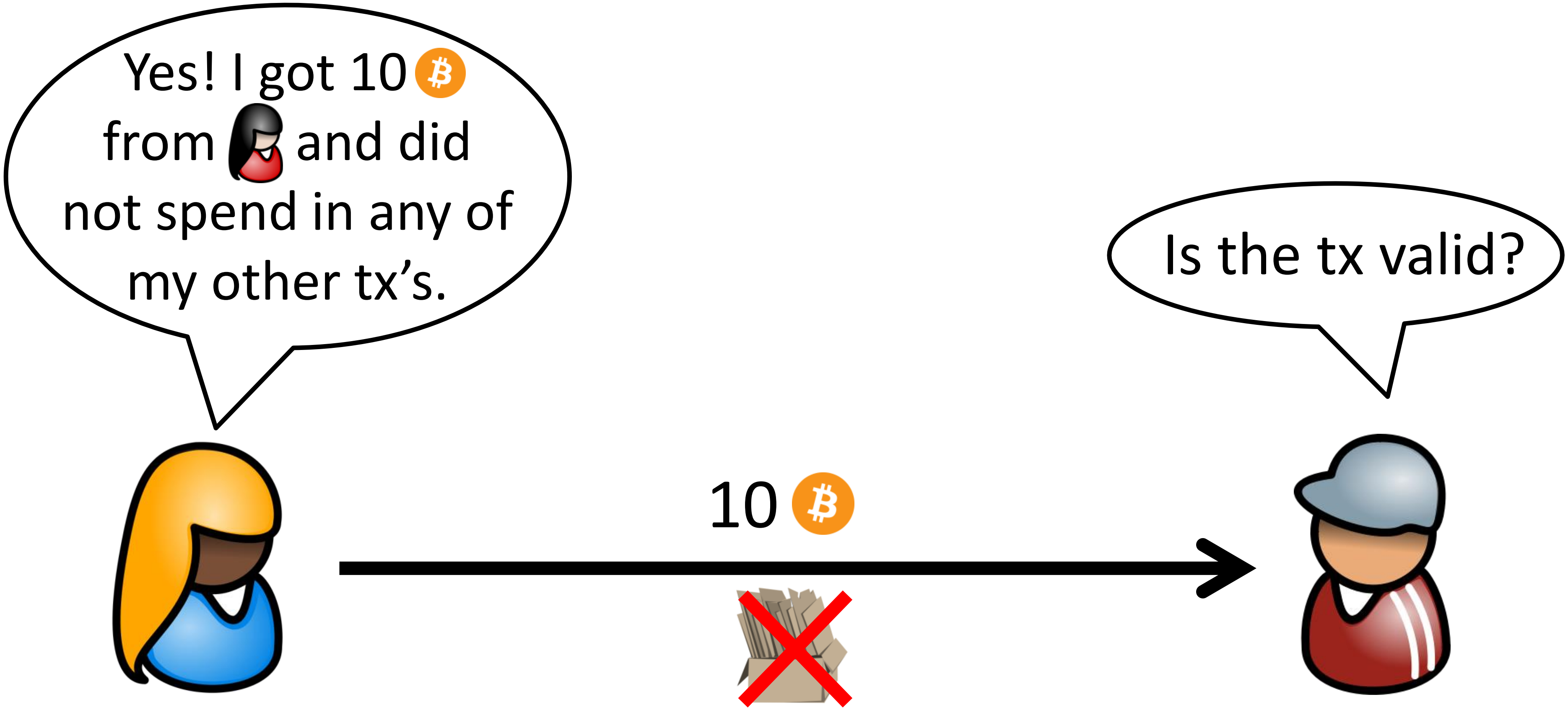
Yes! I got 10 ₿ from 👩 and did not spend in any of my other tx's.

Is the tx valid?

10k ft view: integrity when all is hidden


Yes! I got 10 ₿ from 👤 and did not spend in any of my other tx's.
Is the tx valid?
10 ₿

10k ft view: integrity when all is hidden

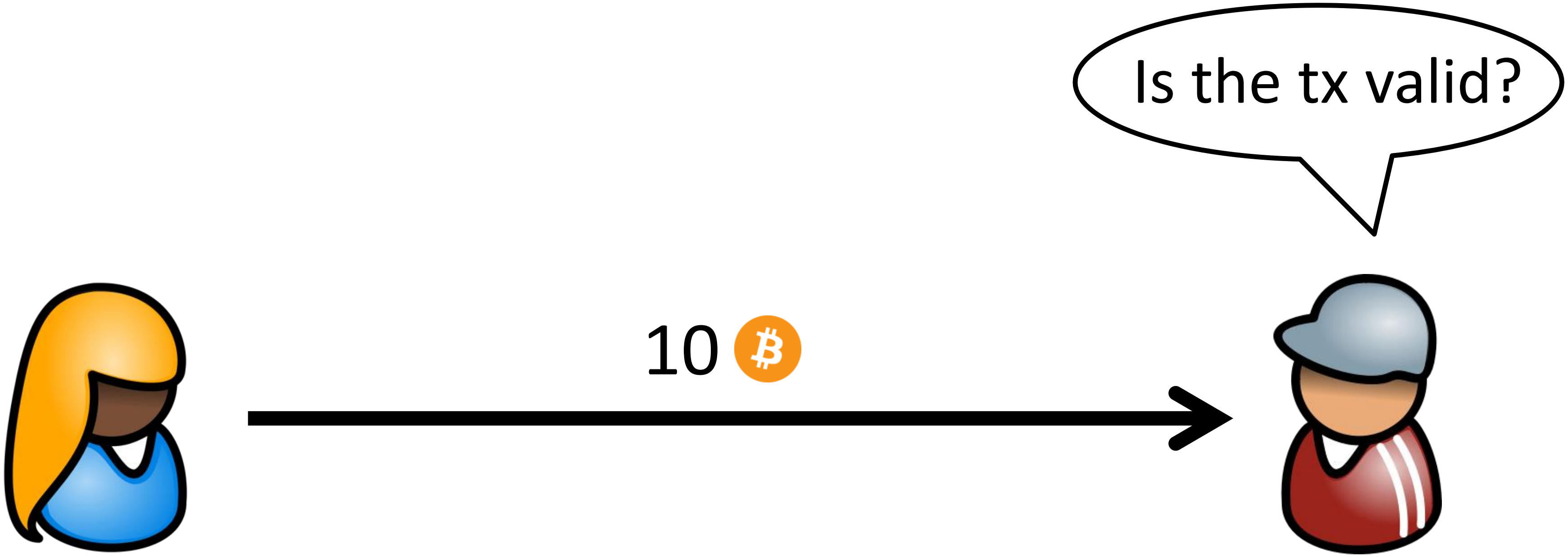


Yes! I got 10 ₿ from 👤 and did not spend in any of my other tx's.

Is the tx valid?

10 ₿

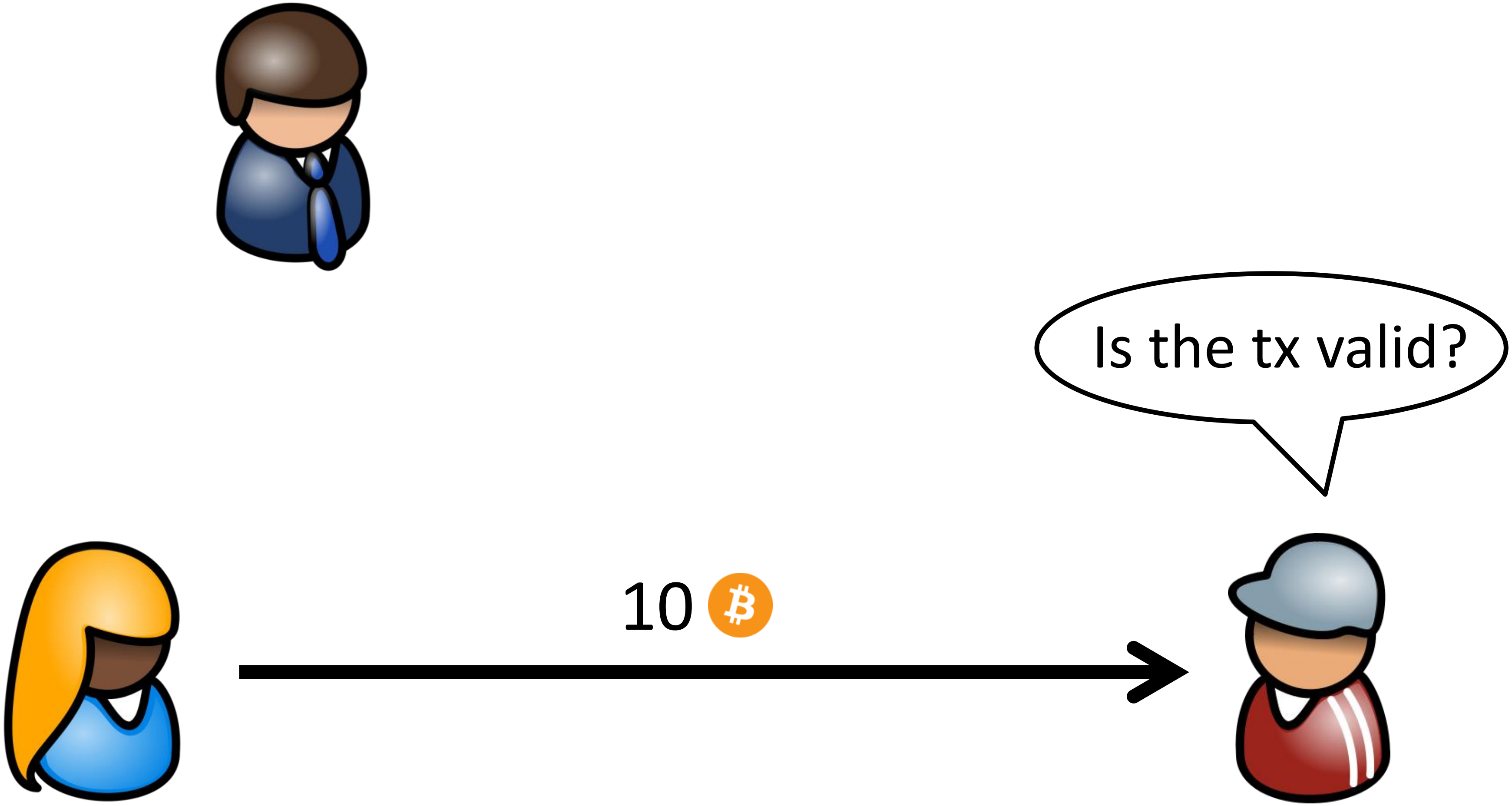
10k ft view: integrity when all is hidden



10 ₿

Is the tx valid?

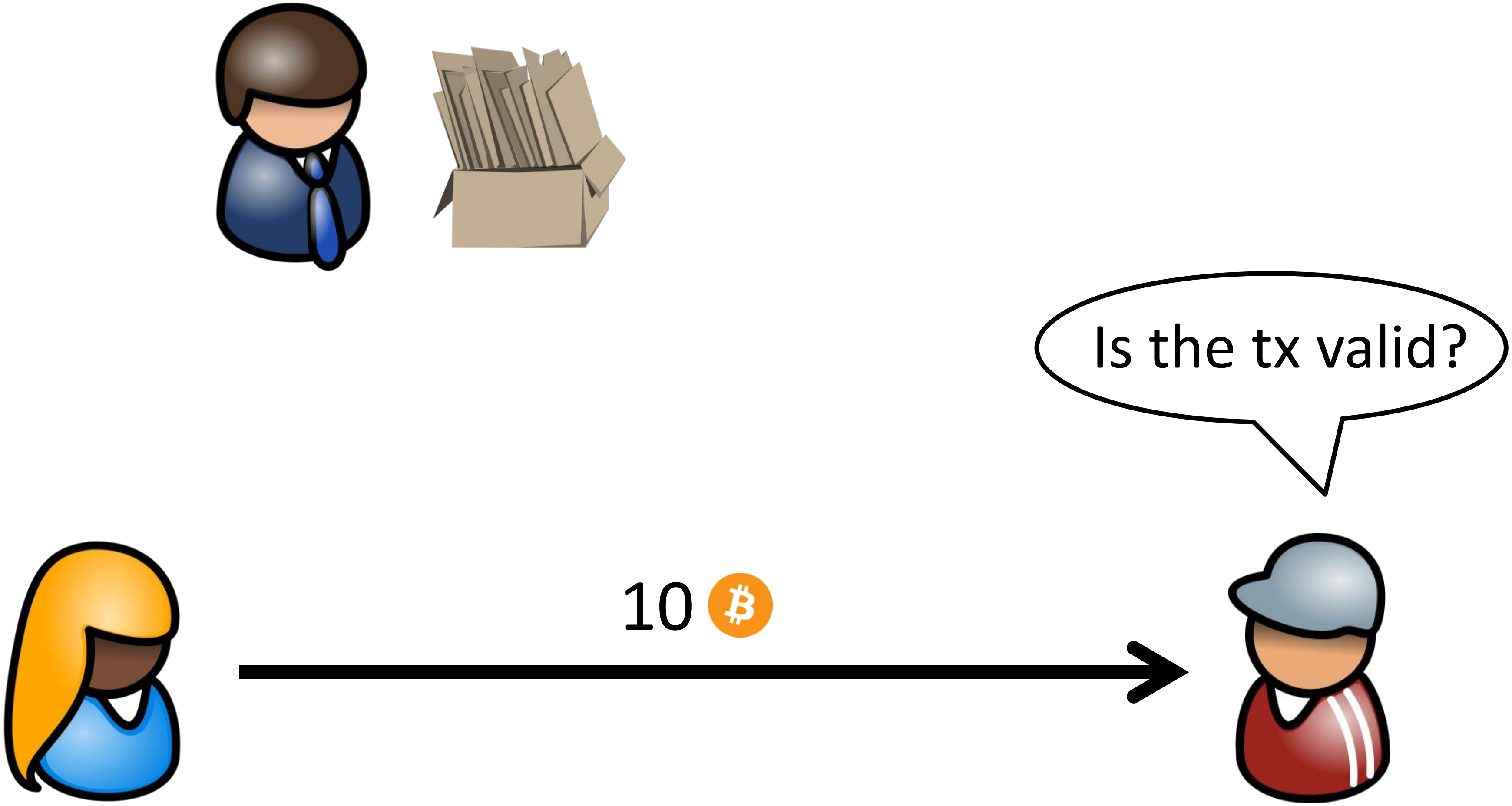
10k ft view: integrity when all is hidden



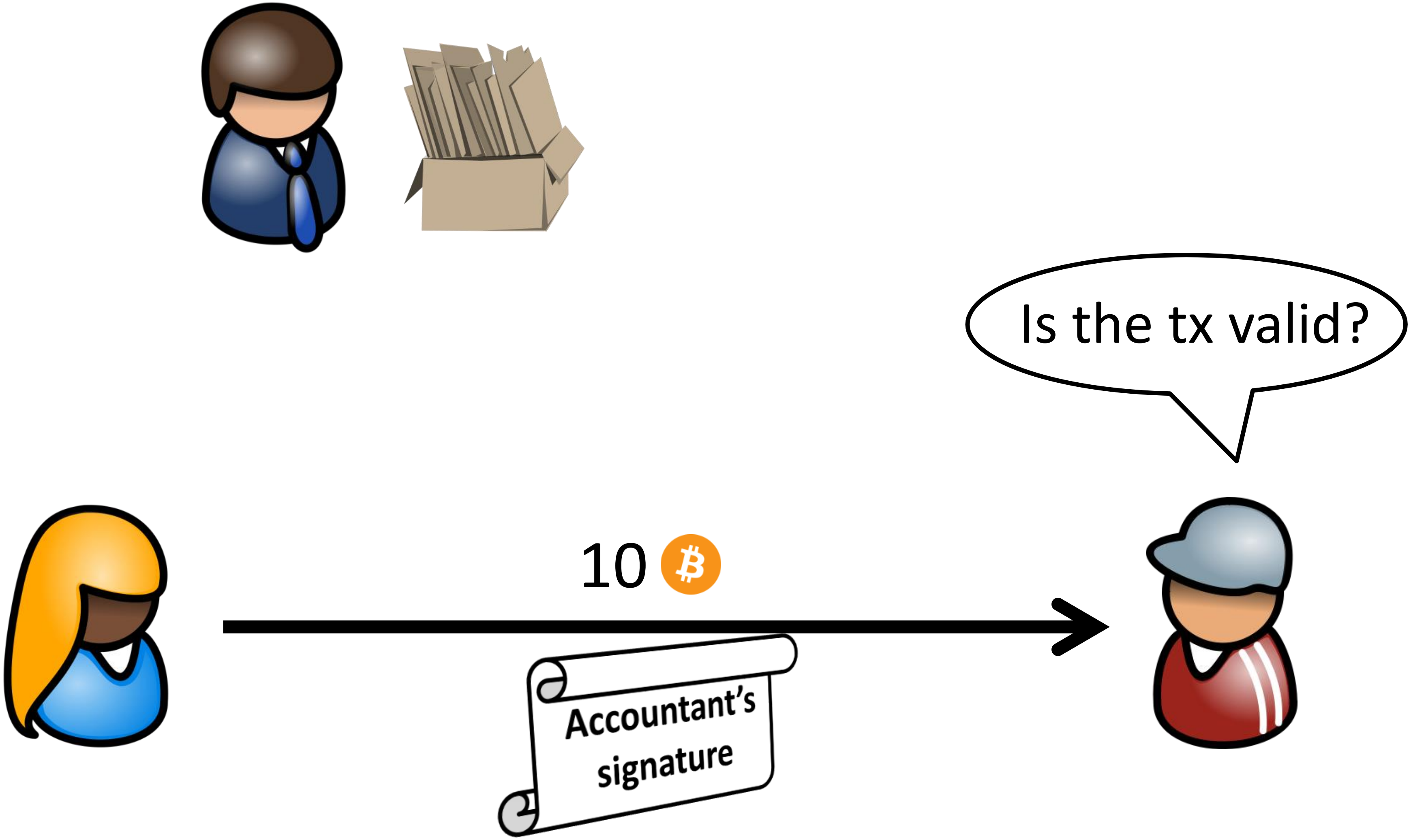
10 ₿

Is the tx valid?

10k ft view: integrity when all is hidden



10k ft view: integrity when all is hidden

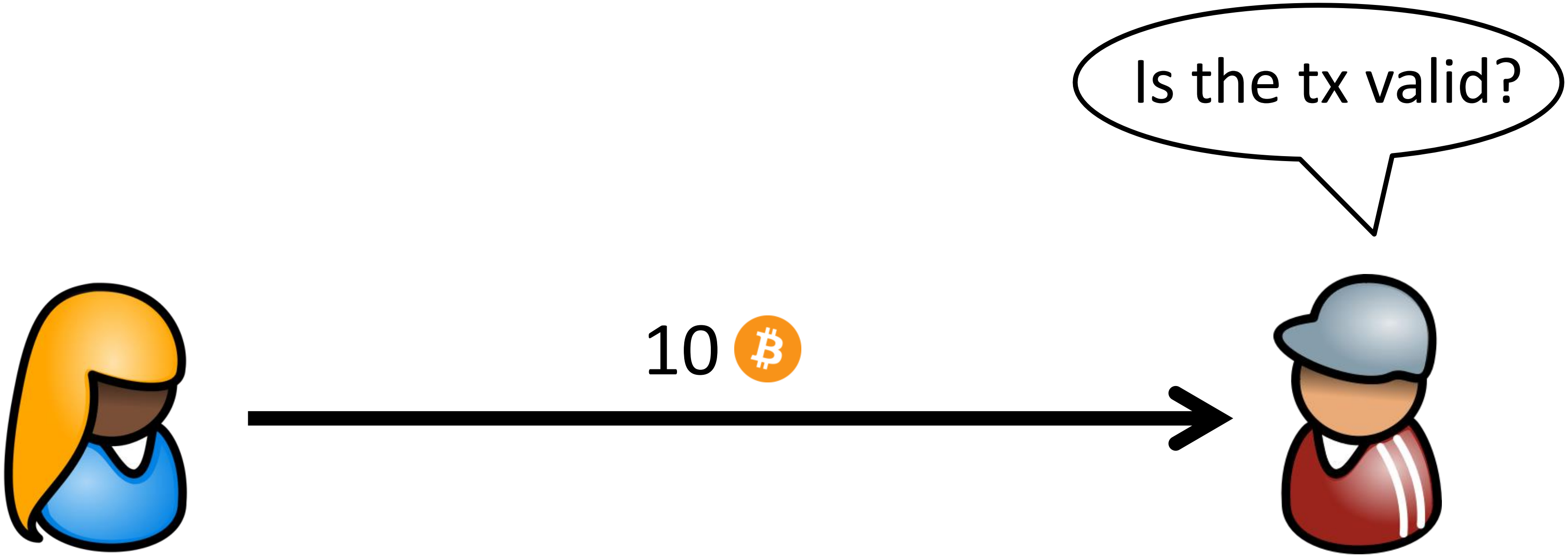


Is the tx valid?

10 ₿

Accountant's signature

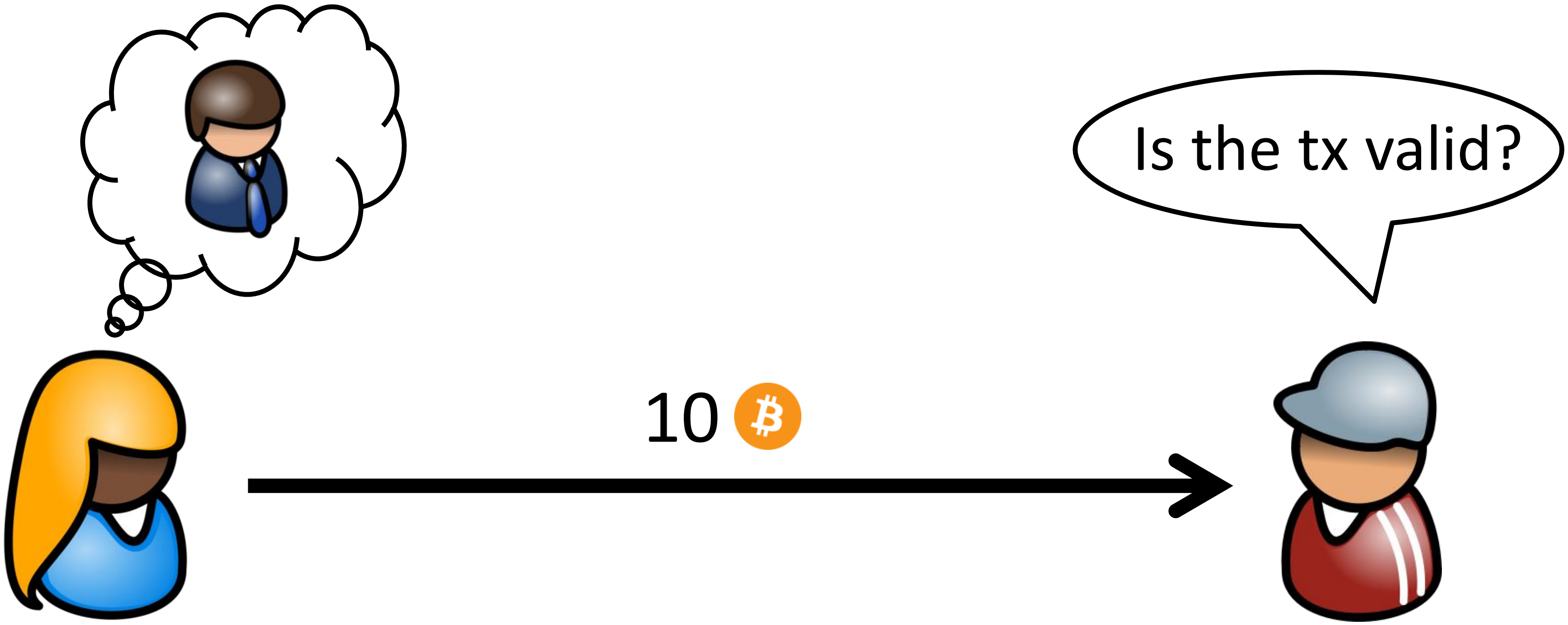
10k ft view: integrity when all is hidden



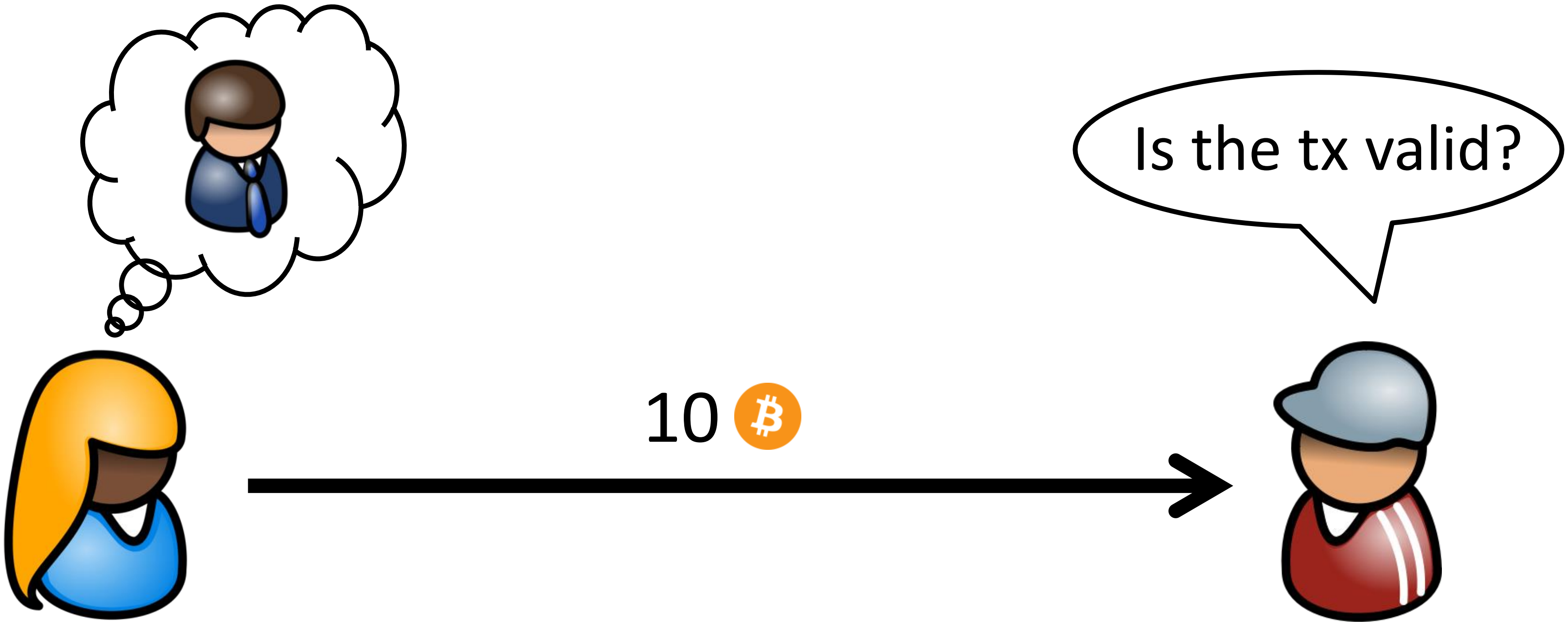
Is the tx valid?

10 ₿

10k ft view: integrity when all is hidden

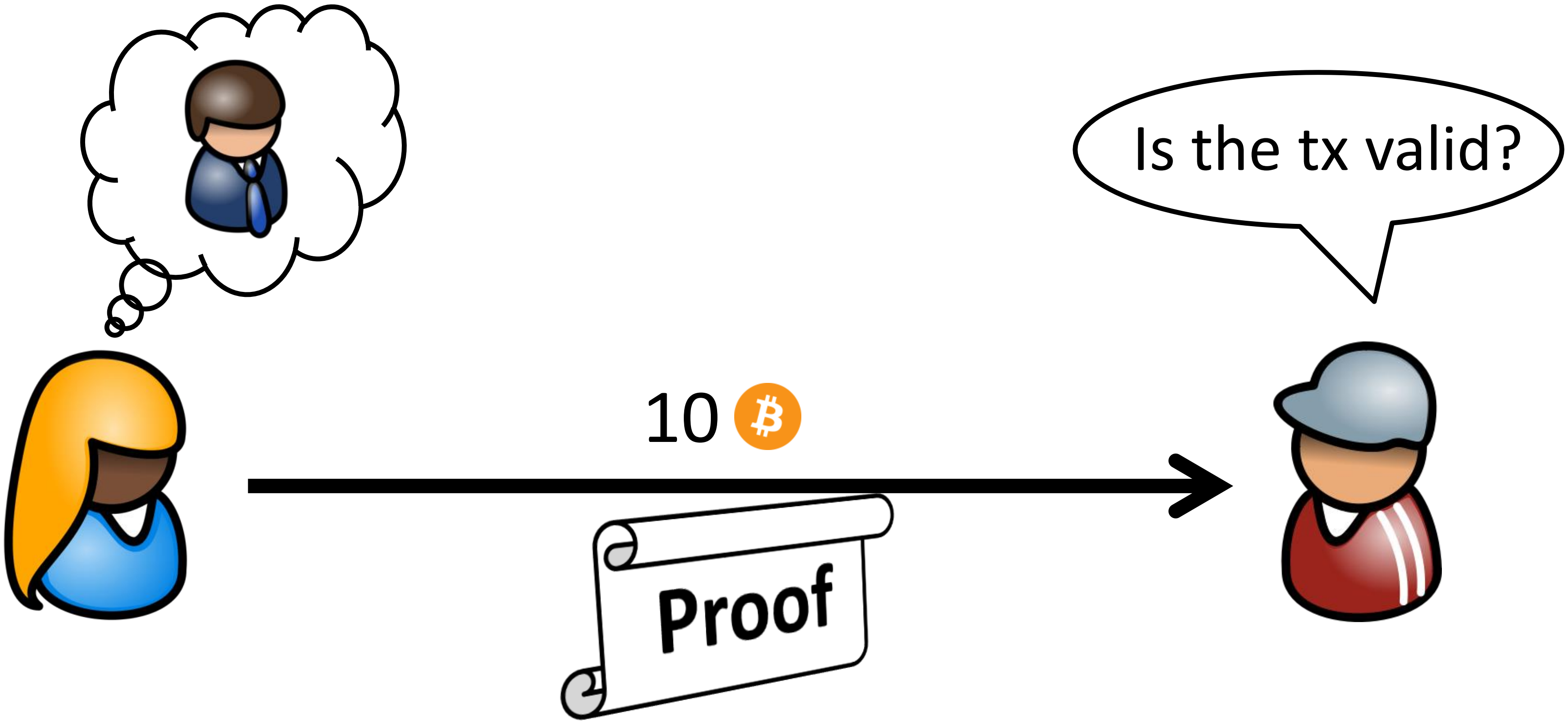


10k ft view: integrity when all is hidden



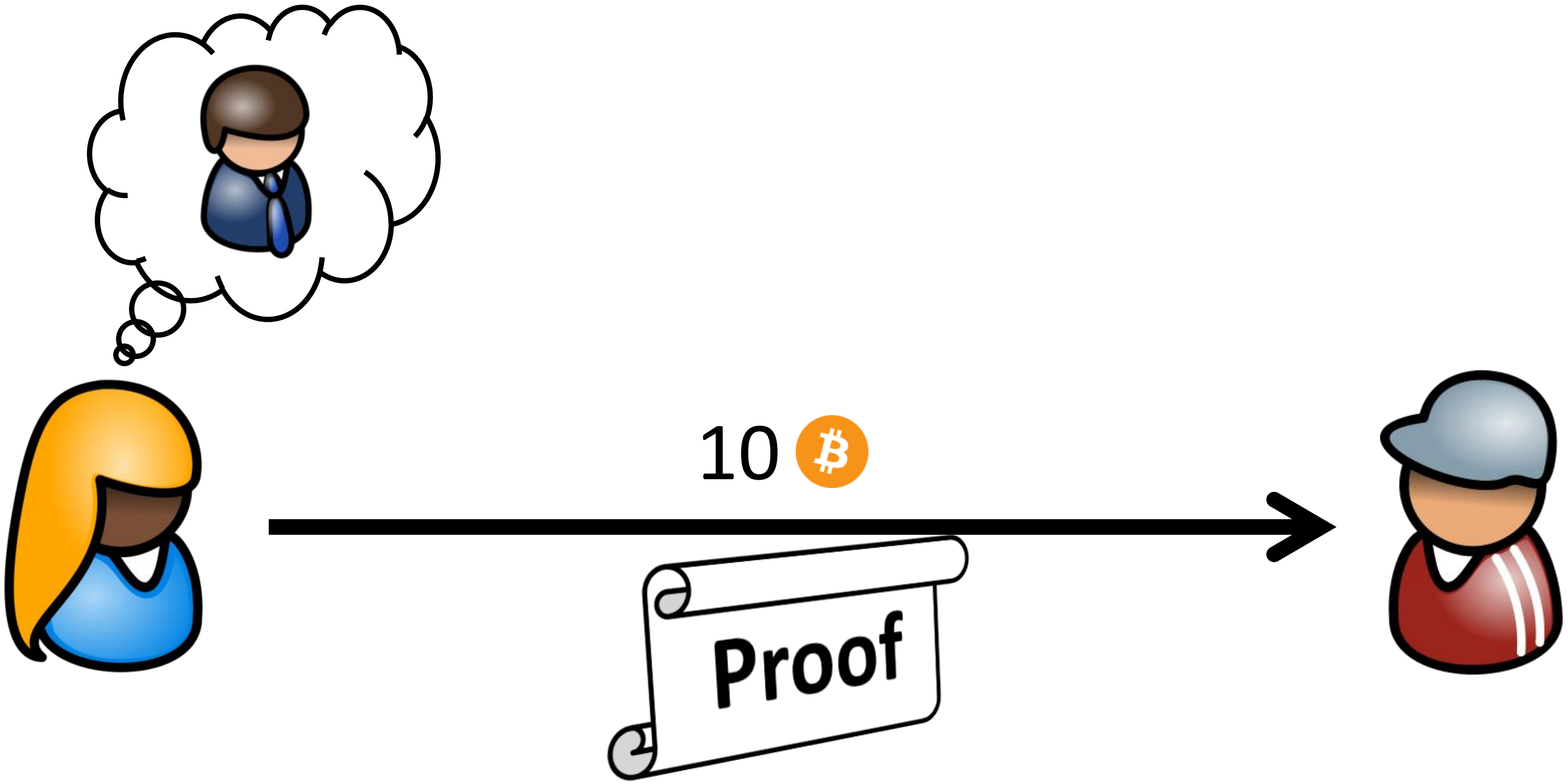
Intuition: “virtual accountant” using cryptographic proofs.

10k ft view: integrity when all is hidden



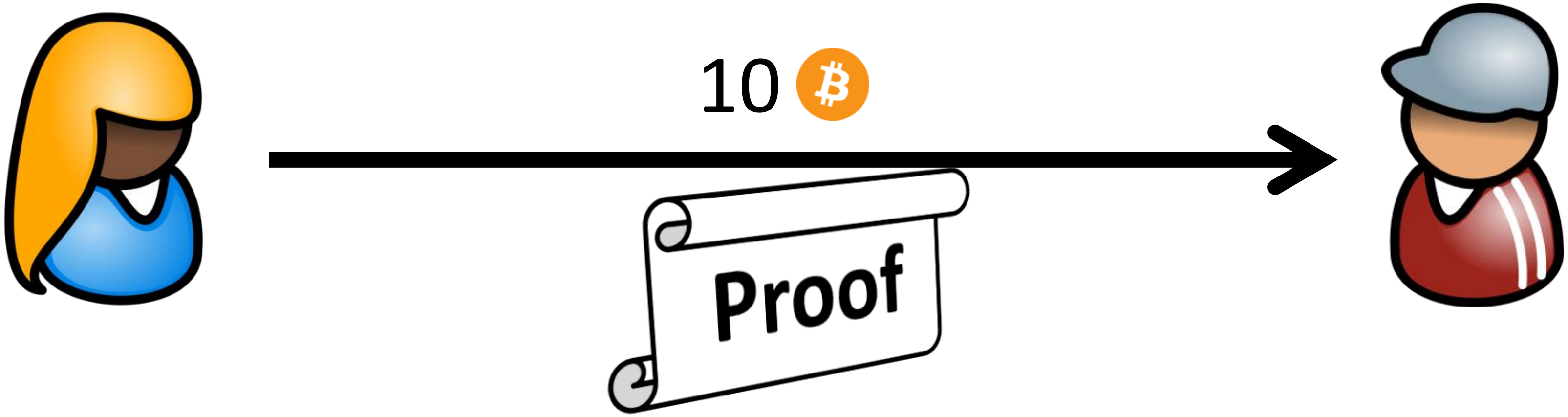
Intuition: “virtual accountant” using cryptographic proofs.

10k ft view: integrity when all is hidden



Intuition: “virtual accountant” using cryptographic proofs.

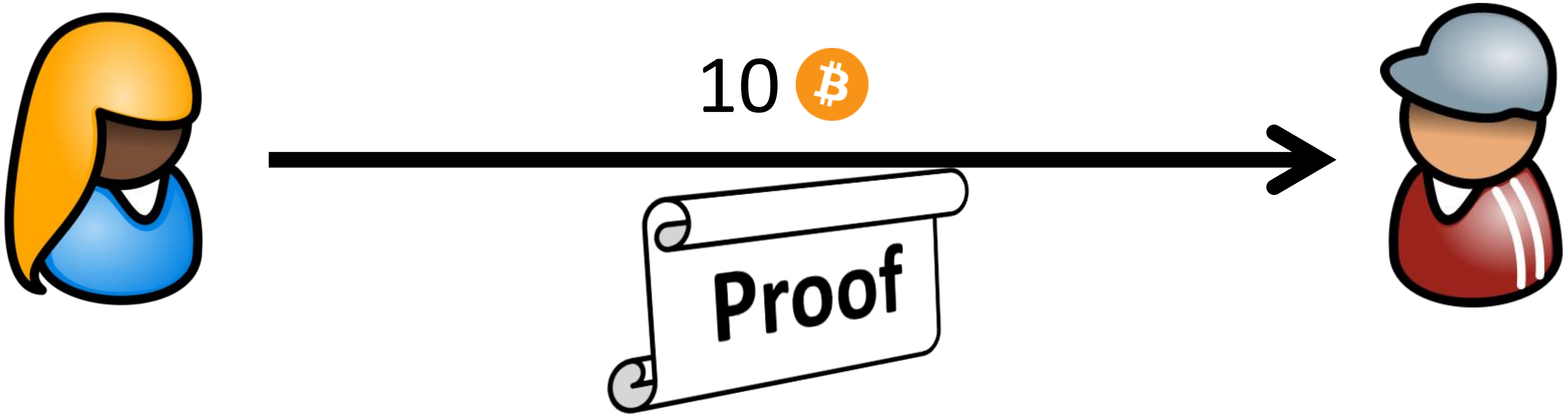
What kind of proof?



Intuition: “virtual accountant” using cryptographic proofs.

What kind of proof?

proof

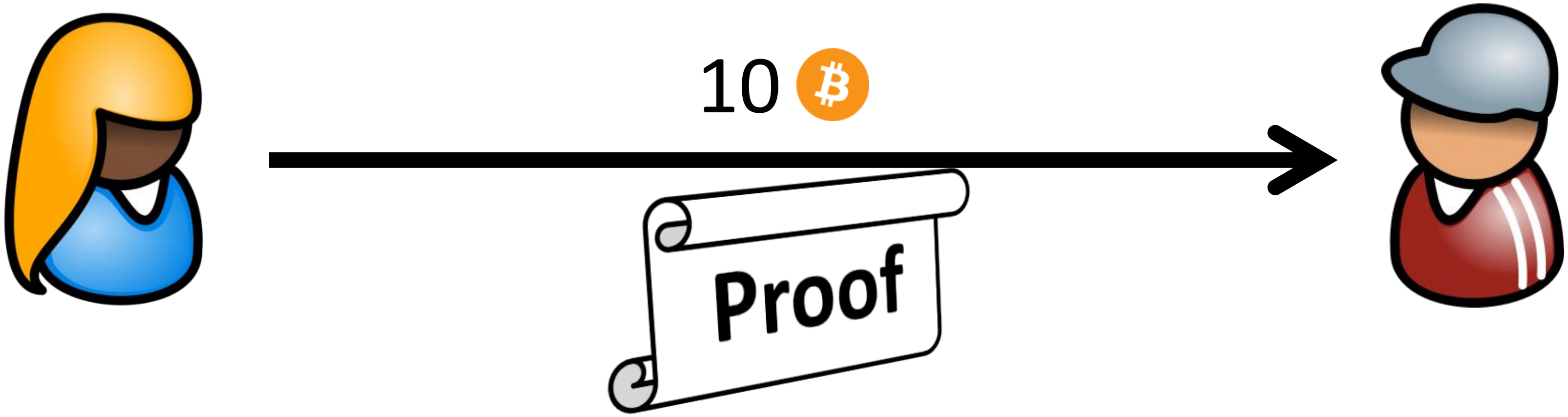


Intuition: “virtual accountant” using cryptographic proofs.

What kind of proof?

zero knowledge

proof

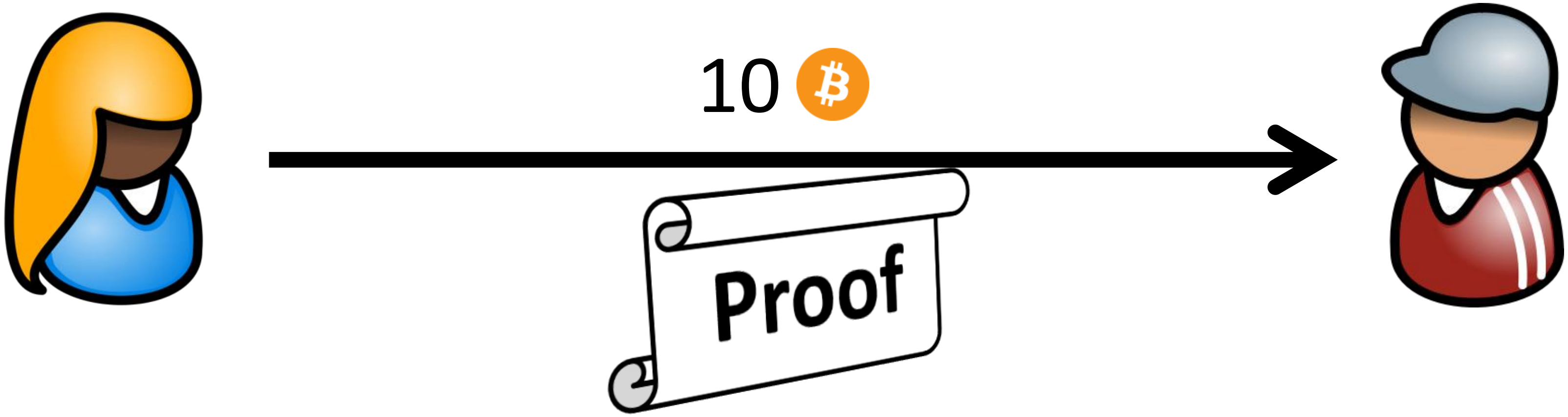


Intuition: “virtual accountant” using cryptographic proofs.

What kind of proof?

zero **k**nowledge

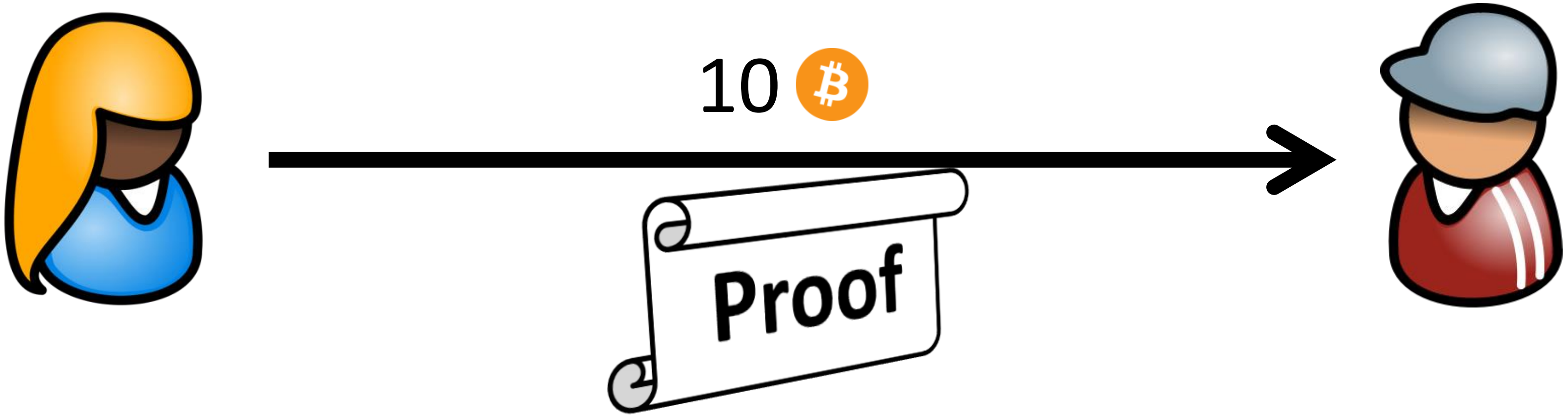
non-interactive proof



Intuition: “virtual accountant” using cryptographic proofs.

What kind of proof?

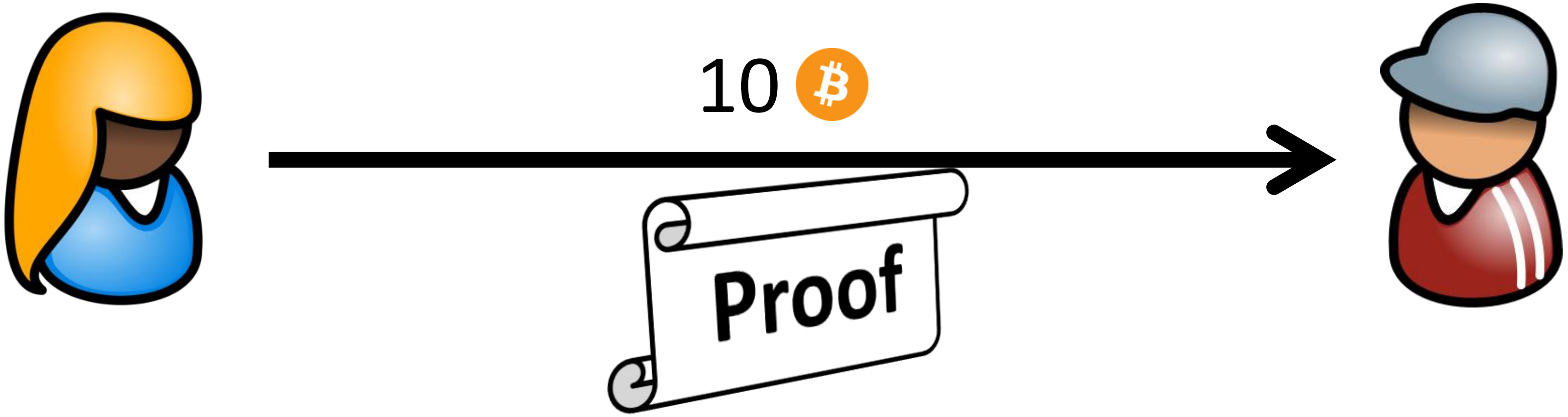
zero **k**nowledge
succinct
non-interactive
proof



Intuition: “virtual accountant” using cryptographic proofs.

What kind of proof?

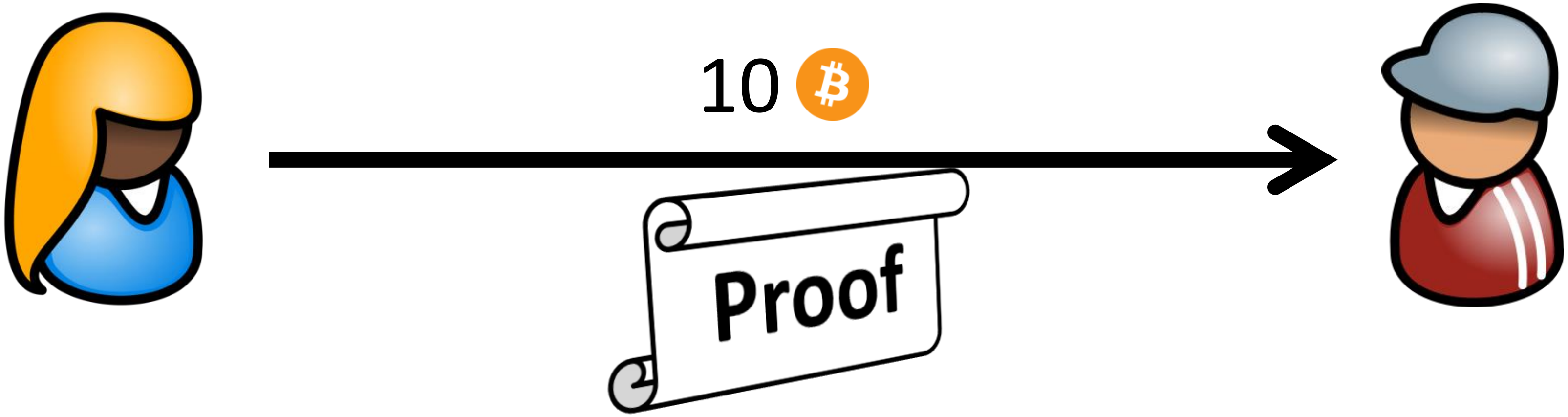
zero **k**nowledge
succinct
non-interactive
proof
of **k**nowledge



Intuition: “virtual accountant” using cryptographic proofs.

What kind of proof?

zero **k**nowledge
succinct
non-interactive
argument
of **k**nowledge

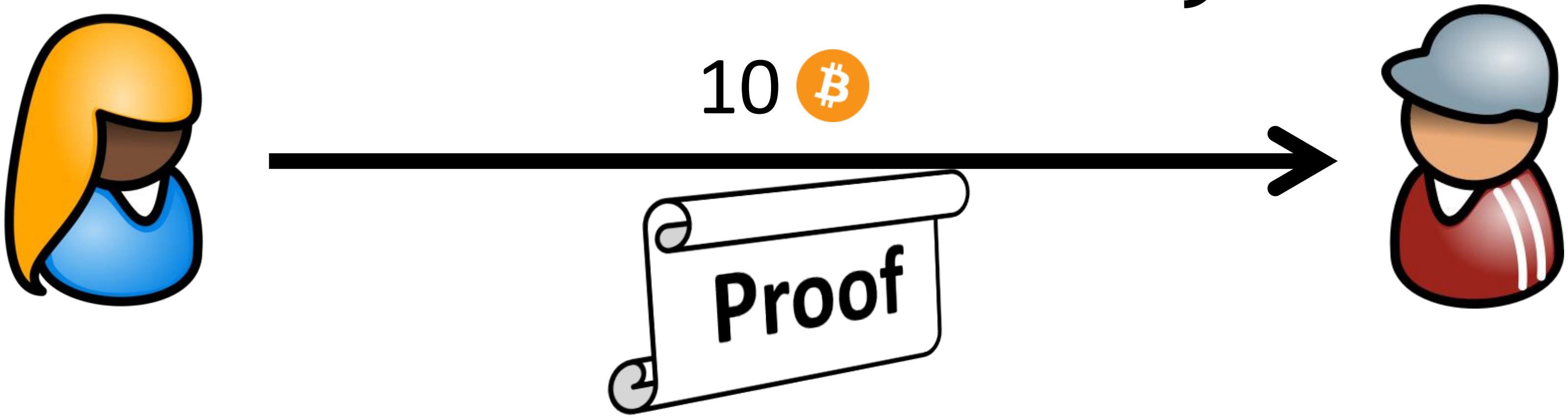


Intuition: “virtual accountant” using cryptographic proofs.

What kind of proof?

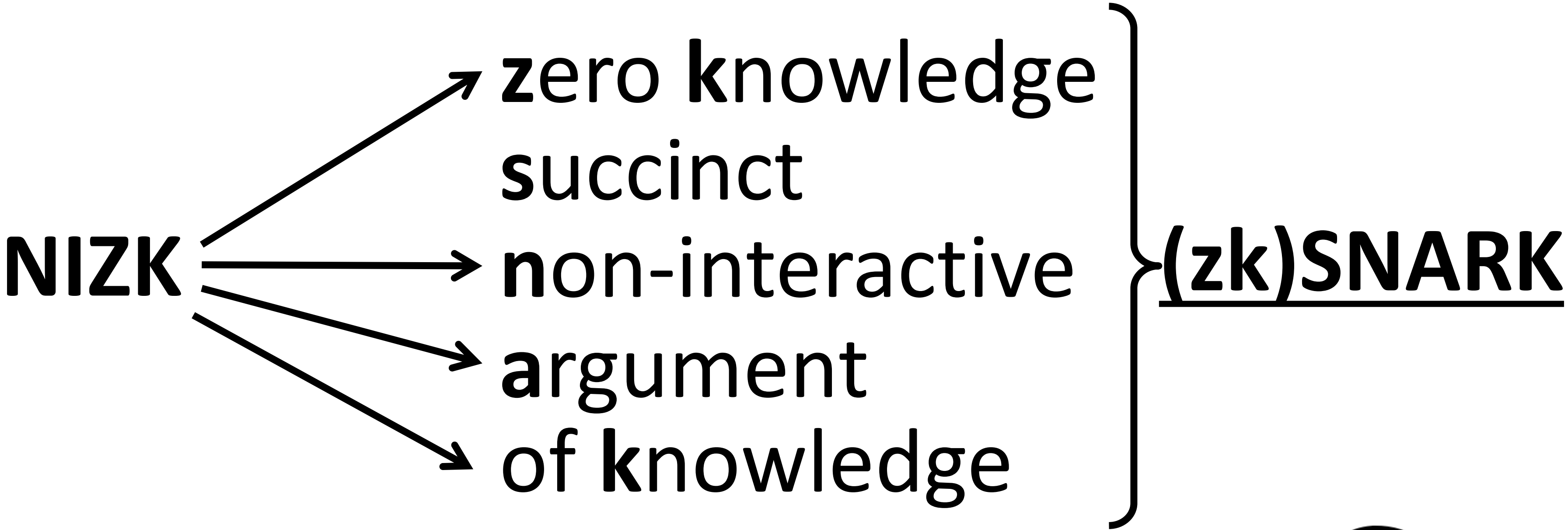
zero **k**nowledge
succinct
non-interactive
argument
of **k**nowledge

(zk)SNARK



Intuition: “virtual accountant” using cryptographic proofs.

What kind of proof?



NIZK

zero **k**nowledge

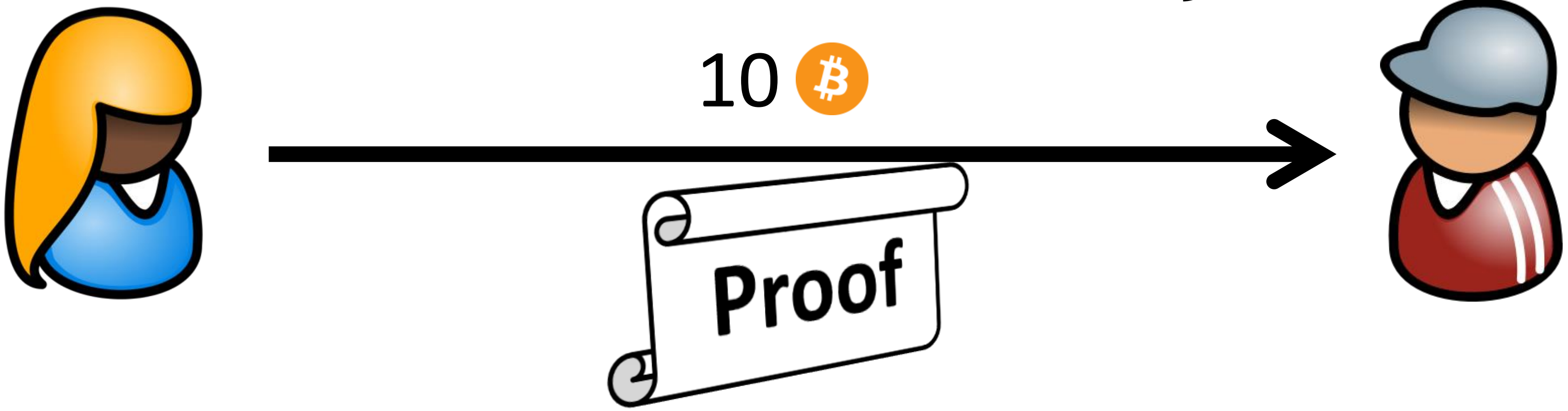
succinct

non-interactive

argument

of **k**nowledge

(zk)SNARK

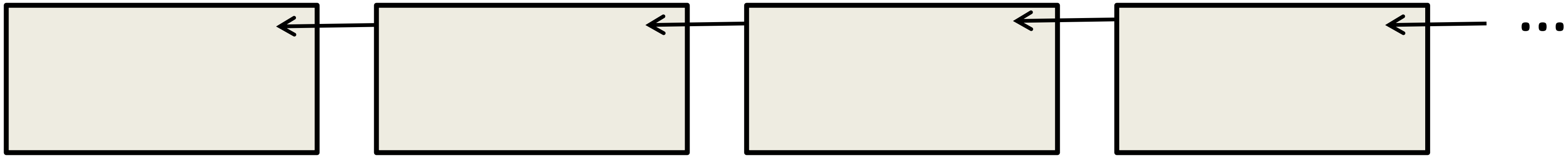


Intuition: “virtual accountant” using cryptographic proofs.

Architecture overview

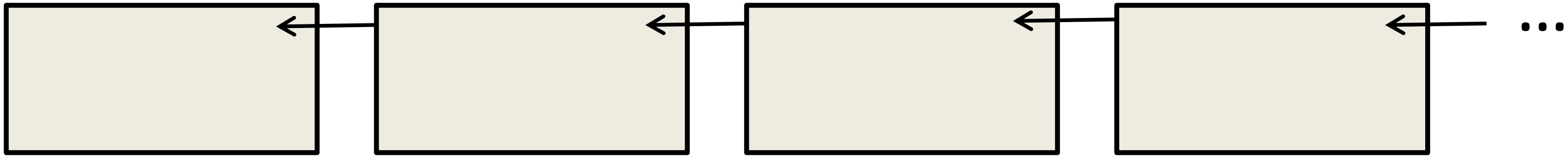
Architecture overview

Public currency ledger (e.g. Bitcoin):



Architecture overview

Public currency ledger (e.g. Bitcoin):

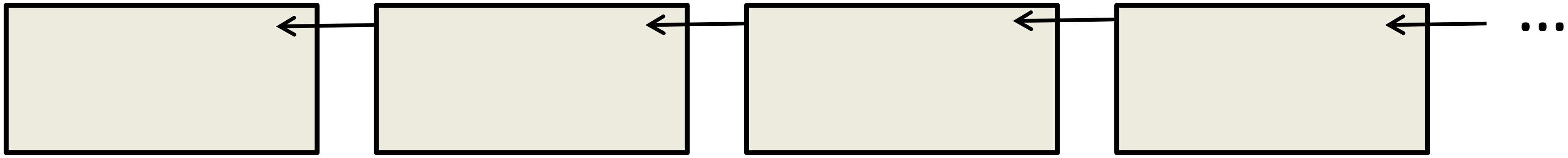


+

Two new transaction types:

Architecture overview

Public currency ledger (e.g. Bitcoin):



+

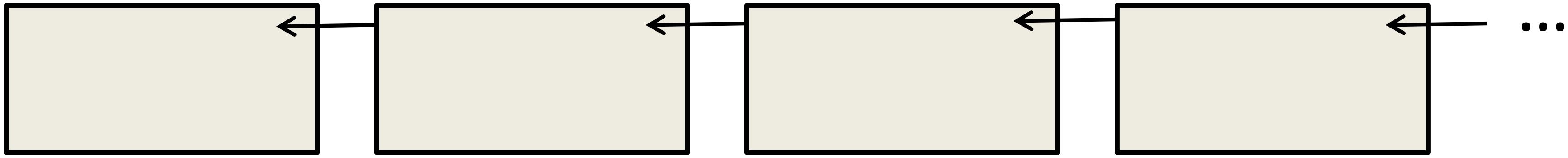
Two new transaction types:

Mint

Spend

Architecture overview

Public currency ledger (e.g. Bitcoin):



+

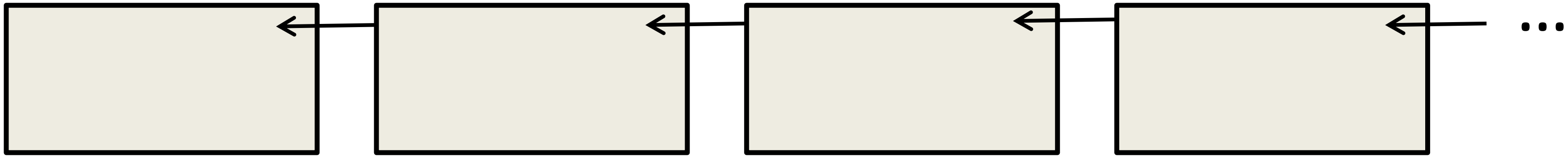
Two new transaction types:

Mint - Consumes (and destroys) Bitcoin
- Creates private coin

Spend

Architecture overview

Public currency ledger (e.g. Bitcoin):



+

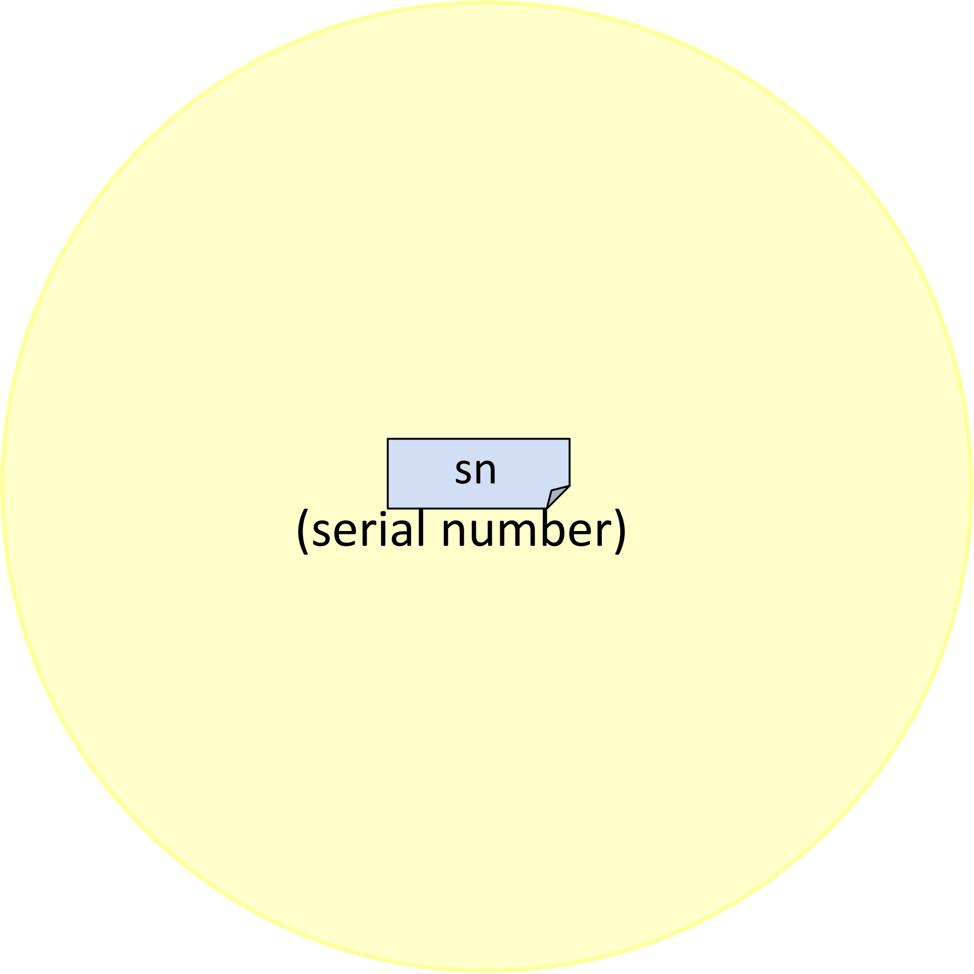
Two new transaction types:

Mint - Consumes (and destroys) Bitcoin
- Creates private coin

Spend - Consumes (and destroys) a private coin
- Creates Bitcoin

Basic anonymous e-cash

Basic anonymous e-cash

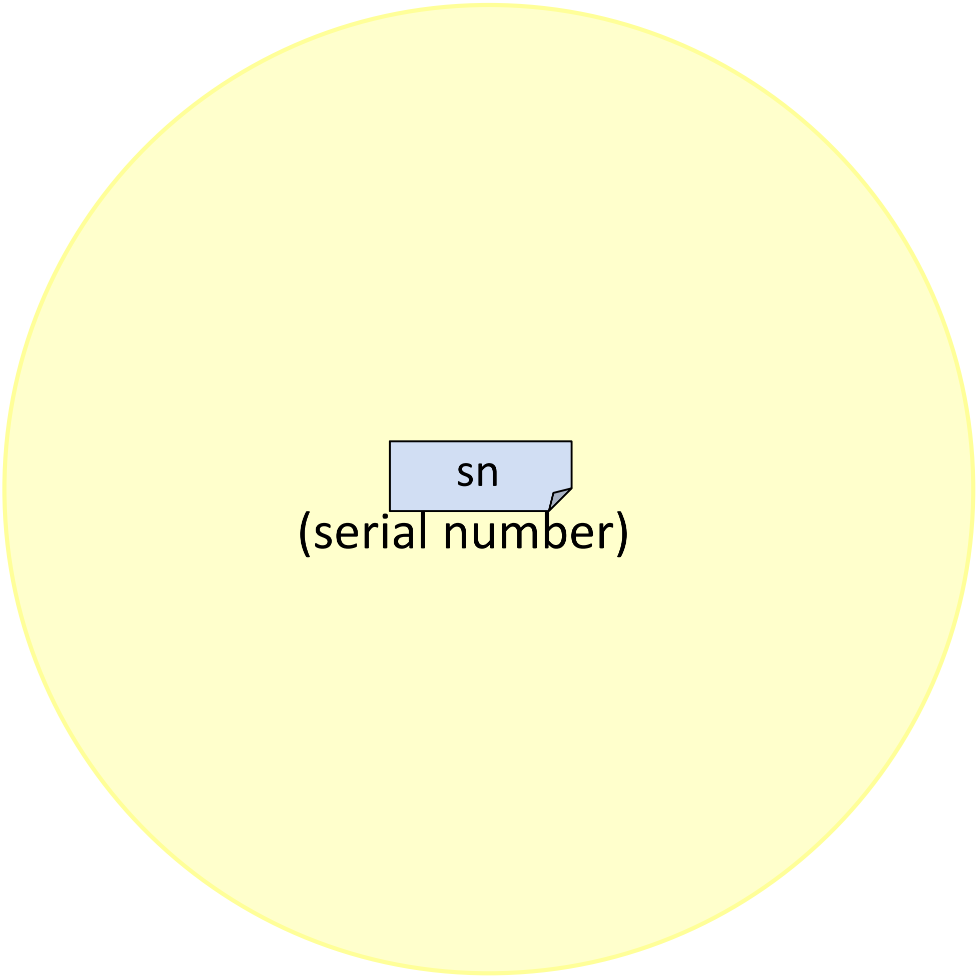


sn

(serial number)

Basic anonymous e-cash

Minting: I hereby spend 1 BTC to create sn

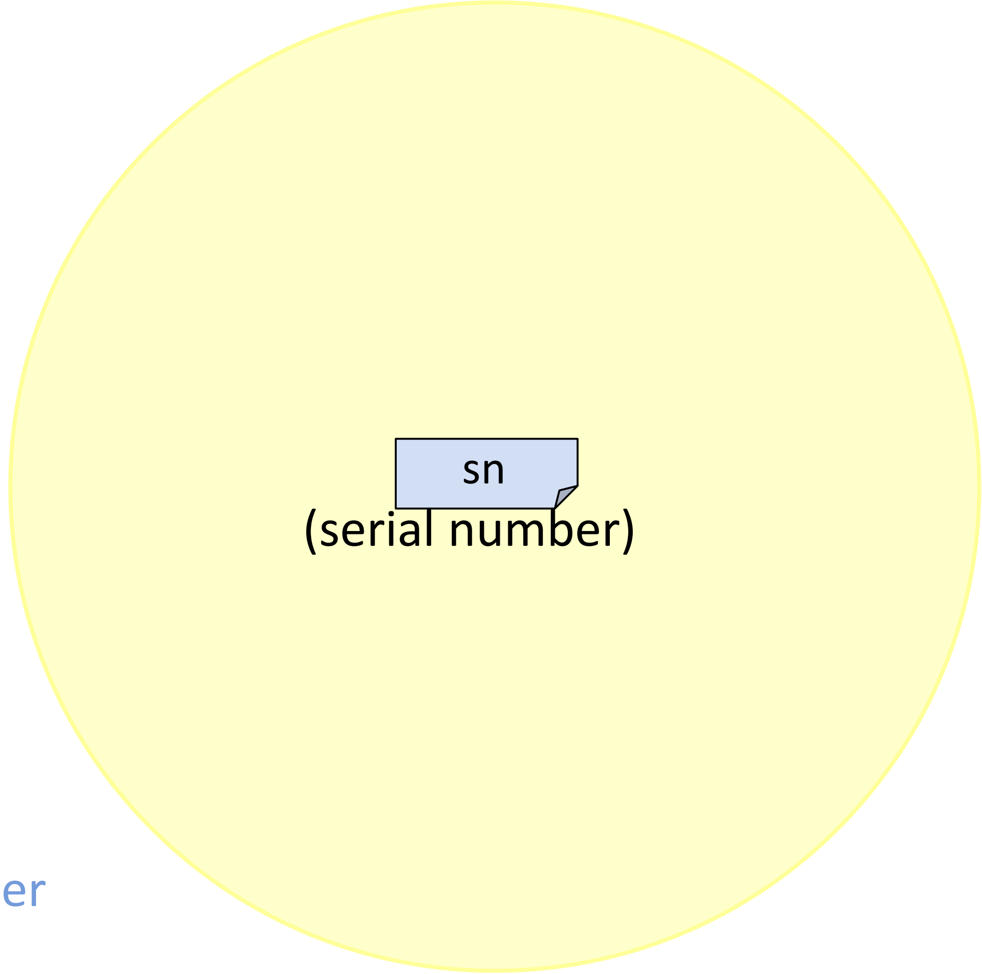


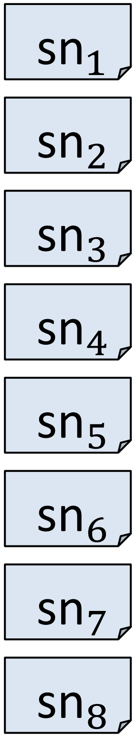
sn

(serial number)

Basic anonymous e-cash

Minting: I hereby spend 1 BTC to create sn





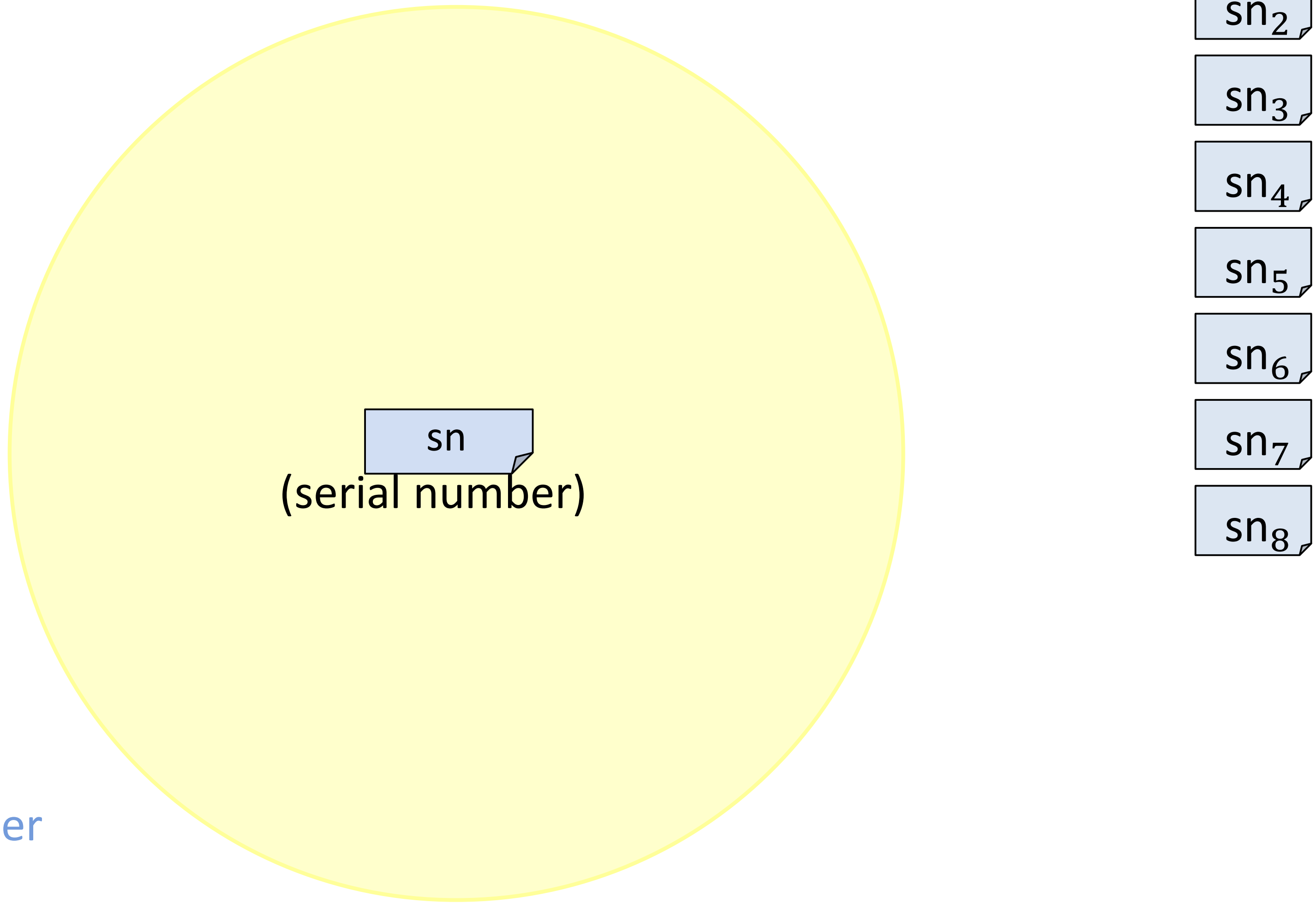
Legend:

▭ In public ledger

Basic anonymous e-cash

Minting: I hereby spend 1 BTC to create sn

Spending: I’m using up a coin with (unique) sn



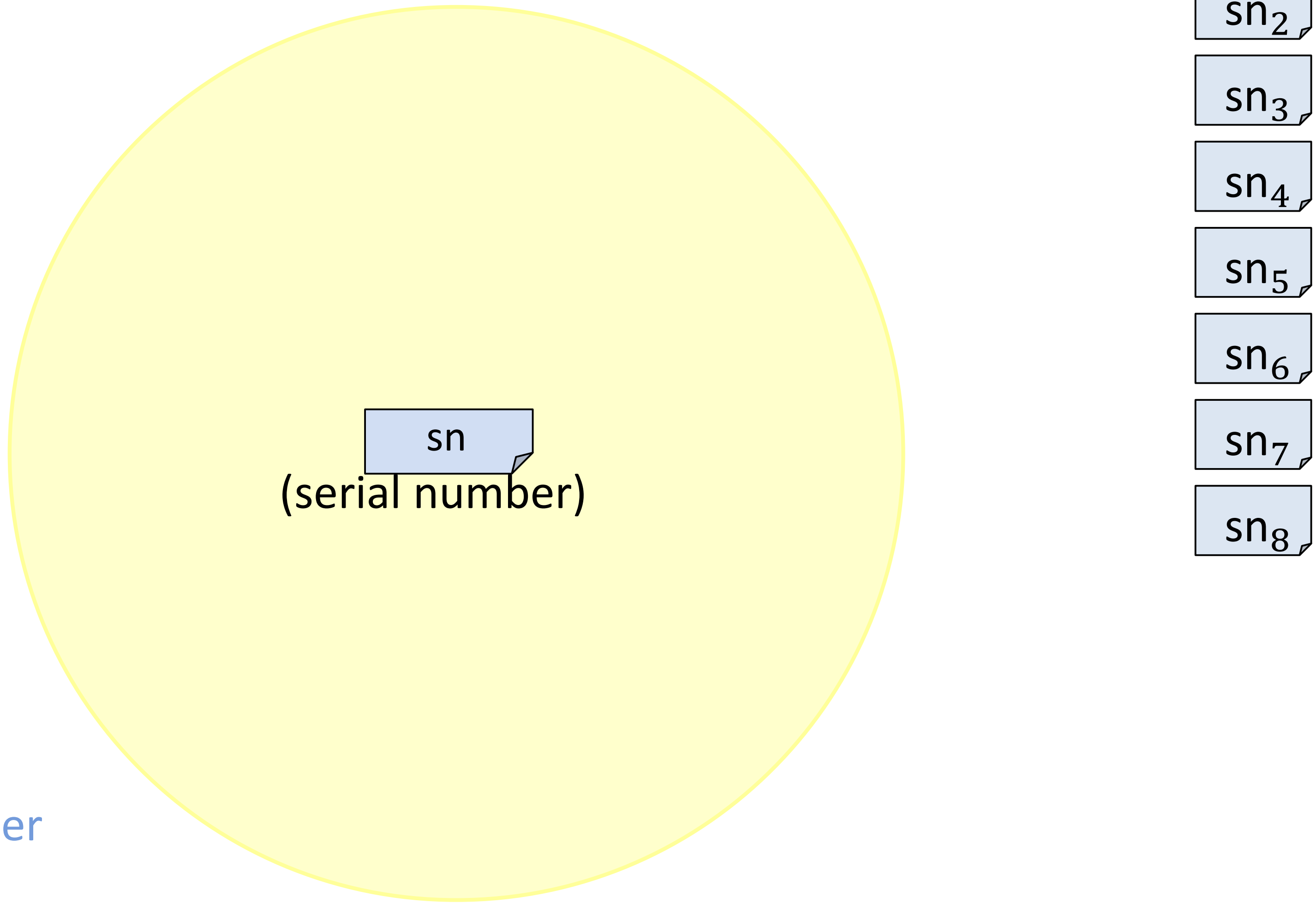
Legend:

▨ In public ledger

Basic anonymous e-cash

Minting: I hereby spend 1 BTC to create sn

Spending: ~~I’m using up a coin with (unique) sn~~



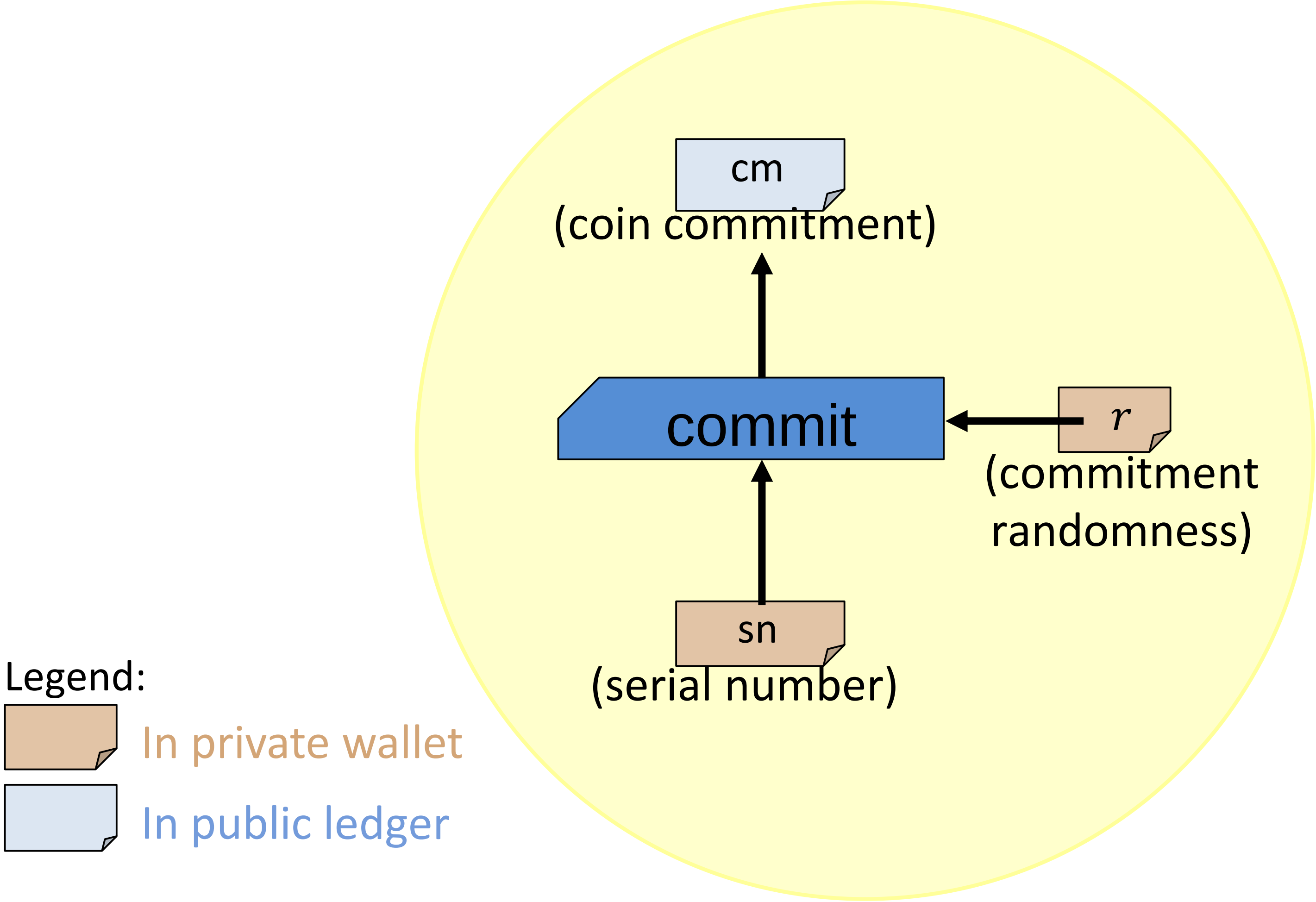
sn
(serial number)

sn_1
sn_2
sn_3
sn_4
sn_5
sn_6
sn_7
sn_8

Legend:

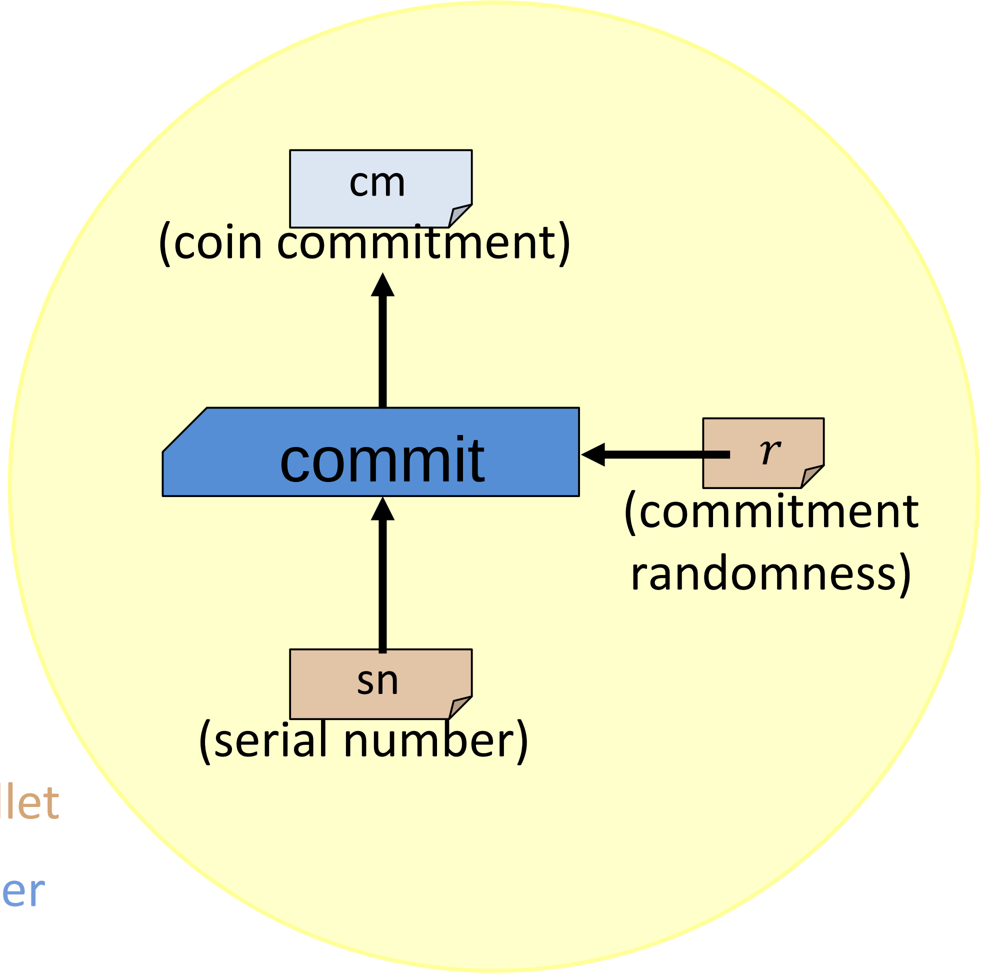
▭ In public ledger

Basic anonymous e-cash [Sander Ta-Shma 1999]



Basic anonymous e-cash [Sander Ta-Shma 1999]

Minting: I hereby spend 1 BTC to create cm



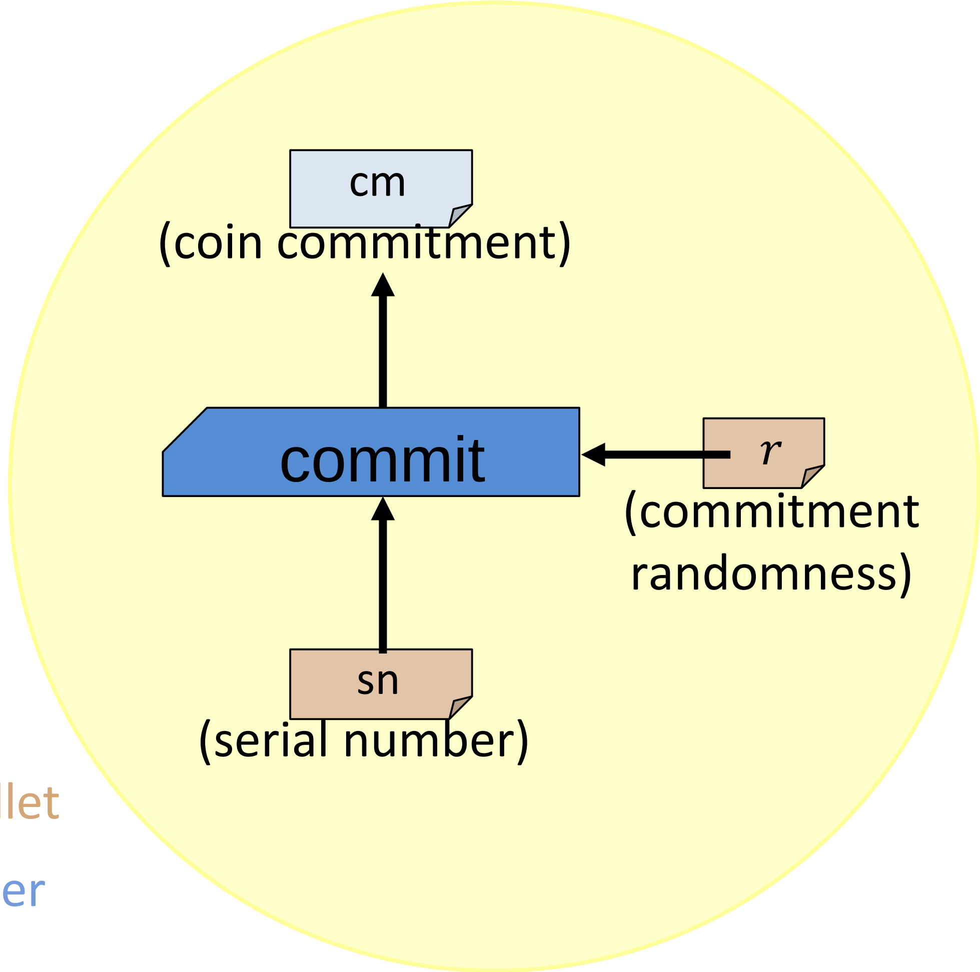
Legend:

🟫 In private wallet

🟦 In public ledger

Basic anonymous e-cash [Sander Ta-Shma 1999]

Minting: I hereby spend 1 BTC to create cm



cm_1

cm_2

cm_3

cm_4

cm_5

cm_6

cm_7

cm_8

Legend:

🟫 In private wallet

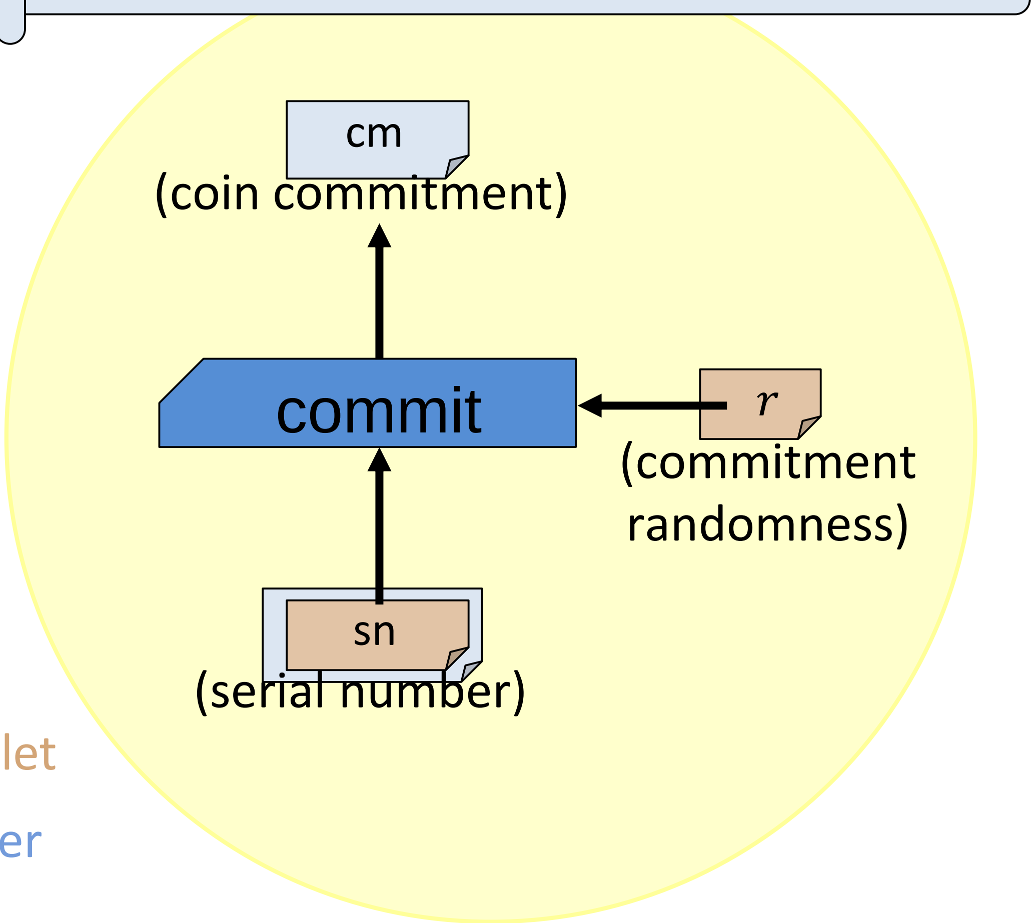
🟦 In public ledger

Basic anonymous e-cash

[Sander Ta-Shma 1999]

Minting: I hereby spend 1 BTC to create cm

Spending: I’m using up a coin with (unique) sn, and here are its cm and r.



cm_1

cm_2

cm_3

cm_4

cm_5

cm_6

cm_7

cm_8

Legend:

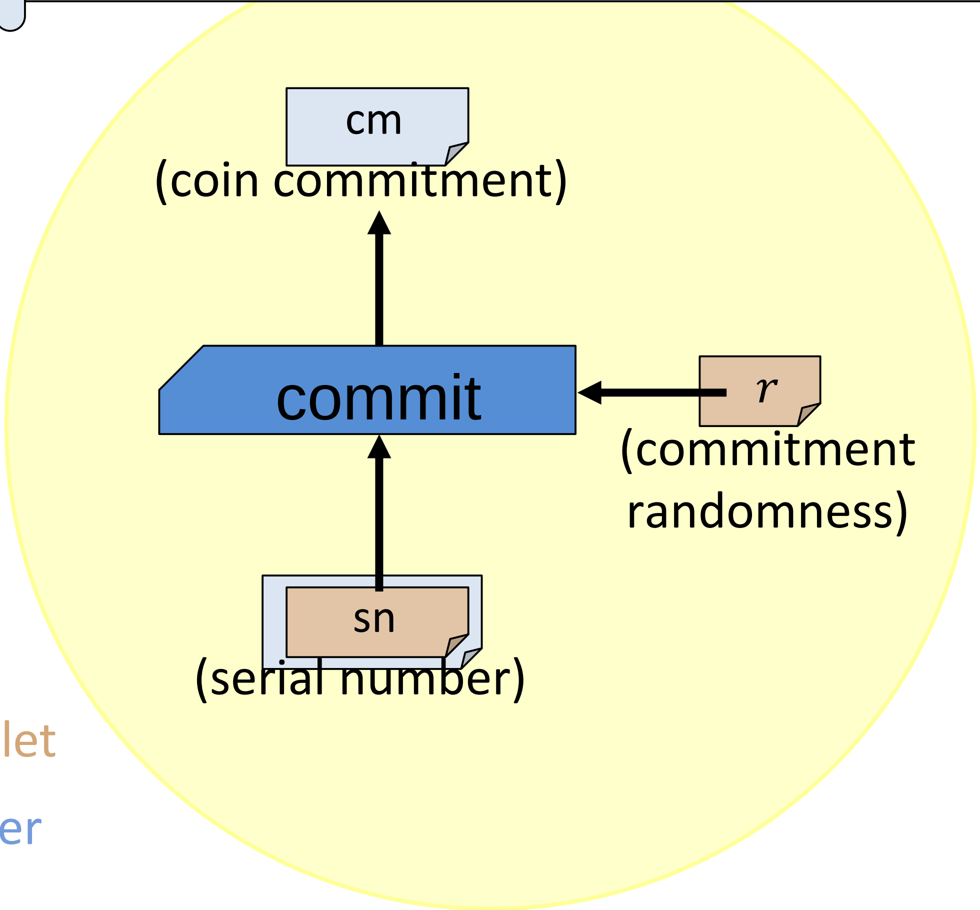
🟫 In private wallet

🟦 In public ledger

Basic anonymous e-cash [Sander Ta-Shma 1999]

Minting: I hereby spend 1 BTC to create cm

Spending: ~~I'm using up a coin with (unique) sn,~~ and here are its cm and r.



cm_1

cm_2

cm_3

cm_4

cm_5

cm_6

cm_7

cm_8

Legend:

🟫 In private wallet

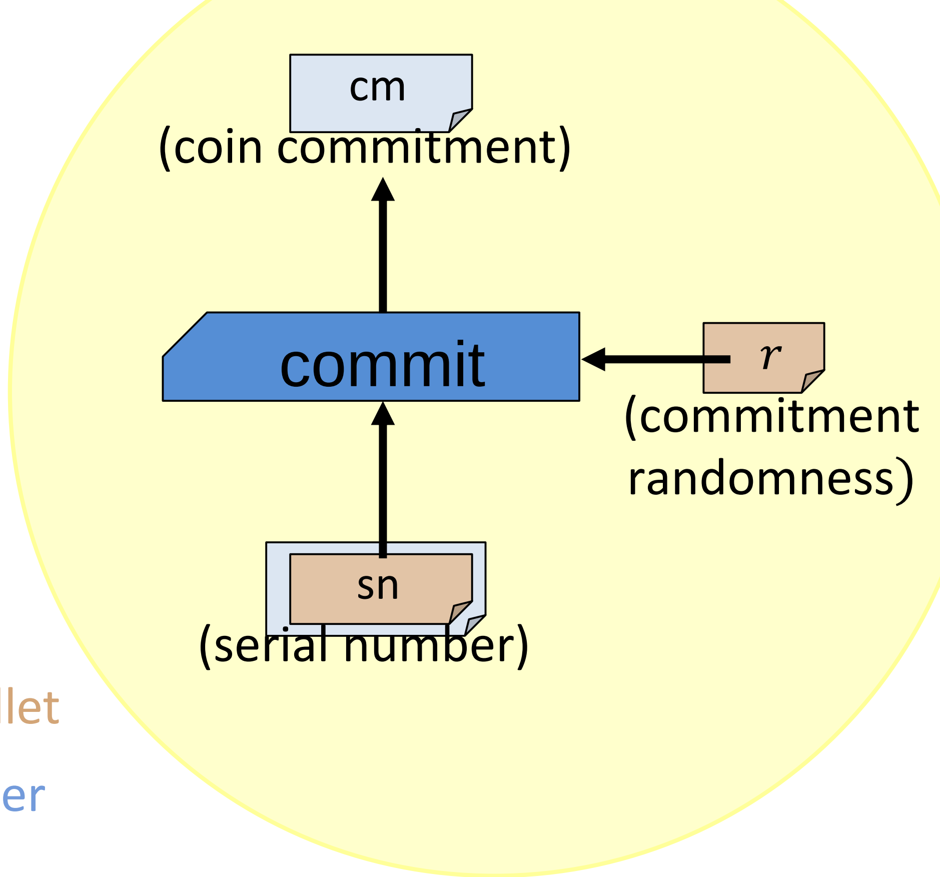
🟦 In public ledger

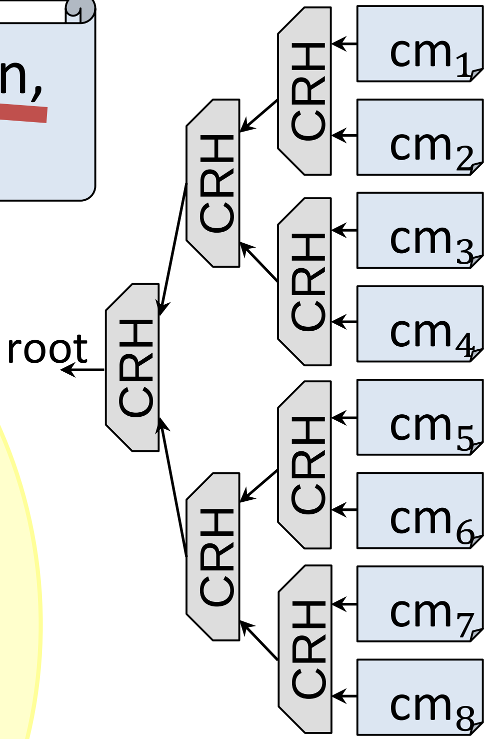
Basic anonymous e-cash

[Sander Ta-Shma 1999]

Minting: I hereby spend 1 BTC to create cm

Spending: ~~I'm using up a coin with (unique) sn,~~ and here are its cm and r.





Legend:

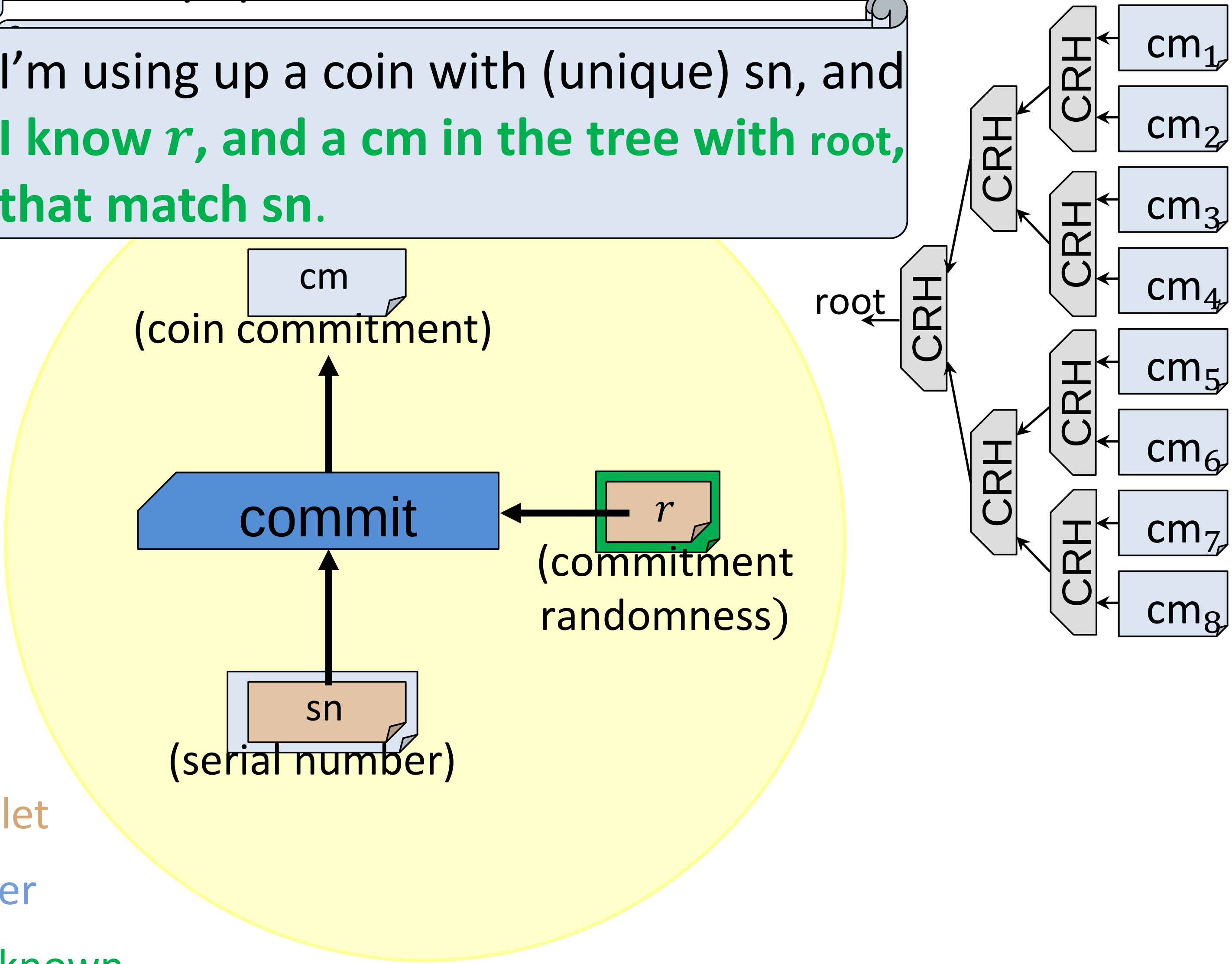
🟫 In private wallet

🟦 In public ledger

Basic anonymous e-cash [Sander Ta-Shma 1999]

Minting: I hereby spend 1 BTC to create cm

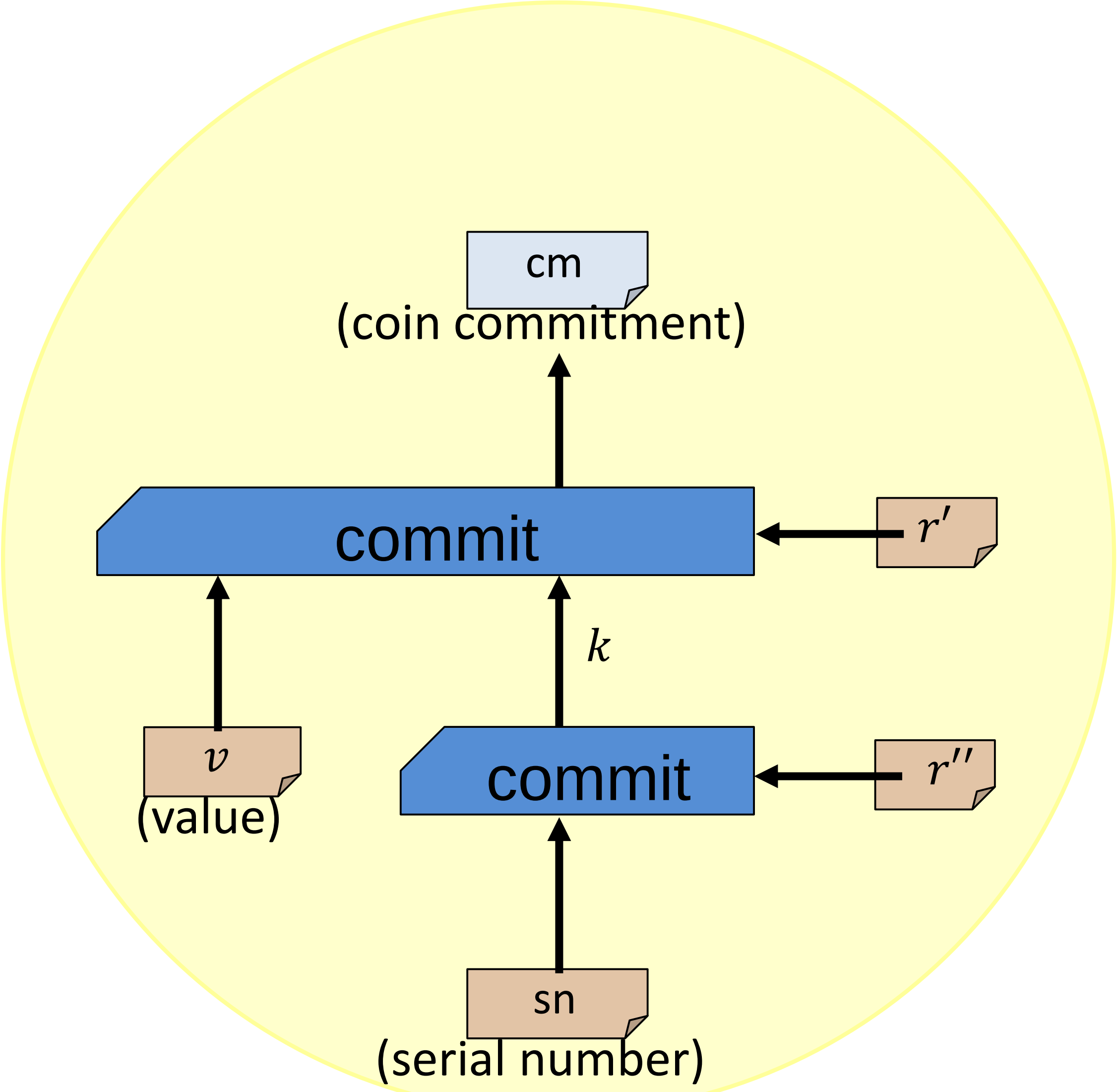
Spending: I'm using up a coin with (unique) sn, and **I know r, and a cm in the tree with root, that match sn.**



Legend:

- ▨ In private wallet
- ▢ In public ledger
- 🟩 Proved to be known

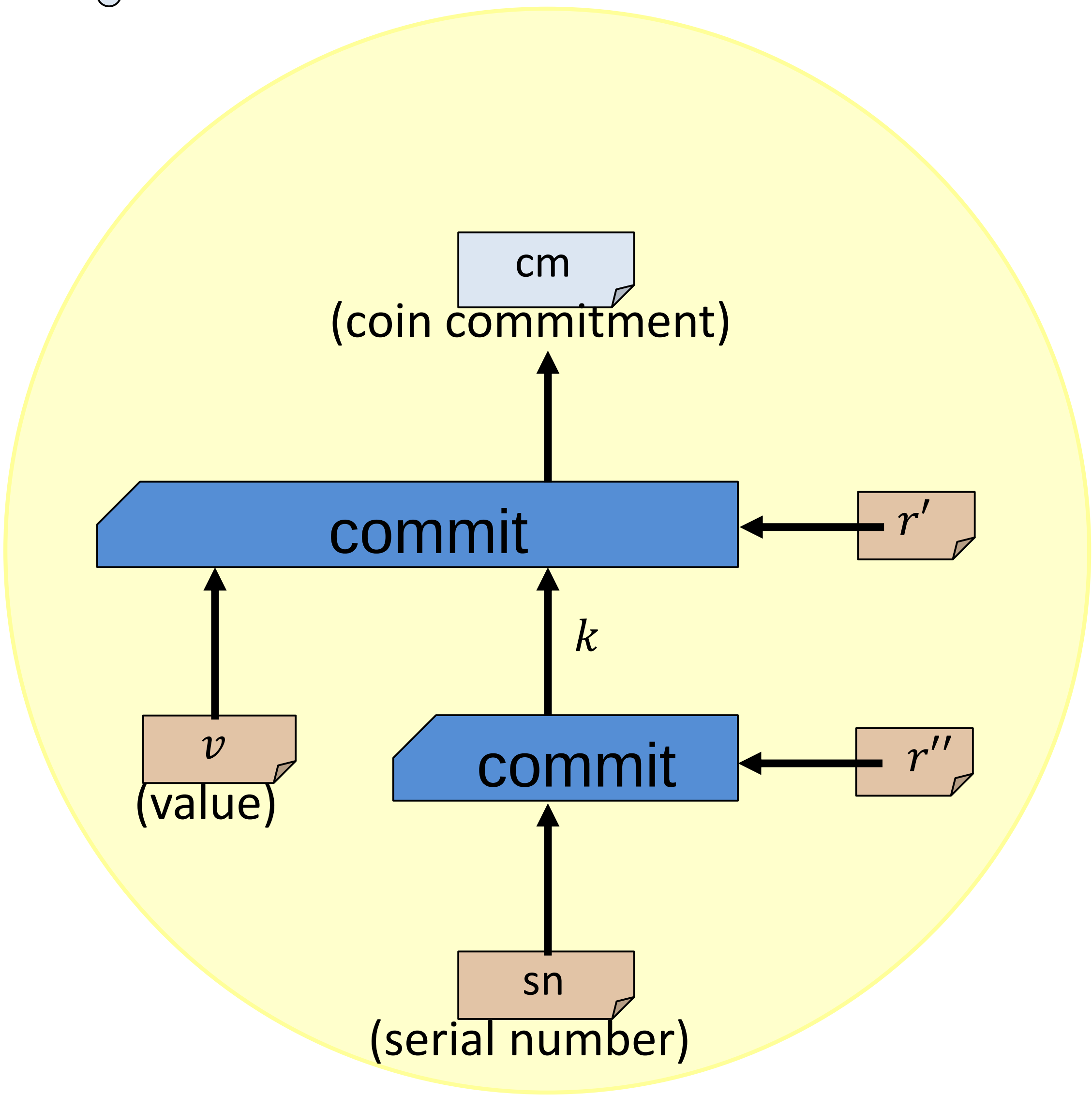
Adding variable denomination



Adding variable denomination

Minting:

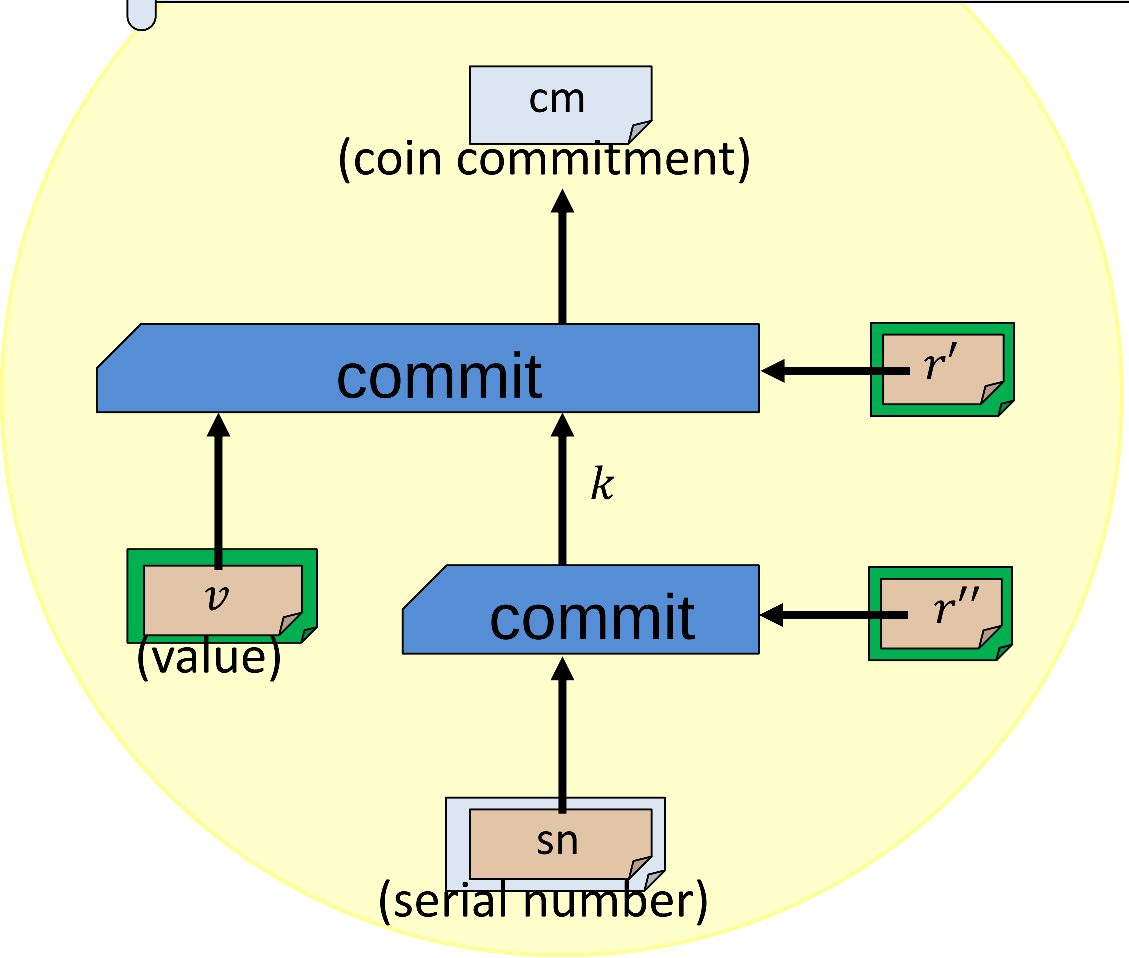
I hereby spend v BTC to create cm,
and here is k, r' to prove consistency.



Adding variable denomination

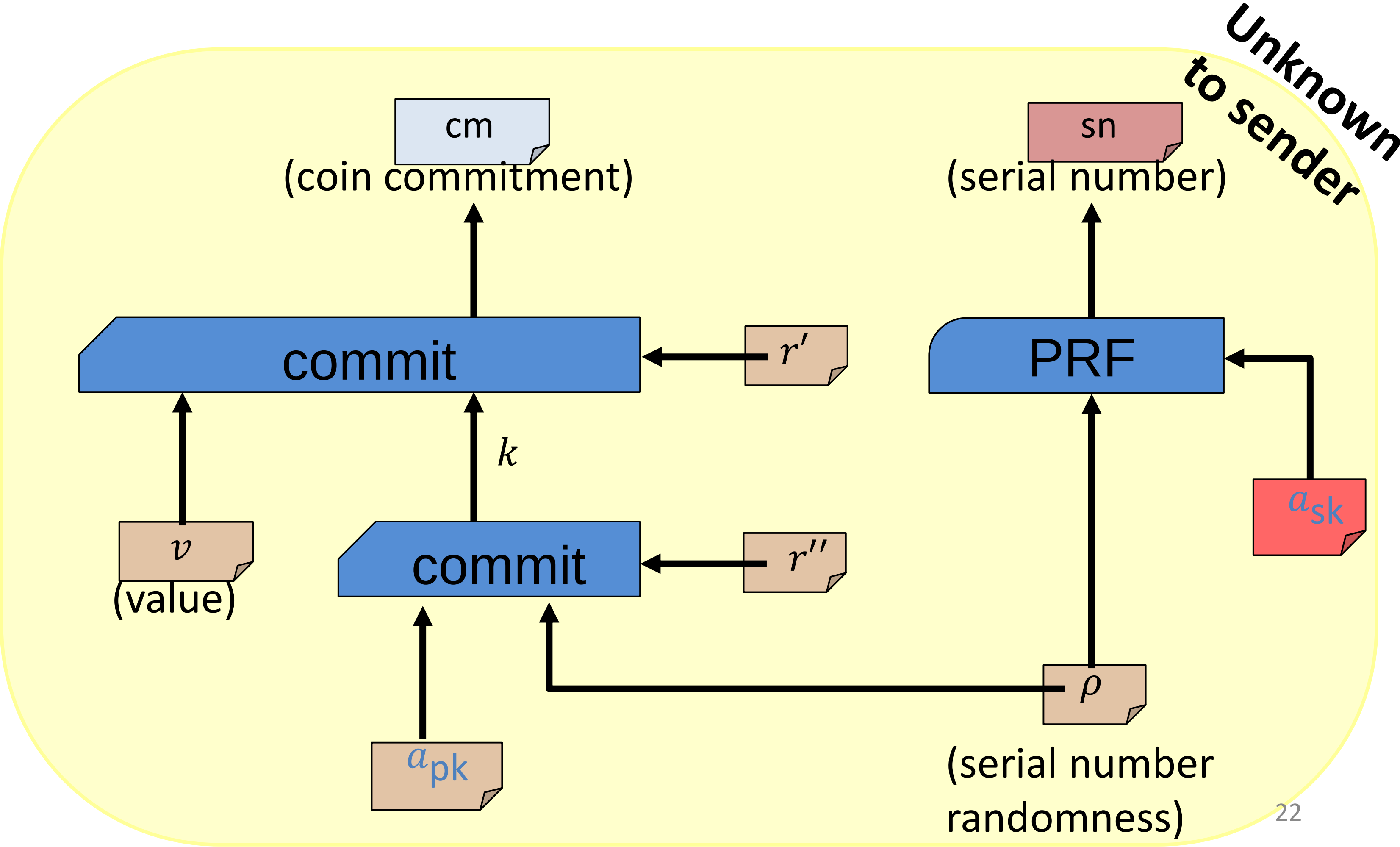
Minting: I hereby spend v BTC to create cm, and here is k, r' to prove consistency.

Spending: I'm using up a coin with value v (unique) sn, and **I know r', r'' that are consistent with cm.**



Adding direct anonymous payments

CreateAddress: recipient creates $a_{\text{pk}}, a_{\text{sk}}$

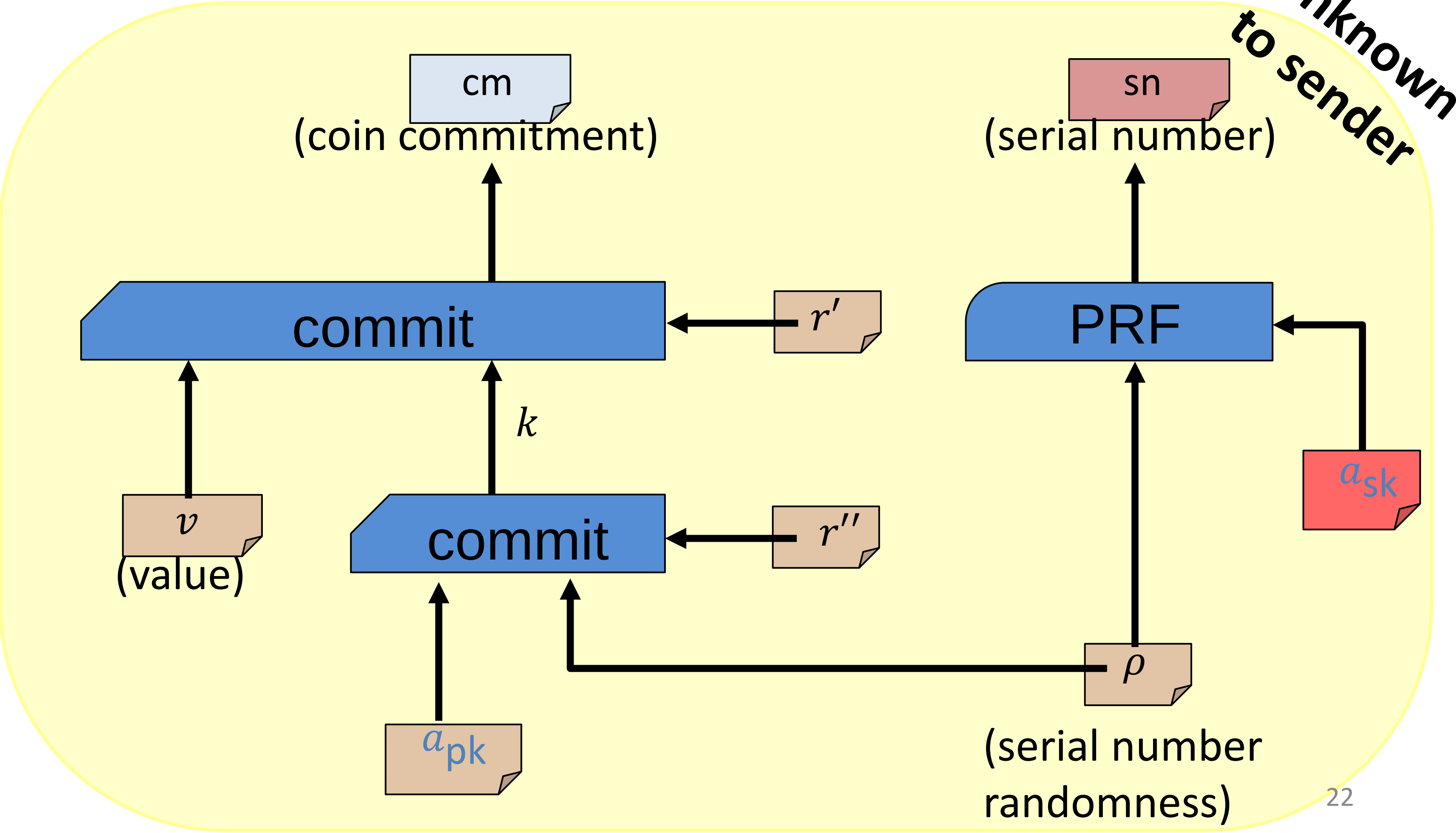


Adding direct anonymous payments

CreateAddress: recipient creates $a_{\text{pk}}, a_{\text{sk}}$

Minting, spending analogous to above.

I'm using up a coin with value v (unique) sn, and **I know $r', r'', \rho, a_{\text{pk}}$ that are consistent with cm.**



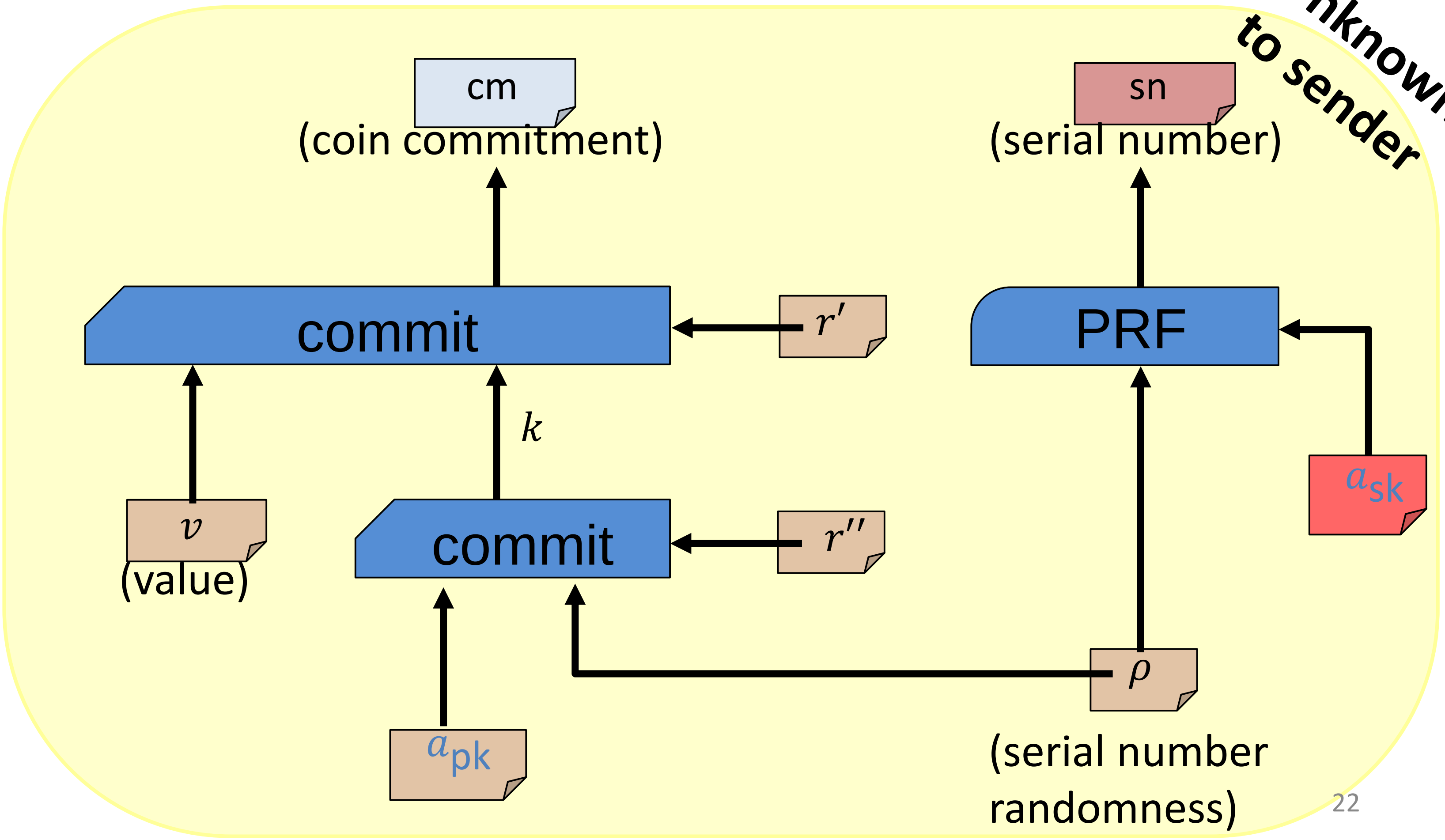
Adding direct anonymous payments

CreateAddress: recipient creates $a_{\text{pk}}, a_{\text{sk}}$

Minting, spending analogous to above.

I'm using up a coin with value v (unique) sn, and **I know $r', r'', \rho, a_{\text{pk}}$ that are consistent with cm**.

Sending?



Unknown to sender

cm

(coin commitment)

sn

(serial number)

commit

r'

PRF

k

a_{sk}

v

(value)

commit

r''

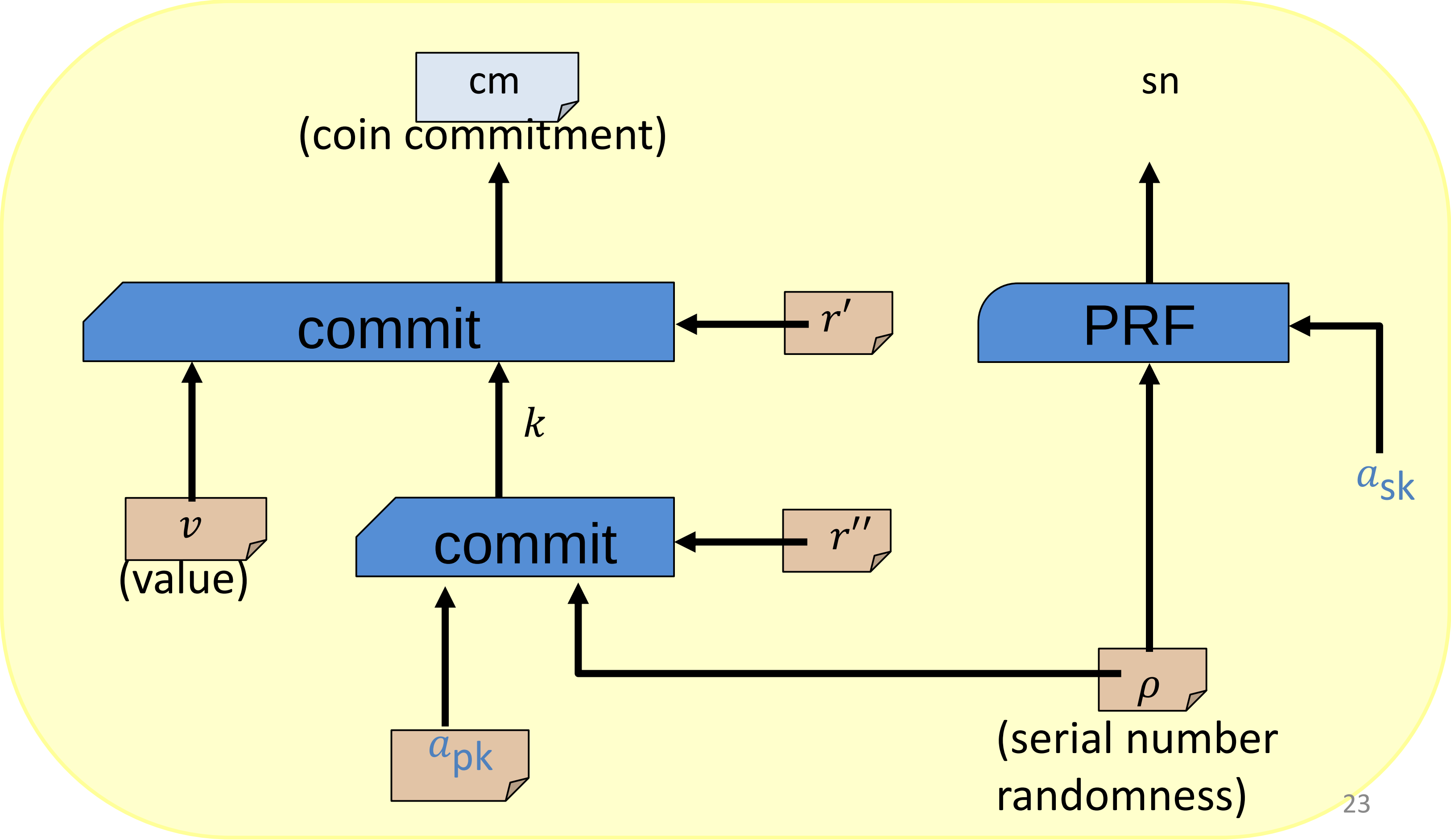
ρ

a_{pk}

(serial number randomness)

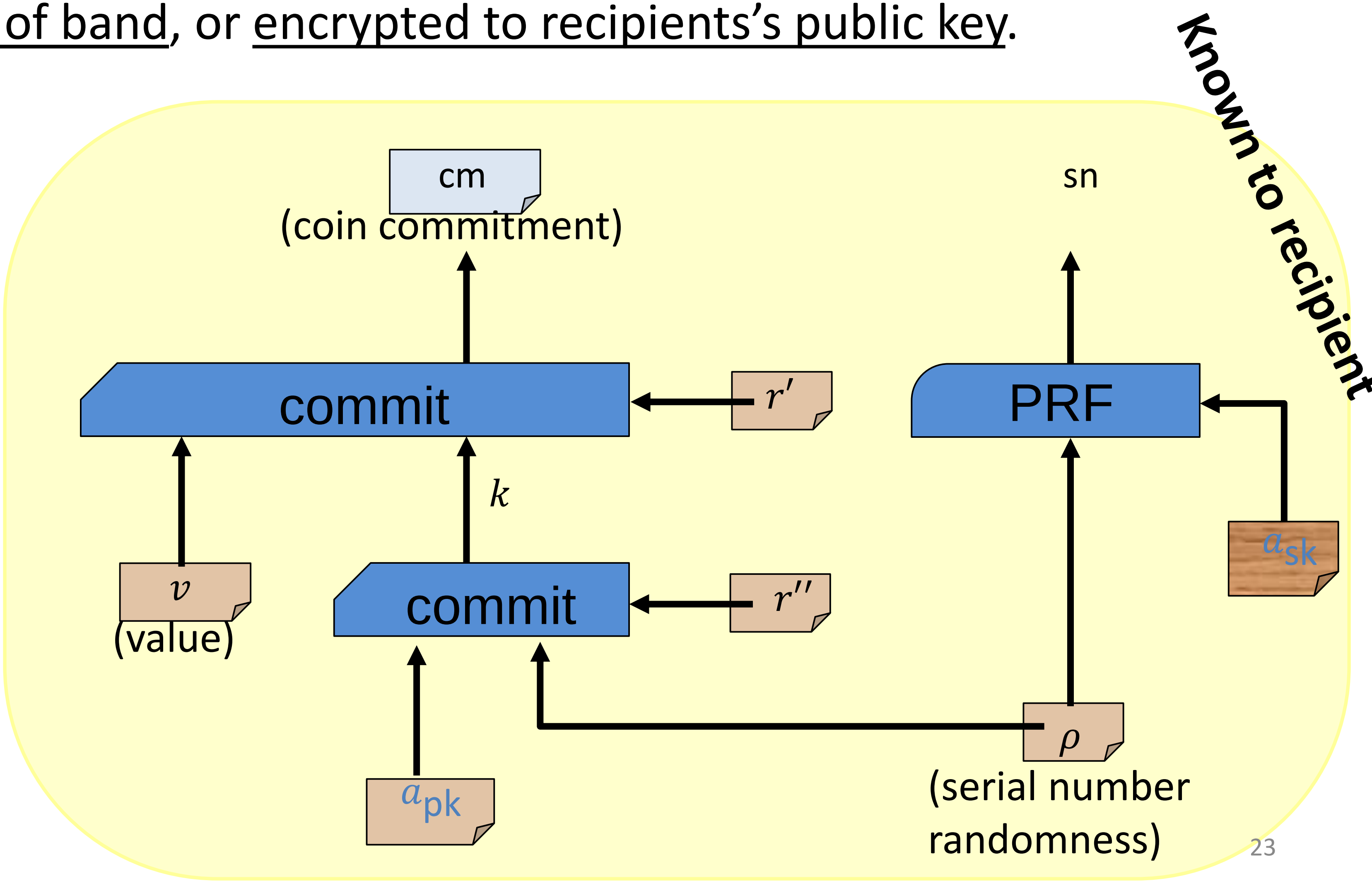
Sending direct anonymous payments

1. Create coin using a_{pk} of recipient.
2. Send coin secrets (v, ρ, r', r'') to recipient
   out of band, or encrypted to recipients's public key.



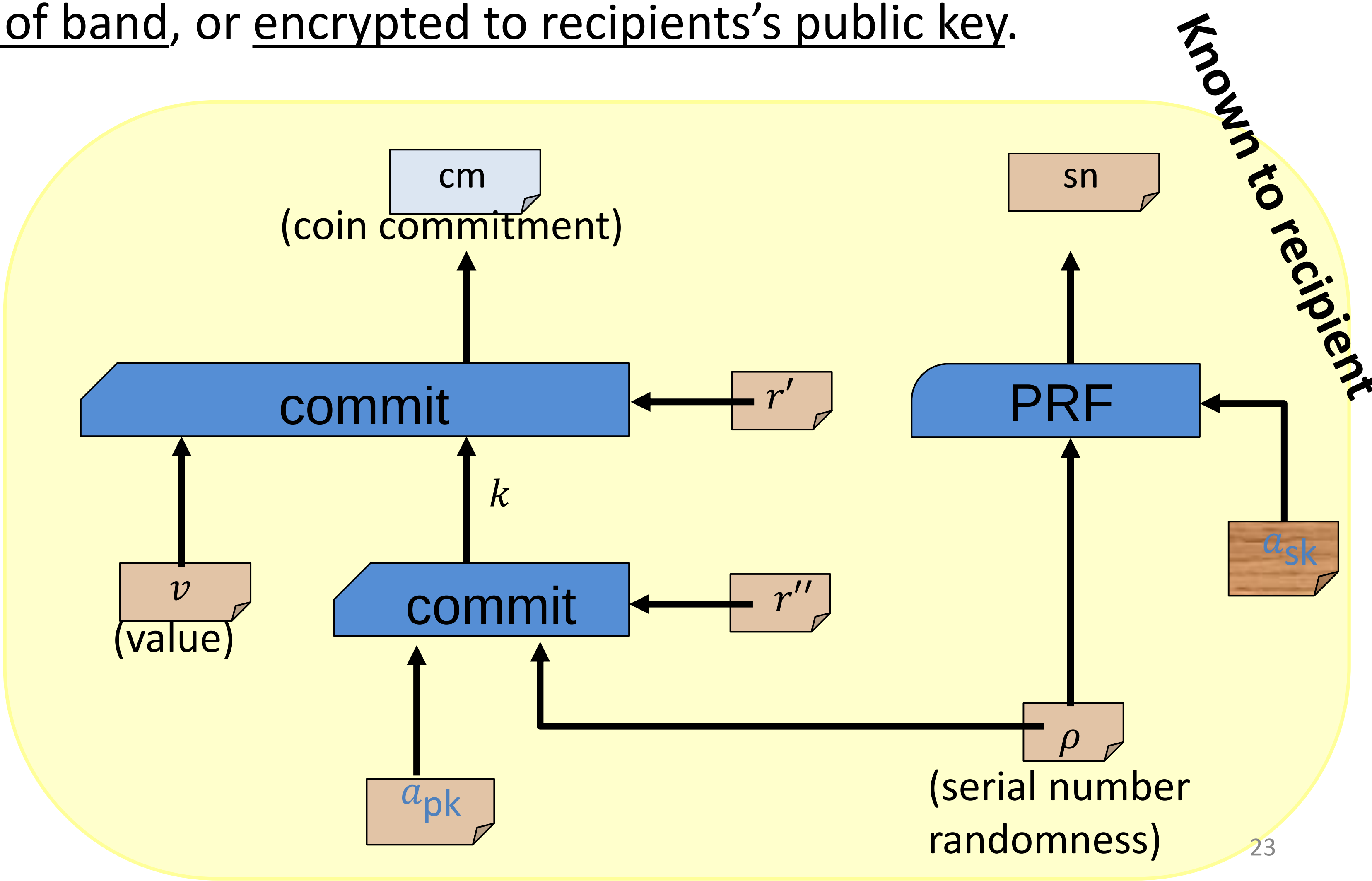
Sending direct anonymous payments

1. Create coin using a_{pk} of recipient.
2. Send coin secrets (v, ρ, r', r'') to recipient
   out of band, or encrypted to recipients's public key.



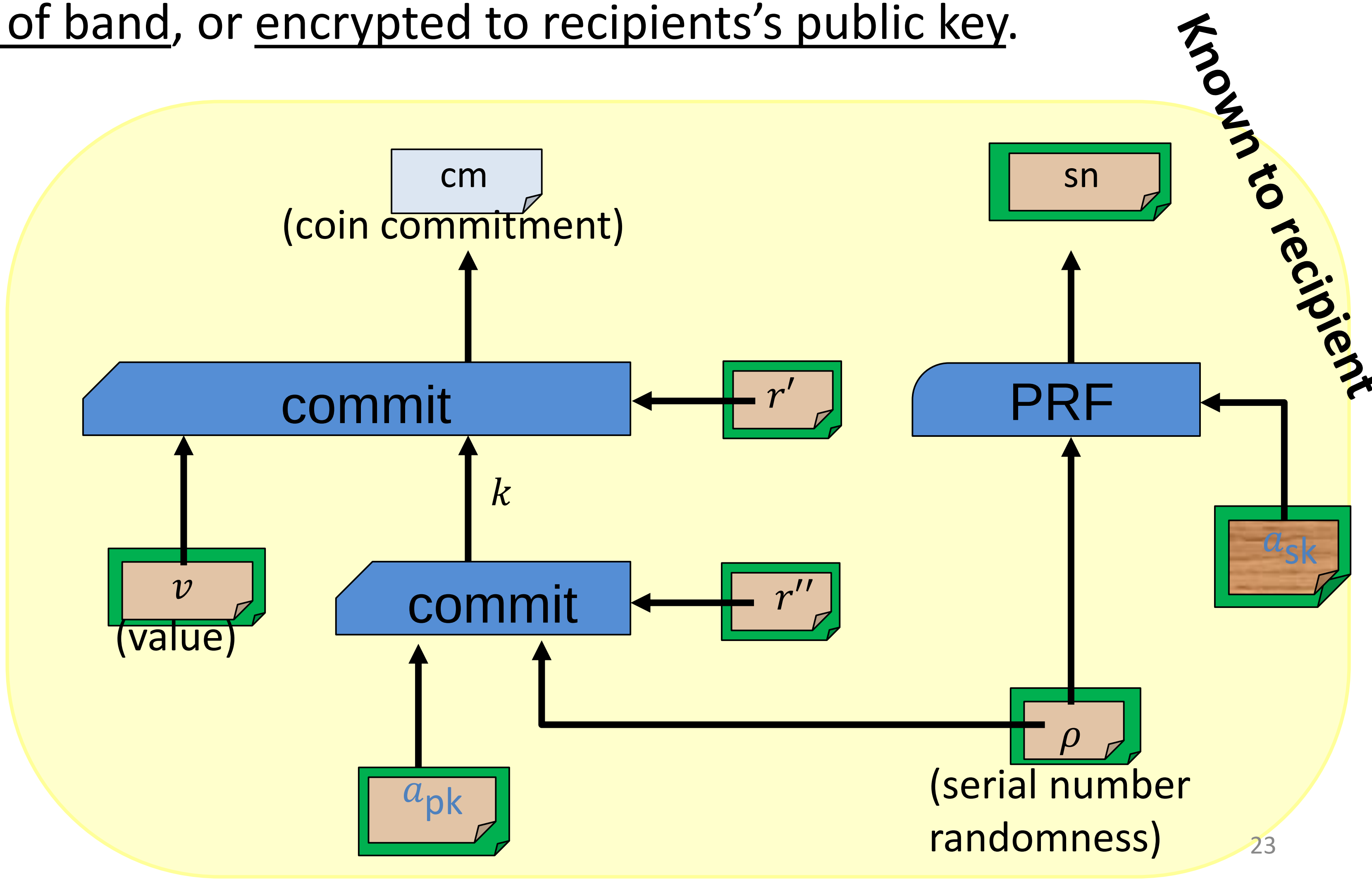
Sending direct anonymous payments

1. Create coin using a_{pk} of recipient.
2. Send coin secrets (v, ρ, r', r'') to recipient
   out of band, or encrypted to recipients's public key.



Sending direct anonymous payments

1. Create coin using a_{pk} of recipient.
2. Send coin secrets (v, ρ, r', r'') to recipient out of band, or encrypted to recipients's public key.



Pouring private coins

Single transaction type capturing:

Sending payments

Making change

Exchanging into BTC

Transaction fees

Pouring private coins

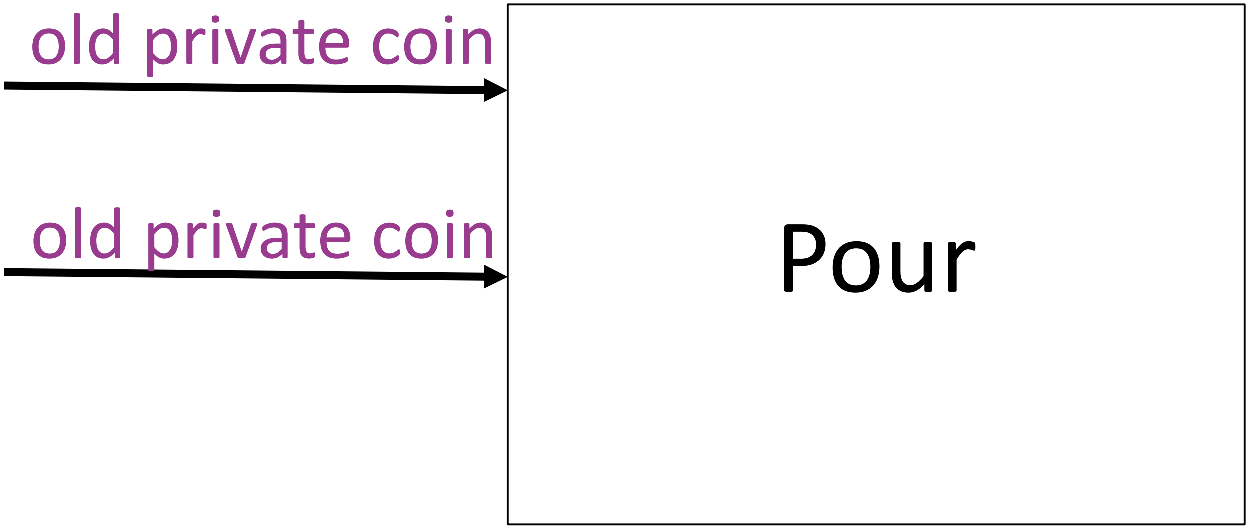
Single transaction type capturing:

Sending payments

Making change

Exchanging into BTC

Transaction fees



Pouring private coins

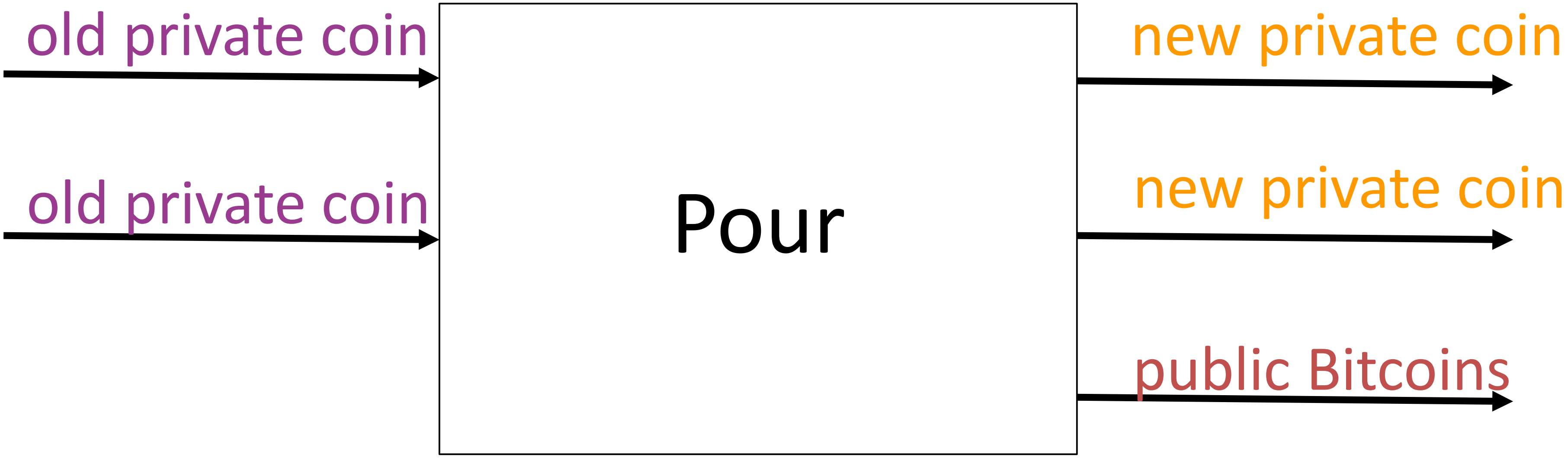
Single transaction type capturing:

Sending payments

Making change

Exchanging into BTC

Transaction fees



Pouring private coins

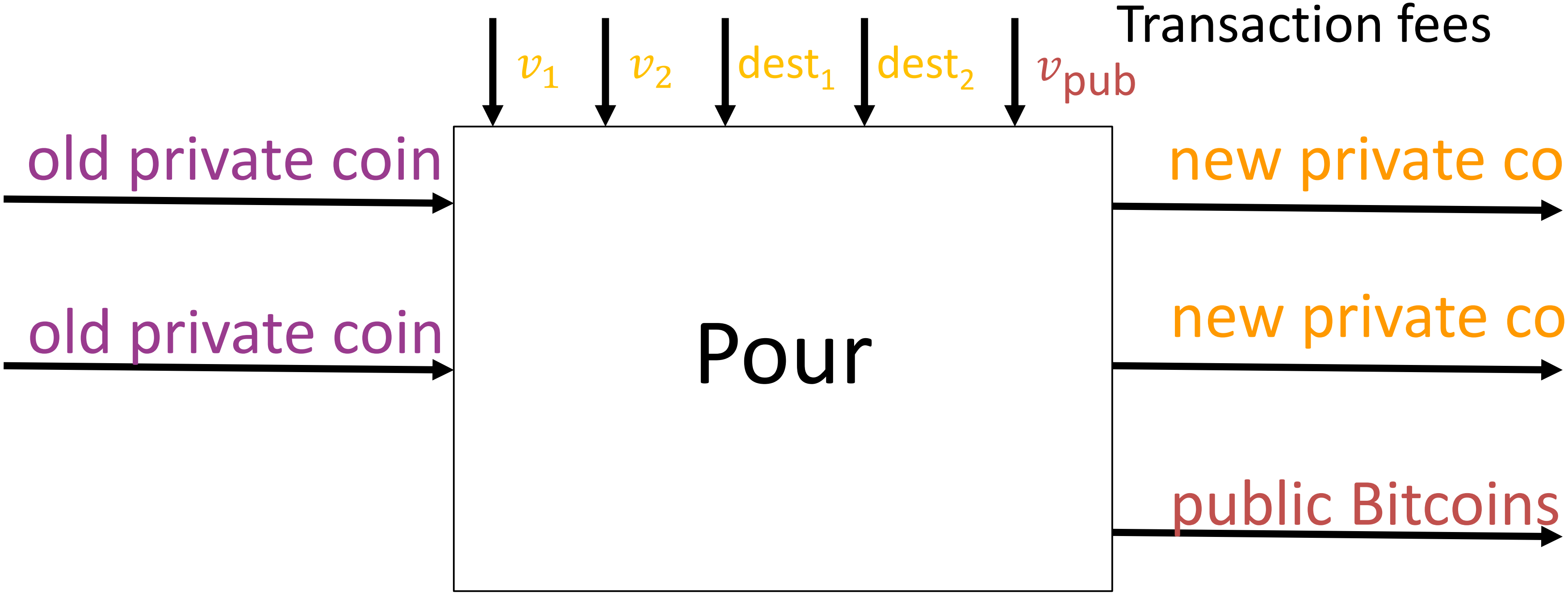
Single transaction type capturing:

Sending payments

Making change

Exchanging into BTC

Transaction fees



Pouring private coins

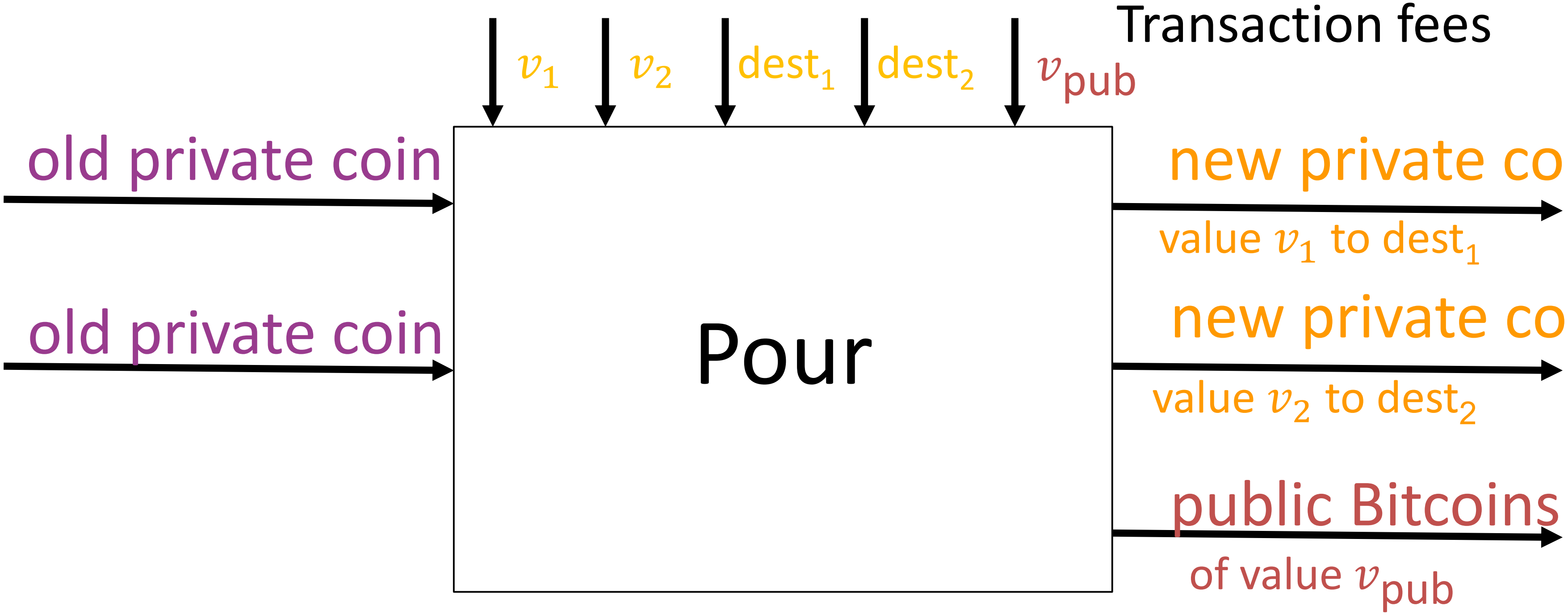
Single transaction type capturing:

Sending payments

Making change

Exchanging into BTC

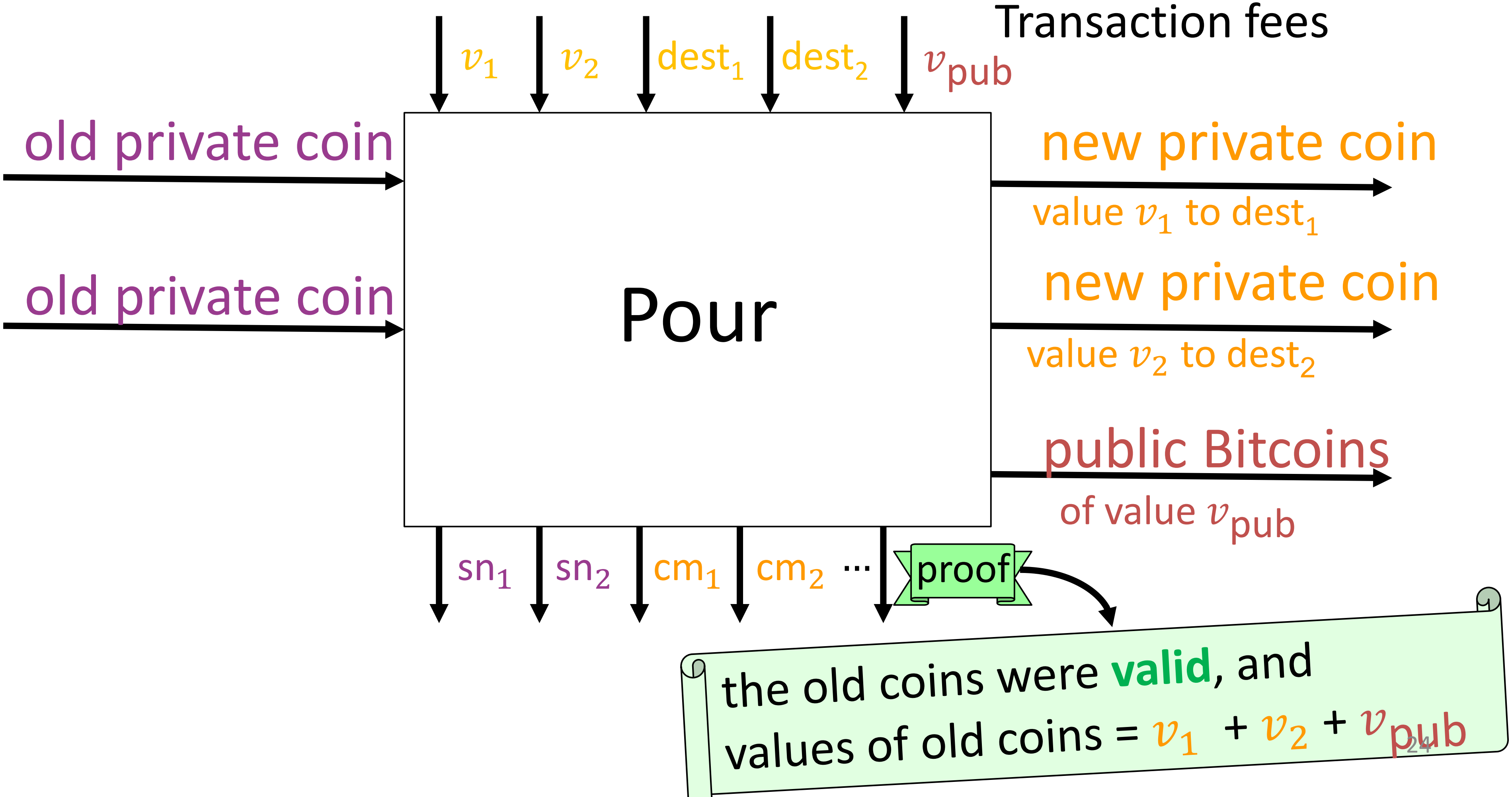
Transaction fees



Pouring private coins

Single transaction type capturing:

- Sending payments
- Making change
- Exchanging into BTC
- Transaction fees



v_1 v_2 dest_1 dest_2 v_{pub}

old private coin

old private coin

Pour

new private coin

value v_1 to dest_1

new private coin

value v_2 to dest_2

public Bitcoins

of value v_{pub}

sn_1 sn_2 cm_1 cm_2 ⋯ proof

the old coins were **valid**, and
values of old coins = $v_1 + v_2 + v_{\text{pub}}$

Pouring private coins

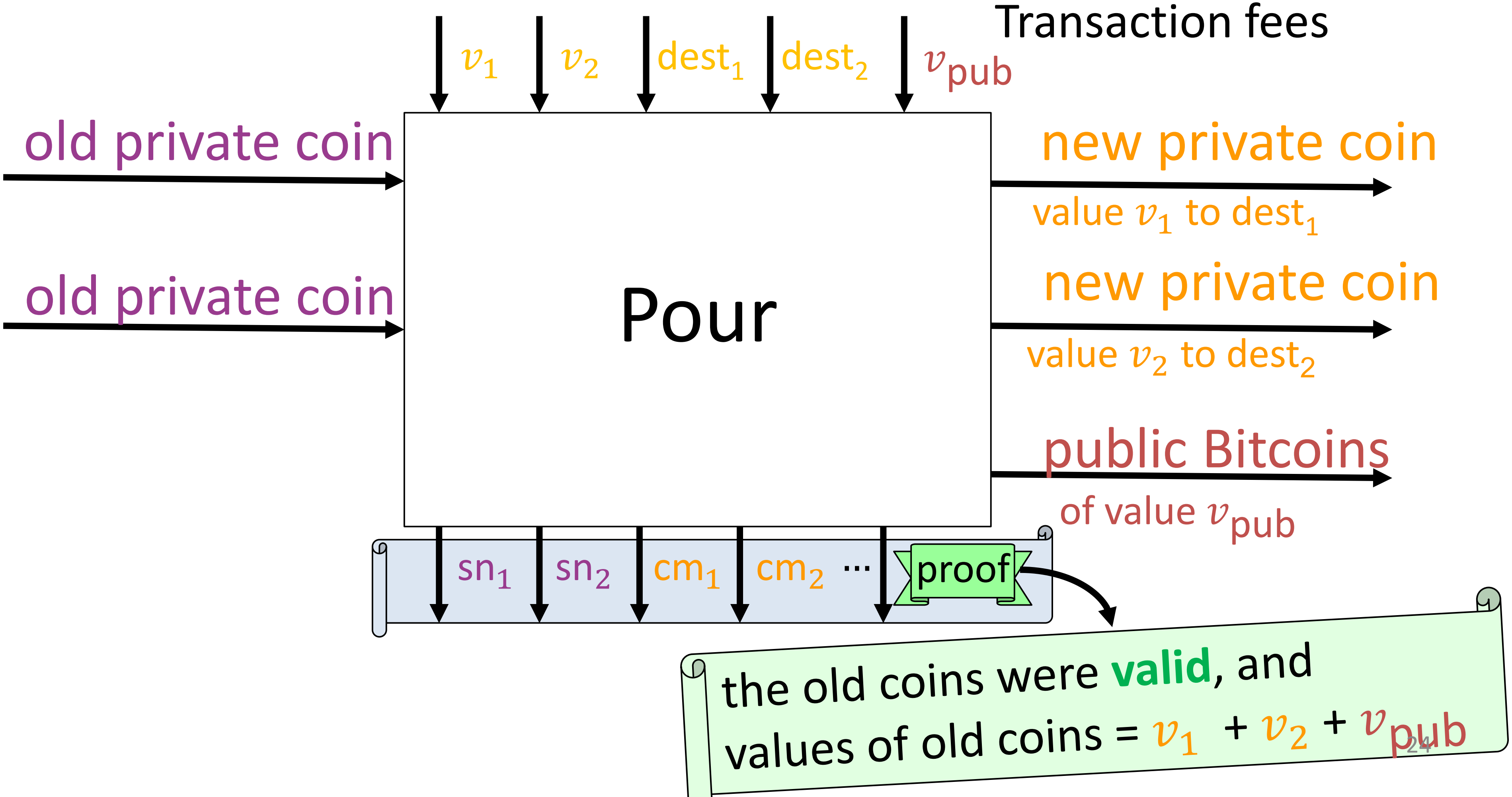
Single transaction type capturing:

Sending payments

Making change

Exchanging into BTC

Transaction fees



the old coins were **valid**, and

values of old coins = $v_1 + v_2 + v_{\text{pub}}$

Outline

1. Bitcoin’s privacy problem

2. Zerocash: privacy-preserving decentralized currency

3. **Zcash: deploying Zerocash in practice…**

Proof-of-concept implementation

Proof-of-concept implementation

`Setup`	<2 min, 896MB params
`Mint`	23 μs 72B tx
`Pour`	46 s, 1KB tx
`VerifyTx`	<9 ms/tx
`Receive`	<2 ms/tx

“academic practical” → “real world practical”

“academic practical” → “real world practical”

Zcash startup: 2+ years of extensive research & development

“academic practical” → “real world practical”

Zcash startup: 2+ years of extensive research & development

Lots of work to bridge the gap:

“academic practical” → “real world practical”

Zcash startup: 2+ years of extensive research & development

Lots of work to bridge the gap:

- Thorough analysis and vetting

“academic practical” → “real world practical”

Zcash startup: 2+ years of extensive research & development

Lots of work to bridge the gap:

- Thorough analysis and vetting

  Uncovered completeness bugs in the protocol ☺

“academic practical” → “real world practical”

Zcash startup: 2+ years of extensive research & development

Lots of work to bridge the gap:

- Thorough analysis and vetting

  Uncovered completeness bugs in the protocol ☺

- External security audits
  (NCC Group, Coinspect, Solar Designer)

“academic practical” → “real world practical”

Zcash startup: 2+ years of extensive research & development

Lots of work to bridge the gap:

- Thorough analysis and vetting

  Uncovered completeness bugs in the protocol ☺
- External security audits
  (NCC Group, Coinspect, Solar Designer)
- Efficiency improvements, protocol changes

“academic practical” → “real world practical”

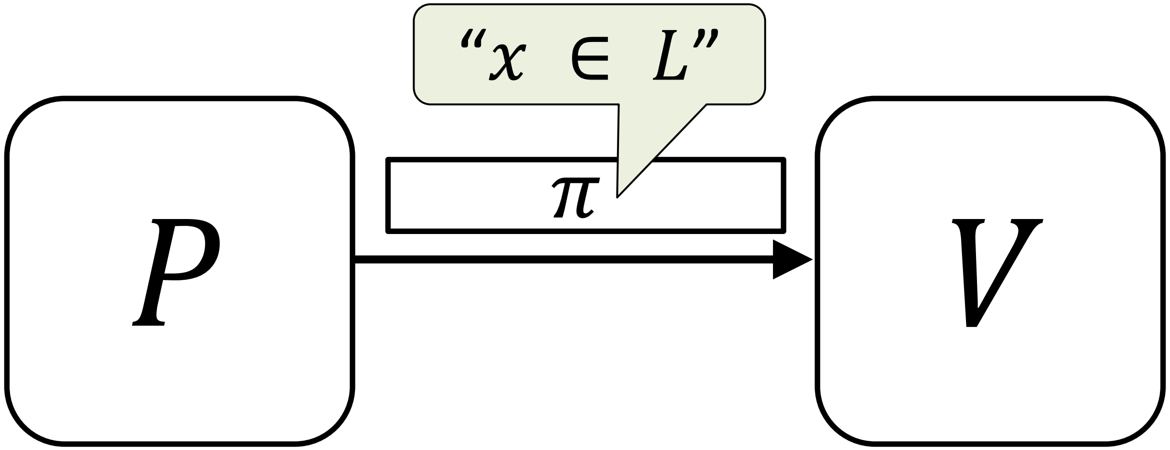
Zcash startup: 2+ years of extensive research & development

Lots of work to bridge the gap:

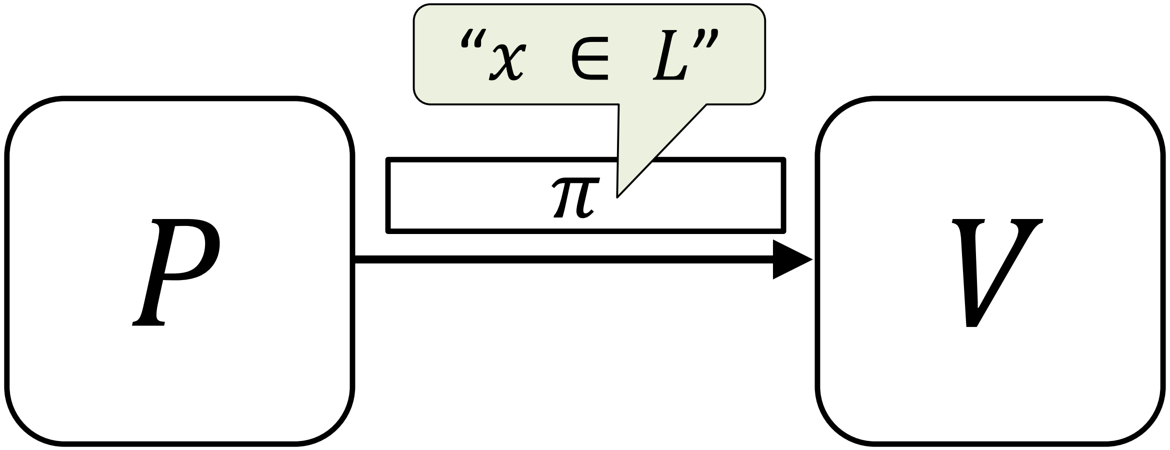
- Thorough analysis and vetting

  Uncovered completeness bugs in the protocol ☺
- External security audits
  (NCC Group, Coinspect, Solar Designer)
- Efficiency improvements, protocol changes
- Creation of clients, integration with wallets & exchanges

Non-interactive zero-knowledge

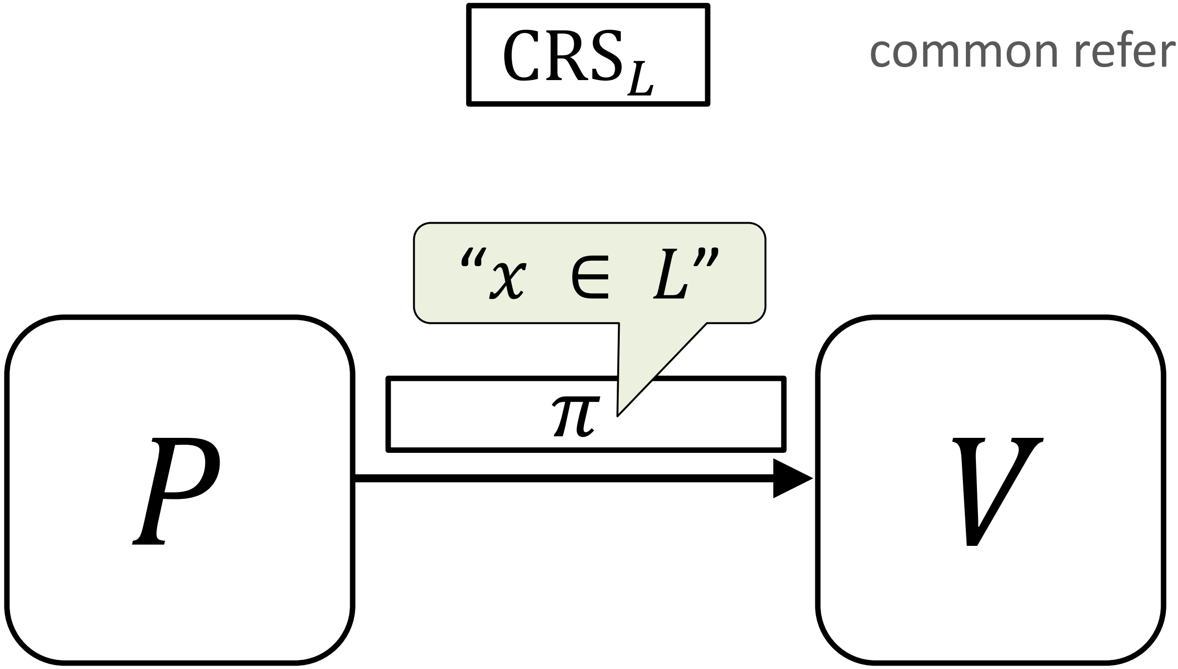


Non-interactive zero-knowledge



Thm: **Impossible** for NP (without any help) [GMR85, GO94]

Non-interactive zero-knowledge

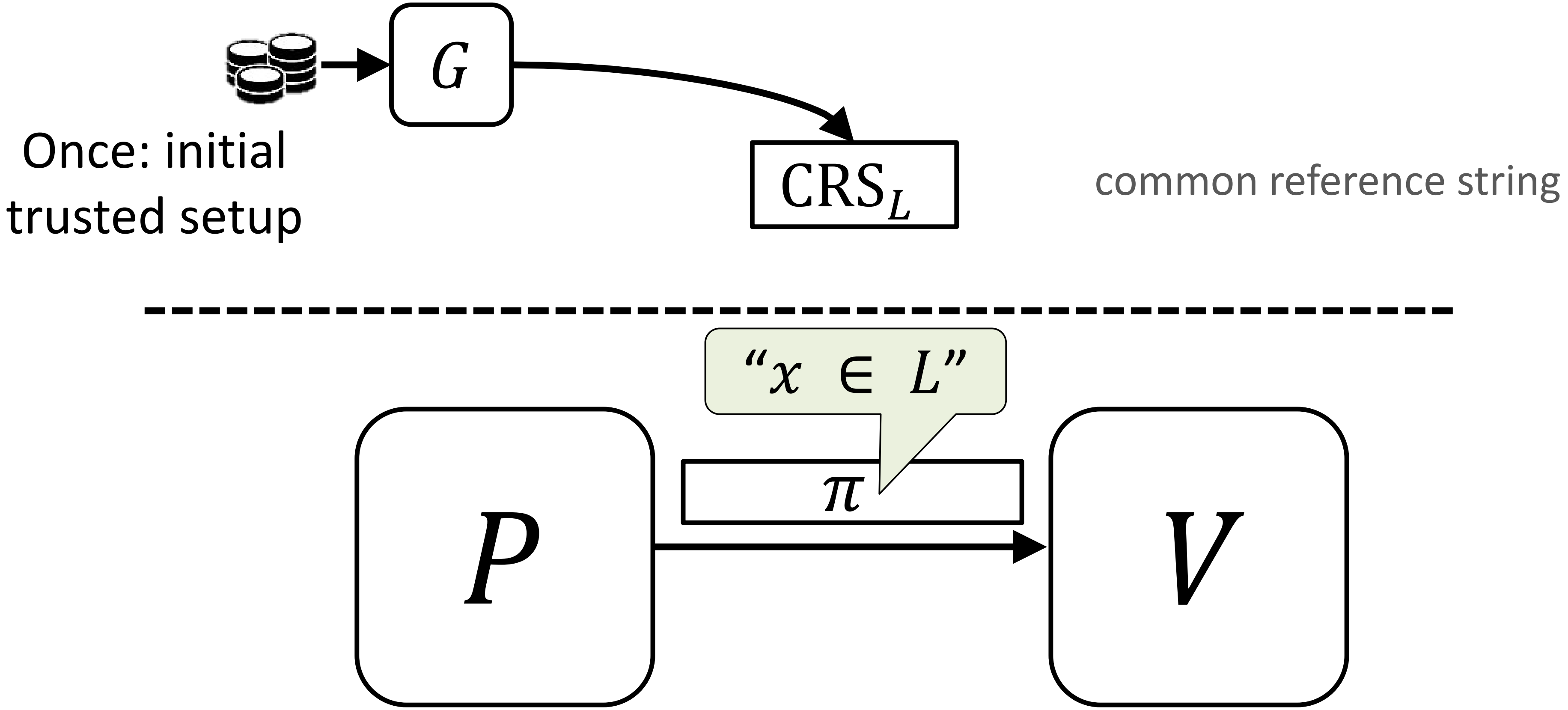


common reference string

Thm: **Impossible** for NP (without any help) [GMR85, GO94]

Thm: **Possible** for NP with help of CRS. [BFM88, NY90, BDMP91]

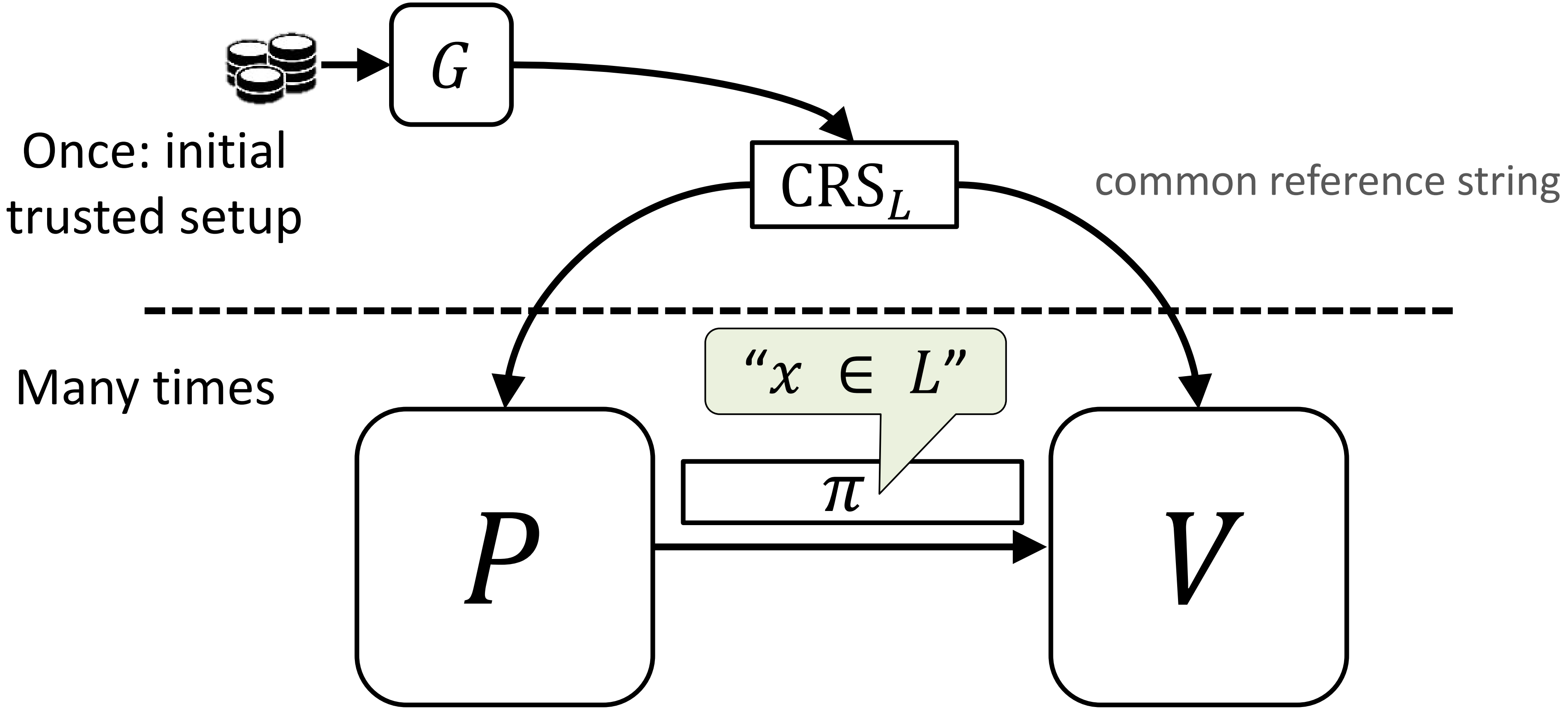
Non-interactive zero-knowledge



Thm: **Impossible** for NP (without any help) [GMR85, GO94]

Thm: **Possible** for NP with help of CRS. [BFM88, NY90, BDMP91]

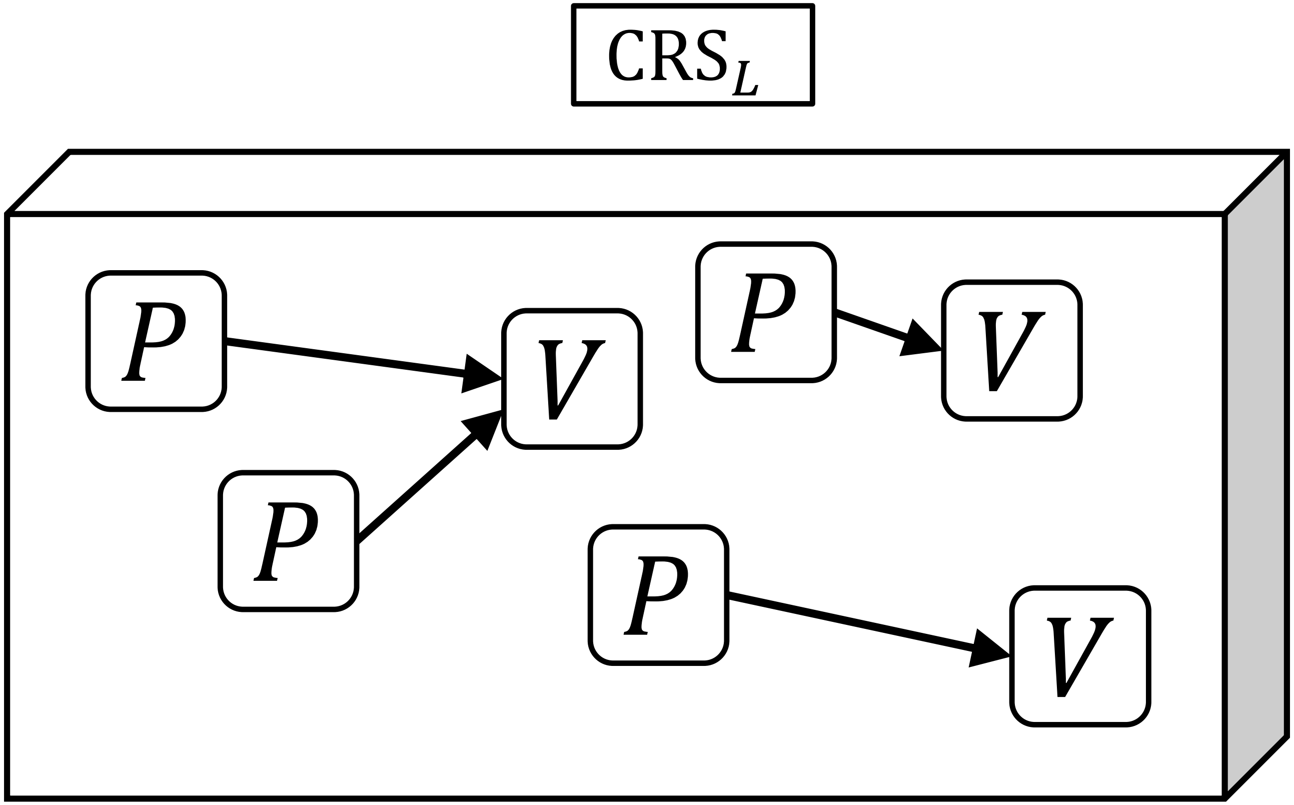
Non-interactive zero-knowledge



Thm: **Impossible** for NP (without any help) [GMR85, GO94]

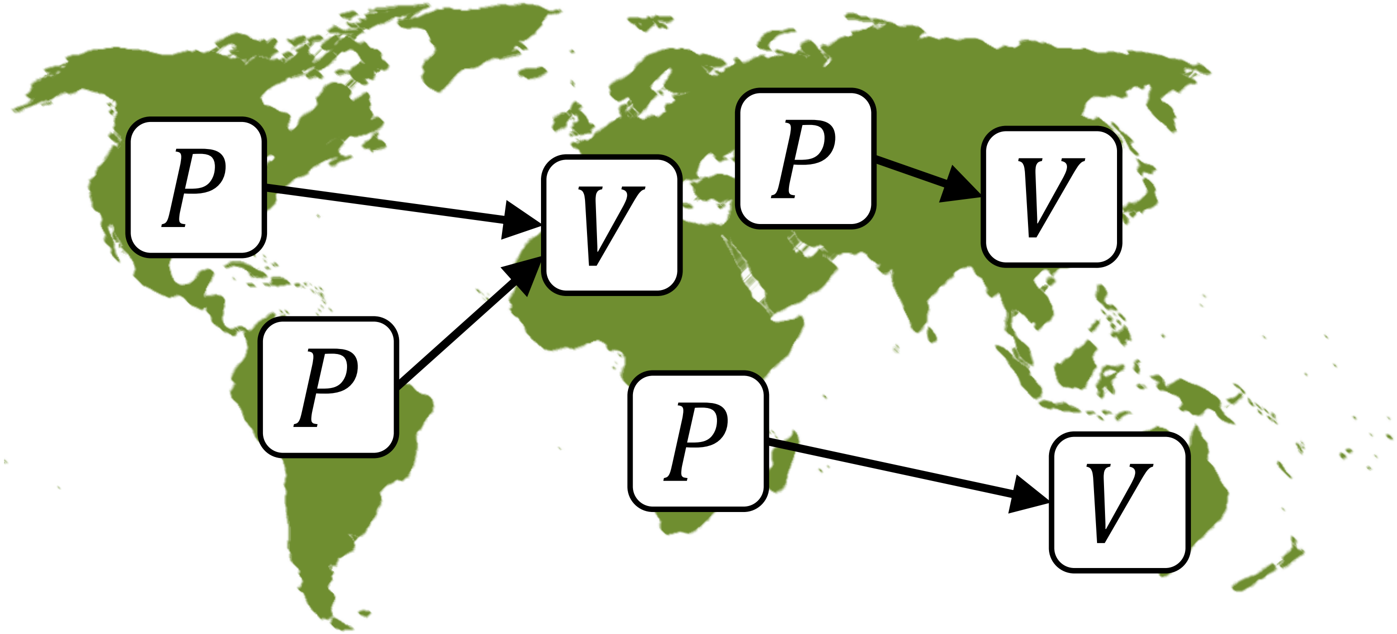
Thm: **Possible** for NP with help of CRS. [BFM88, NY90, BDMP91]

Deploying NIZKs

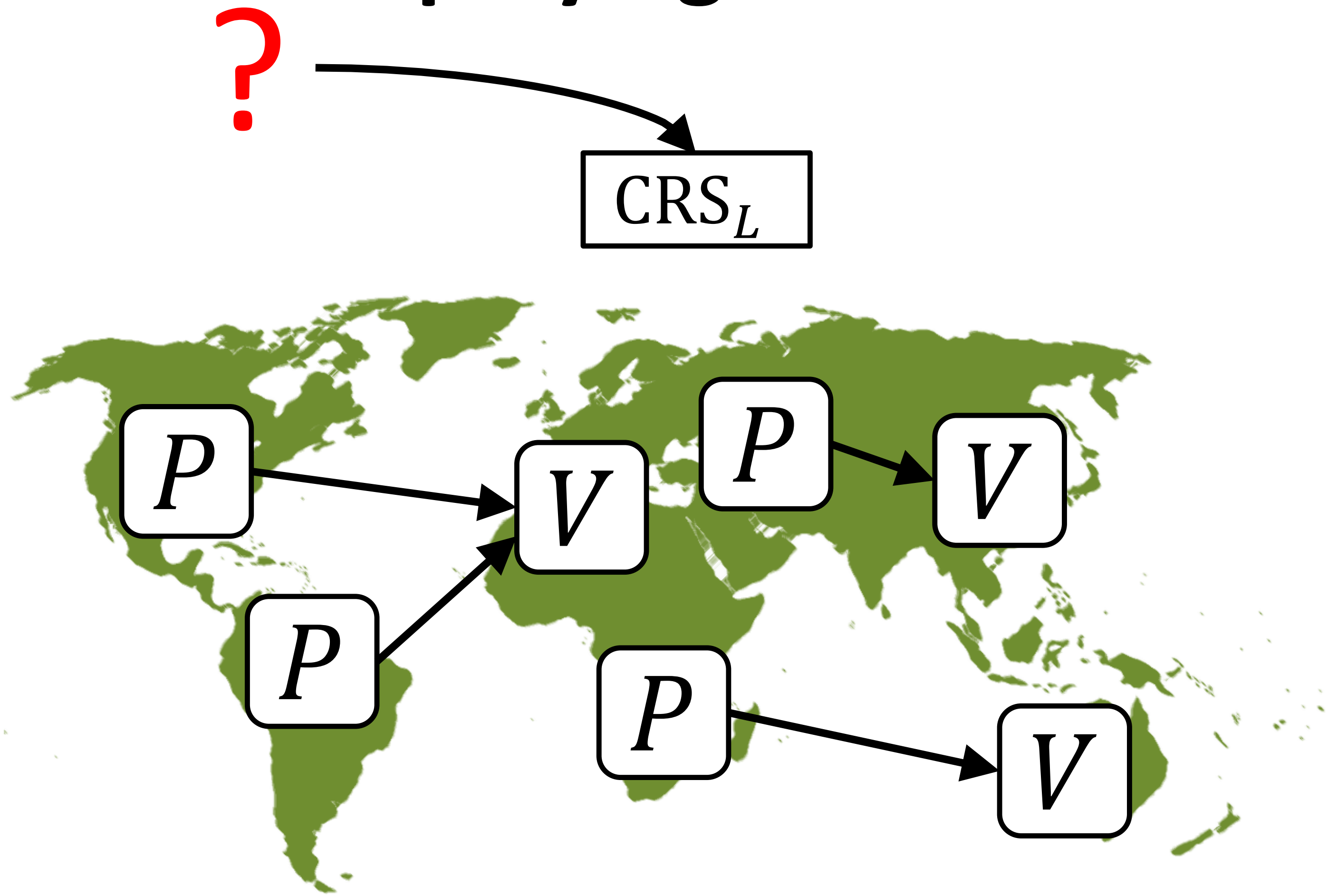


Deploying NIZKs

CRS_L

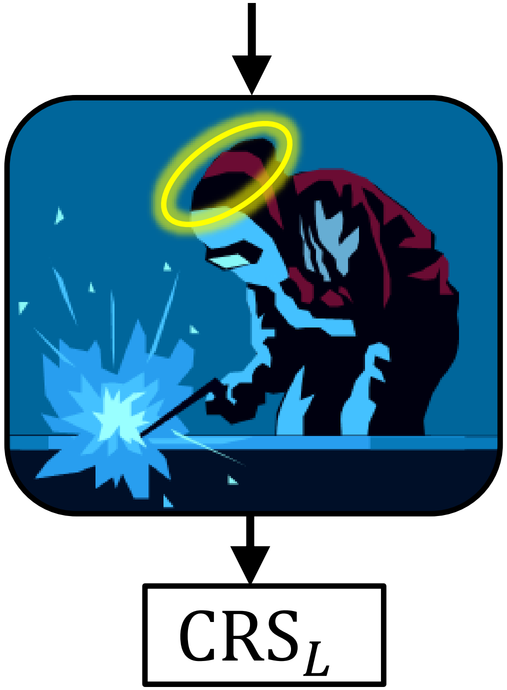


Deploying NIZKs



Q: In practice, who generates the CRS?

Ideal world



CRS_L

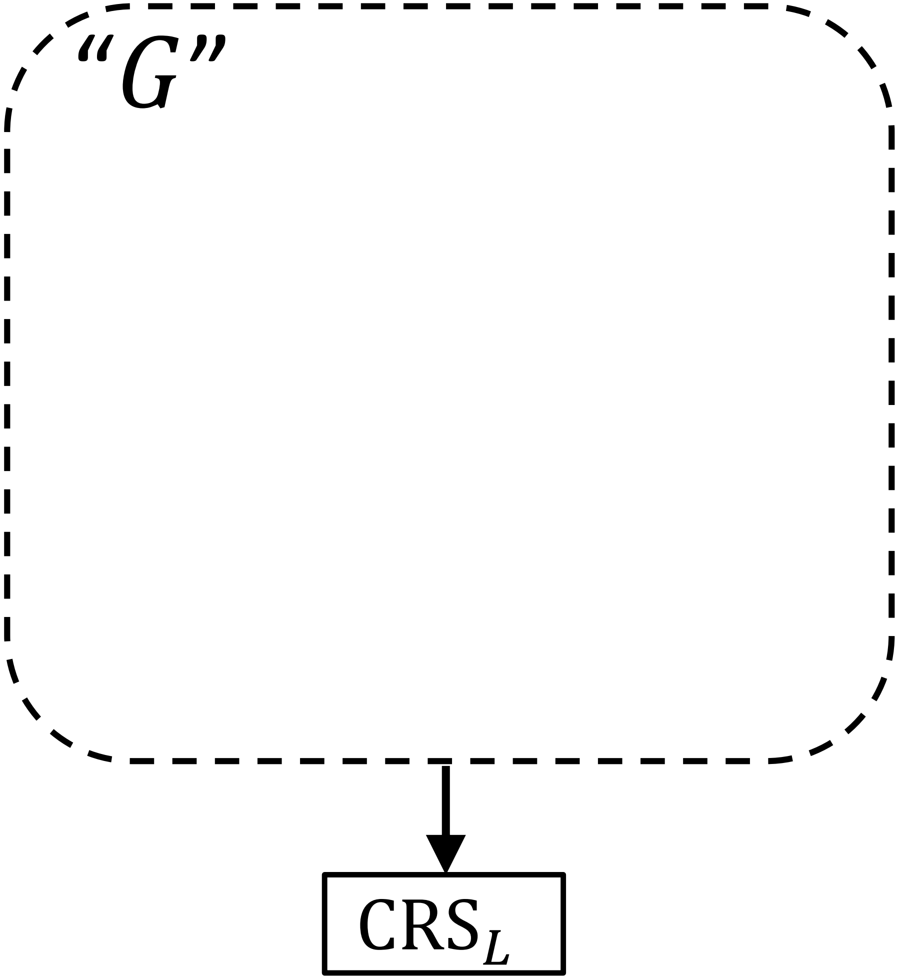
Distributed protocol for CRS of SNARKs

[BCGTV15]

Ideal world

CRS_L

Real world



“G”

CRS_L

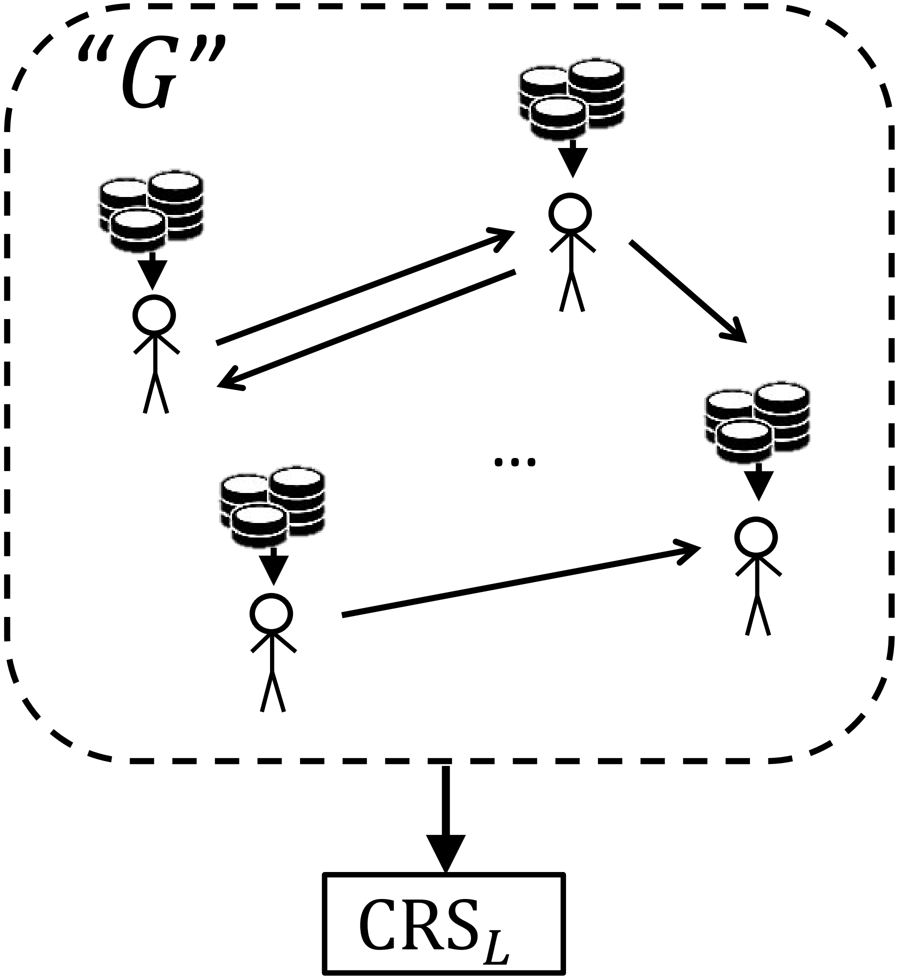
Distributed protocol for CRS of SNARKs

[BCGTV15]

Ideal world

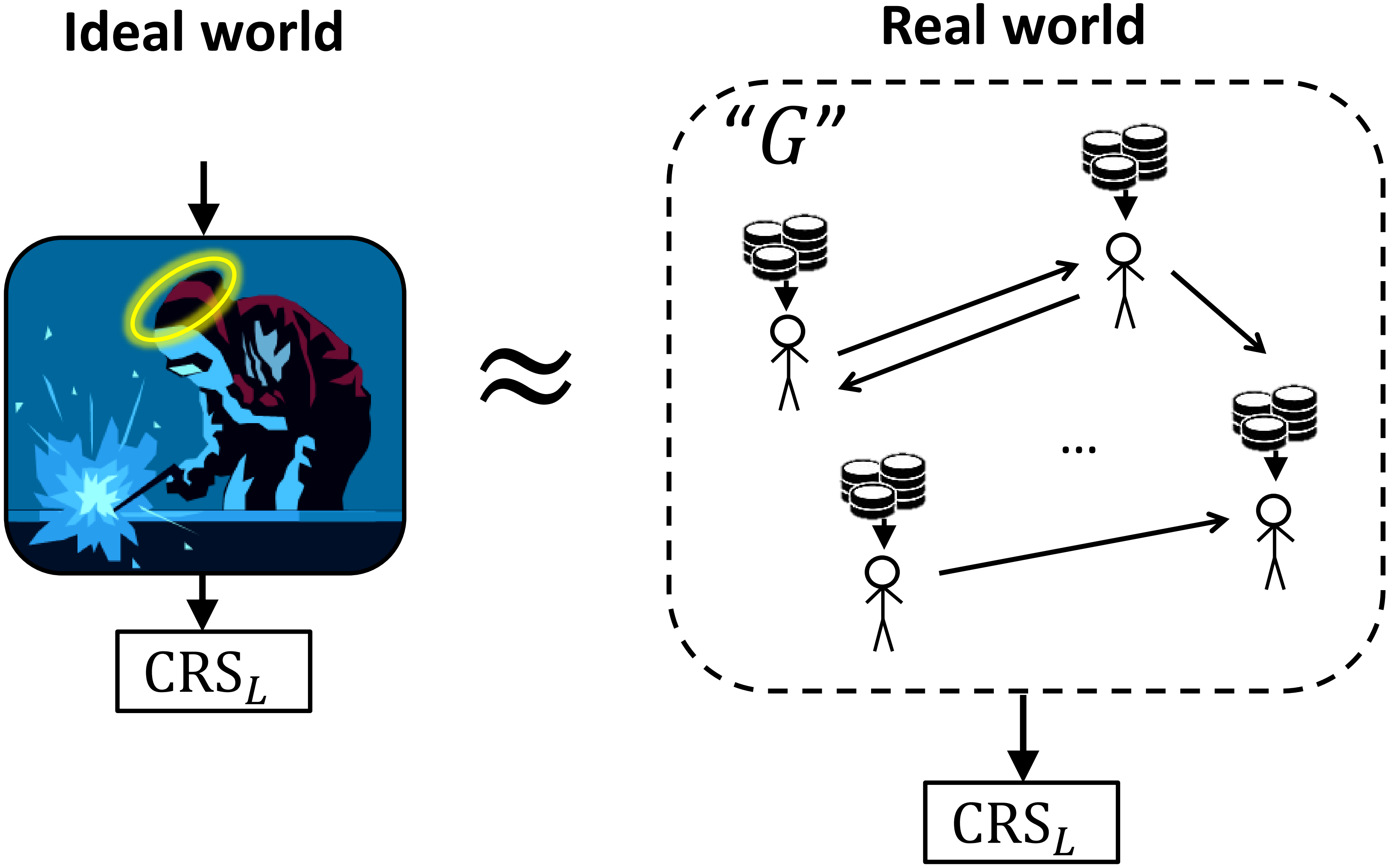
CRS_L

Real world



Distributed protocol for CRS of SNARKs

[BCGTV15]



Ideal world

Real world

$\approx$

"G"

...

CRS_L

CRS_L

Distributed protocol for CRS of SNARKs

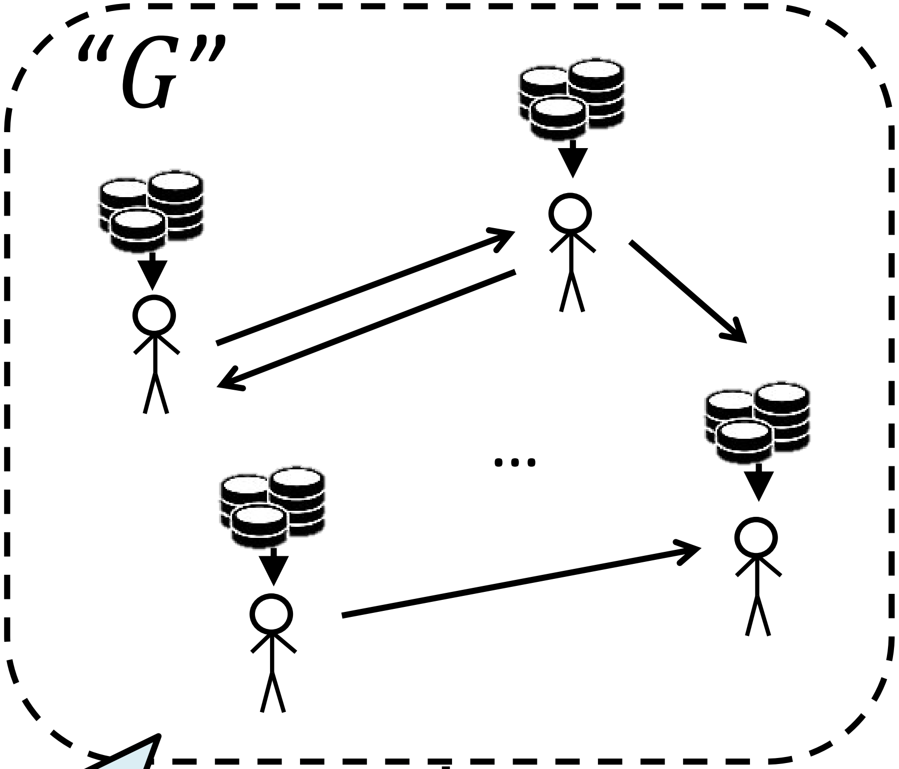
[BCGTV15]

Ideal world

CRS_L

$\approx$

Real world



Up to $n-1$ corruptions

CRS_L

Distributed protocol for CRS of SNARKs

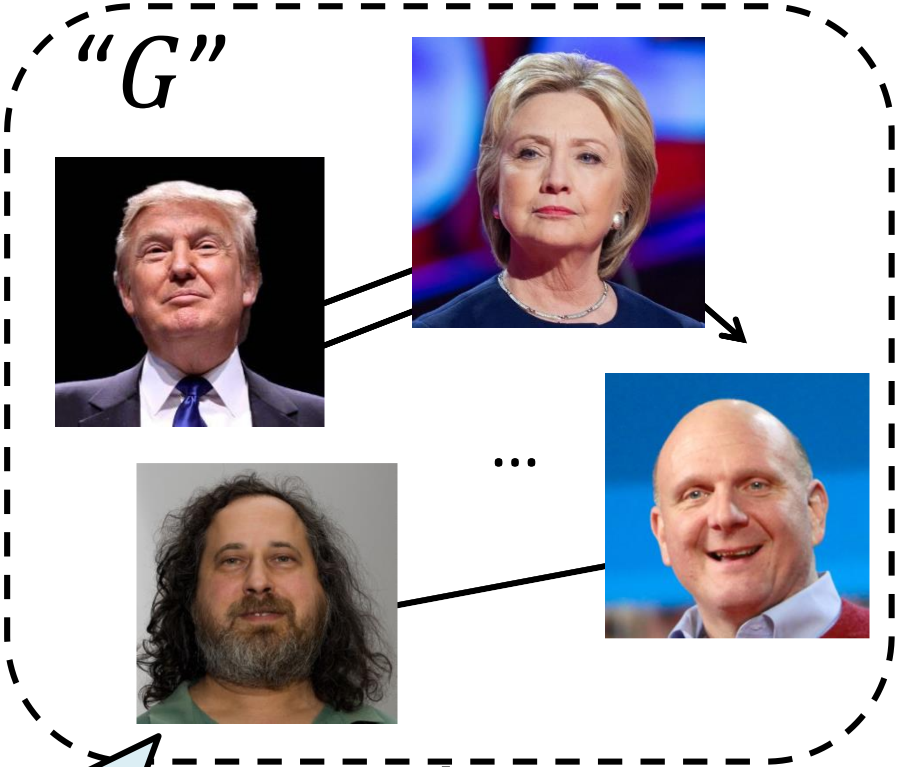
[BCGTV15]

Ideal world

CRS_L

$\approx$

Real world



Up to $n-1$ corruptions

CRS_L

Trusted setup ceremony

(October 21-23 2016)

Trusted setup ceremony

(October 21-23 2016)

- Used a tailored & optimized version of **[BCGTV15]** protocol

Trusted setup ceremony

(October 21-23 2016)

- Used a tailored & optimized version of **[BCGTV15]** protocol
- Ceremony design required extensive threat modelling and security engineering (paper upcoming)

Trusted setup ceremony

(October 21-23 2016)

- Used a tailored & optimized version of **[BCGTV15]** protocol
- Ceremony design required extensive threat modelling and security engineering (paper upcoming)

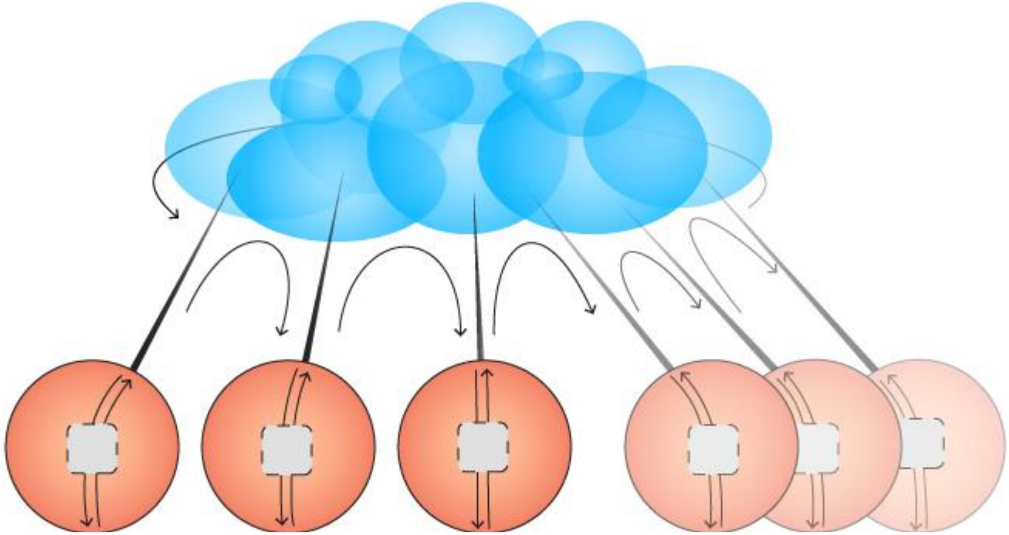


Image credit: z.cash

Trusted setup ceremony

(October 21-23 2016)

- Used a tailored & optimized version of **[BCGTV15]** protocol
- Ceremony design required extensive threat modelling and security engineering (paper upcoming)

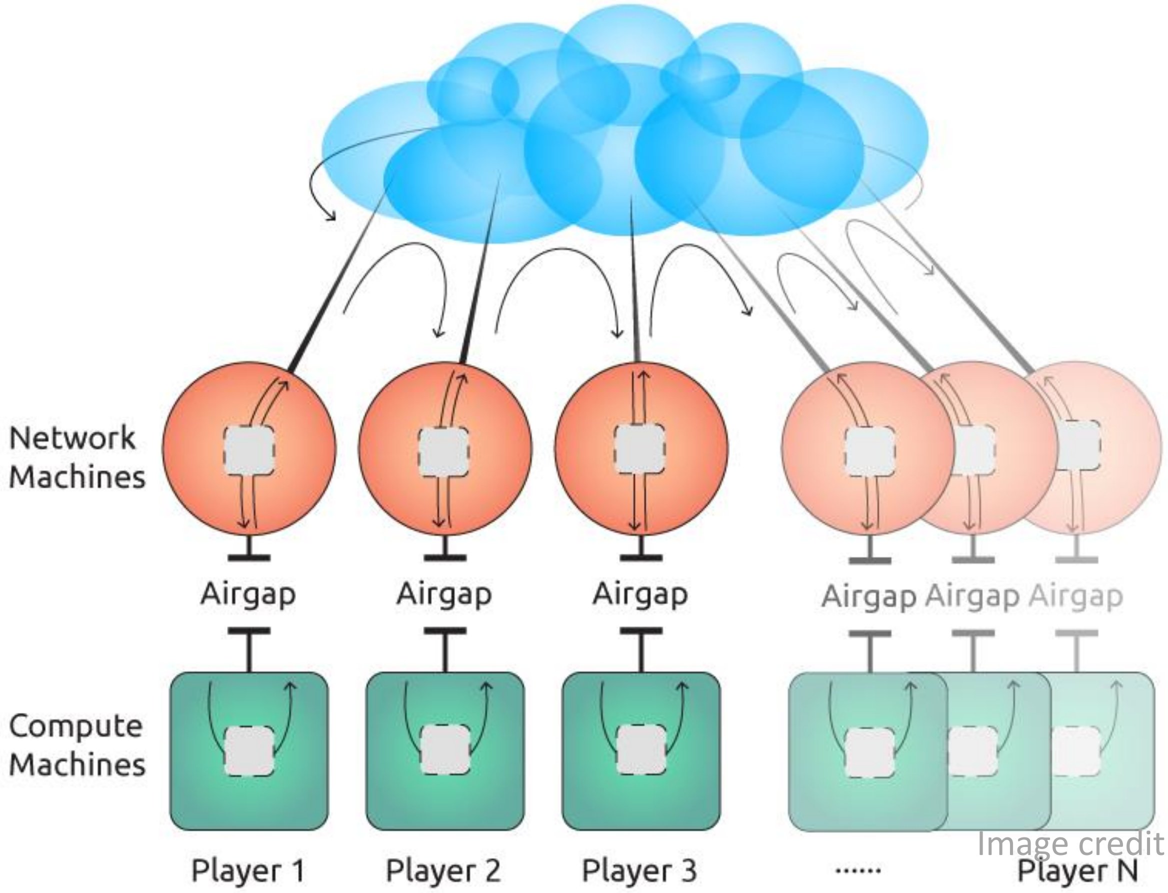


Image credit: z.cash

6 geographically distributed participants

Image from petertodd.org

Destruction of the compute nodes

Destruction of the compute nodes

Images from z.cash and petertodd.org

Destruction of the compute nodes

Images from z.cash and petertodd.org

Publicly verifiable audit trail

Image credit: z.cash

Beyond privacy and fungibility:

Beyond privacy and fungibility: zero-knowledge in public oversight

Beyond privacy and fungibility: zero-knowledge in public oversight

π = “I’m using unspent coins of my own.
My transaction preserves balance.

Beyond privacy and fungibility: zero-knowledge in public oversight

π = “I’m using unspent coins of my own.
My transaction preserves balance.
But I’m **not telling** recipient or amount.”

Beyond privacy and fungibility: zero-knowledge in public oversight

π = “I’m using unspent coins of my own.
My transaction preserves balance.
But I’m **not telling** recipient or amount.”

Beyond privacy and fungibility: zero-knowledge in **public oversight**

π = “I’m using unspent coins of my own.
My transaction preserves balance.
But I’m **not telling** recipient or amount.”

The money went to a **501 (c) non-profit**!

Beyond privacy and fungibility: zero-knowledge in **public oversight**

π = “I’m using unspent coins of my own.
My transaction preserves balance.
But I’m **not telling** recipient or amount.”

The money went to a **501 (c) non-profit**!
But I’m not telling anyone which one.

Beyond privacy and fungibility: zero-knowledge in public oversight

π = “I’m using unspent coins of my own.
My transaction preserves balance.
But I’m **not telling** recipient or amount.”

The money went to a **501 (c) non-profit**!
But I’m not telling anyone which one.

or

Beyond privacy and fungibility: zero-knowledge in public oversight

π = “I’m using unspent coins of my own.
My transaction preserves balance.
But I’m **not telling** recipient or amount.”

The money went to a **501 (c) non-profit**!
But I’m not telling anyone which one.

or

Proof of solvency.

Beyond privacy and fungibility: zero-knowledge in public oversight

π = “I’m using unspent coins of my own.
My transaction preserves balance.
But I’m **not telling** recipient or amount.”

The money went to a **501 (c) non-profit**!
But I’m not telling anyone which one.

or

Proof of solvency. My private keys control 50 000 BTC, but I won’t tell you my address.

Beyond privacy and fungibility: zero-knowledge in public oversight

π = “I’m using unspent coins of my own.
My transaction preserves balance.
But I’m **not telling** recipient or amount.”

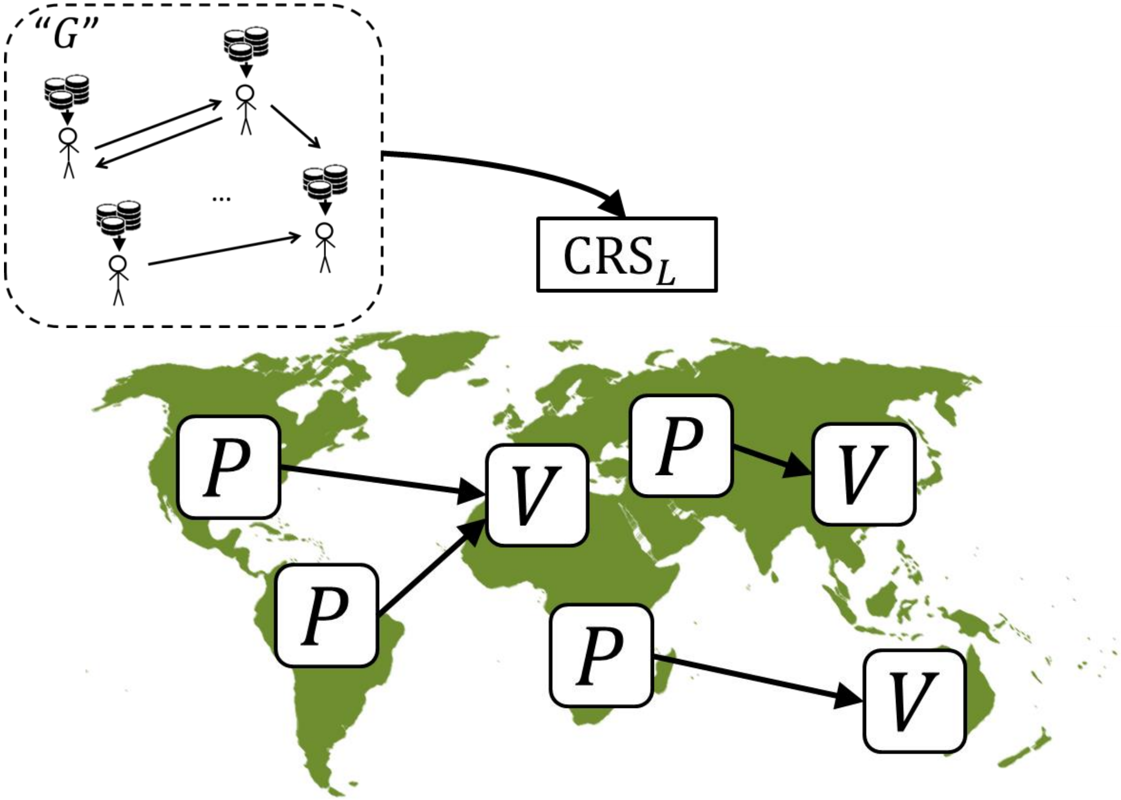
The money went to a **501 (c) non-profit**!
But I’m not telling anyone which one.

or

Proof of solvency. My private keys control
50 000 BTC, but I won’t tell you my address.

Q: Which policies are desirable/feasible?

Thank you!



www.zerocash-project.org