



# CS 5594: BLOCKCHAIN TECHNOLOGIES

Spring 2024

THANG HOANG, PhD

## PRIVACY-PRESERVING BLOCKCHAIN

Anonymity

Zerocoin

Zerocash

# Anonymity Basics

Most slides derived from the original ones of “Bitcoin and Cryptocurrency Technologies” book by Arvind Narayanan, Joseph Bonneau, Edward W. Felten, Andrew Miller, Steven Goldfeder and Jeremy Clark

# Case Study: Bitcoin

Some say Bitcoin provides anonymity

*“Bitcoin is a secure and anonymous digital currency”*

— WikiLeaks donations page

Others say it doesn't

*“Bitcoin won't hide you from the NSA's prying eyes”*

— Wired UK

# Anonymity Basics

**Literally:** anonymous = without a name

Bitcoin addresses are public key hashes rather than real identities

Computer scientists call this pseudonymity

**Anonymity = pseudonymity + unlinkability**



Different interactions of the same user with the system should not be linkable to each other

# Unlinkability Definition

Many blockchain (e.g., bitcoin) services require real identity

Linked profiles can be deanonymized by a variety of side channels

Profile lookup, active time of users, etc.

## What is unlinkability?

Hard to link different addresses of the same user

Hard to link different transactions of the same user

Hard to link sender of a payment to its recipient

# Quantifying Anonymity

Complete unlinkability (among all addresses/transactions) is **hard**

Anonymity set: A set of transactions that an adversary cannot distinguish from your transaction

Adversary knows you made a transaction, they can only tell that it's one of the transactions in the set, but not which one it is

Goal: Maximize the set

How to calculate anonymity set

No general formula, need to analyze each protocol/system case-by-case

Define adversary model

Reason carefully about: what the adversary knows, does not know, and cannot know

# Taint Analysis

Intuitive analysis of anonymity in Bitcoin without rigorous definitions

Calculate how “related” two addresses are

If transactions from address S always end up at address R (no matter how they are directed), then (S,R) has a high taint score

Not a good measure of anonymity

You may have low taint score but, in fact, low degree of anonymity

Adversary is smarter than you may thought

Exploit various strategies to deanonymize

Example – timing attacks



# Anonymity Dilemma

Public blockchains are totally, publicly, and permanently traceable

Without anonymity, privacy is much worse than centralized services

However, achieving anonymity is a dilemma

**Good uses:** hiding sensitive information (e.g., salary, private contract/business)

**Bad uses:** criminal activities (e.g., money laundering)

# Anonymity Dilemma

Can we keep only the good ones?

Common conundrum in computer security and privacy:

Uses that are very different morally are pretty much the same technologically

In fact, the best is to separate technical anonymity properties of systems from legal principles

# Similar Dilemma: Tor

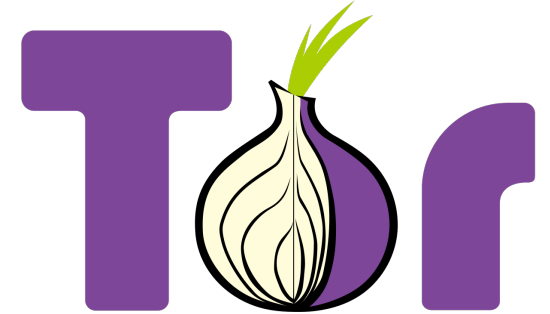
Anonymous communication network

Sender and receiver of message unlinkable

Used by:

- Normal people
- Journalists & activists
- Law enforcement
- Malware
- Child pornographers

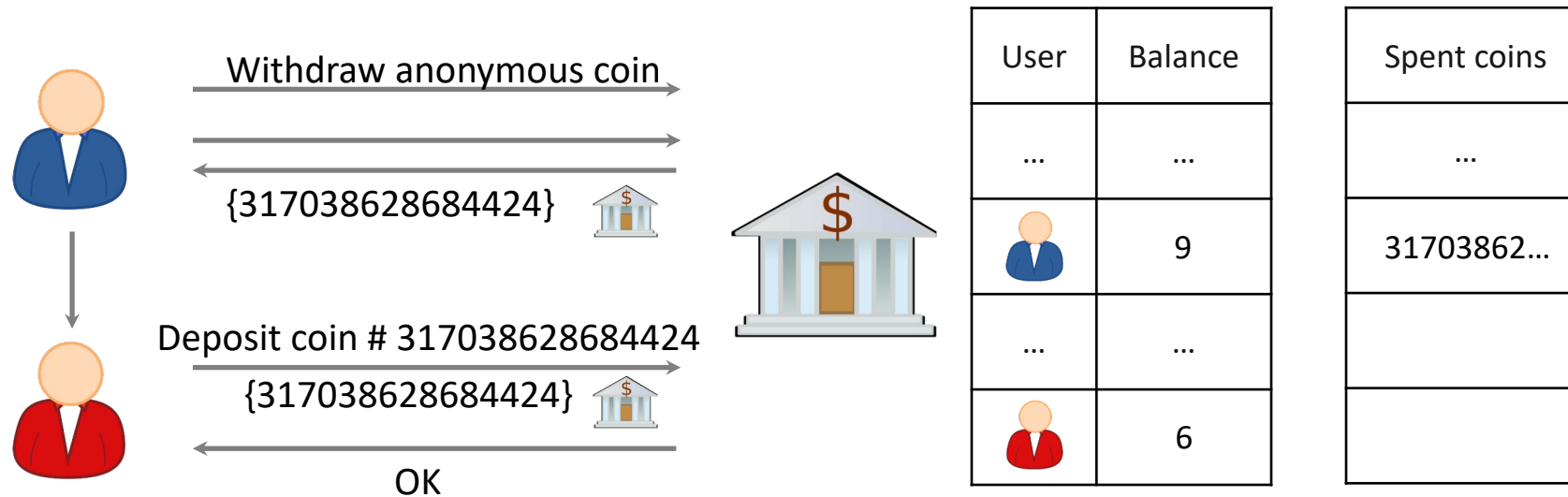
Funded by (among others): U.S. State Department



# Anonymous E-cash

Blind signature by David Chaum, 1982

Two-party protocol to create digital signature without signer knowing the input



Bank cannot link the two users

(Stay tuned to the next lecture)

# Blind Signature

Based on RSA

## Two-party protocol

User A has message  $m$  to be signed

User B has RSA public key  $(n, e)$  and secret key  $d$

## Protocol (simplified)

A chooses random (**blinder**)  $r$  in  $\mathbb{Z}_n^*$  and asks B to sign  $M = m \cdot r^e \pmod n$

B returns  $y = M^d = m^d \cdot r \pmod n$

A sets the signature of  $m = y \cdot r^{-1} \pmod n$

Correctness?

Blindness?

# Anonymity vs. Decentralization

Anonymity and decentralization are in conflict

Interactive protocols with a central authority are hard to decentralize

Decentralization often achieved via public traceability to enforce security

Public traceability is a threat to anonymity

# Deanonymization in Bitcoin

# Deanonymization in Bitcoin

## Bitcoin

**Bitcoin** is a secure and anonymous digital currency. Bitcoins cannot be easily tracked back to you, and are safer and faster alternative to other donation methods. You can send BTC to the following address:

13DFamCvSxG8EG16VyXzdpfqxyooifswYx



Various sites offer a service to exchange other currency to/from Bitcoins. There are also services allowing trades of goods for Bitcoins. Bitcoins are not subject to central regulations and are still gaining value. To learn more about Bitcoins, visit the website (<http://bitcoin.org>) or read more on [Wikipedia](#).

To generate a new, private address for your donation, click the refresh button above.




*Snippet from Wikileaks donation page*



# Deanonymization in Bitcoin

## Bitcoin

**Bitcoin** is a secure and anonymous digital currency. Bitcoins cannot be easily tracked back to you, and are safer and faster alternative to other donation methods. You can send BTC to the following address:

16nLrMAQma6GJ4AavfxXLaZoeCHBBqqzX3 

Various sites offer a service to exchange other currency to/from Bitcoins. There are also services allowing trades of goods for Bitcoins. Bitcoins are not subject to central regulations and are still gaining value. To learn more about Bitcoins, visit the website (<http://bitcoin.org>) or read more on [Wikipedia](#).

To generate a new, private address for your donation, click the refresh button above.



*Snippet from Wikileaks donation page*

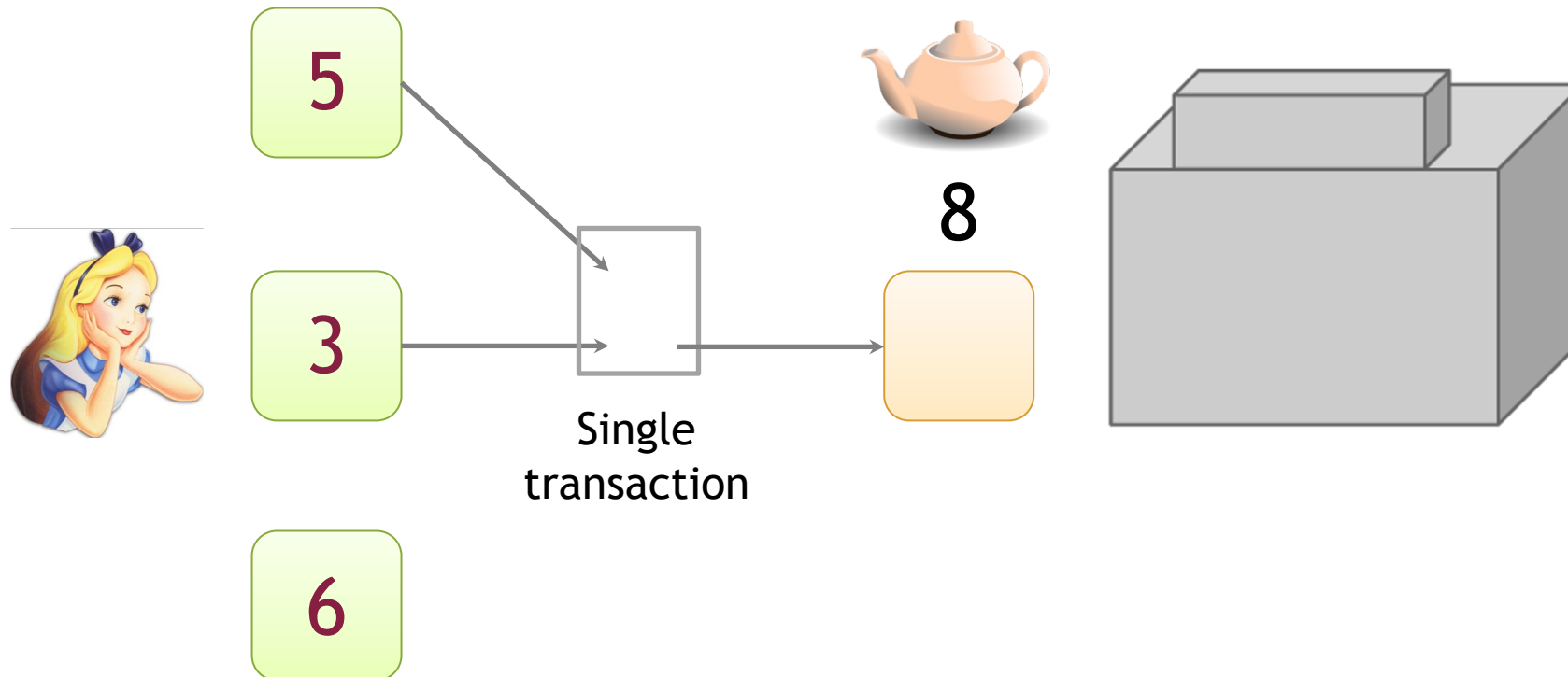
**Question: Can these two addresses be linked??**

# Fresh Address

Best practice: Create new address per transaction and always receive at fresh address

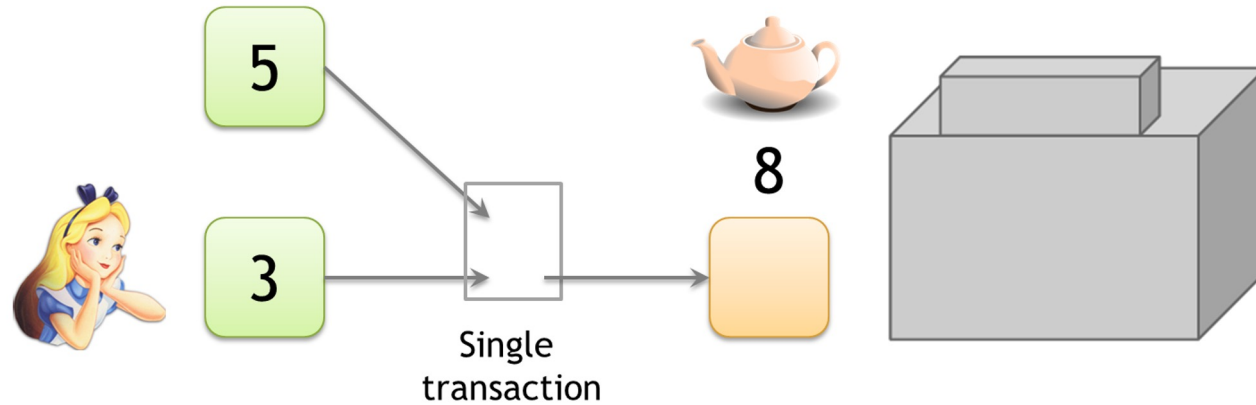
Is it unlinkable?

Example: Alice buys a teapot at Big box store



# Example

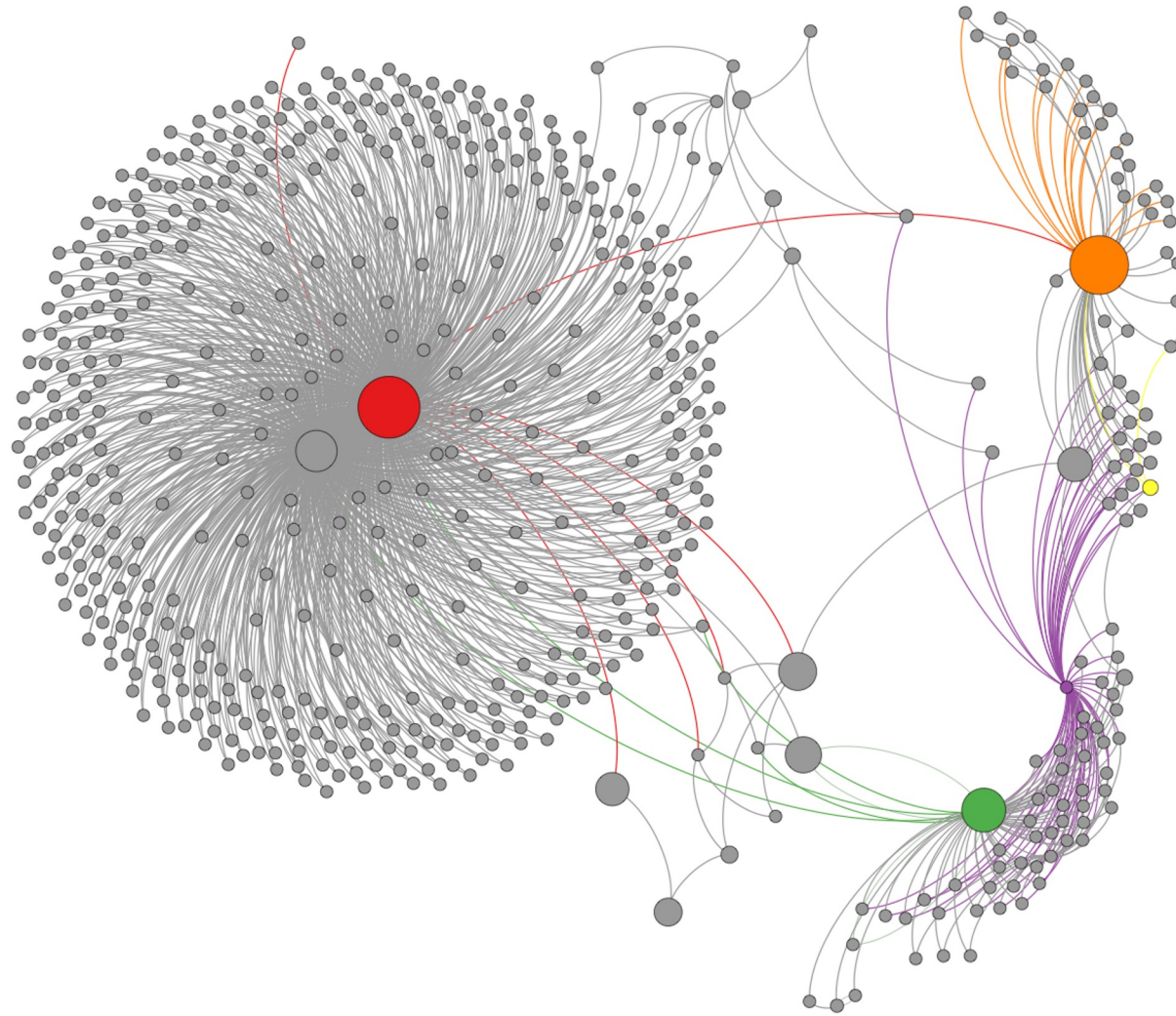
Shared spending is a clear evidence of joint control of different input addresses



Addresses can be linked transitively

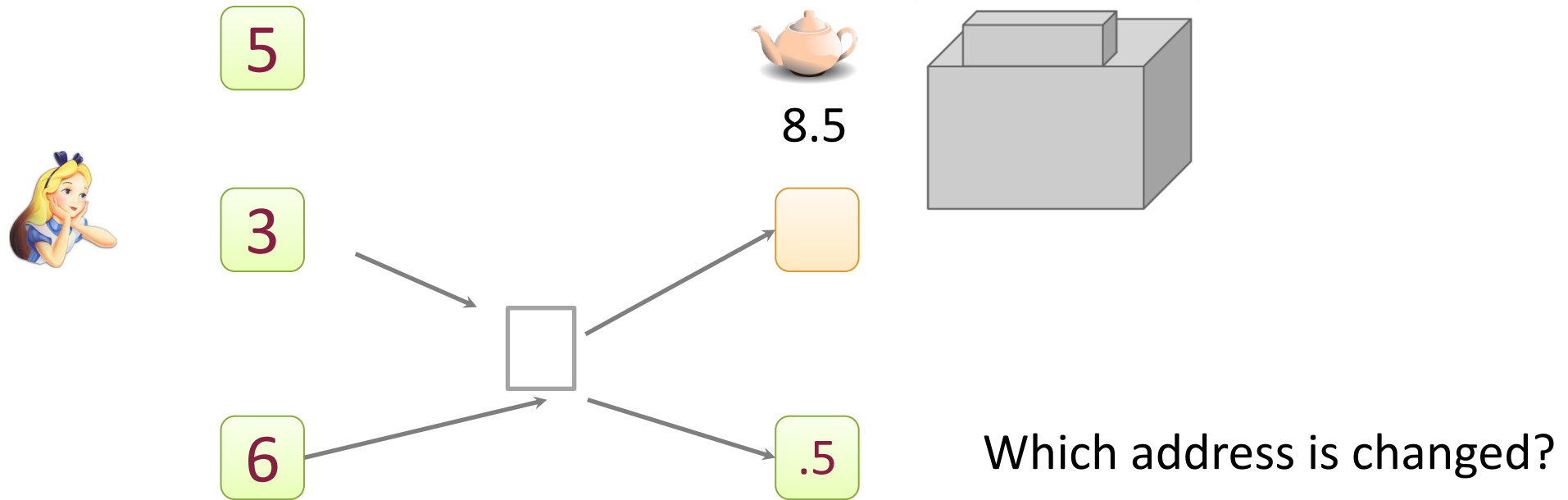
Adversary can repeat and link an entire cluster of transactions belonging to the same entity

# Clustering of addresses



F. Reid and M. Harrigan, *An Analysis of Anonymity in the Bitcoin System*, PASSAT 2011

# Change address



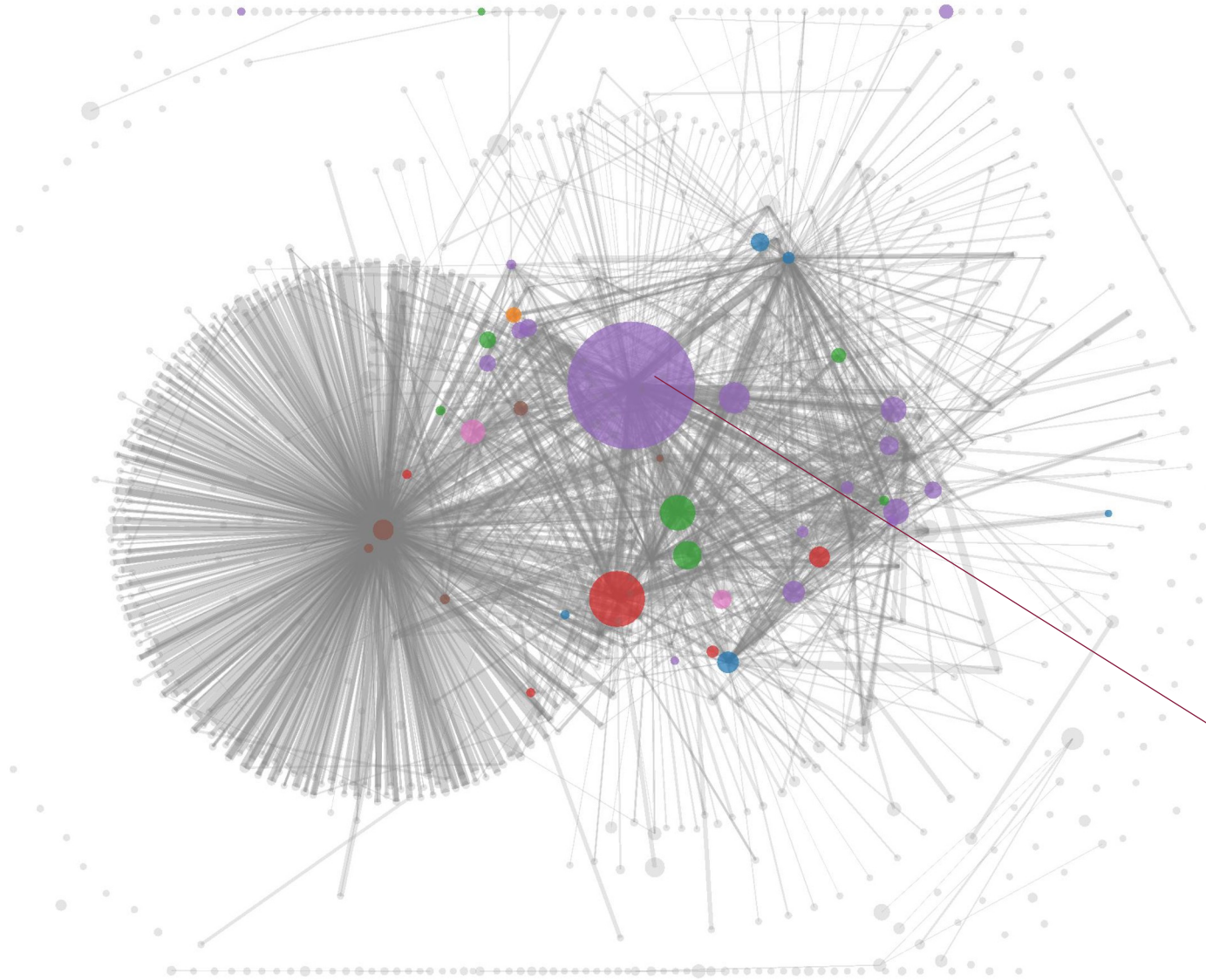
## “Idioms of use”

Idiosyncratic features of wallet software: Generates a fresh address when a change of address is required

- Change addresses are the ones that have never appeared in the blockchain
- nonchange outputs are not new and may have appeared previously

Adversary exploit this to distinguish change address and link with input addresses

# Shared spending + Idioms of Use



Graph is not labeled— identities are not yet attached to the clusters

Educated guess based on bitcoin community:

**Mt.Gox** was largest Bitcoin exchange

Addressed controlled by **Mt.Gox**??

# Tagging by Transacting

Conduct actual transaction to service providers

One of their addresses, which will soon end up in the block chain

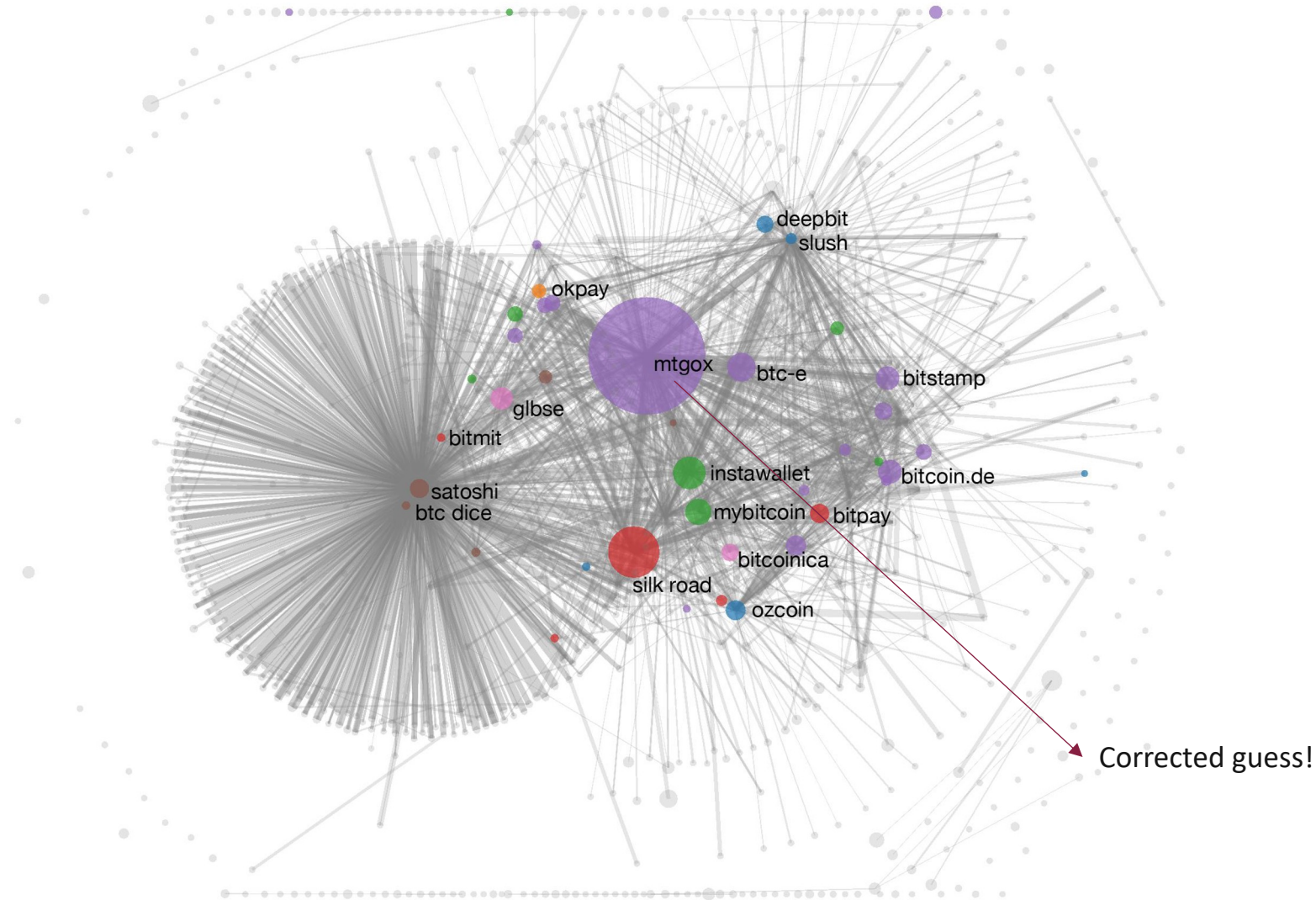


S. Meiklejohn et al., *A Fistful of Bitcoins: Characterizing Payments Among Men with No Names*, IMC 2013

344 actual transactions

- Mining pools
- Wallet services
- Exchanges
- Vendors
- Gambling sites

# Shared spending + Idioms of Use



S. Meiklejohn et al., *A Fistful of Bitcoins: Characterizing Payments Among Men with No Names*, IMC 2013



# From Service Providers to Users

## Can we cluster individuals?

Connect little clusters corresponding to individuals to their real-life identities

- *Direct transacting*: Anyone who transacts with an individual knows at least one address belonging to that individual
- *Via service providers*: service providers ask users for their identities
- *Carelessness*: Post addresses in public forums (e.g., donation request)  
Create link between identity and address

Attacks on privacy become more effective with time

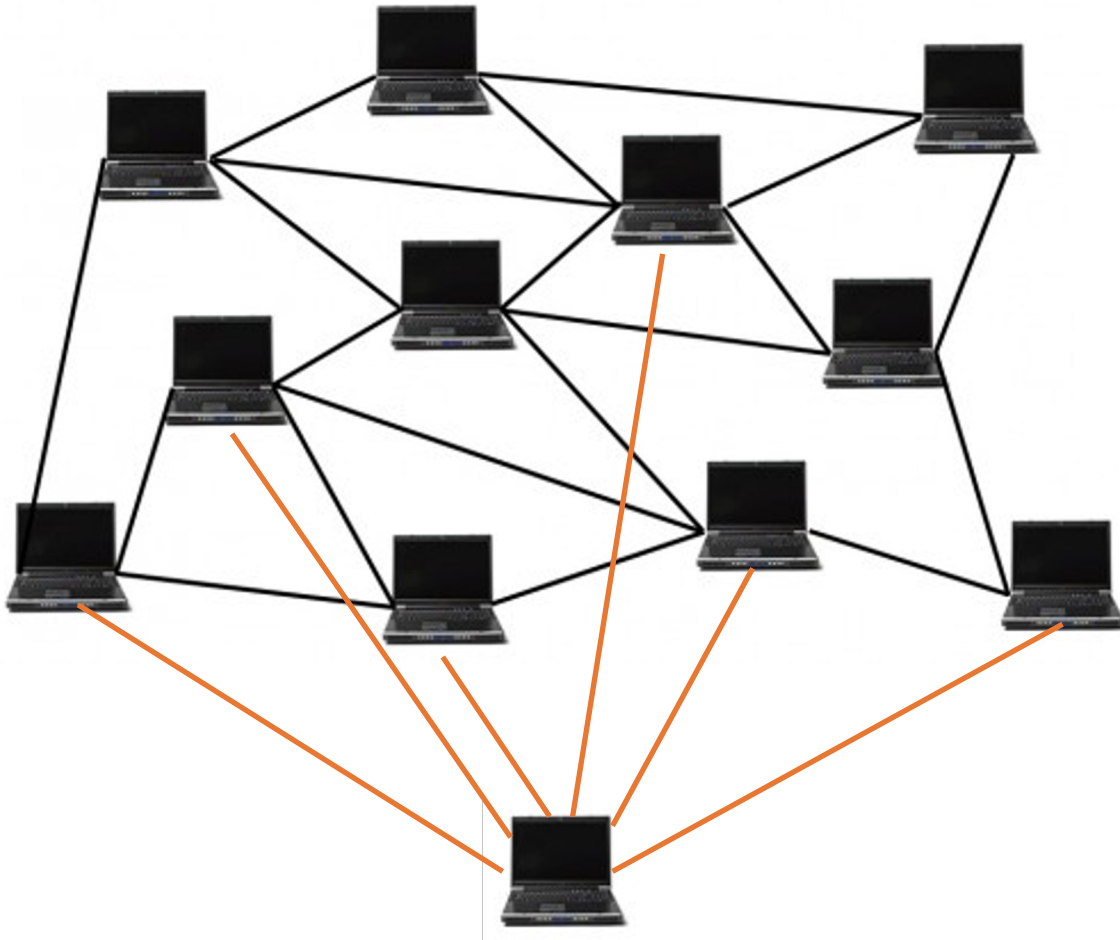
More data to analyze, more auxiliary information to exploit

Most deanonymization techniques are based on transaction graph analysis

# Network-Layer Deanonimization

Transaction graph analysis is based on application layer

Different approach: network layer



*“The first node to inform you of a transaction is probably the source of it”* – Dan Kaminsky  
Black Hat 2011 talk

# Solution: Tor

Tor is used for generic anonymous communication

There are some caveats

- Tor is intended for low-latency applications (e.g., web browsing)

  - Blockchain is high-latency

- Interaction between Tor protocol and on-top protocol may breach anonymity

  - Vulnerabilities found in Bitcoin-over-tor protocol

Mix-net might provide better anonymity

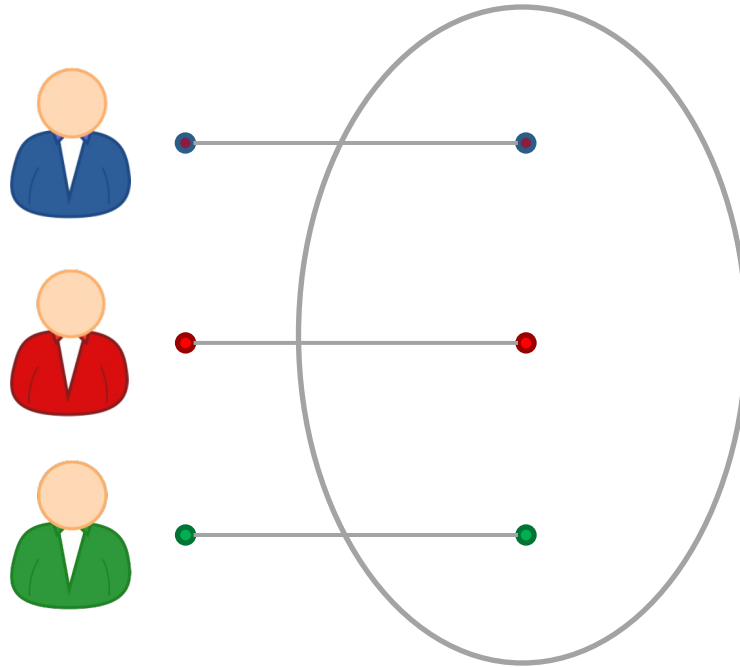
BUT Tor is what's deployed and works with large user base and intensively studied security

# Mix-Net

# Mix-Net

To protect anonymity, use an intermediary

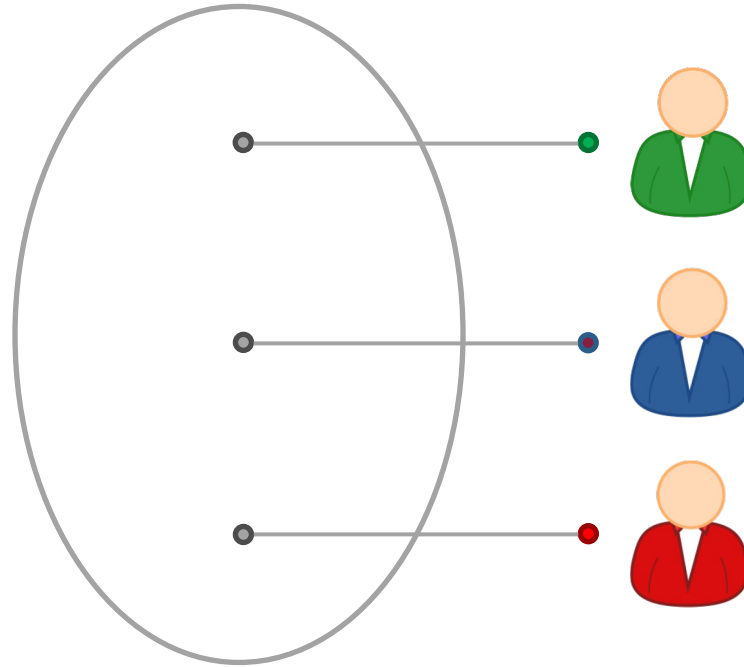
Post transactions to the intermediary...



# Mix-Net

Online wallets  
can do this

Do they provide  
anonymity?!



# Dedicated Mixing Services

Send your coin to the mix, and tell the service the destination address

An ideal dedicated mixing service will

- Promise not to keep records

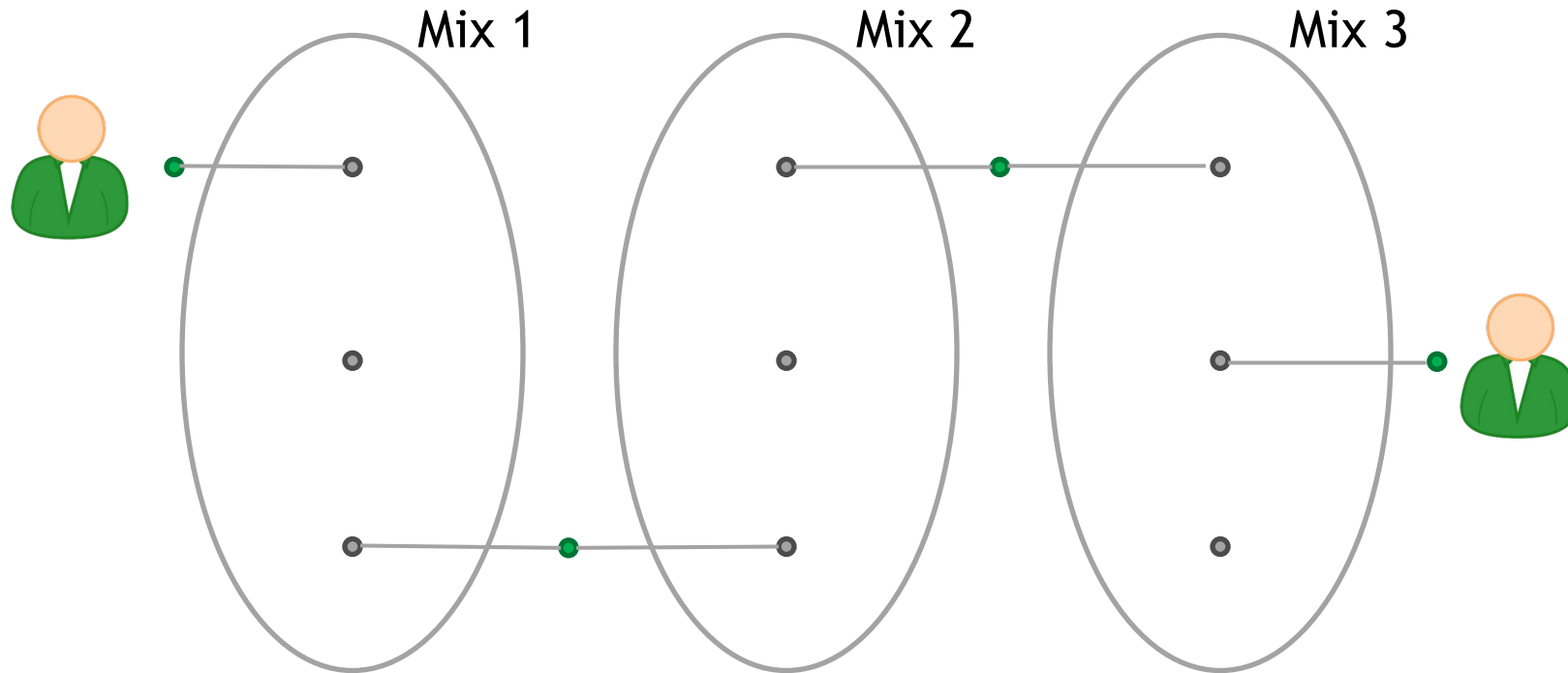
- Don't ask for your identity

Trust?

# Design Principles of Mixing Service

## 1. Use a series of mixes

Mixes should implement a standard API to make this easy



J. Bonneau et al., *Mixcoin: Anonymity for Bitcoin with accountable mixes*, Financial Cryptography 2014



## 2. Uniform transactions

In particular: all mix transactions must have the same value!

“*Chunk size*” – Difficult to select a single size in practice

Reasonable trade-off between efficiency and privacy

## 3. Client side must be automated

Reduce impact of side channel (e.g., timing attacks)

Example – Auto interact with mixers w.r.t fixed time

Privacy-friendly wallet software

# Design Principles of Mixing Service

## 4. Fees must be all-or-nothing

Mixing is a service – needs fee to operate

Problem: mix transactions cannot be in standard chunk sizes

Solution: probabilistic fees

0.1% mixing fee = mix will swallow chunk with 0.1% chance

Tricky to achieve in practice – need to convince users it does not cheat (honest probability, unbiased pseudorandom generator)

**Current mixes follow none of these principles**

J. Bonneau et al., *Mixcoin: Anonymity for Bitcoin with accountable mixes*, Financial Cryptography 2014

# Trusting Mixes?

Many mix services available, but low volumes and small anonymity sets

Many of them are malicious

- Users do not want to use

- Low transaction volume and hence poor anonymity

## Currently no reputable dedicated mix

*“Caution: Mixing services may themselves be operating with anonymity. As such, if the mixing output fails to be delivered or access to funds is denied there is no recourse. Use at your own discretion.”*— Bitcoin Wiki

## Solution?

- Stay in business, build up reputation

- Users can test for themselves

- Cryptographic “warranties”

# Decentralized Mixing

Replace mixing services with a P2P protocol by which a group of users can mix their coins.

## Advantages

- No bootstrapping problem
  - Don't have to wait for reputable centralized mixes to become available
- Theft impossible
  - Users get back coins equal to the one being put to be mixed
- Possibly better anonymity

# CoinJoin

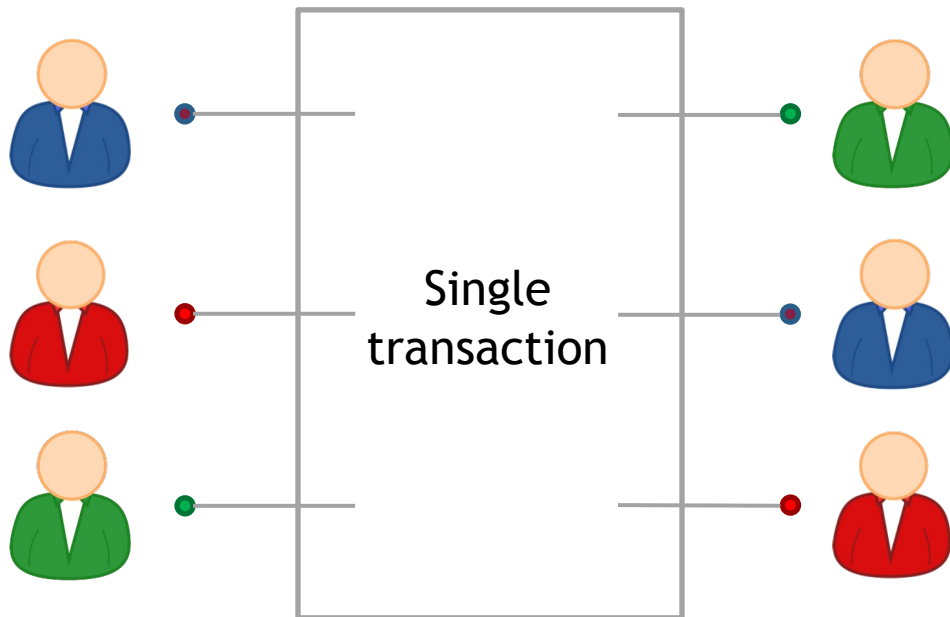
Proposed by Greg Maxwell, Bitcoin core developer

Main proposal for decentralized mixing

Allow a group of users to mix their coins with a single transaction

Each user provides an input and output address, and forms a transaction together

Randomized order of input and output addresses



Each signature is entirely separate

This is 1 mixing round

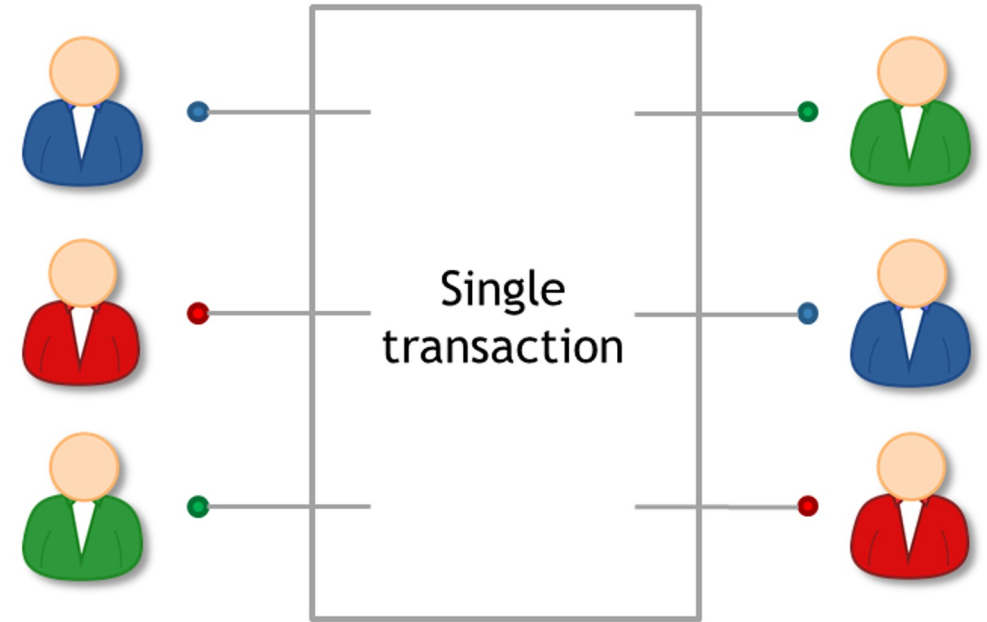
Extensible to multiple rounds to improve anonymity

Mixing principles from before apply on top of basic protocol

# CoinJoin

## Five steps of operations

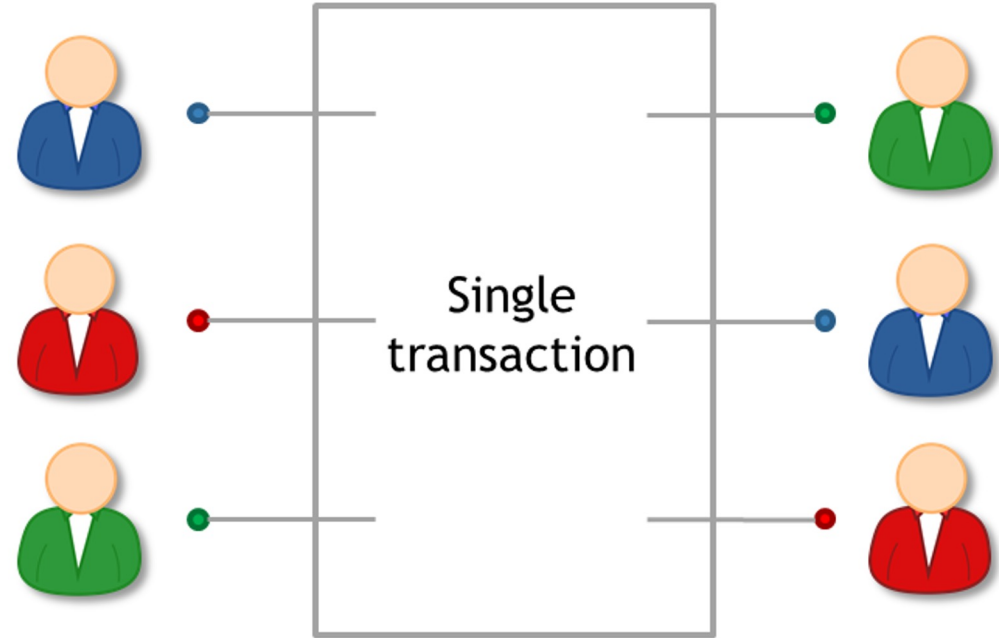
1. Find peers who want to mix
2. Exchange input/output addresses
3. Construct a transaction (by any peer)
4. Send it around, collect signatures  
(Before signing, each peer checks if her output is present)
5. Broadcast the transaction (by any peer)



# CoinJoin

## Some problems:

- How to find peers?
  - Use an untrusted server to let users connect and group together.
- Peers know your input-output mapping?  
(This is a worse than for centralized mixes)
- Denial of service?
  - Participate in the first phase of the protocol, providing input and output, but then refuse to sign in the second phase



Strawman solution:

1. exchange inputs
2. disconnect and reconnect over Tor
3. exchange outputs

Better solution:

Special-purpose anonymous routing mechanism



## Proposed solutions to deal with DoS

Impose cost to participate

Proof of work

Proof of burn

Identify and kick out malicious participants via special cryptographic protocols

Cryptographic “blame” protocol (T. Ruffing et al., *CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin*, PETS 2014)

# Side-Channel Leakages

Anonymity can be broken by side channel leakages

## Example:

Alice receives 43.12312 coins / week, and always transfers 5% to retirement account

"High-level flow" transfer pattern

No mixing strategy can hide the relationship between these two addresses

Merge avoidance (proposed by Mike Hearn) to seal high-level flow leakage

Instead of a single payment transaction

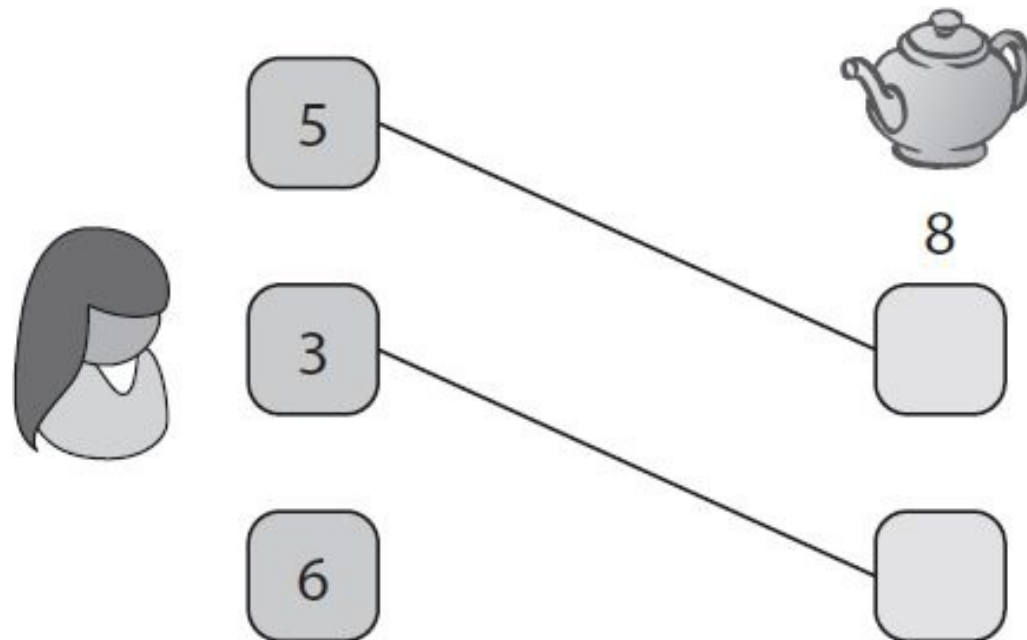
receiver provides multiple output addresses

sender avoids combining different inputs into a single output

# Merge Avoidance

Adversary might not be able to discern a flow if it is broken up into many smaller flows that aren't linked to one another

Merge avoidance prohibits cluster techniques relying on coins being spent jointly in a single transaction



# ZeroCoin

I. Miers et al., ZeroCoin: Anonymous Distributed E-Cash from Bitcoin, IEEE S&P 2013

# Zerocoin

Incorporate anonymity at the protocol level

Offer high level of privacy

Mixing capability baked into protocol

Cryptographic guarantee of mixing

No need to trust anybody – mixers, peers, intermediaries, miners, etc.

Qualitatively better than other mixing techniques with provable security

# Zerocoin

New concept: Basecoin

Basecoin: Bitcoin-like altcoin

Zerocoin: Extension of Basecoin

Key feature of anonymity: Basecoins can be converted into zerocoins and back

Basecoin is the currency for transaction

Zerocoin provides a mechanism to trade basecoins in for new ones that are **unlinkable** to the old ones

➤ Break linkability between original basecoin and new basecoin

# Zerocoin

A Zerocoin is a cryptographic proof that you owned a Basecoin and made it unspendable

Miners can verify these proofs

Gives you the right to redeem a new Basecoin (Somewhat like poker chips in casino)

## Questions

How to construct these proofs?

How to make sure each proof can only be “spent” once?

# Zero-Knowledge Proofs

A way to (mathematically) prove a statement without revealing any other information

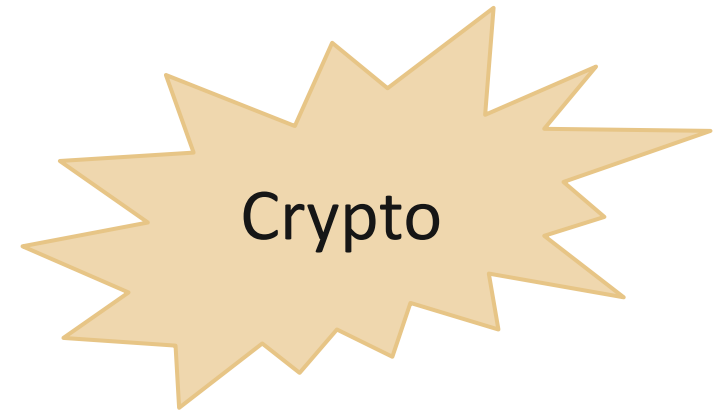
## Example:

“I know an input that hashes to **da39a3ee5e**”

“I know  $x$  such that  $H(x \parallel \text{other known inputs}) < \langle \text{target} \rangle$ ”

Proofs do not reveal any thing about  $x$

Let's assume ZK-proof as a black box for now





# Minting Zerocoins

Zerocoins come in standard denominations (i.e., 1 basecoin)

Anyone can make one!

They have value once put on the block chain

That costs 1 basecoin

How to mint a zerocoin?

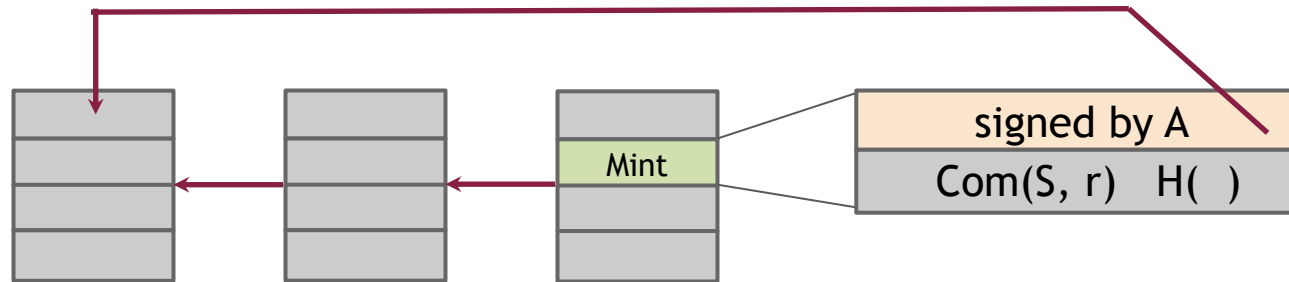
Cryptographic commitment scheme



# Minting a Zerocoin

## Three step to mint a Zerocoin

1. Generate serial number  $S$  and a random secret  $r$
2. Compute  $s = \text{Commit}(S, r)$ , the commitment to the serial number  $S$
3. Publish  $s$  on the block chain by creating a special Tx (Mint) with 1 basecoin as input  
(This burns a basecoin, making it unspendable, and creates a zerocoin)



(Let's keep  $S$  and  $r$  secret for now)


# Spending a Zerocoin

## To spend a Zerocoin

Create a special \*spend\* TX that contains  $S$  (to reveal  $S$ ) and a ZK-proof of the statement

“I know  $r$  such that  $\text{Commit}(S, r)$  is in the set  $\{c_1, c_2, \dots, c_n\}$ ”

set of zerocoins in the block chain



## Miner verifies:

- Your proof – to establish your ability to open one of zerocoin commitments on blockchain without actually opening it
- $S$  never been used in any previous “spend” TX

Output of \*spend\* TX will be a new basecoin

Output address should be your own address

# ZeroCoin is Anonymous

Once a ZeroCoin is spent, the serial number  $S$  becomes public

$S$  can only be redeemed once

One serial number per coin  $\rightarrow$  each coin can only be spent once

Key concept to anonymity:

Since  $r$  is secret, no one can figure out *which* zerocoin corresponds to serial number  $S$

Commit( $S, r$ )



$h_1$



$h_2$

...



$h_N$

No link b/w “mint” TX that committed to  $S$  and “spend” TX that later revealed  $S$  to redeem a coin

# Zerocoin is Efficient

The proof is a giant disjunction over all zerocoins

Yet the proof is relatively small!

$O(\log N)$  in size

Approx. 50 KB in practice

Require trusted setup

*I know  $r$  such that*

$$H(S, r) = h_1$$

**OR**

$$H(S, r) = h_2$$

**OR**

...

**OR**

$$H(S, r) = h_N$$

# Zerocash

Most slides derived from the ones of Prof. Alexandro Chiesa, one of the inventors of Zerocash

# Zerocash

Zerocoin without Basecoin

Two differences:

More efficient crypto techniques for ZK-proof

Proposal to run system without Basecoin

Zerocash is an untraceable e-cash

All transactions are zerocoins

Splitting and merging supported

Put transaction values inside the envelope (commitment)

Ledger merely records the existence of transactions

Immune to side-channel attacks against mixing

# ZeroCash vs. Zerocoin

Zerocoin: mix protocol on top of basecoin

Support regular transactions in case unlinkability is not needed

Augmented with computationally expensive TXs used only for mixing

Splitting and merging of values must be done in Basecoin

TX info is still public in basecoin

**TX history itself reveal significant sensitive information**

ZeroCash: everything is confidential (privacy-preserving cryptocurrency)

TX amounts are also inside the commitments and are NOT visible on blockchain  
(confidential transaction)

Cryptographic proofs to ensure splitting and merging done correctly

(No coin created out of thin air)



# Zerocash: Motivation

Many techniques developed to achieve unlinkability between transactions

Mixing, money laundering

Would unlinkability be enough to ensure privacy when detailed transactions stay public in the blockchain?

TX amount

TX time

Transaction history publicly stored **forever**

Methods of analysis only get **stronger**

# Zerocash

In Zerocash, the public ledger only records the existence of TXs, along with proofs

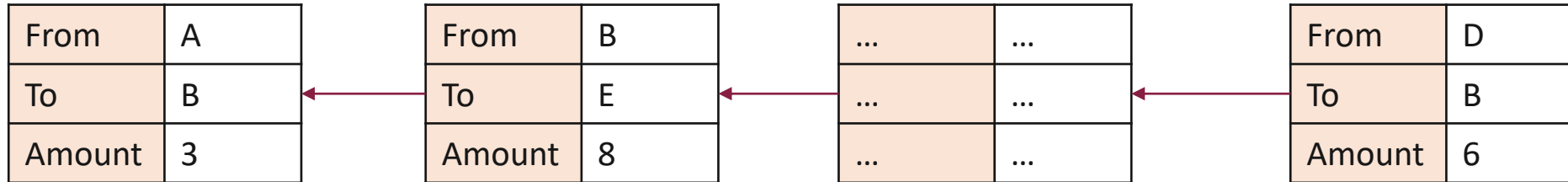
Plain addresses and values NOT stored on blockchain at any point

Miner don't need to know TX amount, but can verify properties/statements needed for the system to operate properly and correctly

TX amount only known by the sender and receiver of that TX

# Zerocash: Motivation

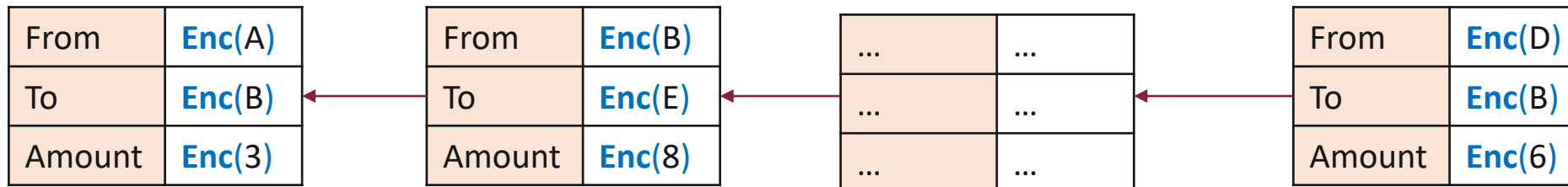
## Bitcoin-like blockchain (recap)



How does everybody know A has 1 BTC to spend?

Check that A receives it, and did not spend it

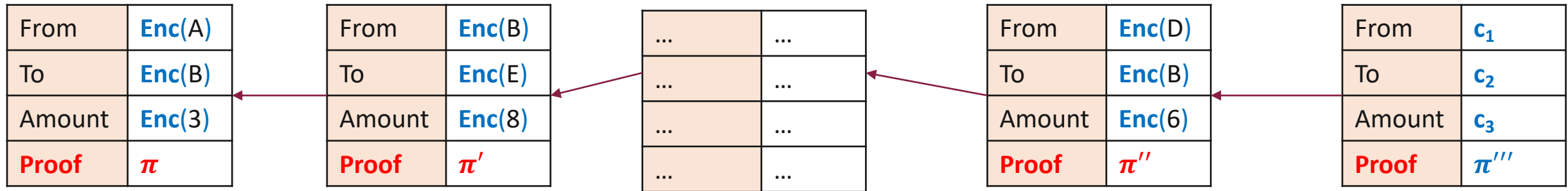
What if users encrypt transaction information to increase privacy?



How to check transaction validity?

Dilemma between privacy and accountability...

# Zerocash: High Level Idea



I am publishing three ciphertexts  $c_1, c_2, c_3$

They contain the encryptions of a sender address,  
a receiver address, and a TX amount, resp

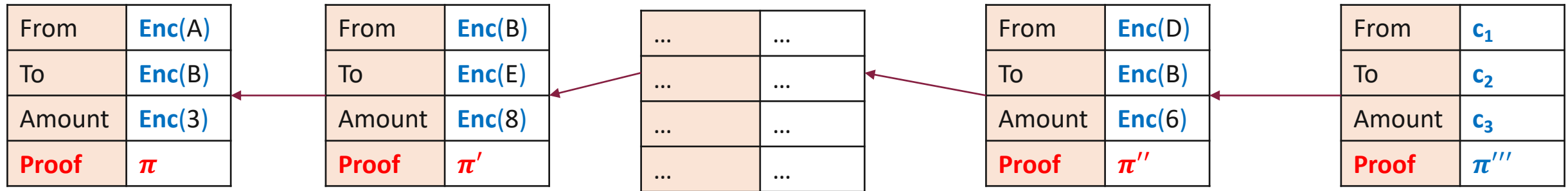
Moreover, the TX amount has not been double spent

I have generated a cryptographic proof  $\pi'''$  that all above are true

Q1: what kind of cryptographic proof?

Q2: what exactly is the statement being proved?

# Zerocash: High Level Idea



Q1: what kind of cryptographic proof?

zero-knowledge

(nothing is revealed by truth of statement)

succinct

(proof size is small, and efficient to verify)

non-interactive

(no need to interact with prover)

argument (proof)

(true statements have proofs, false ones do not)

of knowledge

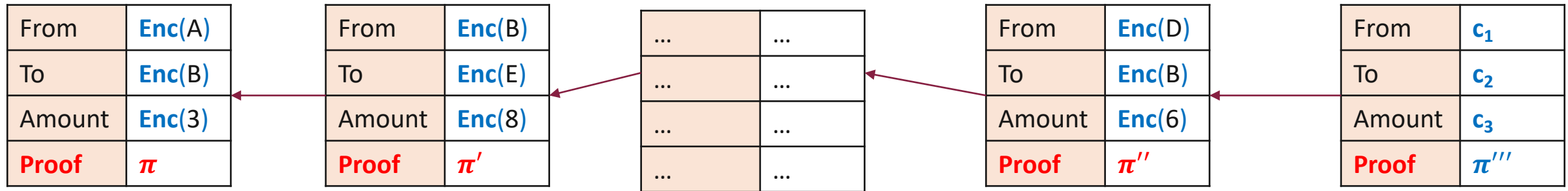
(technical... allows using crypto in statement)

---

zk-SNARK

(efficient constructions available at <https://libsnark.org>)

# Zerocash: High Level Idea

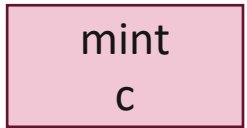
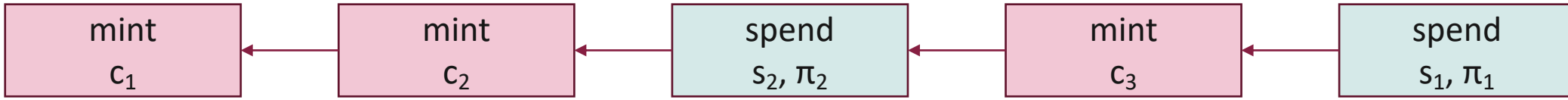


Q2: what exactly is the statement being proved?

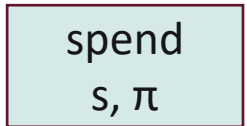
(a little bit complicated)

# ZeroCash: Preliminary Design

ZeroCoin (recap) – Zero-Knowledge proof of knowledge of commitment



Consume 1 basecoin to create value-1 zerocoin with commitment  $c$

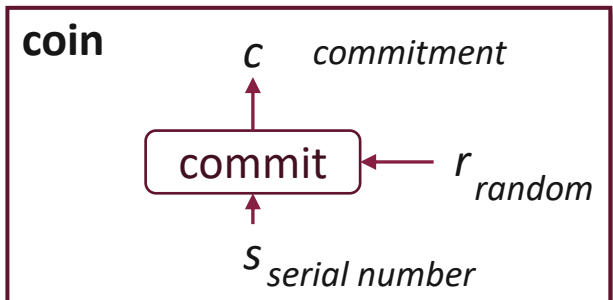


Consume the zerocoin w/ serial number  $s$

Here is a ZK proof  $\pi$  that I know  $r$  s.t.

**exists** •  $c \in$  set of commitments in blockchain

**well-formed** •  $c = \text{commit}(s, r)$



## Pros

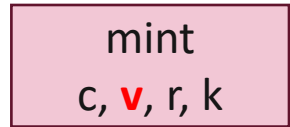
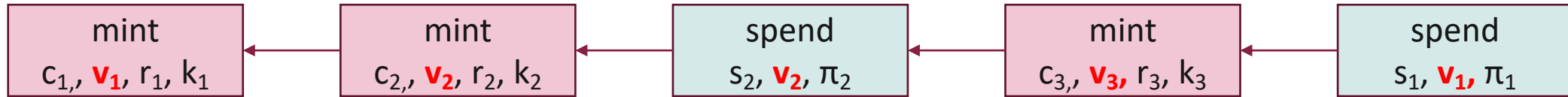
- No double spend
- Others cannot spend my coins
- Spend and mint TXs unlinkable

## Cons

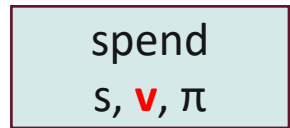
- Fixed denomination

# Zerocash Design

## Attempt #1: Variable denomination

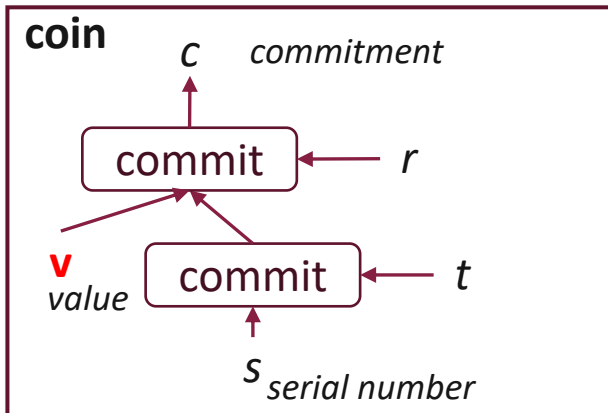


Consume **v**-basecoin to create value-**v** zerocoin with commitment  $c$



Consume a **v**-value coin w/ serial number  $s$   
Here is a ZK proof  $\pi$  that I know  $(r,t)$  s.t.

- exists** •  $c \in$  set of commitments in blockchain
- well-formed** •  $c = \text{commit}(v,k,r)$  and  $k = \text{commit}(s,t)$



### Pros

- No double spend
- Others cannot spend my coins
- Variable denomination

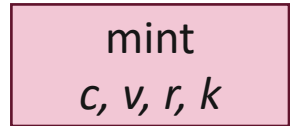
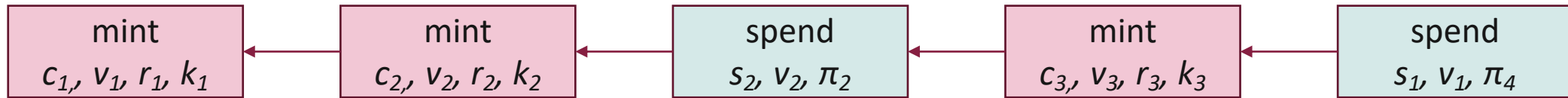
### Cons

- Spend and mint **partially linked**

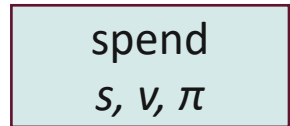


# Zerocash Design

## Attempt #2: payment addresses



Consume  $v$  coins to create value- $v$  zerocoin with commitment  $c$



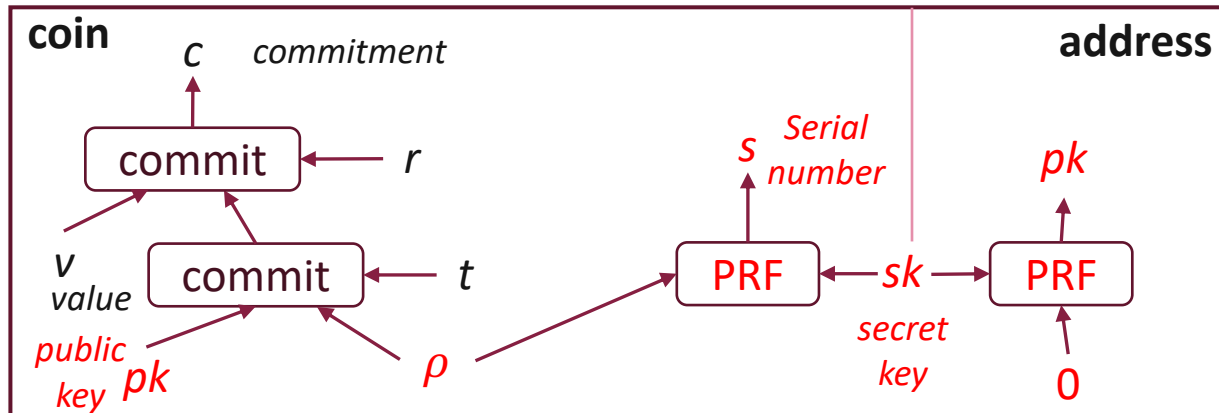
Consume coin w/ serial number  $s$

Here is a ZK proof  $\pi$  that I know  $(c, v, k, r, s, \rho, pk)$  s.t.

**exists** •  $c \in$  set of commitments in blockchain

**well-formed** •  $c = \text{commit}(v, k, r)$  and  $k = \text{commit}(pk, \rho, t)$

**mine** •  $s = \text{PRF}(\rho, sk)$  and  $pk = \text{PRF}(0, sk)$



### Pros

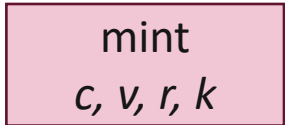
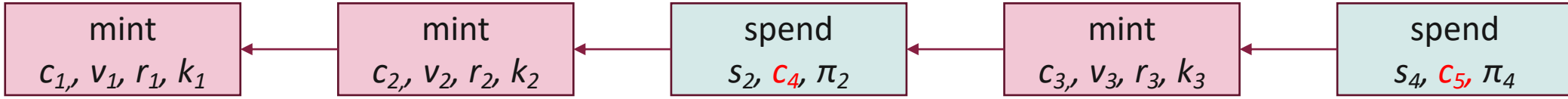
- No double spend
- Others cannot spend my coins
- Spend & mint TX **partially unlinkable**
- Variable denomination

### Cons

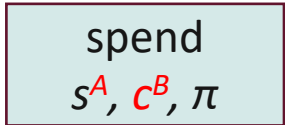
- Reveal  $v$

# Zerocash Design

## Attempt #3: direct payments



Consume v-basecoin to create value-v zerocoin with commitment c

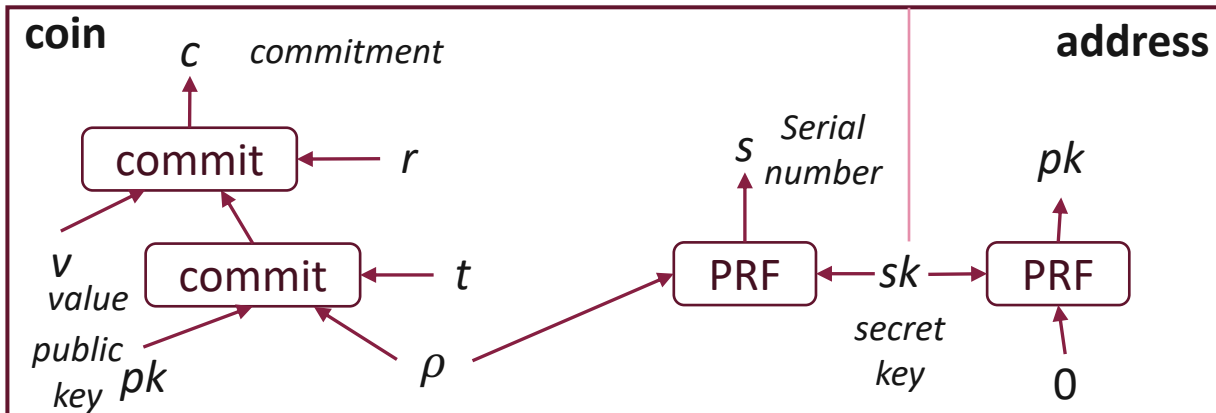


Consume coin w/ serial number s and create coin w/ commitment  $c^B$   
Here is a ZK proof  $\pi$  that I know  $(c^A, v^A, k^A, r^A, s^A, \rho^A, pk^A)$  s.t.

- exists** •  $c^A \in$  set of commitments in blockchain
- well-formed** •  $c^A = \text{commit}(v^A, k^A, r^A)$  and  $k^A = \text{commit}(pk^A, \rho^A, t^A)$
- mine** •  $s^A = \text{PRF}(\rho^A, sk^A)$  and  $pk^A = \text{PRF}(0, sk^A)$
- well-formed** •  $c^B = \text{commit}(v^B, k^B, r^B)$  and  $k^B = \text{commit}(pk^B, \rho^B, t^B)$
- same value** •  $v^A = v^B$

$(c^B, v^B, k^B, r^B, s^B, \rho^B, pk^B)$

Send out-of-band or via blockchain



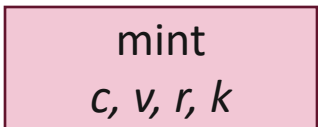
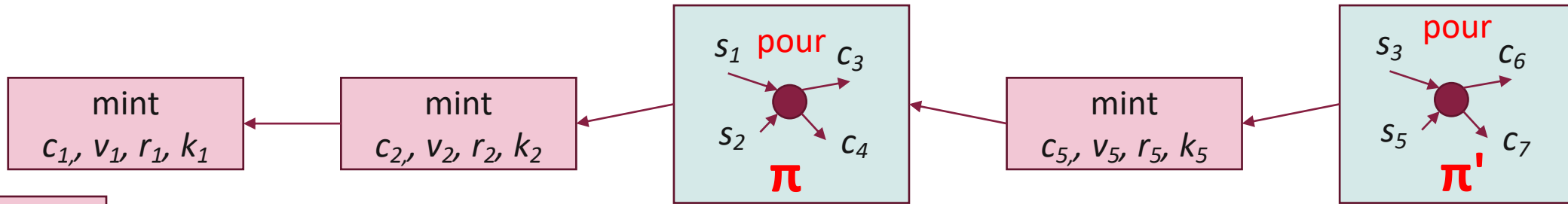
### Pros

- No double spend
- Others cannot spend my coins
- Spend & mint TX **unlinkable**
- Hide sender, receiver, amount

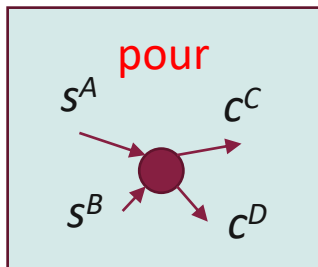
### Cons

- Join & split coins?

# Zerocash (Final) Design



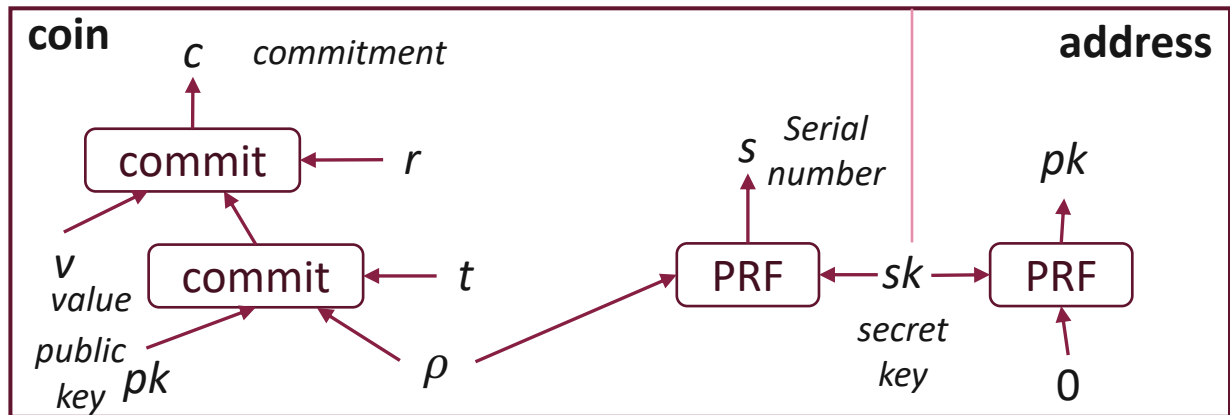
Consume v-basecoin to create value-v zerocoin with commitment c



Consume my input coins w/ serial number sA and sB in order to create two output coins (maybe not mine) w/ commitment cC and cD

Here is a ZK proof  $\pi$  that I know the secrets that demonstrate that

- Input coins were minted at some point in the past
- Output coins are well-formed
- balance is preserved



- Single TX type for
- Simple payments
  - Coin join and split
  - Making change
  - Pay transaction fee

# Zerocash Limitation

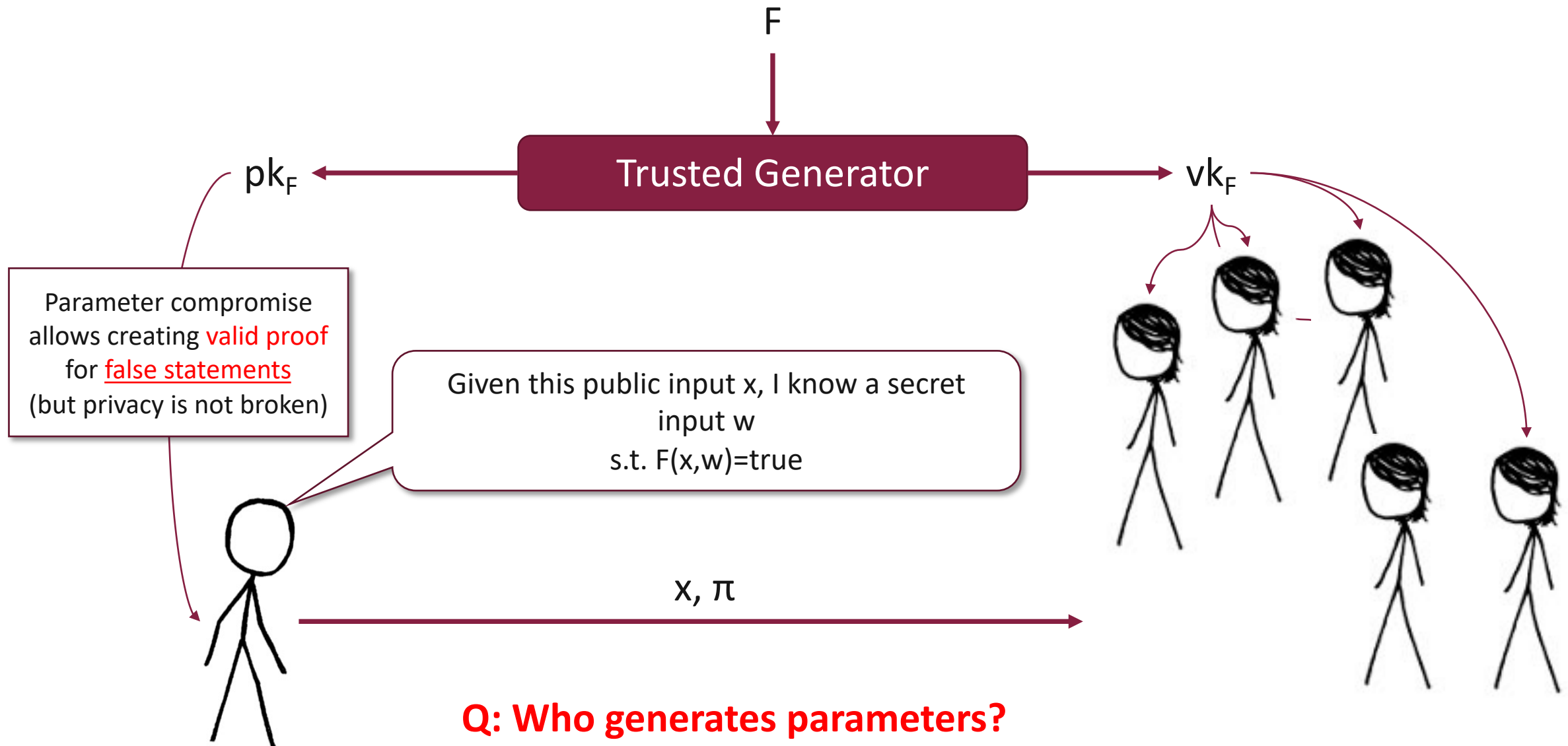
Require a trusted setup due to underlying crypto ZK-protocol

Random, secret inputs are required to generate public parameters for proving/verifying statements

These secret inputs must then be securely destroyed

No one can know them (**anyone who does can break the system**)

# Trusted Setup



A: A set of people via distributed multi-party computation protocol

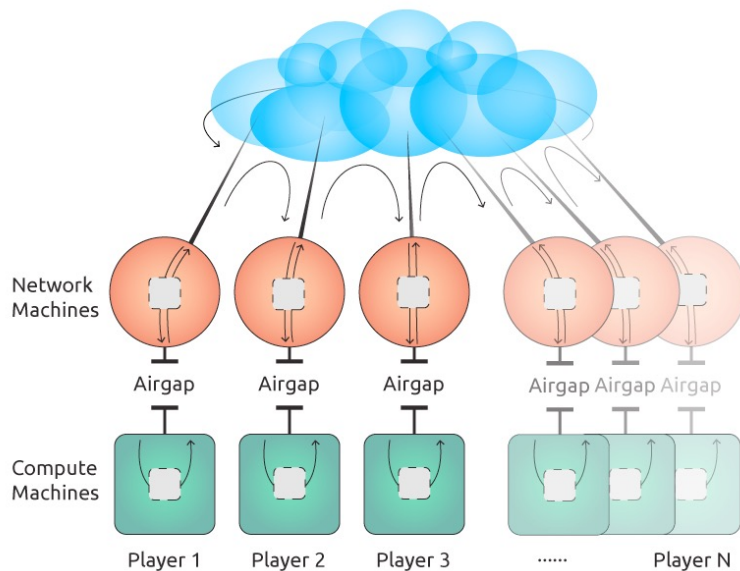
# MPC Ceremony

Run by ZECC during October 22-23, 2016

## Main ingredients:

- N-party MPC protocol that is secure against  $\leq n-1$  corruptions [BCGTV15][BGG16]
- Extensive threat modeling and security engineering

airgap between network machines and compute machines



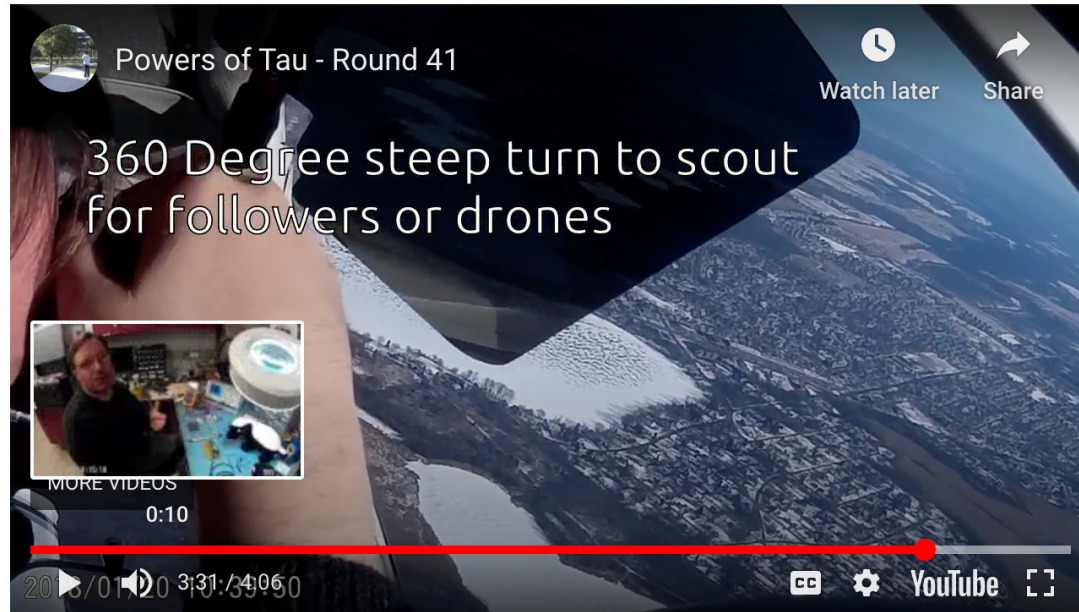
n=6 geographically distributed participants (including one security company, and a mobile station)

publicly-verifiable audit trail, in a hash chain stored on Twitter and the Internet Archive

video documentation from all participants including destruction of compute nodes

# MPC Ceremony

Some folks took randomness generation very seriously...  
Using radioactive material from Chernobyl in an airplane...



Driving through the desert...

Some participants were hacked...

# Summary

## 5 levels of anonymity

System	Type	Anonymity attacks	Deployability
Bitcoin	Pseudonymous	Tx graph analysis	Default
Single mix	Mix	Tx graph analysis, bad mix	Usable today
Mix chain	Mix	Side channels, bad mixes/peers	Bitcoin-compatible
Zerocoin	Cryptographic mix	Side channels (possibly)	Altcoin
Zerocash	Untraceable	None	Altcoin, <b>tricky setup</b>