

# Blockchains & Cryptocurrencies

## **Anonymity in Cryptocurrencies**



Instructors: Matthew Green & Abhishek Jain  
Johns Hopkins University - Spring 2019

# Housekeeping

- Reminder: we have a midterm on 3/13!
  - This will cover all readings and lectures up through 3/11 (next Monday)
- Assignment 2 due 3/11 end of day
- Note: readings listed post-Spring Break are currently out of whack and will be updated soon

# News?

# News?

INSIGHTS

CRYPTOCURRENCY

EXCHANGE [ORGANIZED MARKET]

BITCOIN

## QuadrigaCX mystery deepens after cryptocurrency cold wallets found empty

### Everything you need to know about Facebook's new cryptocurrency

Whatsapp is set to launch a coin, but what does this mean?



By [Laurie Clarke](#) | Mar 04, 2019

Share



Facebook is reportedly planning to launch its own cryptocurrency within Whatsapp, which would allow users to frictionlessly transact with one another through the service without incurring fees.

[According to Bloomberg](#), the new cryptocurrency will initially focus on the remittance market in India, where there are hundreds of millions of Whatsapp users and most financial transfers today are made via mobile payments. It's worth noting however that this coin won't be a typical cryptocurrency, it will be a 'stablecoin'.

Review stuff (from last time)



```
function giveBirth(uint256 _matronId)
    external
    whenNotPaused
    returns(uint256)
{
    Kitty storage matron = kitties[_matronId];

    // Check that the matron is a valid cat.
    require(matron.birthTime != 0);

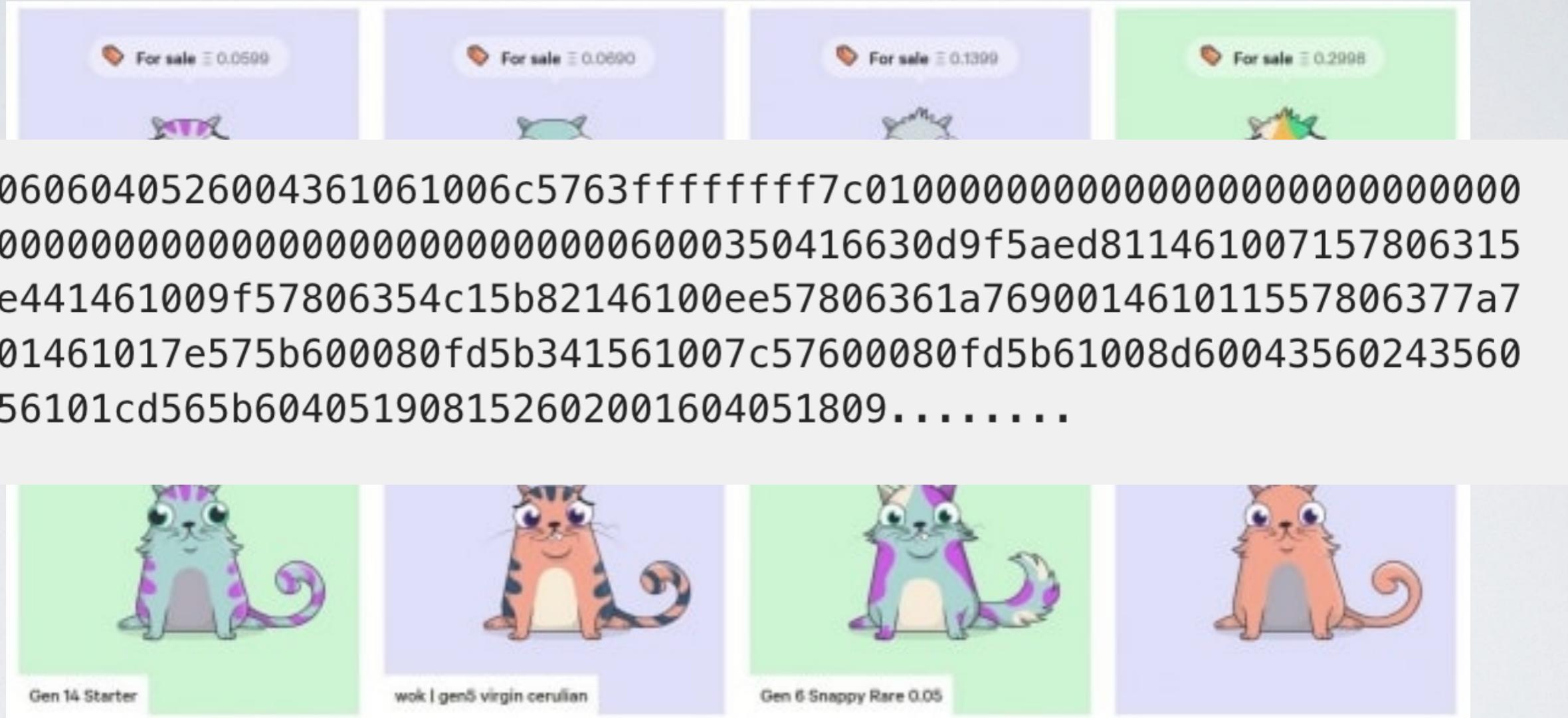
    // Check that the matron is pregnant, and that its time has come!
    require(_isReadyToGiveBirth(matron));

    // Grab a reference to the sire in storage.
    uint256 sireId = matron.siringWithId;
    Kitty storage sire = kitties[sireId];

    // Determine the higher generation number of the two parents
    uint16 parentGen = matron.generation;
    if (sire.generation > matron.generation) {
        parentGen = sire.generation;
    }

    // Call the sooper-sekret gene mixing operation.
    uint256 childGenes = geneScience.mixGenes(matron.genes,
sire.genes, matron cooldownEndBlock - 1);
```

- Cryptokitties



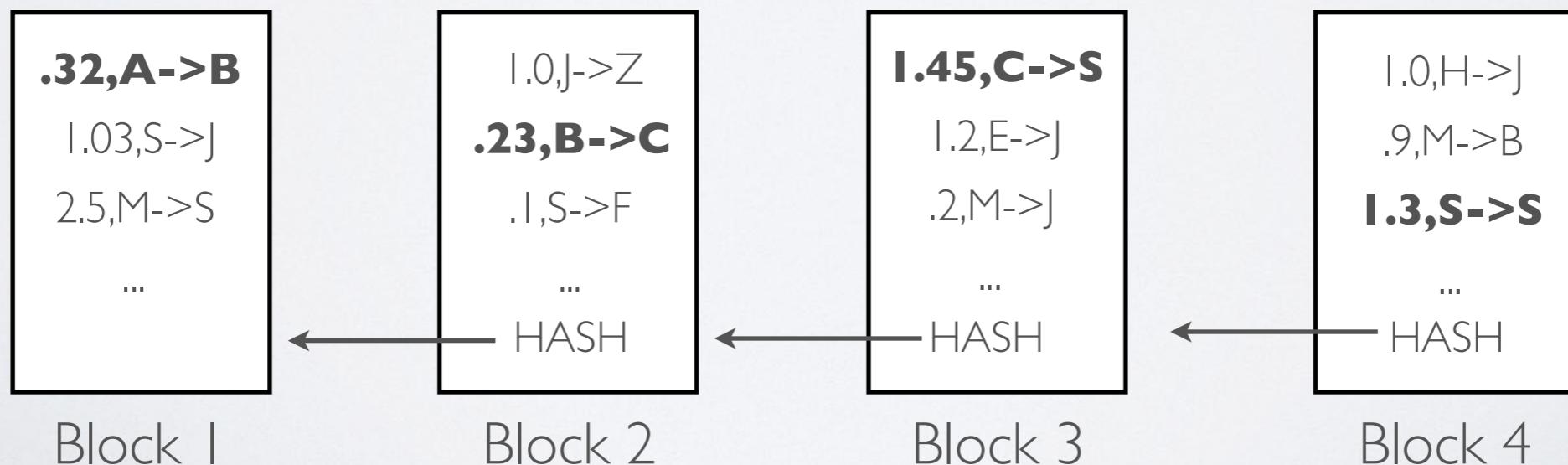
- Reverse engineering Cryptokitties (see link below):
  - Kitty gets “pregnant”, i.e., commit to DNA of sire and matron
  - Then kitty “gives birth” sometime later, and this is when DNA is computed
  - The DNA combination is randomized by the Ethereum block hash at birth time
  - Are there attacks on this type of randomness?

- Decentralized exchanges
  - Allow you to trade Ethereum-native tokens (e.g., ERC20 tokens)
  - If you could perform a “swap” from another blockchain to an Ethereum token, then you could trade that currency too
  - What problems come up here?

# Anonymity

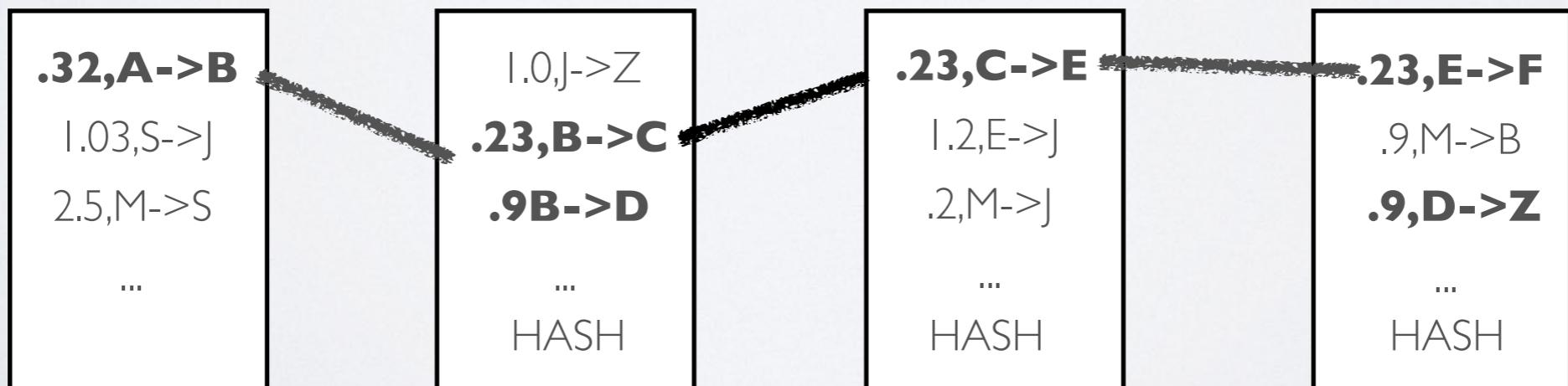
# Bitcoin: review

- Bitcoin's core innovation is the blockchain
  - A decentralized append-only ledger (divided into 'blocks' of many transactions)
  - Massively replicated
  - Everyone can download and see transactions



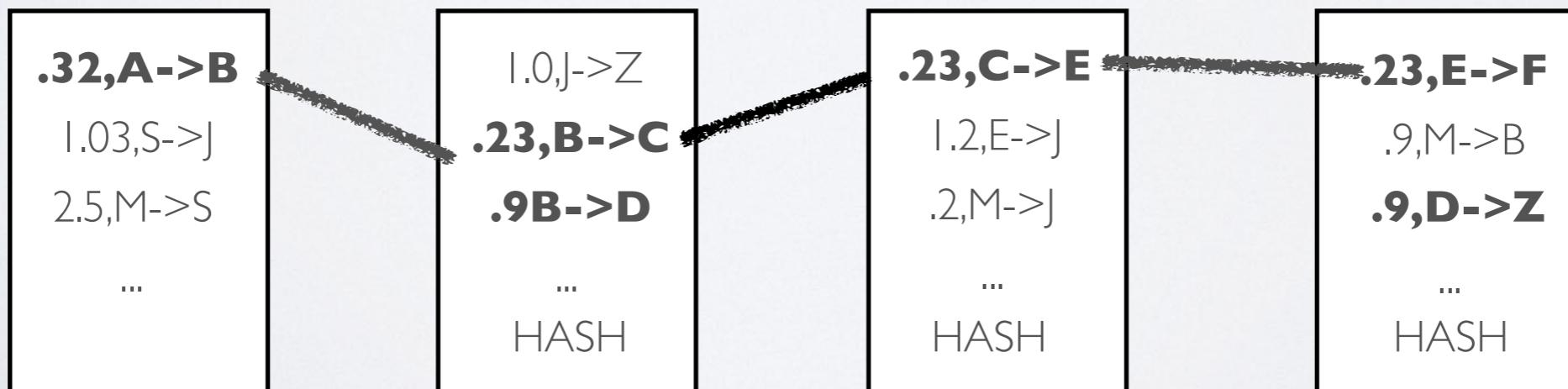
# Bitcoin privacy

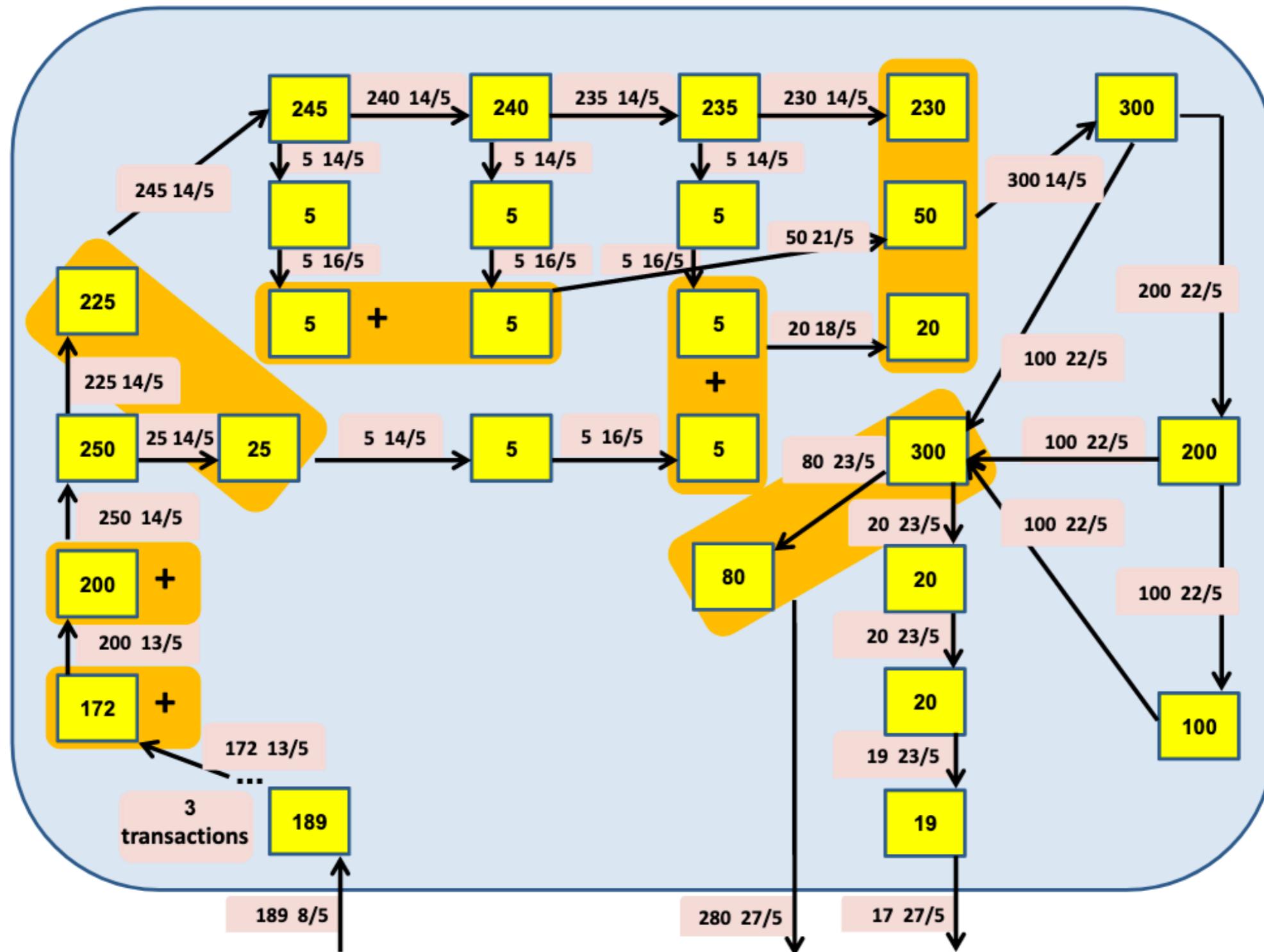
- **Disadvantage: Bitcoin is not very anonymous**
  - All these transactions are visible to the world, your neighbors, etc.
  - If people can link you to your address, you're screwed
  - How does one address this?

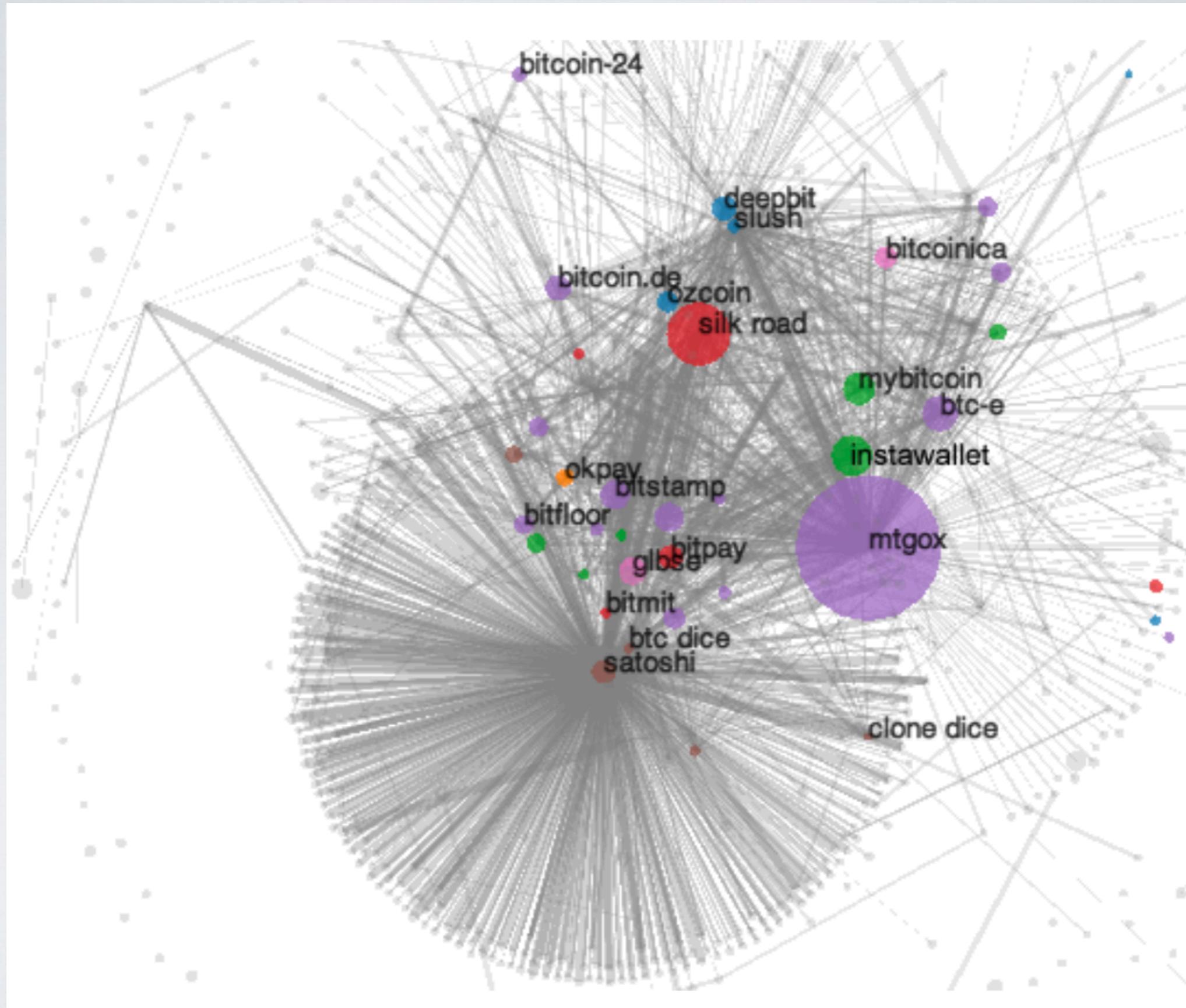


# What does it matter?

- Threats range from privacy loss to extortion
- Other concrete threats:
  - allows front-running on trades
  - transaction censorship by miners







Home ▶ Crypto ▶ Coinbase Sought Neutrino After Losing Control Over Customer Data Being Sold



Coinbase, one of the first "Bitcoin Unicorns," faced major industry backlash after acquiring blockchain analysis firm Neutrino. | Source: Shutterstock

# Coinbase Sought Neutrino After Losing Control Over Customer Data Being Sold

**IN THE SUPREME COURT OF NOVA SCOTIA**

**IN THE MATTER OF:**

Application by Quadriga Fintech Solutions Corp., Whiteside Capital Corporation and 0984750 B.C. Ltd. dba Quadriga CX and Quadriga Coin Exchange (collectively referred to as the “Applicants”), for relief under the Companies’ Creditors Arrangement Act

43. The Applicants identified to the Monitor six (6) cold wallet addresses which Quadriga used in the past to secure bitcoin (the “**Identified Bitcoin Cold Wallets**”), the primary cryptocurrency traded on the Quadriga platform. The addresses of the Identified Bitcoin Cold Wallets identified by the Applicants are as follows:

- (a) 1MhgmGaHwLAvvKVyFvy6zy9pRQFXaxwE9M;
- (b) 1JPtxSGoekZfLQeYAWkbhBhkr2VEDADHZB;
- (c) 1ECUQLuijJbFZAQchcZq9pggd4EwcpuANe;
- (d) 1J9Fqc3TicNoy1Y7tgmhQznWrP5AVLXj9R;
- (e) 1HyYMMCdCcHnfjwMW2jE4cv9qVkJDFUzVa; and

# Bitcoin Address

Addresses are identifiers which you use to send

## Summary

## Transactions

|          |  |                  |                                     |   |
|----------|--|------------------|-------------------------------------|---|
| Address  | <a href="#">1MhgmGaHwLAvvKVyFvy6zy9pRQFXaxwE9M</a>       | No. Transactions | 2111                                |  |
| Hash 160 | <a href="#">e31458ca4d933e612c97d9444a824fab6c7711c9</a> | Total Received   | <a href="#">14,073.11987636 BTC</a> |  |
|          |  | Final Balance    | <a href="#">19.54328527 BTC</a>     |  |

in the past to secure bitcoin (the “**Identified Bitcoin Cold Wallets**”), the primary

## Summary

## Transactions

|          |  |                  |                                     |   |
|----------|--|------------------|-------------------------------------|---|
| Address  | <a href="#">1JPtxSGoekZfLQeYAWkbhBhkr2VEDADHZB</a>       | No. Transactions | 1741                                |  |
| Hash 160 | <a href="#">becea8f53732df6354c6bf8cc8bdee6e77390f94</a> | Total Received   | <a href="#">12,436.02100587 BTC</a> |  |
|          |  | Final Balance    | <a href="#">33.19556316 BTC</a>     |  |

(c) [1ECUQLuioJbFZAQchcZq9pggd4EwcpuANe](#);

(d) [1J9Fqc3TicNoy1Y7tgmhQznWrP5AVLXj9R](#);

(e) [1HyYMMCdCcHnfjwMW2jE4cv9qVkJDFUzVa](#); and

# Bitcoin Address

Addresses are identifiers which you use to send

## Summary

## Transactions

4dd314417356d8f1f908724b7f77518c2bb45dad06ea70c7c8137325ea9de463

2019-02-06 17:24:07

32GK5iiR47WSmB88FVYoNYgYj8JRdSzuBZ  
3GHRJxQjKTV25bQhs8HSCdmRA19PZNbvtG  
3KKxgqeK9XJkA2DGBvy2bux69cirPju2nv  
3JZ7Lp3csarRZbSRVWtcGmHWJwwjh6QVZA  
3NfVNUIY3Ykxas3ws2zJmCJikYMpv3aEvt  
3JjqxJStfsNAbcV1cQmYiQTT6K1i9Mbgzv  
35LHRCs3JLM8skEzusKkTPMACFPuKHYvhV  
3B4jcfjuaUAZk3khbBcvFt1VMVB8WxDohM  
3L8UtV59MrUmyoHozNDFuqz6SjQKTv1zis  
38ACZBZPWprtEN8ZGbW1XxvJcFtGH1Hj3E



1JPtxSGoekZfLQeYAWkbhBhkr2VEDADHZB

0.60857038 BTC

0.60857038 BTC

4d0a8f0663b902c397198bcc06b44d24993056a47f6a4ddaa645843af7a5376f

2019-02-06 17:21:18

3K3pxgb2fFM6BAG2XvTnZNzptmXPSJjLAm  
3DR9PddmHk1vE59RCRLvBBV2ZZKhKCQE6a  
3Jrwdqr2qHcmcQQcSPTe9jQDEEhTHQWytD  
31wySWG3iB7FN5L39RUNjzUk29MwBB1mbk  
3LHWCTZ8hvLWMSASCpseDQa7Hk1g1k9Fmp  
35gDrq7xK43Y5PUukPpJU3vWT8scGBU5bN  
39m3QkS49vYjRMwMmAHS6dNjB51WY5uaS  
3Fj6QWJfdvfDvYzEggGvxcmfxsCAKD2WYE  
39kQ5nqSPrgKgqvCzmQAUEmJDmzeEmkJwh  
35EvkfcF2oRzYAhqmxbCCc1ECVhZyiasYK  
35bohu2URxJWLNmWDQAmadaCoF5QZ6oq5cX  
3BWbVZPX8xeAVD2j6dbY2ozYZw6r7xCWQK  
32JeSgdHrUGZmmjs0Tf5qouJXRtUvgvvgb  
3QtBeESnsNqLCatzUtYfkQ5Fbrfe2kgv39



1JPtxSGoekZfLQeYAWkbhBhkr2VEDADHZB

2.80083949 BTC

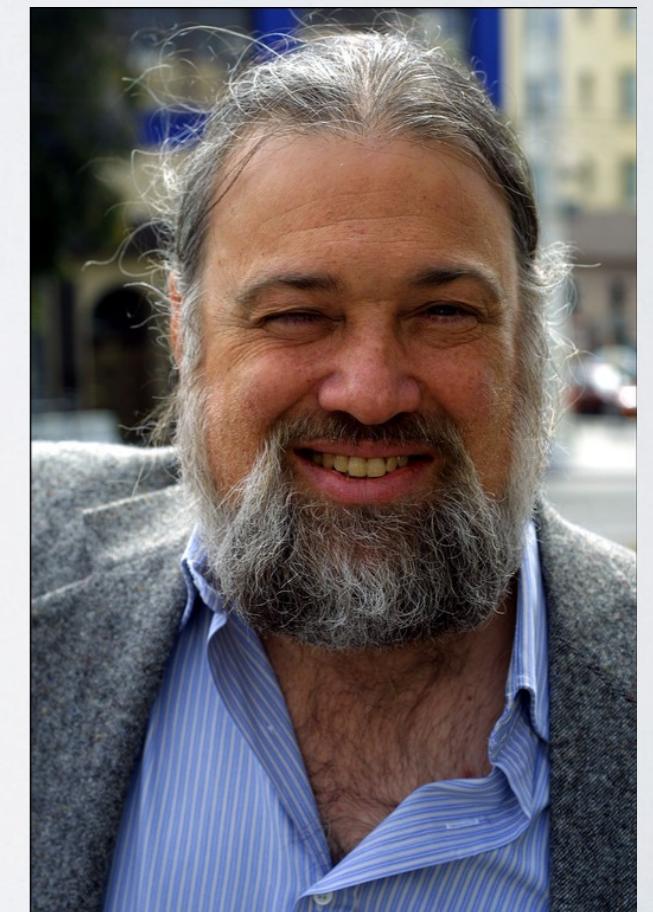
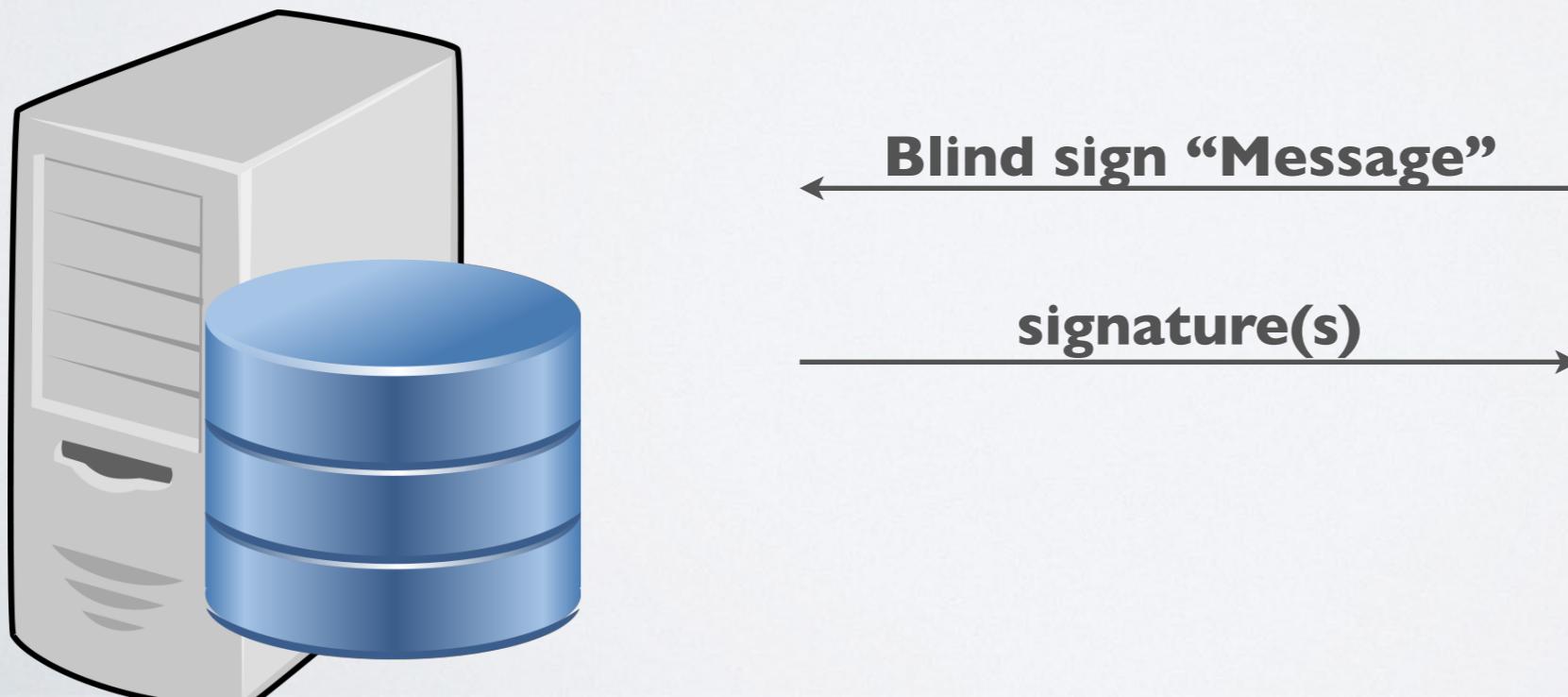
(e) 1HyYMMCdCcHnfjwMW2jE4cv9qVkJDFUzVa; and

# Electronic Cash (e-cash)

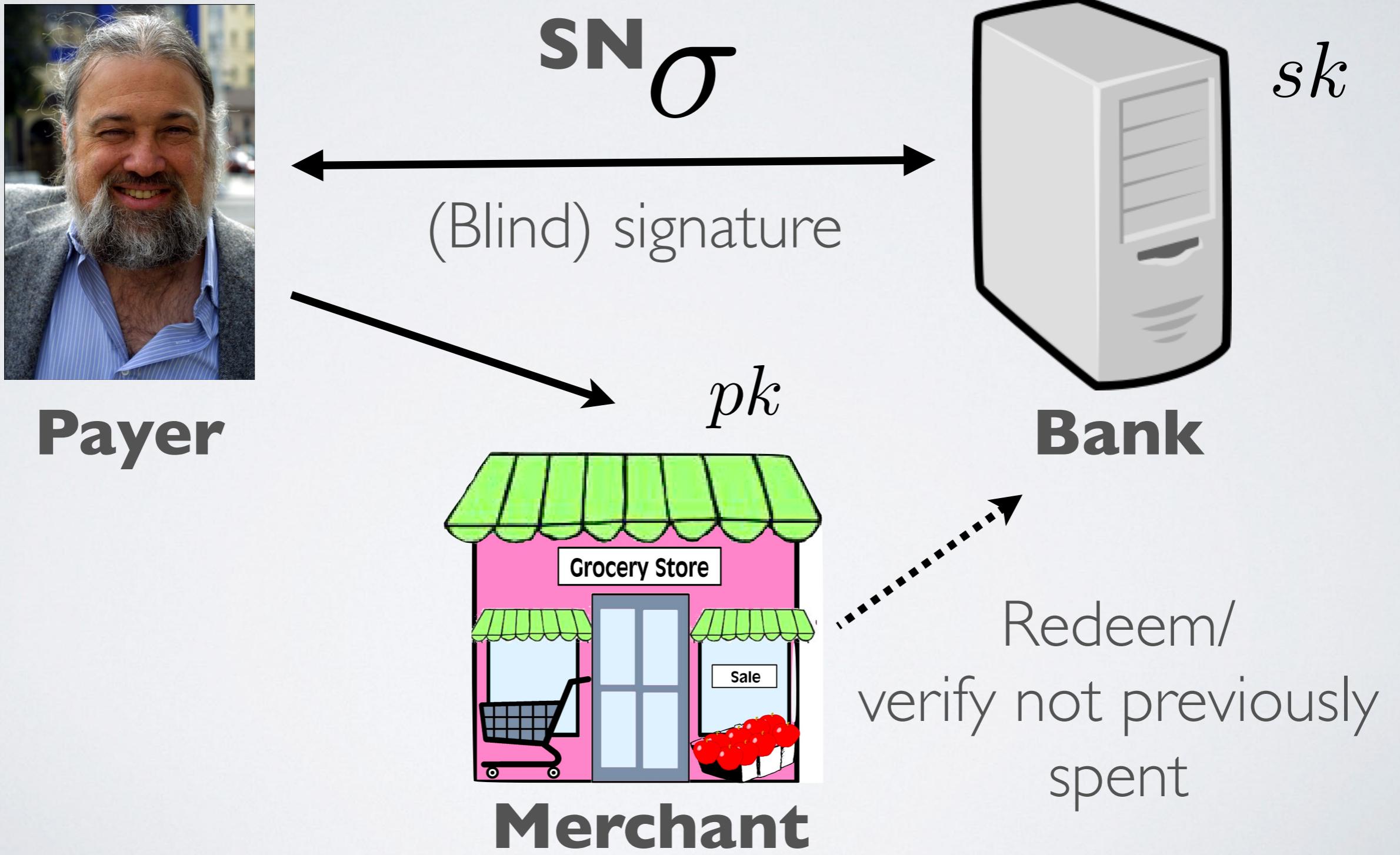
- Dates to Chaum [Chaum82] (many subsequent works)
  - Untraceable electronic cash
  - Withdraw ‘coins’ from a **central bank** (using blind signatures)
  - Even the bank can’t track the coins

# Blind signatures

- Allows a server (“signer”) with a secret key to produce a signature for a user who has a message
  - Server does not learn the message being signed (blind)
  - User gets only one signature



# Chaum (CRYPTO '83)



# “Textbook” RSA signature

- RSA signature:
  - Compute a public modulus  $N = pq$ , secret exponent “d”, public exponent “e”  
Output  $pk = (N, e)$ ,  $sk = d$
  - Signature: encode message as  $m \in \{1, \dots, N - 1\}$   
compute signature  $\sigma = m^d \bmod N$
  - Verify a message and signature using pk, as:  $m = \sigma^e \bmod N$

# Textbook RSA blind signature

- RSA blind signature:
  - Key generation is unchanged
  - User computes a random element  $r$   
calculates  $S = m \cdot r^e \pmod{N}$   
sends  $S$  to server
  - Server computes:  $S' = S^d \pmod{N}$   
sends  $S'$  back to user
  - User computes  $\sigma = S' \cdot r^{-1}$

Note: much more secure if  $m = \text{Hash}(\text{message})!$

Why doesn't e-Cash work  
(well) on a blockchain?

# Why doesn't e-Cash work (well) on a blockchain?

- Lack of centralized issuer
  - Technology fundamentally relies on a bank that holds a secret key
  - Who is going to take on this role in a decentralized system?
    - Volunteer nodes come and go, and aren't trustworthy
    - Anyone with this key can make unlimited tokens
  - Seems fundamentally problematic

# Bitcoin privacy solutions (early)

- **Generate a new address for each transaction**
  - Hierarchical wallet (BIP32) makes this easy, “pluralizes” one secret “key” into many addresses
  - Sounds good, except for “change”:
    - Leftover funds from a spend have to go somewhere
    - If they go to a new address, this links the old/new addresses

# Bitcoin privacy solutions (early)

- **Mixes/Laundry services**

- Create a centralized server; many people send coins
- Mixer shuffles those together, sends the right amount back to each user (less a fee), thus unlinking the sources of transactions
  - Risk 1: Mixer can “exit” and steal your cash
  - Risk 2: Mixer keeps track of the sources/destinations
  - Risk 3: Low volume of mixing can make tracing easy

# Bitcoin privacy solutions (early)

- **CoinJoin**

- Proposed by Maxwell; variants even earlier by “killerstorm” on BitcoinTalk\*
- Solves the “scamming mixer” problem
- Idea: each transaction has multiple inputs and outputs
  - Have a mixer author one single transaction that consumes N equal-value inputs, produces N outputs
  - All N parties then sign it

# Bitcoin privacy solutions (early)

- **CoinJoin** (cont'd)

- Idea is that if you trust the mixer to formulate the TX (keep your data private), the mixer can't steal your funds
  - Since this is a single (atomic) transaction, it either goes through completely or all funds remain untouched
  - Parties keep custody of their funds
  - Could combine with Chaumian e-cash to get privacy (e.g., TumbleBit)
    - \* <https://bitcointalk.org/index.php?topic=150681.0>

# Mixes/Coinjoin weaknesses

- Philosophical questions...?
- Still requires some centralized server
- Anonymity set:
  - “Anonymity set” for a transaction refers to the number of transactions that could plausibly represent the previous link in a transaction graph
  - Mixes/Coinjoin: size  $N$  for  $N$  users in a given time period  
(assumes all users are honest. If they aren't: 🤡)
  - Longer time periods mean larger  $N$ , but less convenient

# Mixes/Coinjoin weaknesses

- Improving anonymity set size?
  - Can we combine multiple, sequential mixings/CoinJoins to increase the anonymity set size?
    - E.g.,  $2 \times N$ -size CoinJoins gives  $N^2$  anonymity set
  - Any weaknesses?

# Decentralizing E-Cash

- The challenge:
  - Without a central bank we have no more **secret keys** which rules out digital signatures
  - We need a fundamentally different ingredient.



# Zerocoins (MGGR14)

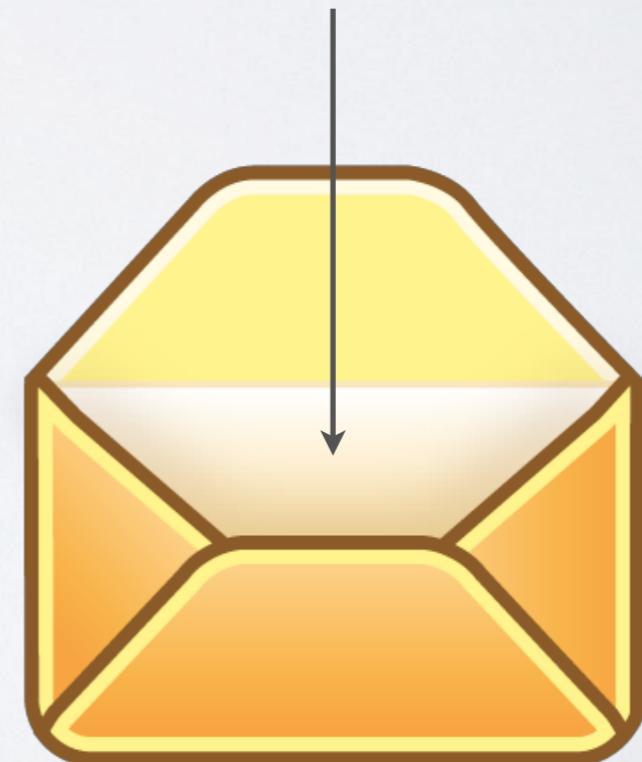
- Proposed as an extension to Bitcoin in 2014
  - Requires changes to the Bitcoin consensus protocol!
- I can take Bitcoin from my wallet
  - Turn them into 'Zerocoins'
  - Where they get 'mixed up' with many other users' coins
- I can redeem them to a new fresh Wallet



# Minting Zerocoins

- Zerocoins are just numbers
  - Each is a digital commitment to a random serial number
  - Anyone can make one!

82384827347|0|2983

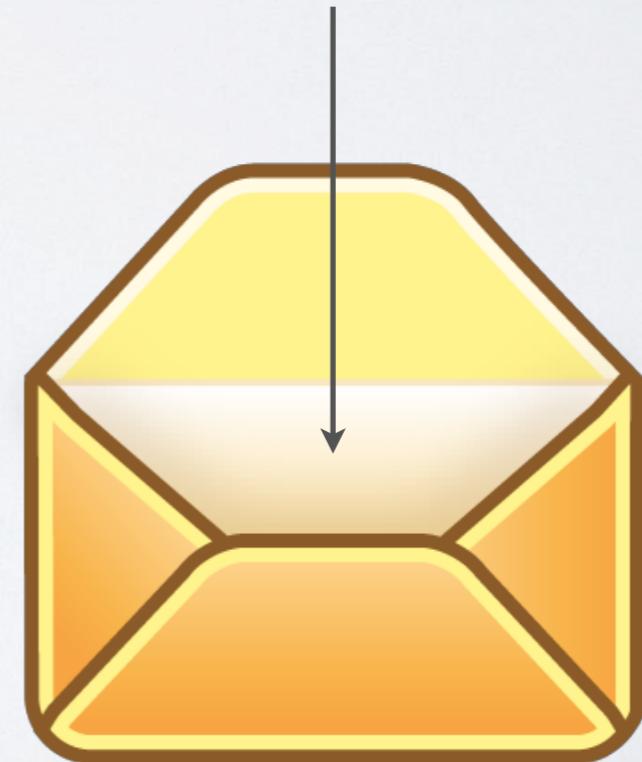


# Minting Zerocoins

- Zerocoins are just numbers
  - Each is a digital commitment to a random serial number SN
  - Anyone can make one!

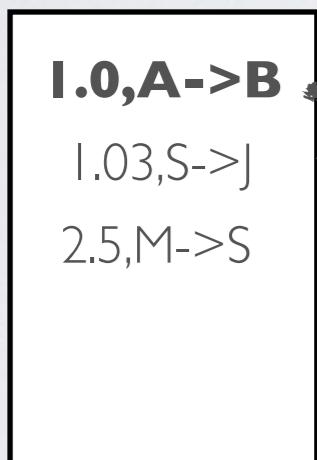
82384827347|0|2983

$$C = \text{Commit}(SN; r)$$

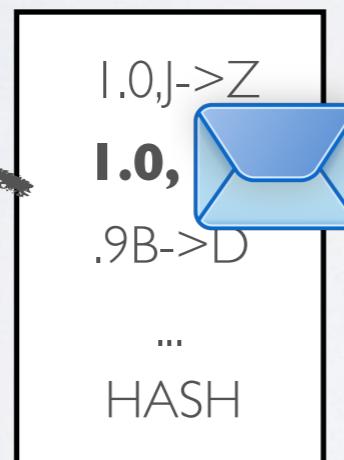


# Minting Zerocoins

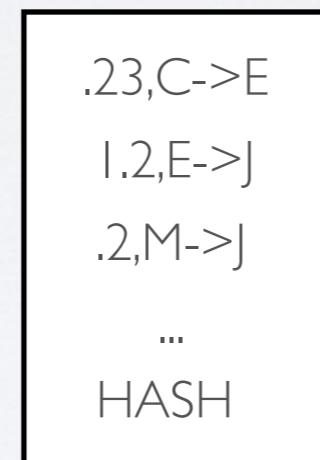
- Zerocoins are just numbers
- They have value once you write them into a valid transaction on the blockchain
- Valid: has inputs totaling some value e.g., 1 bitcoin



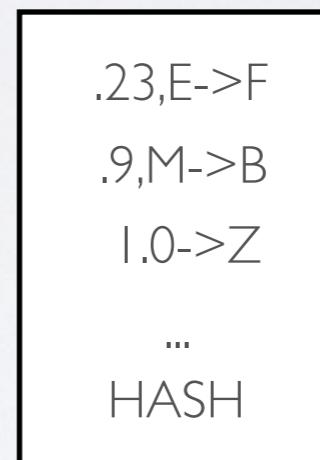
Block 1



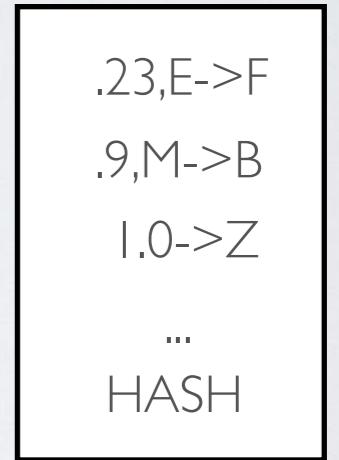
Block 2



Block 3



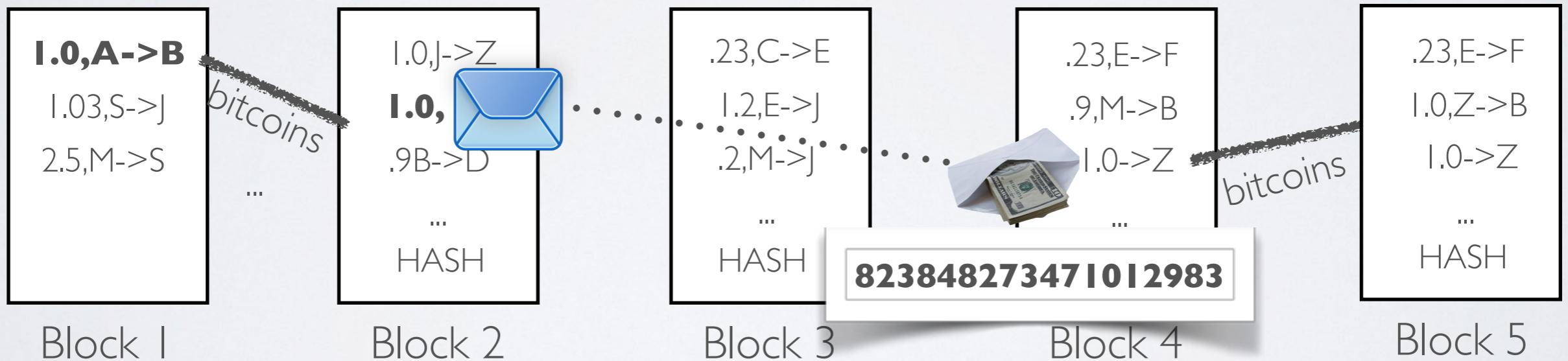
Block 4



Block 5

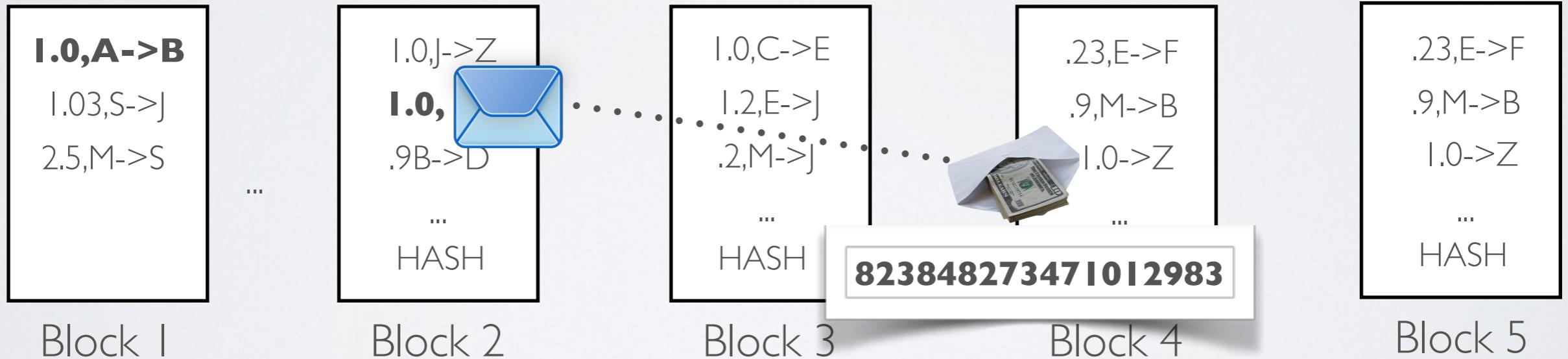
# Redeeming Zerocoins

- You can redeem zerocoins back into bitcoins
- Reveal the serial number &  
Prove that it corresponds to some Zerocoins on the chain
- In exchange you get one bitcoin (if SN is not already used)



# Spending Zerocoins

- Why is spending anonymous?
- It's all in the way we 'prove' we have a Zerocoins
- This is done using a zero knowledge proof



# Spending Zerocoins

- Zero knowledge [Goldwasser, Micali 1980s, and beyond]
  - Prove a statement without revealing any other information
  - Here we prove that:
    - (a) there exists a Zerocoins in the block chain
    - (b) we just revealed the actual serial number inside of it
  - Revealing the serial number prevents double spending
  - The trick is doing this efficiently!

# Spending Zerocoins

- Zero knowledge [Goldwasser, Micali | 1980s, and beyond]
  - Prove a statement without revealing any other information (other than that a statement is true)

# Spending Zerocoins

- Zero knowledge [Goldwasser, Micali 1980s, and beyond]
  - Prove a statement without revealing any other information
  - Here we prove that:
    - (a) there exists a Zerocoins (commitment) in the block chain
    - (b) the thing we revealed is the opening to that commitment
  - Revealing the serial number prevents double spending
  - The trick is doing this efficiently!

# Spending Zerocoins

- Possible proof statement (not efficient, see CryptoNote):
  - Public values: list of Zerocoins commitments  $C_1, C_2, \dots, C_N$   
Revealed serial number  $SN$
  - Prove you know a coin  $C$  and randomness  $r$  such that:
$$C = C_1 \vee C = C_2 \vee \dots \vee C = C_N \wedge C = Commit(SN; r)$$
- Problem: using standard techniques, this ZK proof has cost/size  $O(N)$

# Spending Zerocoins

- Zerocoins (actual protocol)
  - Use an efficient RSA one-way accumulator
  - Accumulate  $C_1, C_2, \dots, C_N$  to produce a short value  $A$
  - Then prove knowledge of a short witness s.t.  $C \in \text{inputs}(A)$
  - And prove knowledge that  $C$  opens to the serial number

Requires a DDL proof (**~25kb**)  
for each spend. In the block chain.

# Spending Zerocoins

- Zerocoins (actual protocol)
  - Use an efficient RSA one-way accumulator
  - Accumulate  $C_1, C_2, \dots, C_N$  to produce a short value  $A$
  - Then prove knowledge of a short witness s.t.  $C \in \text{inputs}(A)$
  - And prove knowledge that  $C$  opens to the serial number

Requires a DDL proof (**~25kb**)  
for each spend. In the block chain.

# CryptoNote & RingCT

- 2012: CryptoNote
  - Originally launched as part of the ByteCoin currency, predates Zerocoin
  - Anonymous creators, did a pre-mine
  - Was forked multiple times into many different currencies, including bitmonero -> Monero
  - Protocol ideas later improved into RingCT, which hid amounts as well as inputs (used in Monero today)

# CryptoNote idea

- Each transaction lists a set of  $N$  inputs (all same value).
- One is the real input, and  $N-1$  are “cover traffic”, AKA “mixins”
  - A ZKP proves that the transaction creator has a key to one of the input transaction, but not which one
    - This requires a proof (“ring signature”) of size  $O(N)$
  - Post proof plus a “key image” (function of the secret key) to prevent that transaction being spent twice

# RingCT idea

- Extends this idea to support variable-value transactions
- Add to the commitment a transaction value  $V$

$$C = \text{Commit}(SN\|V; r)$$

- Encrypt the commitment and randomness under a recipient's public key, store the ciphertext on the chain

$$\text{Ciph} = \text{Encrypt}(pk, SN\|r\|V)$$

- Must also prove that the output value of the next commitment is less than or equal to input transaction(s)

# Anonymity set comparison

- Anonymity set in CoinJoin:
  - **M**: where **M** is number of inputs in the transaction (bounded by TX size)
- Anonymity set in ByteCoin/RingCT:
  - **N**: where **N** is the number of inputs allowed in a transaction (bounded by TX size, 7-11 historically)
- Anonymity set in Zerocoins:
  - **P**: where **P** is number of total Zerocoins minted on the blockchain thus far\* (independent of TX size)

# Next time

- So far we've given a very high level view of Zerocoin, CryptoNote/RingCT
  - Next time we'll talk about Zerocash (Zcash) and MimbleWimble
  - As well as other online techniques like Tumblebit