From Zero Knowledge to Private Transactions

Alessandro Chiesa

UC Berkeley

StarkWare

Zcash

Plan for today:

1. in peer-to-peer systems there is a conflict between privacy & integrity

2. zero knowledge proofs enable achieving <u>both</u> privacy & integrity

3. frontiers of ZKP on blockchains

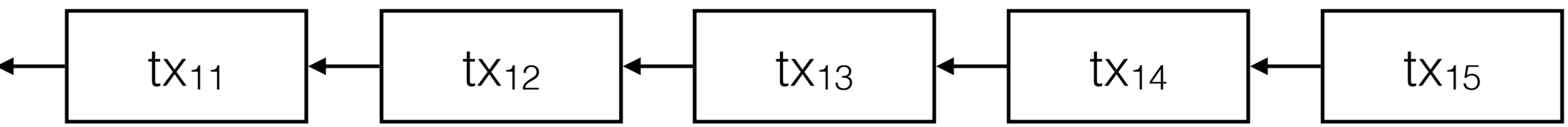
The Conflict Between Privacy & Integrity in Peer-to-Peer Systems

A Motivating Example: Bitcoin

Bitcoin has three main parts.

(1) consensus

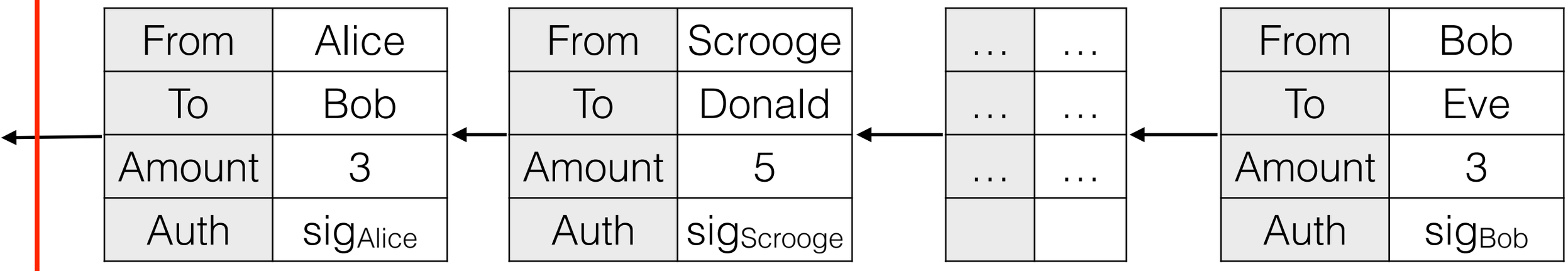
a peer-to-peer protocol that achieves an append-only ledger



(2) incentives to participate

nodes that participate get *donations/fees* from processed transactions

(3) transactions are signed payment messages



Every payment transaction reveals: **sender**, **receiver**, **amount**.

This should raise some worries!

Payment History Reveals A Lot

- medical information (specialty of your doctors)

  ➡ insurance companies could use it to increase premium or even deny coverage

- current and past locations (your travel patterns)

  ➡ gold mine for stalkers, burglars, assassins, …

- merchant cash flow (suppliers, daily sales, …)

  ➡ intelligence for competitors

One of the reasons for why banks are highly regulated.

E.g., in the US the Gramm-Leach-Bliley Act states that:

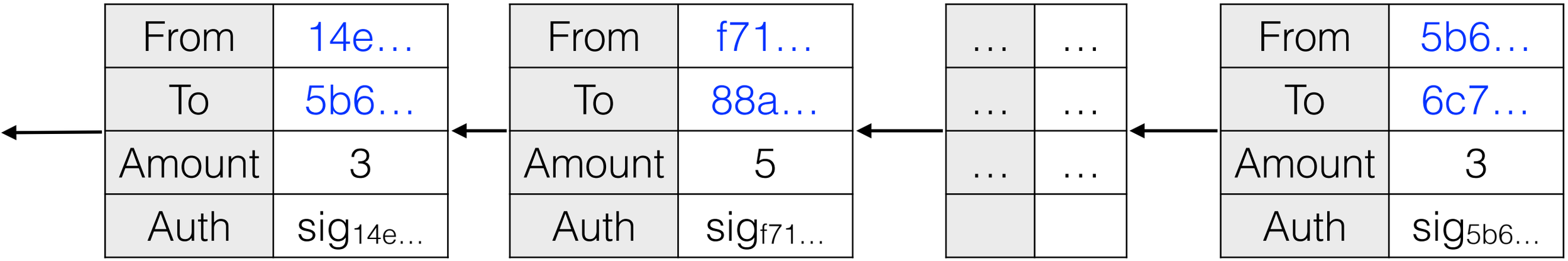
The GLB Act requires financial institutions … to explain their information-sharing practices to their customers and to ***safeguard sensitive data****.* [www.ftc.gov]

Transactions In Bitcoin

(simplified)

But in Bitcoin payment transactions do not include names!

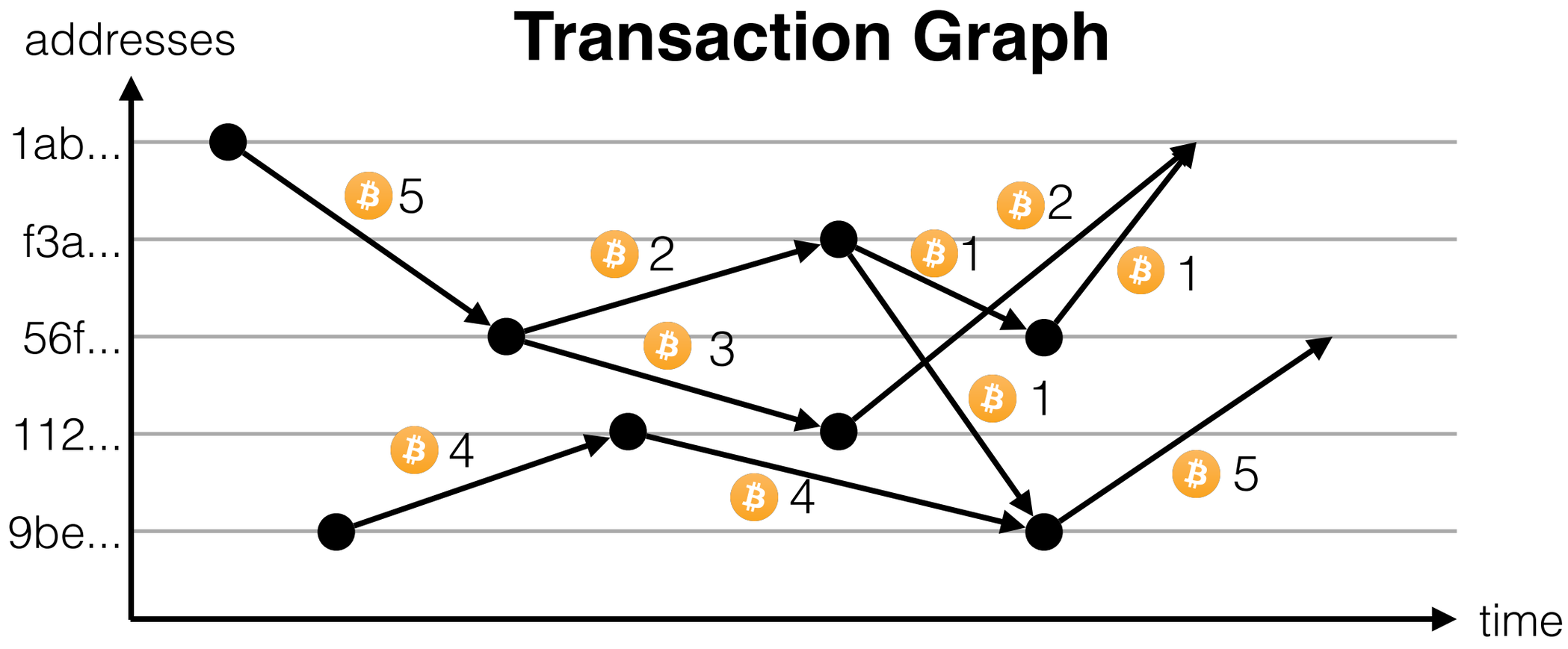
Instead they include *cryptographic addresses*:



The Transaction Graph

Your addressess are known by everyone you interact with.

And literally anyone can analyze the ledger:

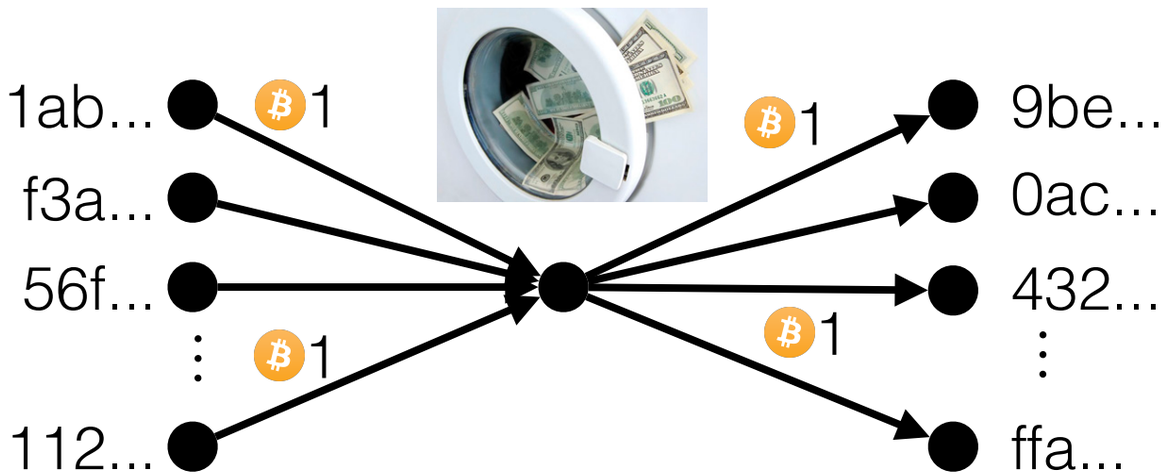


transaction graph + side-info → addresses become names of people!

Obfuscation

Use new address for each payment.

Launder money with others.



"Seems" harder to analyze.

But these "obfuscation" approaches leave tracks.

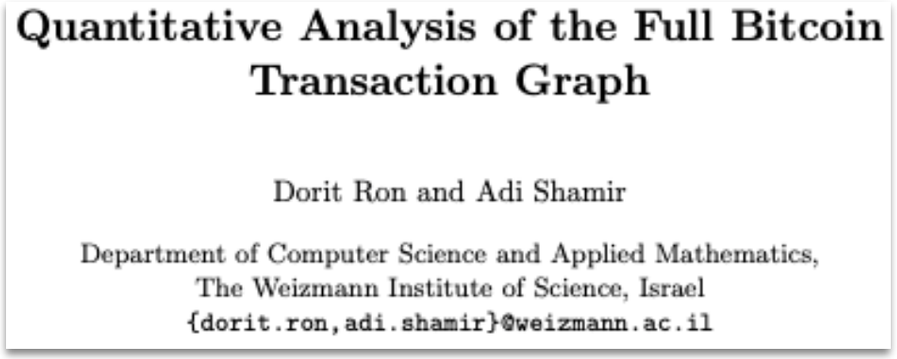
Bitcoin history is publicly stored **forever**.

Methods of analysis only get **stronger**.

Transaction Graph Analytics

Numerous de-anonymization studies by researchers for Bitcoin and altcoins:

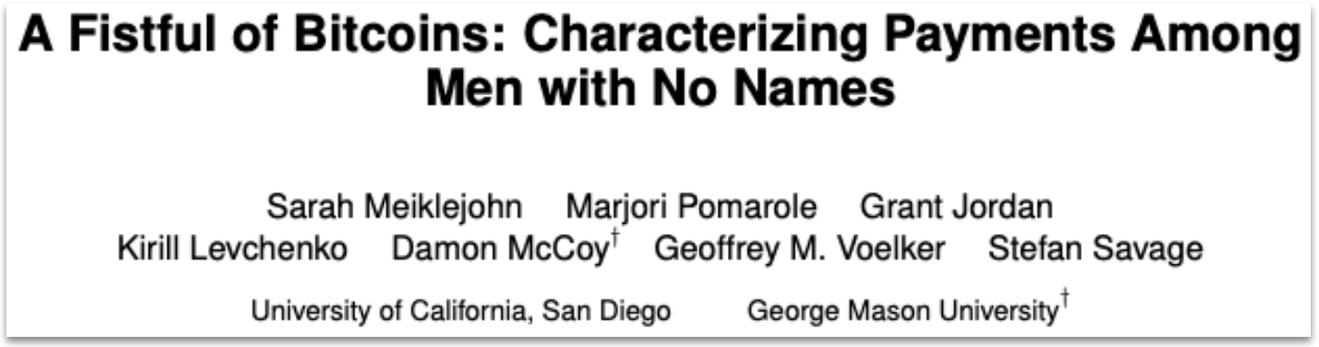
[RM11] [BBSU12] [RS12] [RS13] [MPJLMVS13] [RS14] ... [MMLN17] [KFTS17] ...



Quantitative Analysis of the Full Bitcoin Transaction Graph

Dorit Ron and Adi Shamir

Department of Computer Science and Applied Mathematics,
The Weizmann Institute of Science, Israel
{dorit.ron,adi.shamir}@weizmann.ac.il



A Fistful of Bitcoins: Characterizing Payments Among Men with No Names

Sarah Meiklejohn Marjori Pomarole Grant Jordan
Kirill Levchenko Damon McCoy† Geoffrey M. Voelker Stefan Savage

University of California, San Diego George Mason University†

Several companies provide analytics services:

CIPHERTRACE

Cryptocurrency Risk Intelligence

Protecting Financial Institutions from Money Laundering Risks

ELLIPTIC

Bringing Compliance to Cryptoassets

Blockchain analytics for crypto businesses and financial institutions. Manage financial crime risk, achieve regulatory compliance, and grow with confidence.

Chainalysis

Building trust in blockchains

We create transparency for a global economy built on blockchains, enabling banks, business, and governments to have a common understanding of how people use cryptocurrency.

elementus

The Universal Blockchain Query Engine

Deanonymization: Good or Bad?

The FBI famously used "blockchain forensics" to support investigations that shut down the darknet market *Silk Road* (which sold illegal drugs, data, contraband, ...).

In *United States v. Coinbase, Inc*: coinbase was ordered by the government to disclose information about **nearly all its customers**, and it did so. The IRS has stated that it is using this information for tax fraud investigations.

At the same time **every user** is at risk of deanonymization and exploitation:
- Bitcoin transactions are like "*twitter for your bank account*" [Ian Miers]
- Bitcoin transaction data can be exploited by anyone (domestic or foreign)

Protecting user privacy is a must.

And achieing privacy via p2p systems is a socially desirable goal. →

Fungibility

a dollar is a dollar, regardless of its history

Recognized as crucial property of money 350+ years ago.

(*Crawfurd v. The Royal Bank*, 1749)

Bitcoin is **NOT** fungible
because a coin's history is public.

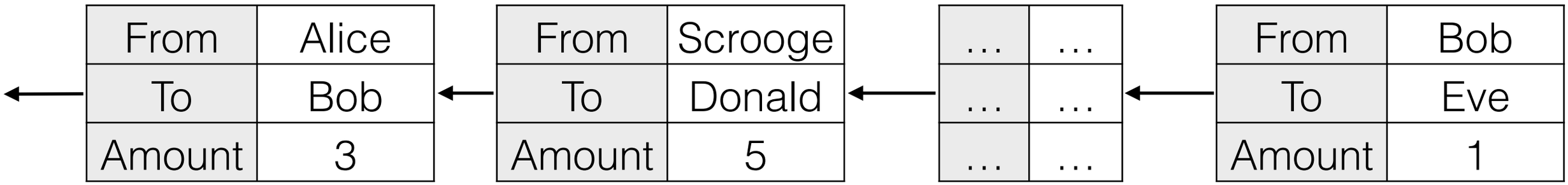
In particular, **a coin's value is ill-defined**:

- different people may value the same coin differently
- the same person may value different coins differently

This leads to the heuristic "new coins are more valuable than old ones".

But how should people agree on the correct value? A central party?

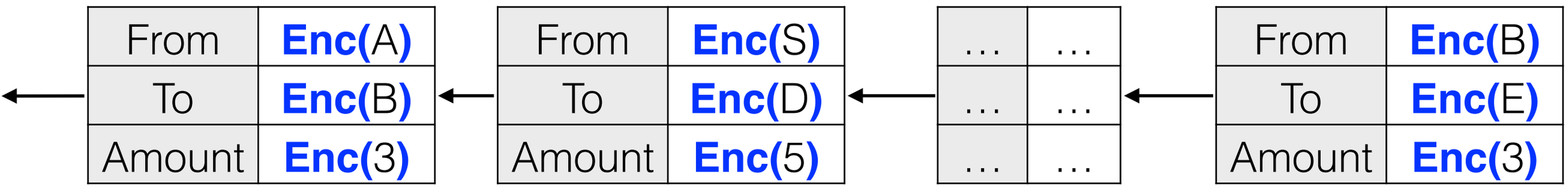
If privacy is so important then why isn't Bitcoin private?



How does the world know that Bob has 1 Bitcoin to spend?

Check that Bob received it, and that Bob did not spend it.

What if users encrypted their payment transactions?



Not clear how to check a payment's validity.

privacy and integrity are in conflict

Zero Knowledge Proofs Enable Achieving Privacy **and** Integrity

Today's Example:
The Zerocash Protocol

A cryptographic protocol achieving a ledger-based currency that is

privacy-preserving

Anyone can publish a payment transaction to anyone else, while **provably hiding the payment's sender, receiver, amount**.

Bonus: the protocol is efficient

The privacy-preserving payment transactions:

- take few seconds to produce
- are less than 1KB in size
- take a few milliseconds to verify

Ideas introduced in this protocol are now endemic to privacy in blockchains.

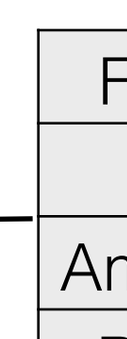
The Basic Intuition

From	**Enc(**A**)**
To	**Enc(**B**)**
Amount	**Enc(**1**)**
Proof	π

From	**Enc(**S**)**
To	**Enc(**D**)**
Amount	**Enc(**2**)**
Proof	π'

From	**Enc(**B**)**
To	**Enc(**E**)**
Amount	**Enc(**1**)**
Proof	π''

From	$\mathbf{c_1}$
To	$\mathbf{c_2}$
Amount	$\mathbf{c_3}$
Proof	π'''



I am publishing three ciphertexts $\mathbf{c_1}$,$\mathbf{c_2}$,$\mathbf{c_3}$.

They contain the encryptions of a sender address,
a receiver address, and a transfer amount respectively.

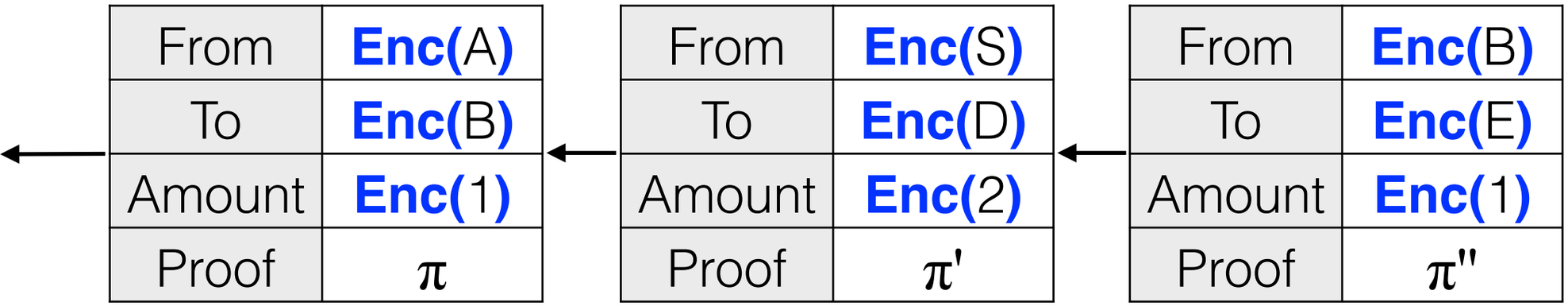
Moreover, the amount transfered has not been double spent.

I have generated a cryptographic proof π''' that all of this is true.

Q1: what kind of cryptographic proof?

Q2: what exactly is the statement being proved?

Beyond The Intuition



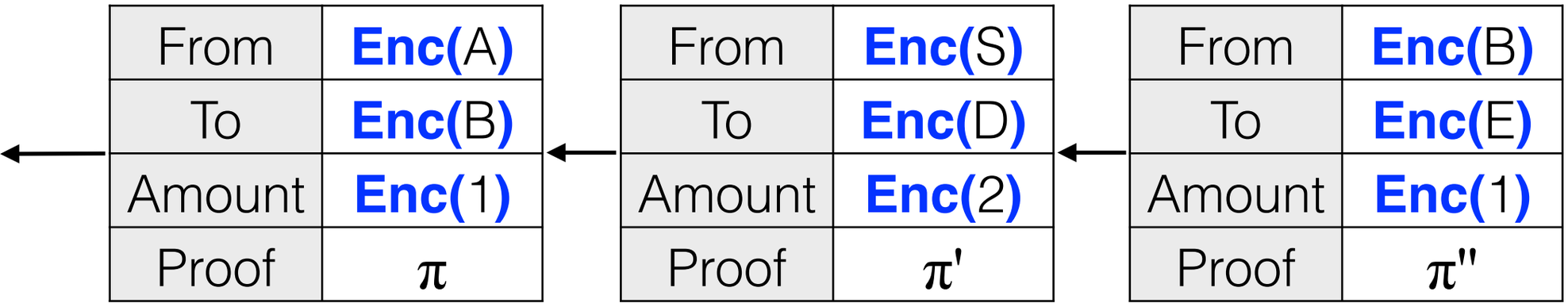
Q1: what kind of cryptographic proof?

zero knowledge	(nothing revealed beyond truth of statement)
succinct	(proof is very short and cheap to verify)
non-interactive	(need to write it down!)
proof	(true statements have proofs, false ones do not)
of knowledge	(technical… allows using crypto in statement)
zkSNARK	

There has been tremendous theory+practice progress over the last 10 years for zkSNARKs.
In particular, there are many popular open-source libraries for these tools:

`libsnark, bellman, arkworks, dalek, gnark, ethstark, openzkp, ...`

A Little Beyond Intuition



Q2: what exactly is the statement being proved?

This is not trivial. Let’s have some design fun.

Our Goal

Functionality: anyone can send a payment to anyone else.

Integrity: users cannot double spend their coins.

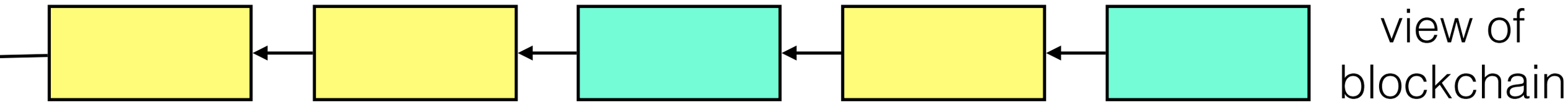
Privacy: all transactions "look the same".

The main tool is **ZKPs**: $\mathrm{P}(\mathrm{f},\mathrm{x},\mathbf{w})$ outputs $\boldsymbol{\pi}$ attesting that "*for public* f *and* x*, I know secret* $\mathbf{w}$ *s.t.* $\mathrm{f}(\mathrm{x},\mathbf{w})=\mathrm{true}$" and $\mathrm{V}(\mathrm{f},\mathrm{x},\boldsymbol{\pi})$ checks it.

Two additional basic tools:

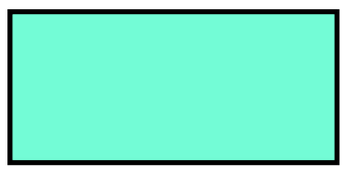
- **commitments**: cm=COMM(m;r) *hides* m and *binds* to m
- **pseudorandom functions:**
  y=PRF(m;sk) looks random to someone not knowing sk

Attempt #0: template



Transaction types

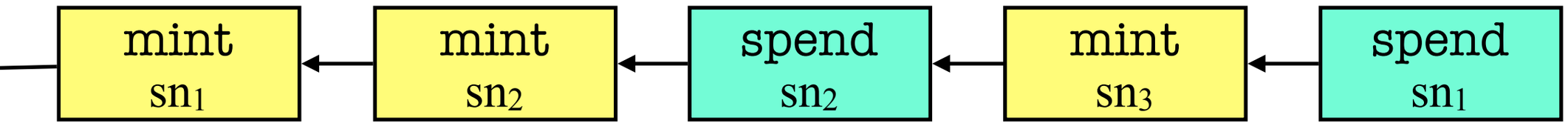
type 1



type 2

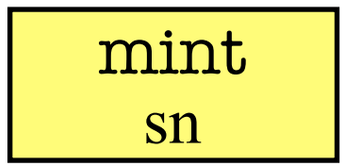
coin

Attempt #1: plain serial numbers

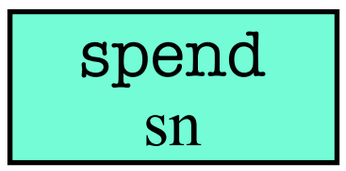


view of blockchain

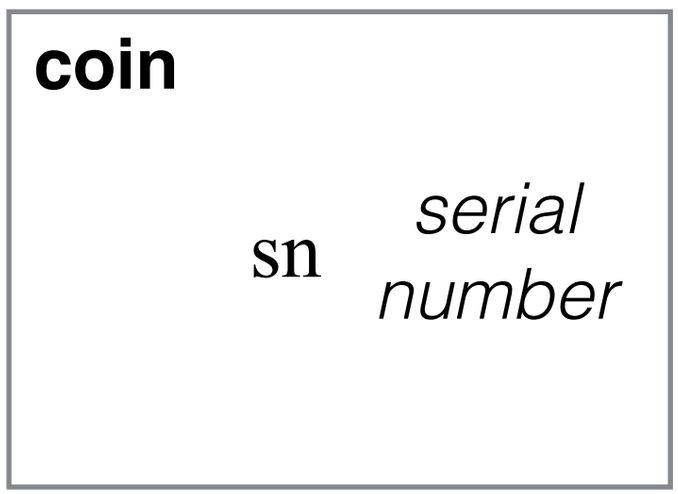
Transaction types



Consume 1 BTC to create a value-1 coin w/ serial number sn.



Consume the coin w/ serial number sn.



Good:

cannot double spend

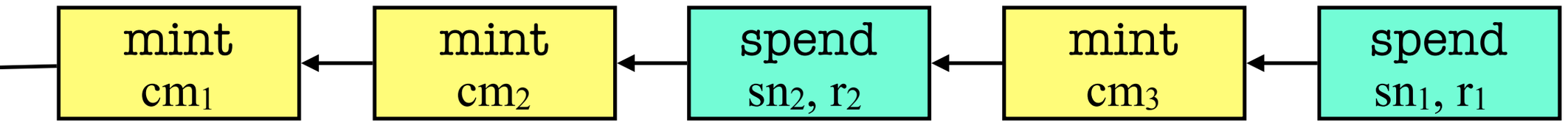
Bad:

spend linkable to its mint

anyone can spend!

…

Attempt #2: committed serial numbers

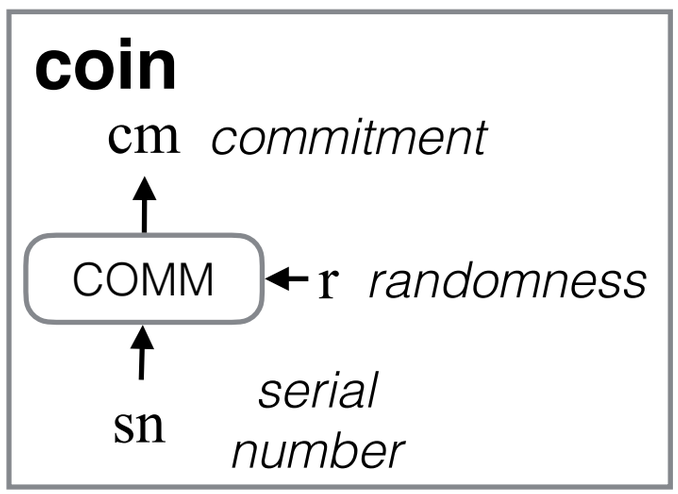


view of blockchain

Transaction types

mint cm — Consume 1 BTC to create a value-1 coin w/ commitment cm.

spend sn, r — Consume the coin w/ serial number sn, by revealing the secret randomness r.



Good:

cannot double spend

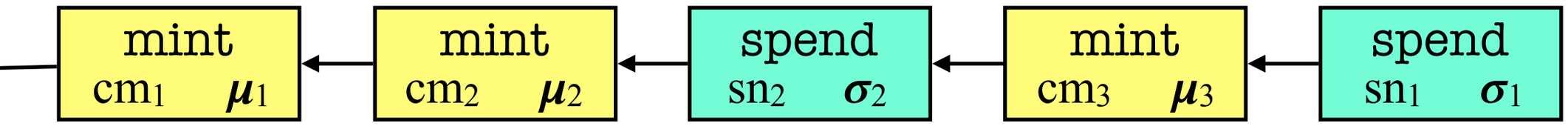
others can't spend my coins

Bad:

spend linkable to its mint

…

Attempt #3: ZKPoK of commitment



view of blockchain

Transaction types

mint
cm μ

Consume 1 BTC to create a value-1 coin w/ commitment cm.

Here is ZKP μ that for cm I know secret (sn,r) s.t. cm=COMM(sn;r).

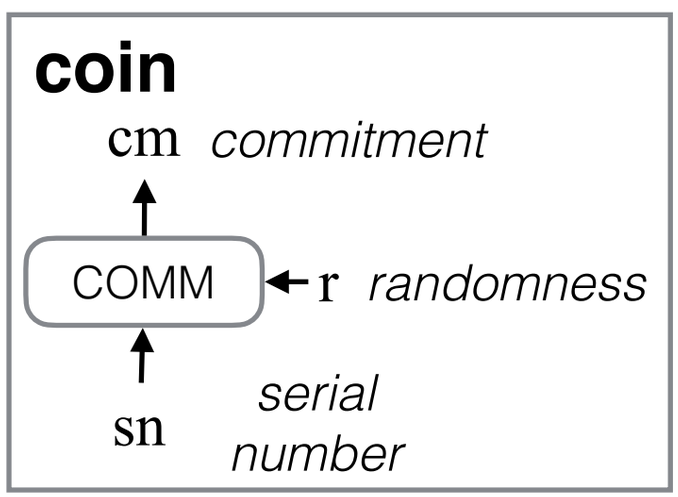
spend
sn σ

Consume the coin w/ serial number sn.

Here is ZKP σ that for sn I know secret randomness r s.t.

exists • cm ∈ "list of all prior commitments"

well-formed • cm=COMM(sn;r)



Good:

cannot double spend

others can't spend my coins

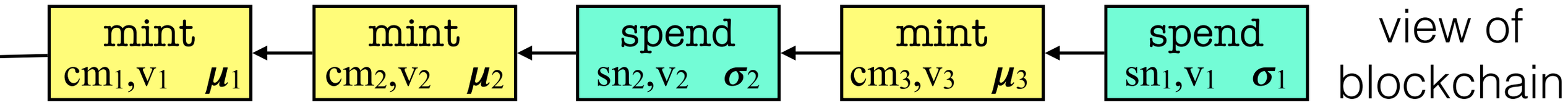
spend and mint unlinkable

Bad:

fixed denomination

…

Attempt #4: variable denomination



Transaction types

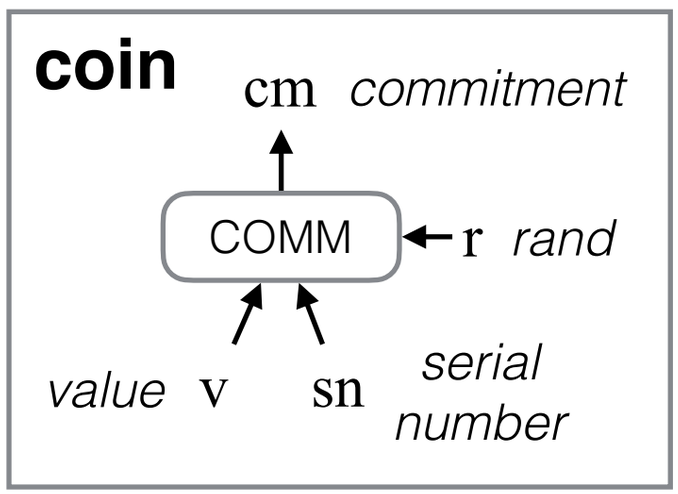
mint cm,v μ

Consume v BTC to create a value-v coin w/ commitment cm.
Here is ZKP μ that for (cm,v) I know secret (sn,r) s.t. cm=COMM(v,sn;r).

spend sn,v σ

Consume the value-v coin w/ serial number sn.
Here is ZKP σ that for (sn,v) I know secret (cm,r) s.t.

exists • cm ∈ "list of all prior commitments"
well-formed • cm=COMM(v,sn;r)



Good:

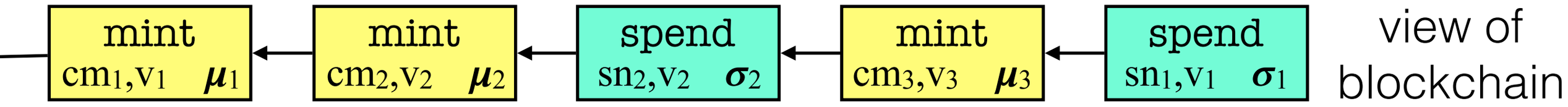
cannot double spend
others can't spend my coins
spend and mint unlinkable
variable denomination

Bad:

only hides sender

…

Attempt #5: payment addresses

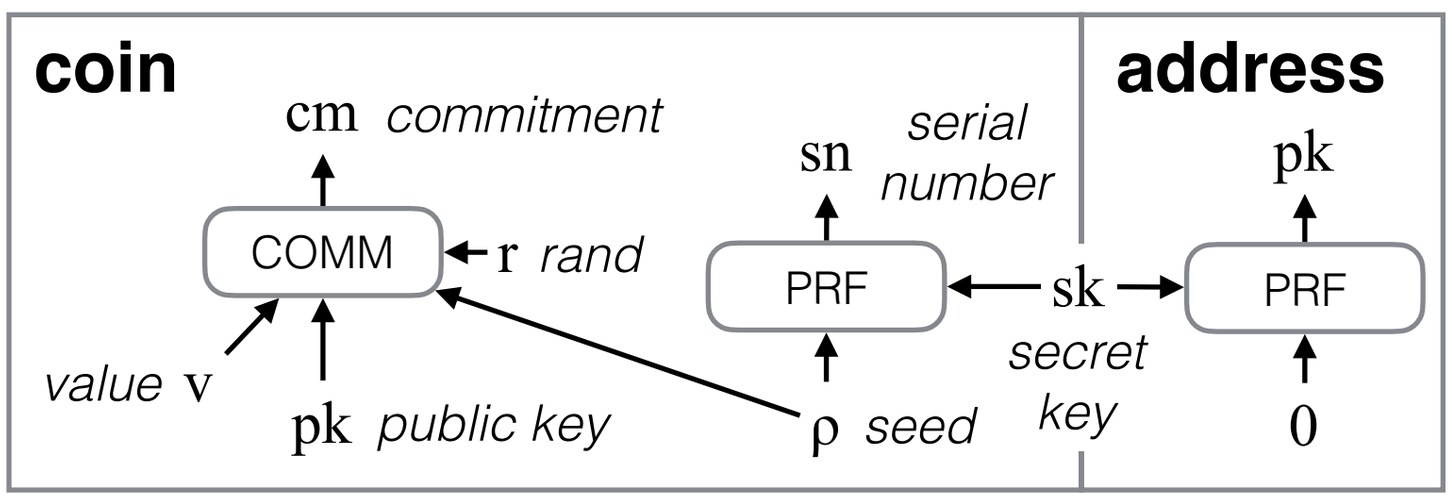


Transaction types

mint cm,v μ — Consume v BTC to create a value-v coin w/ commitment cm.
Here is ZKP μ that for (cm,v) I know secret (sn,r) s.t. cm=COMM(v,sn;r).

spend sn,v σ — Consume the value-v coin w/ serial number sn.
Here is ZKP σ that for (sn,v) I know secret (cm,r,ρ,pk,sk) s.t.

exists • cm ∈ "list of all prior commitments"
well-formed • cm=COMM(v,pk,ρ;r)
mine • sn=PRF(ρ;sk) & pk=PRF(0;sk)



Good:

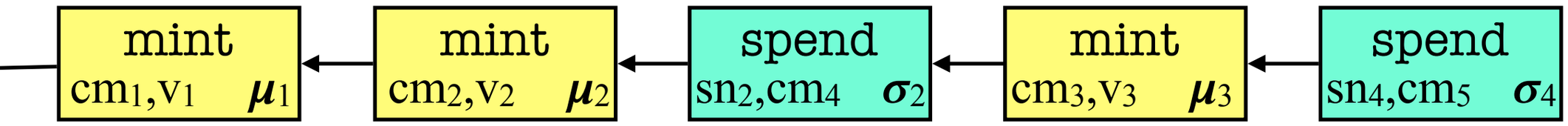
cannot double spend
others can't spend my coins
spend and mint unlinkable
variable denomination

Bad:

still only hides sender

…

Attempt #6: direct payments



view of blockchain

Transaction types

mint cm,v μ

Consume v BTC to create a value-v coin w/ commitment cm.
Here is ZKP μ that for (cm,v) I know secret (sn,r) s.t. cm=COMM(v,sn;r).

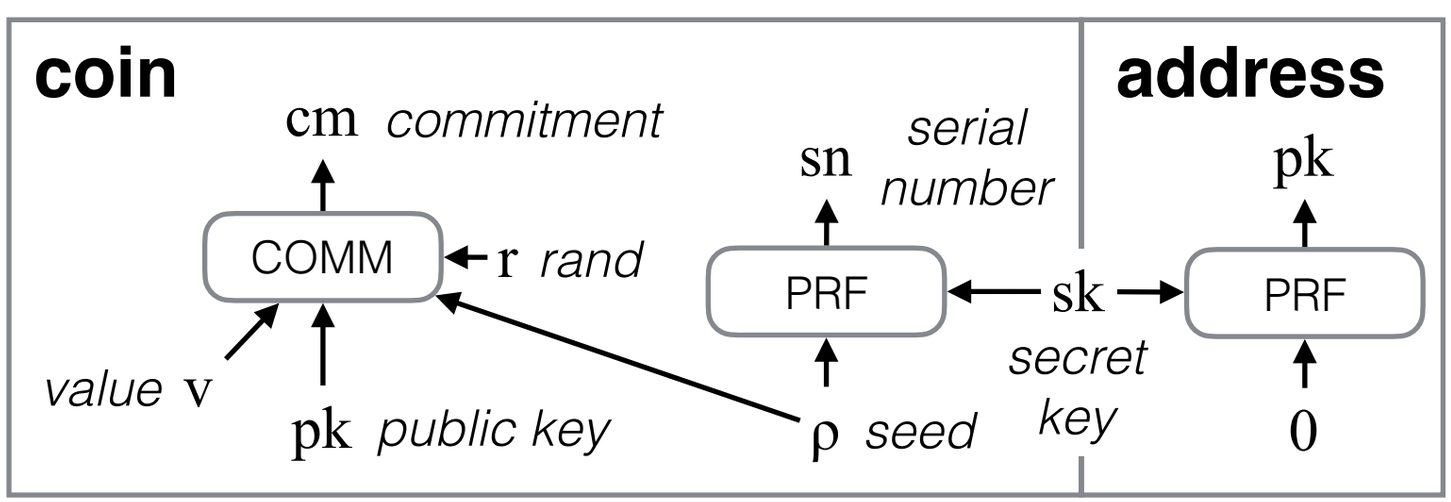
spend sn^A,cm^B σ

Consume coin w/ serial number sn^A & create coin w/ commitment cm^B.
Here is ZKP σ that for (sn^A,cm^B) I know secret (cm^A,v^A,r^A,ρ^A,pk^A,sk^A) s.t.

exists • $cm^A \in$ "list of all prior commitments"
well-formed • $cm^A=COMM(v^A,pk^A,\rho^A;r^A)$
mine • $sn^A=PRF(\rho^A;sk^A)$ & $pk^A=PRF(0;sk^A)$
well-formed • $cm^B=COMM(v^B,pk^B,\rho^B;r^B)$
same value • $v^A=v^B$

(cm^B,v^B,r^B,ρ^B,pk^B)

send out-of-band
or via blockchain

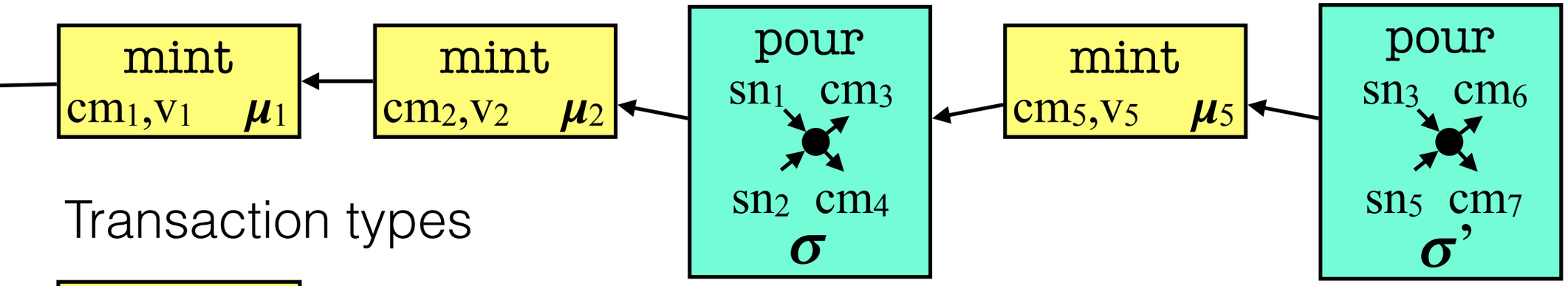


Good:

cannot double spend
others can't spend my coins
spend and mint unlinkable
variable denomination
hides sender, receiver, amt

Bad: join and split coins?

Sketch of Final Design



view of blockchain

Transaction types

mint
cm,v μ

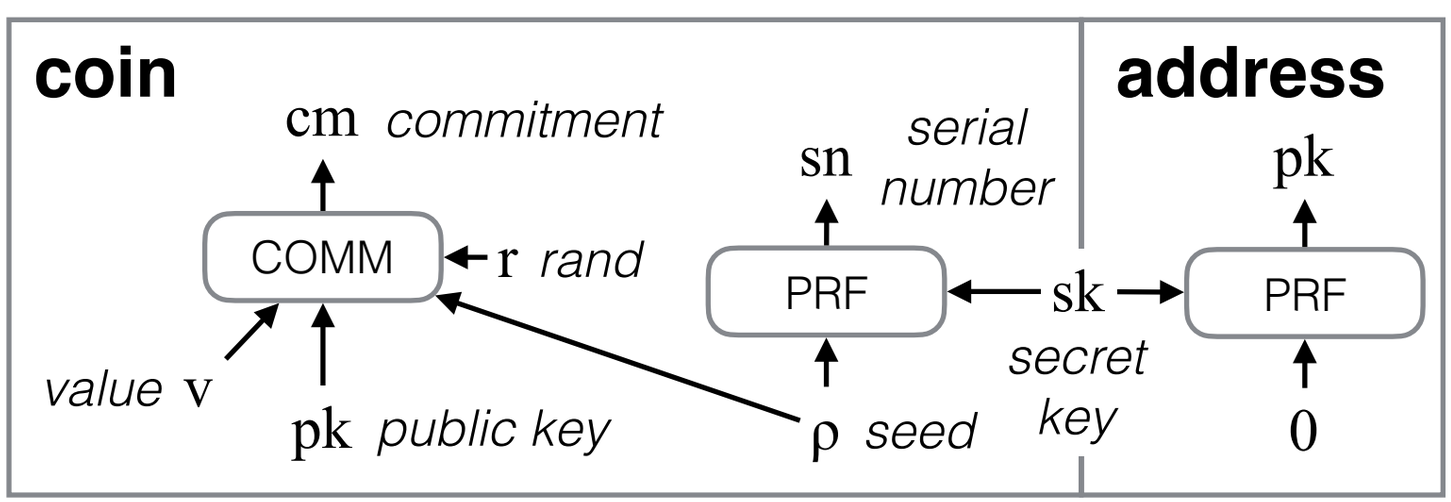
Consume v BTC to create a value-v coin w/ commitment cm.
Here is ZKP $\boldsymbol{\mu}$ that for (cm,v) I know secret (sn,r) s.t. cm=COMM(v,sn;r).

pour
sn^A cm^C
sn^B cm^D
$\boldsymbol{\sigma}$

Consume (my) **input** coins w/ serial numbers sn^A and sn^B in order to create two **output** coins (maybe not mine) w/ commitments cm^C and cm^D.

Here is ZKP $\boldsymbol{\sigma}$ that I know secrets that demonstrate that

- the input coins were minted at some point in the past,
- the output coins are well-formed,
- balance is preserved.



Single tx type for:

- ✓ simple payments
- ✓ join coins
- ✓ split coins
- ✓ making change
- ✓ pay transaction fees

Research → Real World

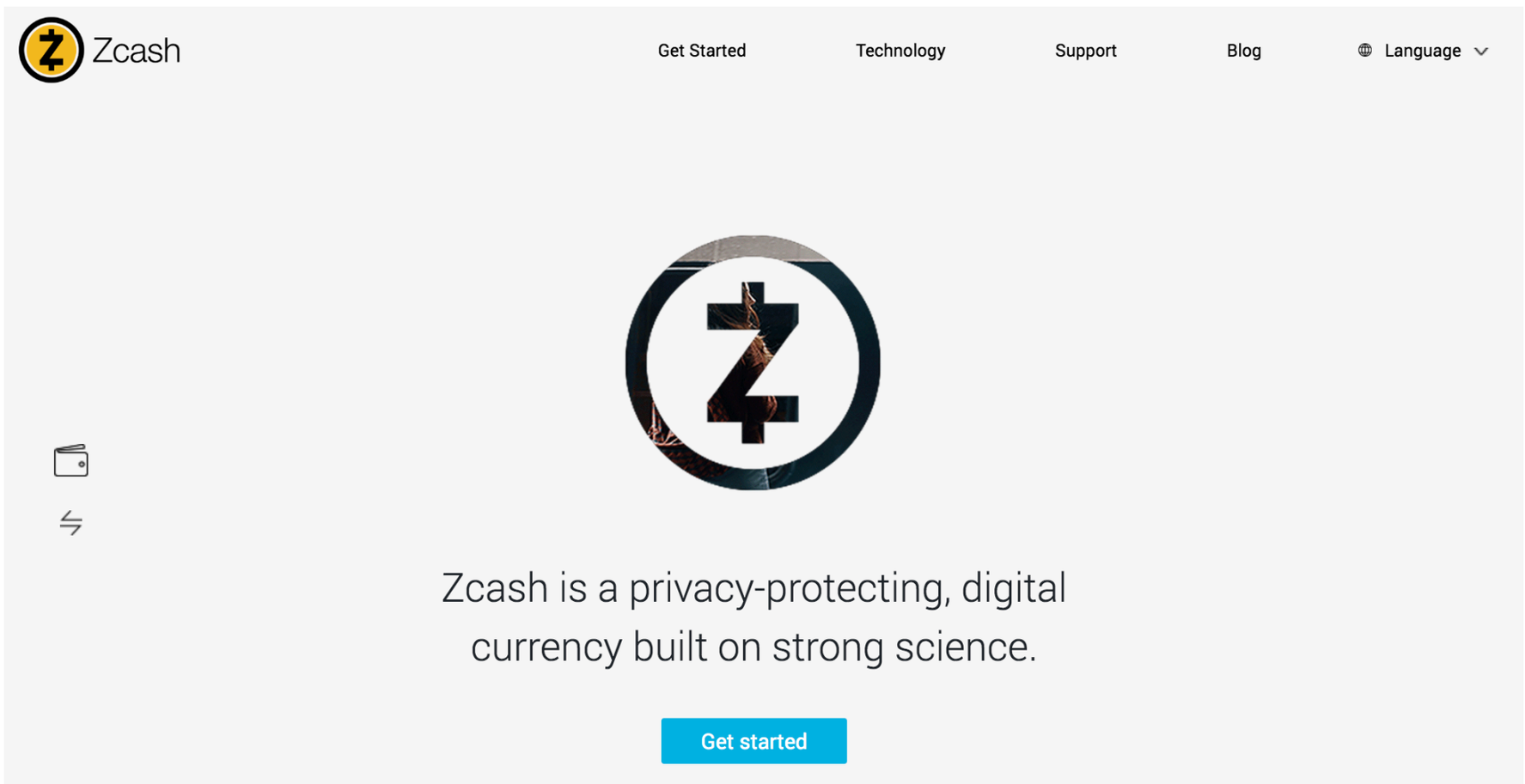
2013.11: design of Zerocash protocol

2014.05: proof-of-concept implementation of Zerocash protocol

2015.02: commercial venture (Electric Coin Company)

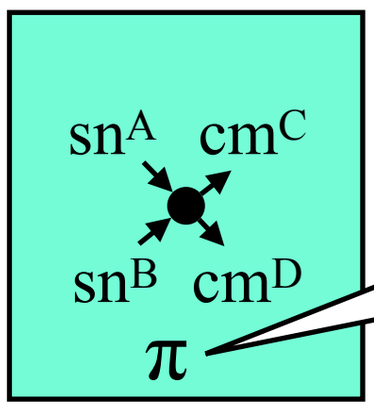
2016.10: launch of ZCASH

The first large-scale deployment of ZKP technology.



Frontiers

Beyond Simple Payments



I'm consuming my **unspent** coins in order to create new coins in a way that **value is preserved**.
I'm not revealing the value, sender, or receiver.

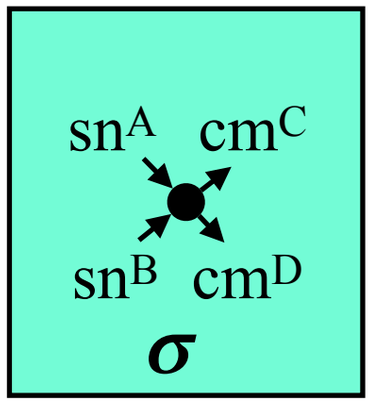
& the receiver was a 501(c) organization but I am not revealing which one

& the value transfered is less than 10K

Exciting research direction:

Which policies are desirable (and feasible!) to balance privacy/fungibility and oversight/integrity?

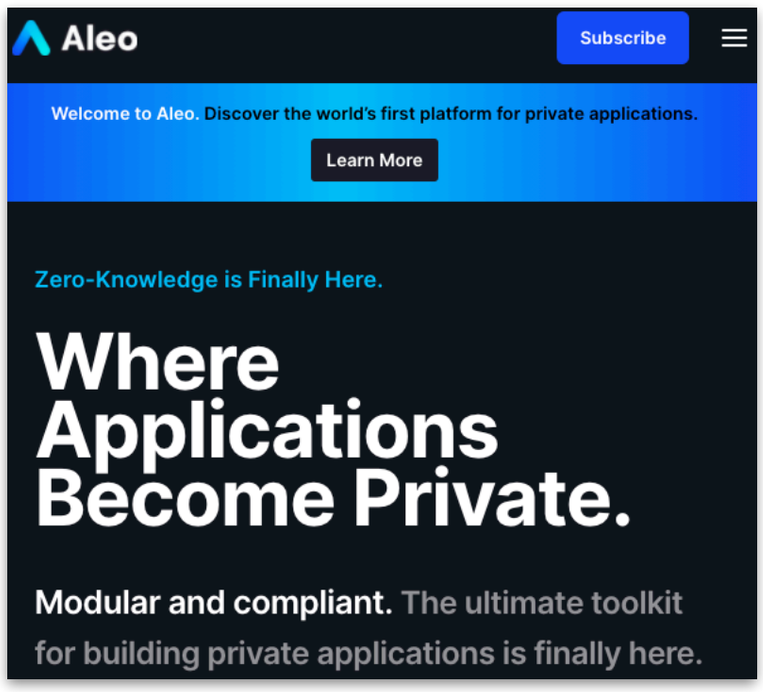
Recent Work:
Zero knowledge EXEcution (ZEXE)



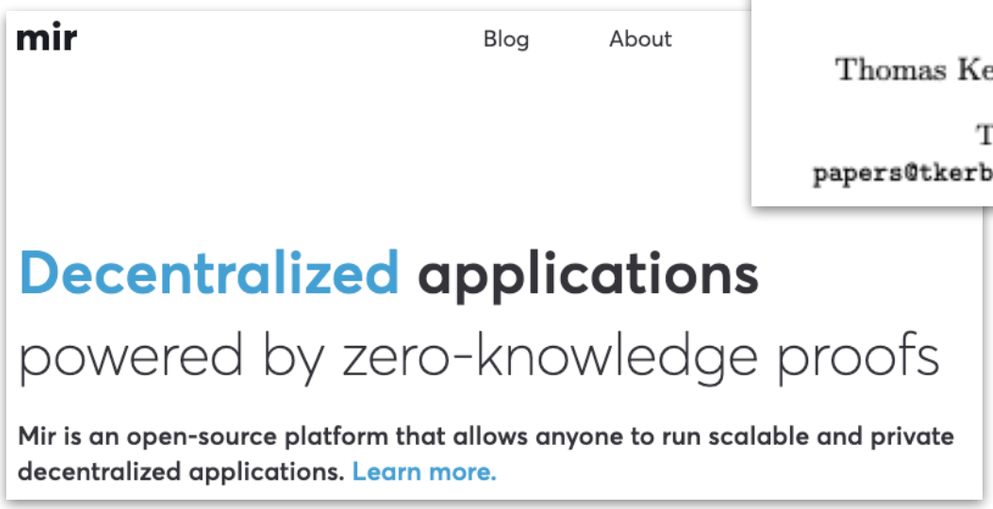
Each transaction contains:
- serial numbers of **consumed** (input) **records**
- commitments of **created** (output) **records**

A zkSNARK attesting that the output records were created according to a **private offline computation**.

The transaction hides the record's contents, owners, and thei relation. (Data and function privacy.) Think "privacy-preserving Bitcoin scripts".



Exciting real world applications!




KACHINA – Foundations of Private Smart Contracts

Thomas Kerber, Aggelos Kiayias, and Markulf Kohlweiss

The University of Edinburgh and IOHK
papers@tkerber.org akiayias@ed.ac.uk mkohlwei@ed.ac.uk

Thanks!