The Science of Bitcoin

Eli Ben-Sasson

You probably heard

- Bitcoin is a *crypto-currency*
- Invented by the mysterious *Satoshi Nakamoto* in 2008, deployed 2009
- Market cap peak ~14B$ (in 2013), currently ~ 3.3B$
- Used also for illegal commerce on *Silkroad* and by *Dread Pirate Roberts*

This talk is ...

- … a computer science perspective of Bitcoin
  - *Cryptography*
  - *Decentralized consensus reaching (Byzantine agreement)*
  - *Anonymity in decentralized payment systems*
- Other interesting aspects not covered today
  - Economics (value/price of bitcoin, if any)
  - Law (crime, regulation, legal status of bitcoin)
  - Politics: internal (among bitcoin players) and external (New money vs. Old money)
  - Ideology (Libertarian crypto-anarchy meets Wallstreet)
  - …

Rest of talk

- **Non-scientific description of Bitcoin**
- Computer Science and Bitcoin
  - Bitcoin's academic pedigree
  - Analysis of Bitcoin's stability
  - Decentralized Consensus (Byzantine agreement)
  - Zerocash: Improving Bitcoins anonymity

  [joint work with Alessandro Chiesa, Christina Garman, Matt Green, Ian Miers, Eran Tromer and Madars Virza]

More reading [Bonneau et al. 2015]:

Research perspectives and challenges for Bitcoin and cryptocurrencies

http://www.jbonneau.com/doc/BMCNKF15-IEEESP-bitcoin.pdf

Money

- Wikipedia: “Money is any item or verifiable record that is generally accepted as payment for goods and services and repayment of debts in a particular country or socio-economic context”
- Money in modern economies
  - is mostly ***bank money***, not ***currency*** (notes/coins)
  - bank money is mostly ***electronic***
  - ***Monetary policy*** managed by government
  - Small set of big trusted parties – ***banks*** – maintain and update electronic ledgers
  - ***stability*** due to legislation, regulation and bank’s incentive to preserve reputation

Two basic challenges of decentralized e-money

Decentralized: no central authority, no regulation, no legislation

1. **Ownership and transfer of money** 2
   - Who owns how much?
   - How do you pay someone?
   - How to prevent forgery/theft/cheating/…?
2. **Monetary policy** 3
   - How is money created? At what rate?
   - Who gets new money?

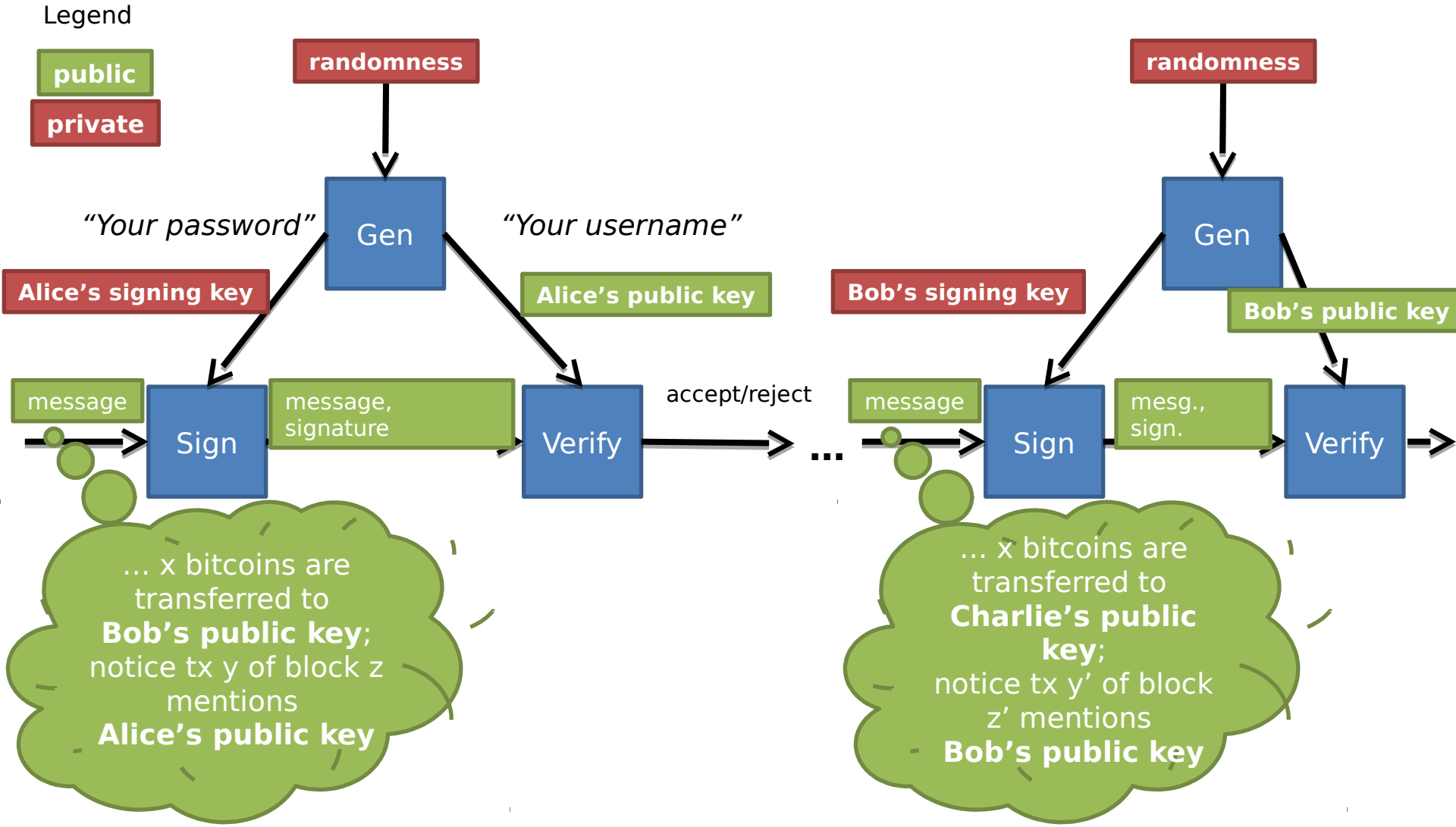
Bitcoin uses ***cryptography*** to implement a ***simple monetary policy*** that incentivizes players to ***simulate*** a ***stable payment ledger*** called the ***blockchain*** 1

1 Bitcoin's blockchain …

- is public, accessible on the web
- is a sequence of blocks $B_0, B_1, \ldots, B_{356,900}*, \ldots$ (* 17/5/2015)

  One block every 10 minutes
- Each block B_t contains transactions (txs), ~ 100-1000 tx/block
- Typical tx: "Alice pays Bob *x* BTC which she received in tx *y* of block *z*"
- Given blockchain, easy to verify that
  - Alice got funds as she claims, and
  - didn't spend them yet (no double-spending)
- But also easy to steal funds, by impersonation
- Theft prevented by digital signatures

2 Ownership via digital signatures



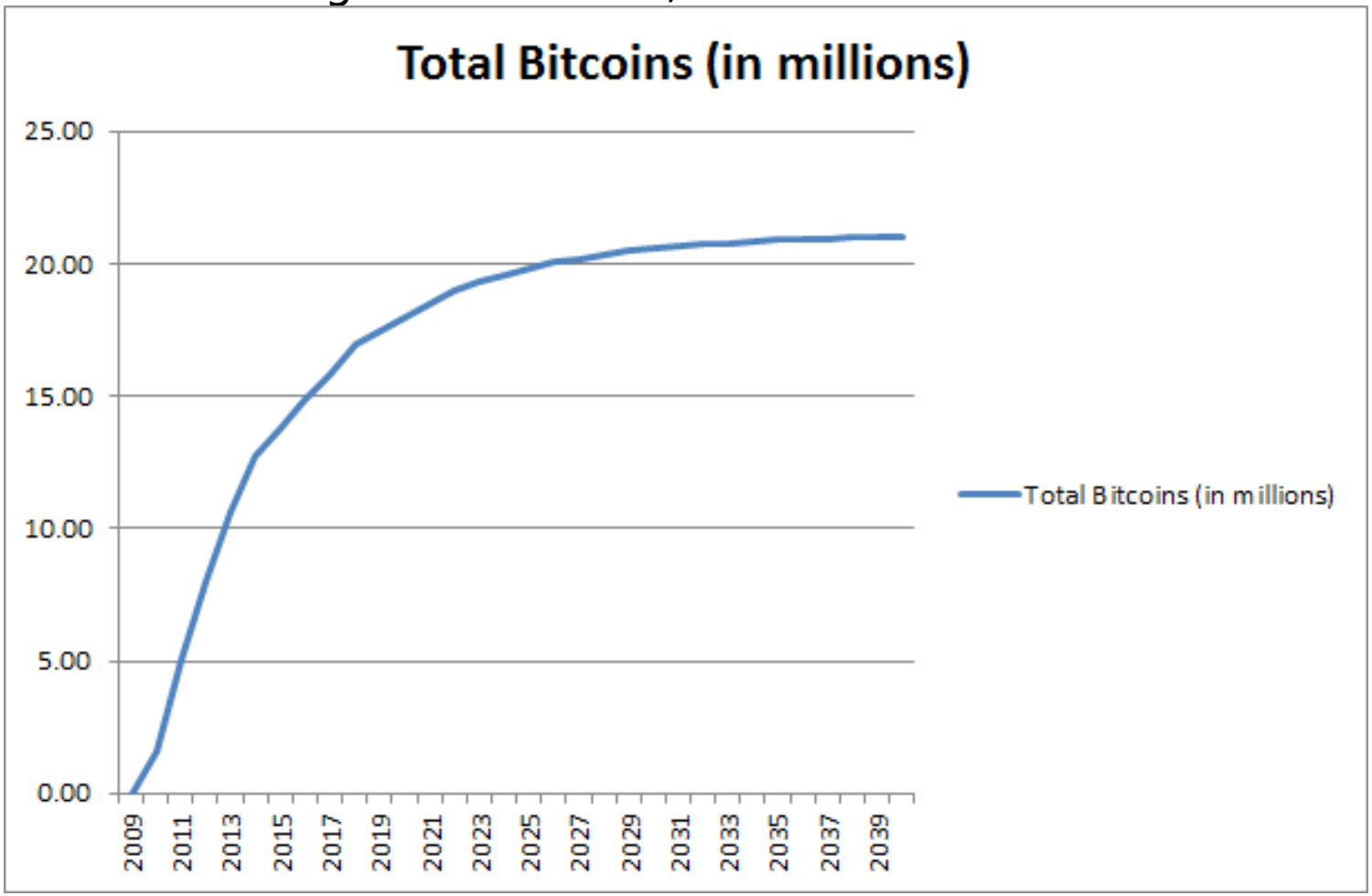
2 Ownership via digital signatures

- You are your key (on Bitcoin)
- Lost key = lost money
- Stolen key(s) = stolen money
  - Tx can include more complicated statement, like multi-signatures:

“to transfer this coin, 3 out of the following 5 public keys must sign the tx”

3 Bitcoin’s fixed monetary policy

- Every 10 minutes 25 Bitcoins are “mined” and given as reward
- Reward amount halves every 4 years
- # Bitcoins is a geometric sum, its limit is ~21M coins



3 Bitcoin’s fixed monetary policy

- Every 10 minutes 25 Bitcoins are “mined” and given as reward
- Reward amount halves every 4 years
- # Bitcoins is a geometric sum, its limit is ~21M coins

Reward given for increasing blockchain length
The Game:

1. To add block, solve hard puzzle defined by
   - *“hash” (fingerprint) of last block in longest block-chain*
   - *Block of new valid tx’s (properly signed, no double-spends, etc.)*
   - *Block contains reward tx: “pay my public key 25 BTC”*
2. Different nodes work on different puzzles due to:
   - *Different block of new valid tx’s*
   - *Different local view of the longest blockchain*
3. First one to solve puzzle broadcasts solution+new block;
4. Other nodes accept block only if (1) contains only valid tx’s, (2) introduces no double-spends, (3) part of longest blockchain

Notice: reward redeemable only if block accepted to blockchain

3 Puzzles and proof-of-work

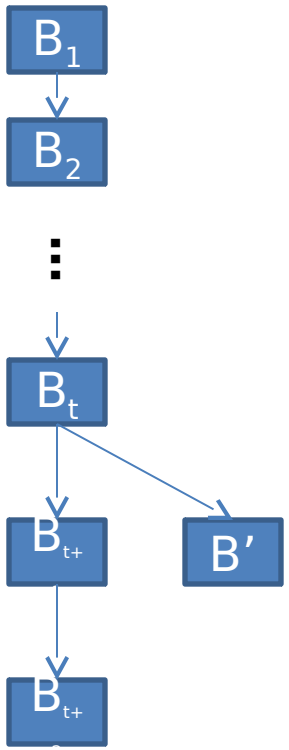
- Pseudorandom function (PRF)

$$H: \{0,1\}^{2n} \rightarrow \{0,1\}^{n},$$

call *n* the chunk-size *(In Bitcoin H is SHA256, n=256)*

  - Given x, easy to compute y=H(x) (say, time < 100n)
  - Given y, hard to find x s.t. y=H(x) (say, time > $2^{n/2}$)
- H compresses k-chunk file ($c_1,c_2,\ldots,c_k$) to single chunk:
  - Compress chunk pairs: $c'_i=H(c_{2i-1}, c_{2i})$
  - Repeat with (k/2)-chunk file ($c'_1,c'_2,\ldots,c'_{k/2}$)
- ***Bitcoin's Puzzle:***
  - Given input *F=H(last block, new block)*
  - find random string *R* s.t. *H(F,R)* starts with *d* zeros
  - For random R, Pr[success]=2^{-d}
  - Currently d~67 (called difficulty level)

Blockchain consensus



- Reward valid only if incorporated in b
- Protocol: “go with lo est chain”
- Satoshi: “If majority of players are honest, blockchain prefix will converge w.p. 1 at t=∞”
- Practically, waiting 6 blocks (1 hour) works well

Sybil attack: one machine can simulate many users

Rest of talk

- Non-scientific description of Bitcoin
- **Computer Science and Bitcoin**
  - Bitcoin's academic pedigree
  - Analysis of Bitcoin's stability
  - Decentralized Consensus (Byzantine agreement)
  - Zerocash: Improving Bitcoins anonymity

  [joint work with Alessandro Chiesa, Christina Garman, Matt Green, Ian Miers, Eran Tromer and Madars Virza]

More reading [Bonneau et al. 2015]:

Research perspectives and challenges for Bitcoin and cryptocurrencies

http://www.jbonneau.com/doc/BMCNKF15-IEEESP-bitcoin.pdf

Bitcoin’s CS pedigree

- **E-cash [Chaum ’82]:** anonymous e-money, using “blind signatures”, ...
- **Proof-of-work [Dwork, Naor, `92]:** anti-spam mechanism, ...
- **Consensus in distributed systems** (aka the *Byzantine Agreement problem*)
  - Studied since the early 1980’s
  - No solution for most general case [FLT 85’]; many solutions for realistic models

Stability of Bitcoin consensus

- Stability can mean
  - ***Eventual consensus***: as $t \to \infty$, honest nodes will agree on prefix of blockchain
  - ***Exponential convergence***:

    $$\Pr[\text{fork of depth } n] < 2^{-O(n)}$$

  - ***Liveness***: new blocks added, even when no more rewards exists (trans. fees?)
  - ***Fairness***: Miner with c fraction of hash-power gets c fraction of reward
  - …

Basic attacks

- ***51% attack***: Party with c>1/2 fraction of hash power can destabilize block-chain
- ***Selfish mining*** [Eyal & Sirer 2013]: Party with c>1/3 can get unfair (c'>c) fraction of reward
- **Other attacks: Goldfinger** [Kroll et al. 2013], observed thru altcoin infanticide, **Feather-forking** [Miller 2013], **Denial of Service**, …
- ***Paradox***
  - Mining-pools reached c>1/2, Bitcoin still stable
- ***Possible explanation***
  - External factors: price of hardware needed to mount attack, effect of attack on bitcoin value,…

The power of Hash

- Satoshi's vision on proof-of-work puzzles
  - one person – one machine
  - all machines are equal
  - Ergo, Bitcoin consensus is a democratic process
- Wikipedia: "as of 2015 a miner who is not using purpose-built hardware is unlikely to earn enough to cover the cost of the electricity used in their efforts"

  [Current global hash-rate ~ 350 Petahash/second (!)]
- Challenges:
  - stable "democratic" consensus
  - non-wasteful puzzles

Alternative Consensus Protocols

- **Bitcoin** [Nakamoto 2009]
  majority of hash-power controls block-chain
- **Proof-of-burn** [Stewart 2012]
  pay coins to join reward lottery
- **Proof-of-coin-age** [King, Nadal 2012] majority of “old” coins
- **Proof-of-deposit** [Kwon 2014]
  majority of savings funds
- **Proof-of-activity** [Bentov et al. 2014]
  majority of tx volume
- ...

Beneficial puzzles

Wikipedia: "as of 2015 even if all miners used energy efficient processes, the combined electricity consumption would be equal to the consumption of about 135,000 American homes"

"Better" puzzles

- **Primecoin** [King 2013]: find cryptographically useful prime numbers
- **Permacoin** [Miller et al. 2014]: store large data
- **Memory-hard puzzles**: resist large-scale hardware (?): scrypt (used in Litecoin, Dogecoin), cukoo-hashing [Tromp 2014]
- **Mining-Pool-resistant puzzles** [Miller et al.; Sirer & Eyal 2014]

…

Zerocash – Decentralized Anonymous Payments

- Joint work with
  - Alessandro Chiesa [ETH+UC Berkley]
  - Christina Garman [John Hopkins]
  - Matt Green [John Hopkins]
  - Ian Miers [John Hopkins]
  - Eran Tromer [TAU]
  - Madars Virza [MIT]

What properties should money have?

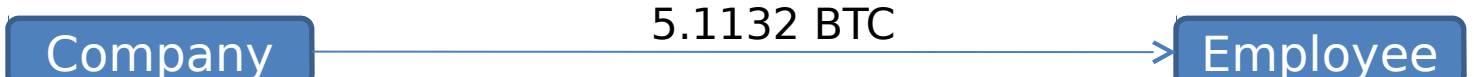
- Scarcity
- Transportability
- Divisibility
- Privacy
- Fungibility*
- Durability
- Accepted

Fungible* being of such nature as to be freely exchangeable or replaceable, in whole or in part, for another of like nature

Anonymity in Bitcoin

- Imagine Bitcoin is the only currency
  - Salaries in Bitcoin, shopping in Bitcoin,…

Got my first paycheck …



Tx: 1CD9RaegDQLTexFZSXSrNBASZQv1qkBnmT, 5.132BTC,
1DkkHZKTCNdPsPFU52X8V8HjYm4foBEFkx

Payer pseudo-ID | Tx amount | Payee pseudo-ID

… celebrated it with coffee

Tx: 1DkkHZKTCNdPsPFU52X8V8HjYm4foBEFkx, 0.01 BTC,
1CD9RaegDQLTexFZSXSrNBASZQv1qkBnmT

Payer pseudo-ID | Tx amount | Payee pseudo-ID

Anonymity in Bitcoin

- Imagine Bitcoin is the only currency
  - Salaries in Bitcoin, shopping in Bitcoin,…

  - Barista learns Employee's salary, CEO learns Empolyee's coffee place, …
  - More can be gained with deeper analysis

[Reid Martin 11] [Barber Boyen Shi Uzun 12] [Ron Shamir 12] [Ron Shamir 13] [Meiklejohn Pomarole Jordan Levchenko McCoy Voelker Savage 13] [Ron Shamir 14]

Methods of analysis only get **stronger**.

Your Bitcoin history is publicly saved **forever**

Lack of privacy consequences

- Imagine Bitcoin is the only currency
  - Salaries in Bitcoin, shopping in Bitcoin,…

- Limits Bitcoin adoption:
  - Consumer income and purchases **visible** to friends, neighbors and co-workers.
  - Merchant cash flow **exposed** to competitors.
- A threat to Bitcoin **fungibility**:
  - Bad (e.g., stolen) coins may **taint** good ones

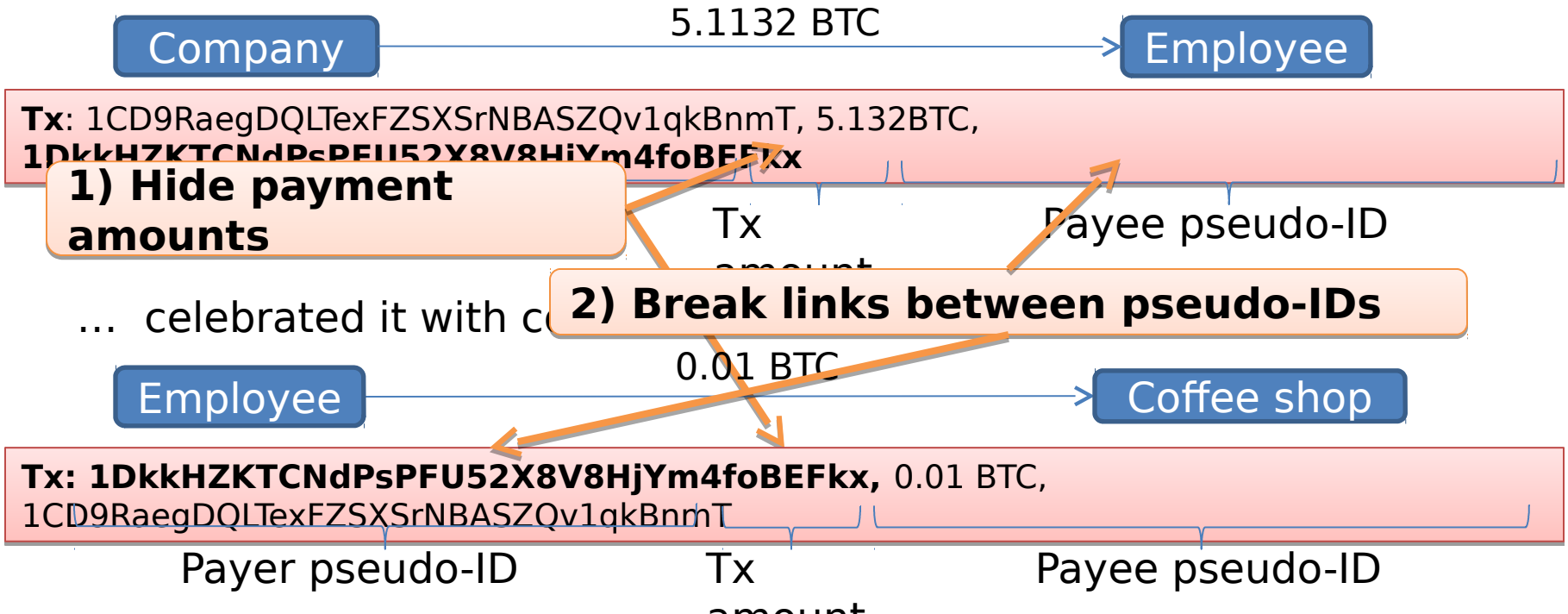
Previous anonymity solutions

- Imagine Bitcoin is the only currency
  - Salaries in Bitcoin, shopping in Bitcoin,…

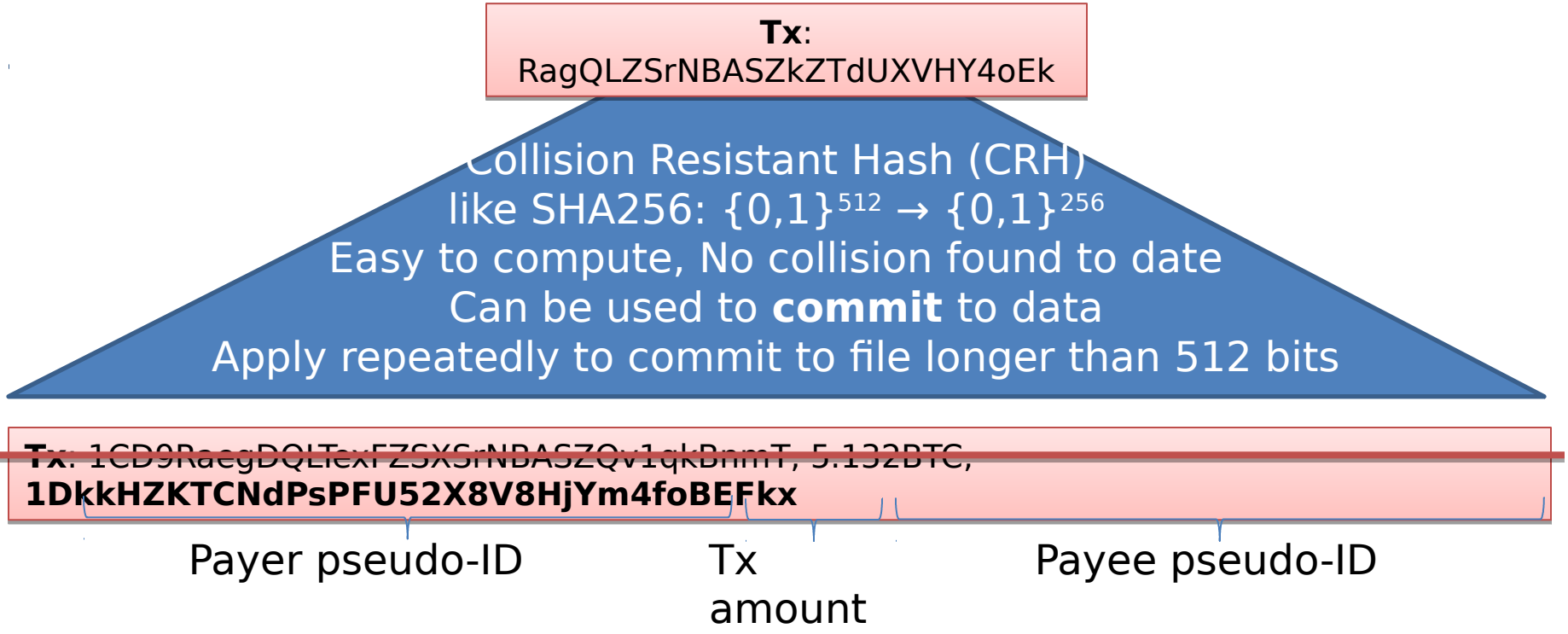
- Replace pseudo-identities often
  - Problems: traceable, complicated to maintain
- Use a “mix” / CoinJoin [Greg Maxwell]
  - Problems: need to find and trust co-mixers, prone to DoS attacks, payment amount revealed
- Trust a large party (central wallet)
  - Problems: no different from bank, Mt. Gox …
- ZeroCoin[Miers, Garman, Green, Rubin 13], PinnochioCoin [Danezis, Fournet, Kohlweiss, Parno 13]
  - Problems: payment amount revealed, scalabiltiy

To achieve anonymity, need to

Got my first paycheck ...

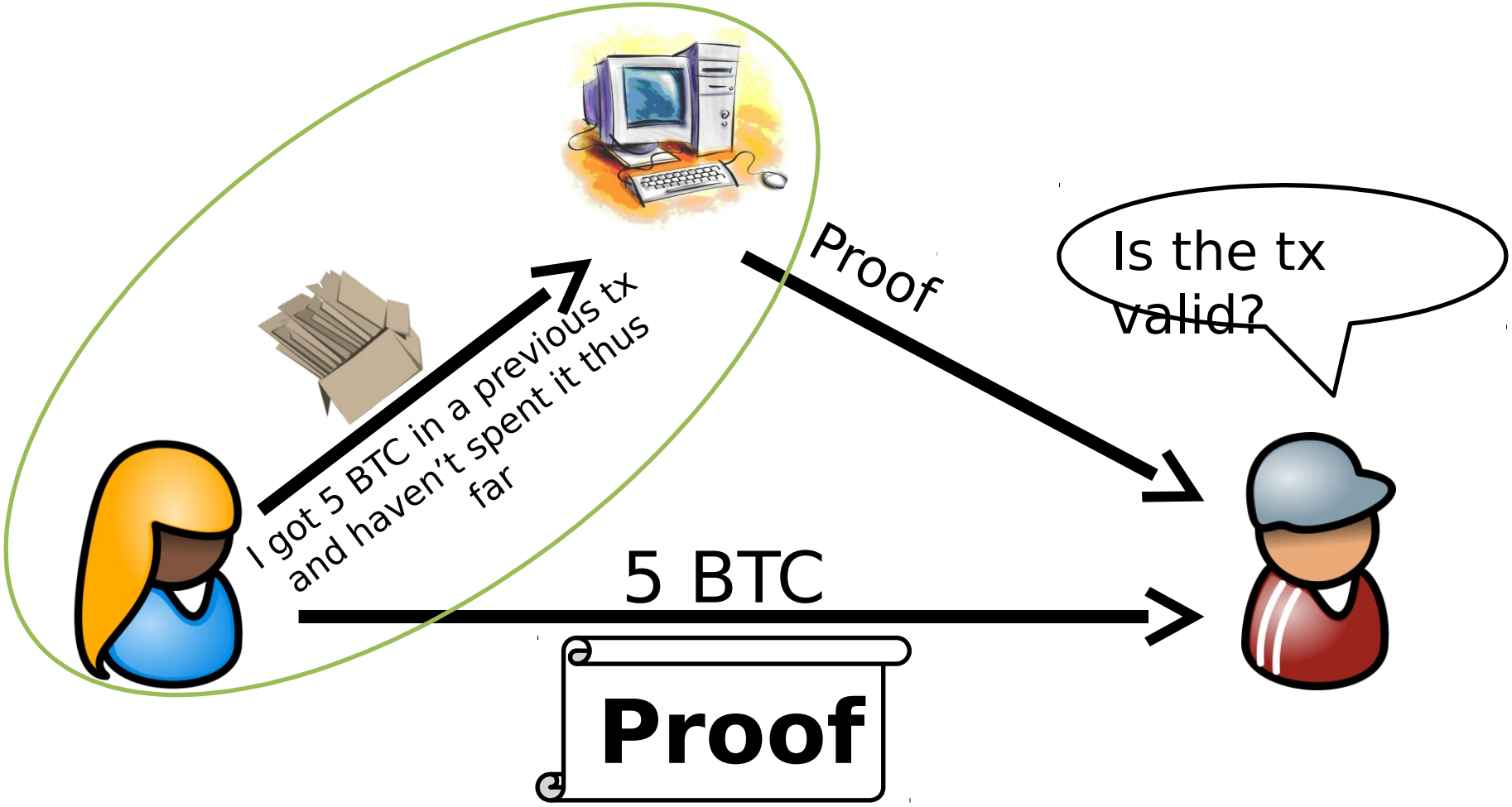


Achieving anonymity is easy…



… Maintaining payment system **integrity** is hard

Integrity when all is hidden

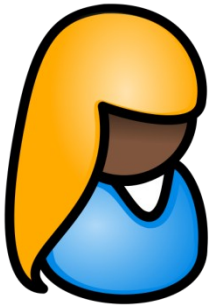


What kind of proof?

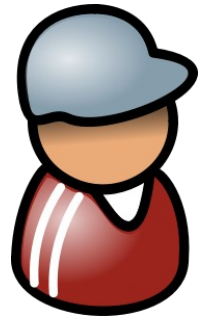
zero **k**nowledge
succinct
Non-interactive
argument
of **k**nowledge

NIZK

(zk)SNARK



5 BTC



Proof

1. Where do we get a SNARK?

(i) Theoretical constructions

[BFLS91, Kilian92,M… …Groth10,GLR11,Lipmaa12,BC1
DFH12,BCIOP13,GG… …3,BCTV14b,Lipmaa14,
KPPSST14,BFR14,W… …CTV14a, BCCGT14]

"SNARKs for C": Execution of C programs can be verified in 230 bytes and verified in 5 ms.

(ii) Working

[PGHR13, BCGTV13, BCTV14b, KPPSST 14, ZPK14, CFHKKNPZ14, BCTV14a, BCTV14b]

Our implementation of choice: ***libsnark***

- Fast.
- Versatile: circuits, RAMs, bootstrapping …

github.com/scipr-lab/libsnark

2. How to use this **tool** to build an **anonymous payment scheme**?

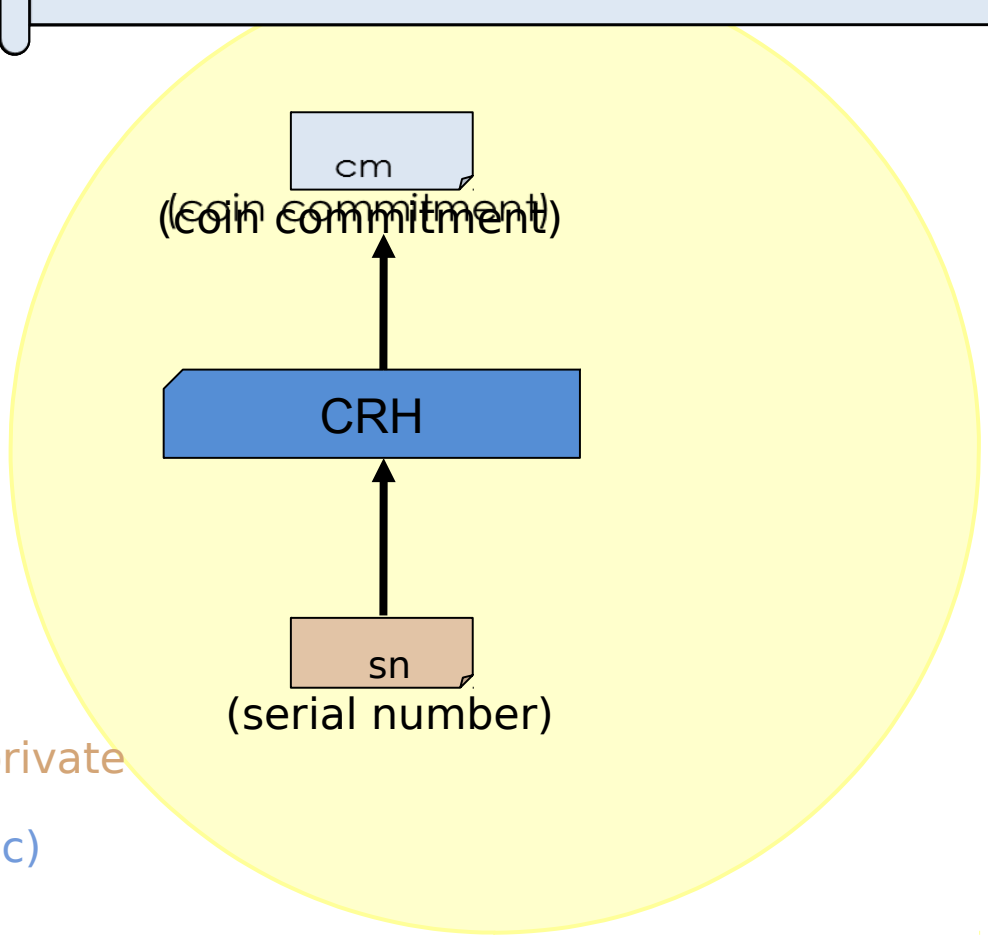
Controlling coins in a public ledger

Minting:

Spending:

I hereby spend 1 BTC to create cm

I'm using up a coin with (unique) sn,
and CRH(sn)=cm, which appears in ledger



cm_1 cm_2 cm_3 cm_4 cm_5 cm_6 cm_7 cm_8

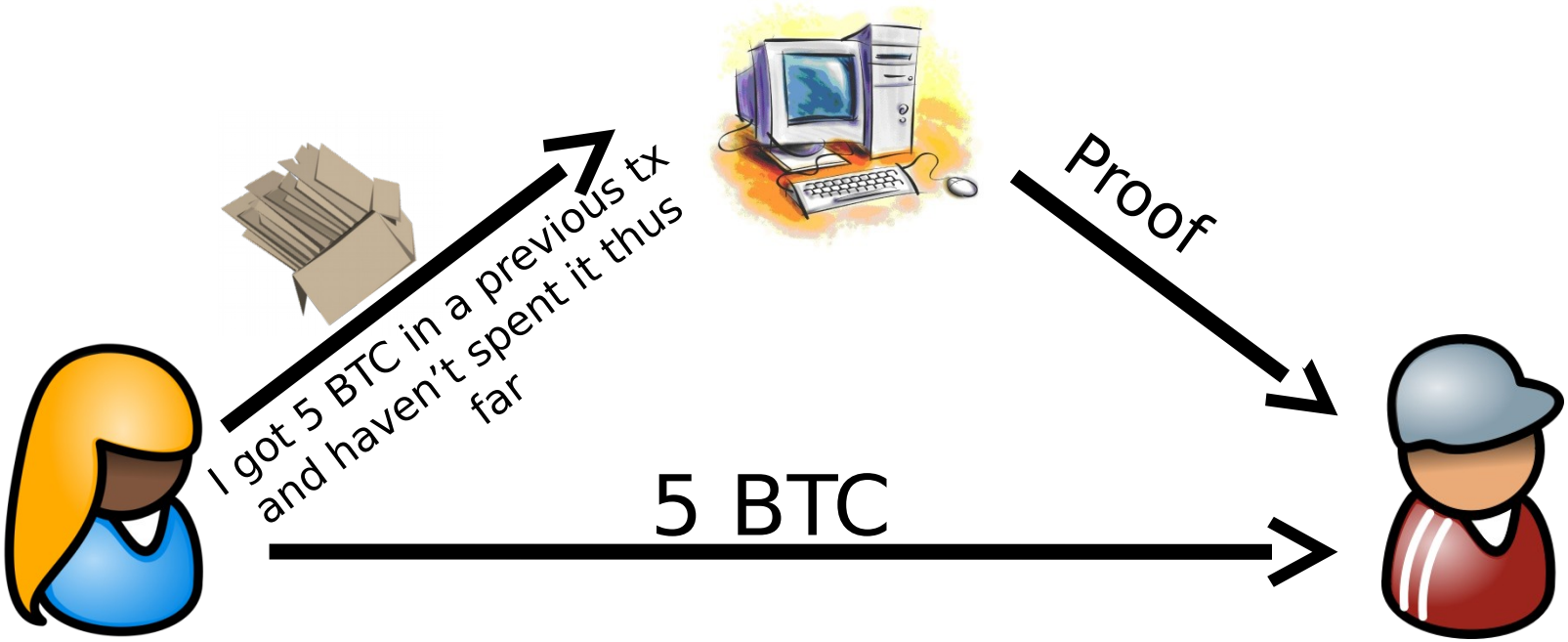
Legend:

Temporarily private

Explicit (public)

ZK proofs of existential statements

- x: Explicit input (public)
- y: Witness (private)
- C: computation (Arithmetic circuit)
- Existential statement: Exists y s.t. C(x,y)=1 (aka NP statement)
- ZK proof: proves statement but reveals nothing about y



Basic anonymous e-cash

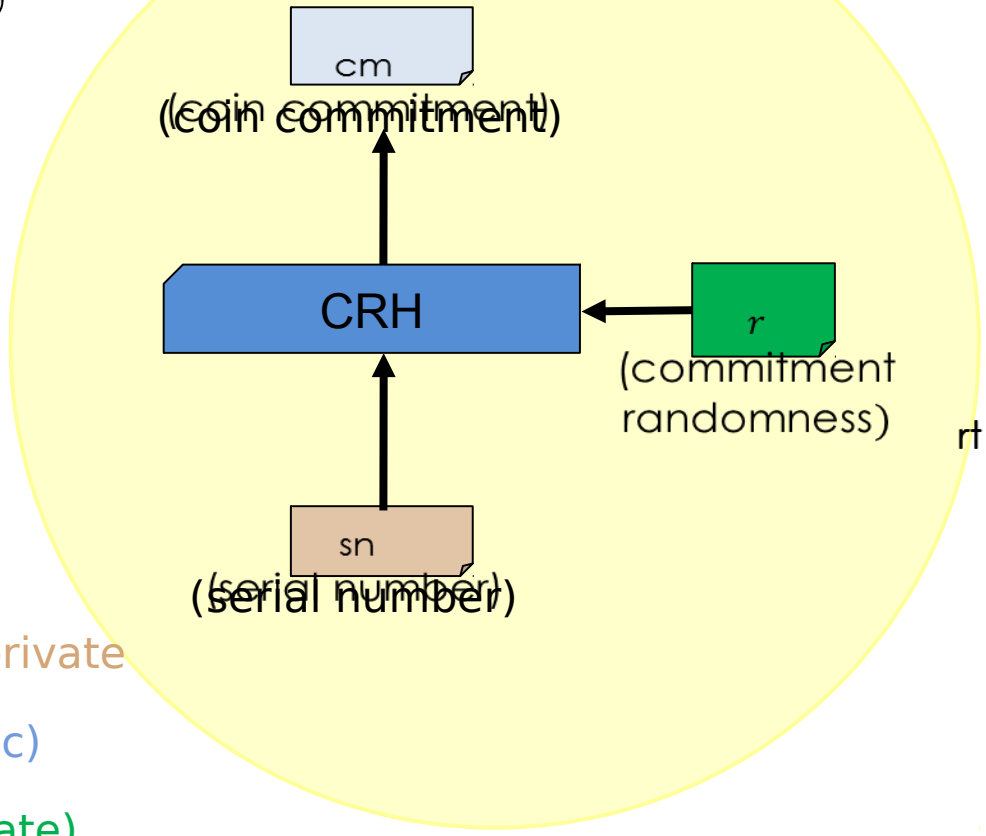
Minting:

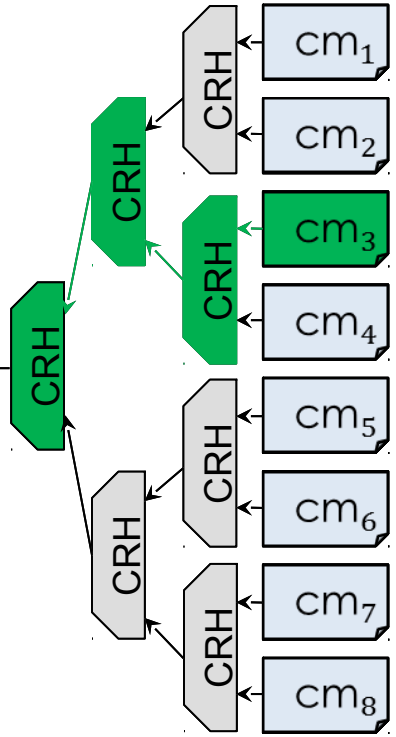
I hereby spend 1 BTC to create cm

[Sander Ta-Shma 1999]

Spending Existential statement:

x=(sn, rt). There exists y=(r, path) s.t. C(x,y)=1 where C(x,y)=1 iff path from rt leads to leaf labeled by CRH(sn,r)





Legend:

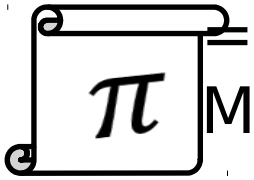
Temporarily private

Explicit (public)

Witness (private)

Beyond privacy and fungibility:

zero-knowledge in public oversight



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

But I’m **not revealing** recipient or amount.”

The money went to a **charity organization**!

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?

Zerocash Efficiency and trust assumptions

- zk-SNARK takes
  - **46 sec.** to generate on i7-4770 @ 3.4 Ghz w/ 16 GB RAM
  - **6 ms** to verify
  - **288 Bytes** long (at 128-bit level of security)
- To generate, requires “proving key” that
  - is **0.9 Gb** long
  - generated (once) in **2 min.** by **trusted** party before deployment
  - key generation algorithm uses **trapdoor** (it must be destroyed)
  - malicious party holding trapdoor can **forge** transactions

Tx: 1CD9F
1DkkHZK

New [BCGTV Oakland S&P `15]
Practically implemented multi-party computation for setup.
If even one player is honest, Then setup is good

Anonymity solution	Trust who?	Trust when?	If trusted party is compromised …			
			Forgery	Theft	Anonymity broken	DoS
Mix	Mix operator	Each Tx	No	Yes	Yes	Yes
CoinJoin	Tx participants	Each Tx	No	No	Yes	Yes
Zerocash	CRS generator	Only setup	Yes	No	No	No

Summary and Discussion

- Bitcoin: first successful decentralized crypto-currency
- uses ***cryptography*** to implement a simple ***monetary policy*** that incentivizes players to ***simulate a stable payment ledger*** called the ***blockchain***
- Bitcoin's success leads us (computer scientists) to try and explain it, improve it and criticize it (ongoing work)