

# **BIT COIN - 3**

**Reference**

**Bitcoin and Cryptocurrency Technologies**

**By Arvind Narayanan, Joseph Bonneau, Edward Felten,  
Andrew Miller, Steven Goldfeder**

Main reference: *Bitcoin and Cryptocurrency Technologies*,

By Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder

Slides are mainly taken (or adapted) from slides of the authors of the text

# Anonymity basics

# What do we mean by anonymity?

- Literally: *anonymous* = ‘without a name’
  - Two interpretations:
    - 1 – without using your real name
    - 2 – without using any name at all
- Bitcoins: Addresses are public key hashes rather than real identities
  - By first interpretation, Bitcoin is anonymous
  - By second interpretation, it is not anonymous!  
Bitcoin addresses → pseudo-identity
  - Computer scientists call this *pseudonymity*

# Bitcoin provides anonymity?

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

— WikiLeaks donations page

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

— Wired UK

Risk that transactions can be linked back to the people that made them

# What do we mean by anonymity?

Literally: ***anonymous*** = ‘*without a name*’

- Two interpretations:
  - 1 – without using your real name
  - 2 – without using any name at all

Bitcoins: Addresses are public key hashes rather than real identities

- By first interpretation, Bitcoin is anonymous
- By second interpretation, it is not anonymous!  
Bitcoin addresses → pseudo-identity
- Computer scientists call this ***pseudonymity***

# Anonymity vs. Privacy

Anonymity is **insufficient** for privacy

Anonymity is **necessary** for privacy

Anonymity is **unachievable** in practice

Re-identification attack → anonymity breach → privacy breach

Just ask Justice Scalia (US Supreme Court)  
“It is silly to think that every single  
datum about my life is private”

# What do we mean by anonymity?

**Literally: anonymous = without a name**

Bitcoin addresses are public key hashes rather than real identities

Bitcoin addresses → pseudo-identity

**Unlikability:**

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

**Anonymity = pseudonymity + unlinkability**

# Wikileaks

WikiLeaks, an international organization for anonymous whistleblowers, accepts *anonymous* donations via Bitcoin claiming that

- “*Bitcoin is a secure and anonymous digital currency. Bitcoins cannot be easily tracked back to you, and are a [sic] safer and faster alternative to other donation methods.*”
- Note the refresh button that every time you refresh you use a different address

**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:

16Vo6XJb2fD4vwVcct3FmqZSMzPGBMMvTQ 

# Wikileaks

- Note the refresh button that every time you refresh you use a different address
- It is trivial to create a new address every time

**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:

16Vo6XJb2fD4vwVcct3FmqZSMzPGBMMvTQ 

So, unlinkable?

# Pseudonymity vs anonymity in forums

## Reddit: pick a long-term pseudonym

- . American social news aggregation, web content rating discussion website

## 4Chan: make posts with no attribution at all

- . anyone can post comments and share images. Users do not need to register an account before participating in the community

# Defining unlinkability in Bitcoin

## 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

Anonymity = pseudonymity + unlinkability

*Question: Why is unlinkability needed?*

*Because pseudonymity **is not enough** for  
“privacy” in Bitcoins!*

- Bitcoin Block chain (and transactions) is public
- Linking Bitcoin addresses to real identities is possible
- This might allow to find identities
  - Easy: Many Bitcoin services require real identity
  - More complicated: Linked profiles can be deanonymized by a variety of side channels

# Unlinkability

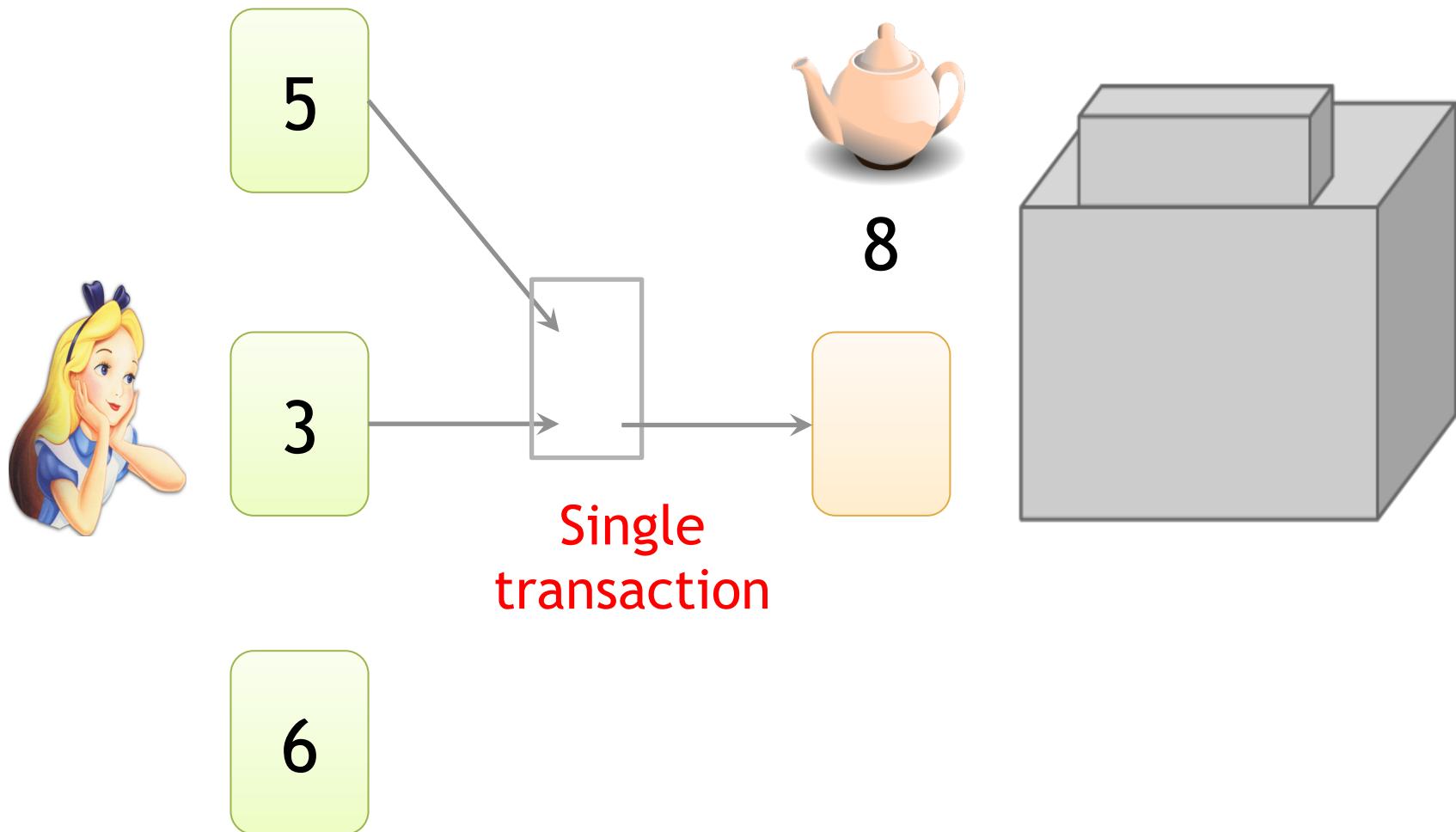
key properties that are required for Bitcoin activity to be unlinkable:

1. It should be hard to link together different addresses of the same user.
2. It should be hard to link together different transactions made by the same user.
3. It should be hard to link the sender of a payment to its recipient (this means the ultimate recipient if some trick has been used by including fake transitions)

Best practice: always receive at fresh address  
So, unlinkable?

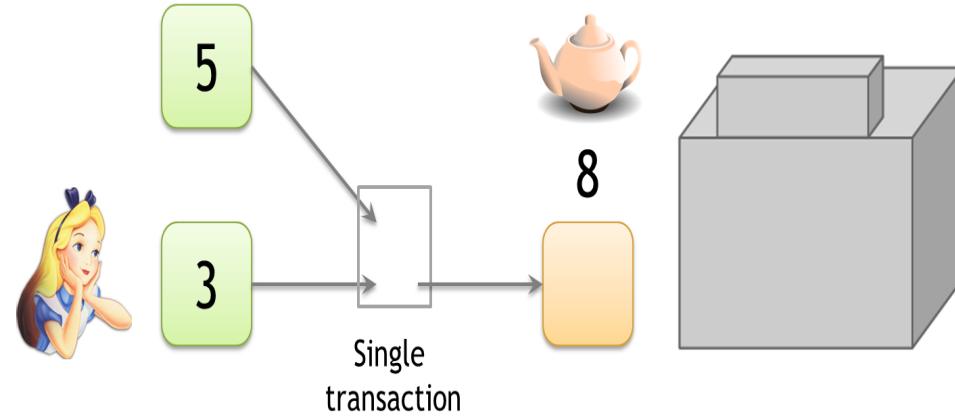
# How to de-anonymize Bitcoin?

# Alice buys a teapot at Big box store



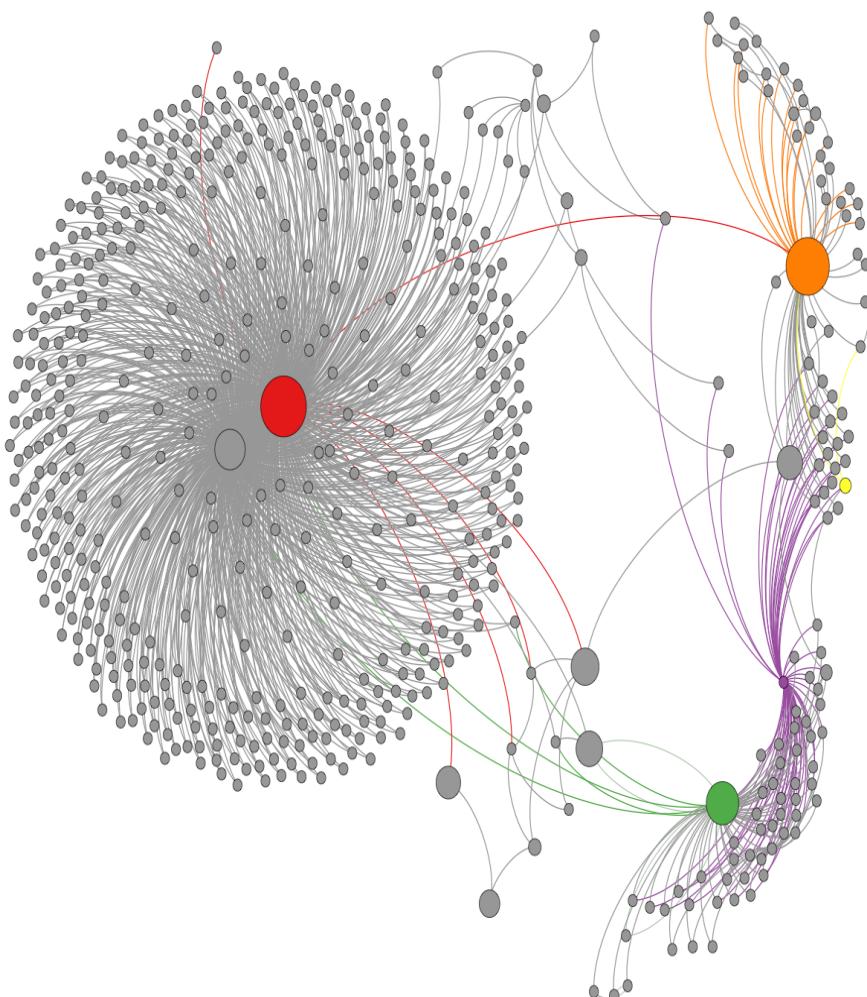
# Linking addresses

Shared spending is evidence of joint control



Addresses can be linked transitively

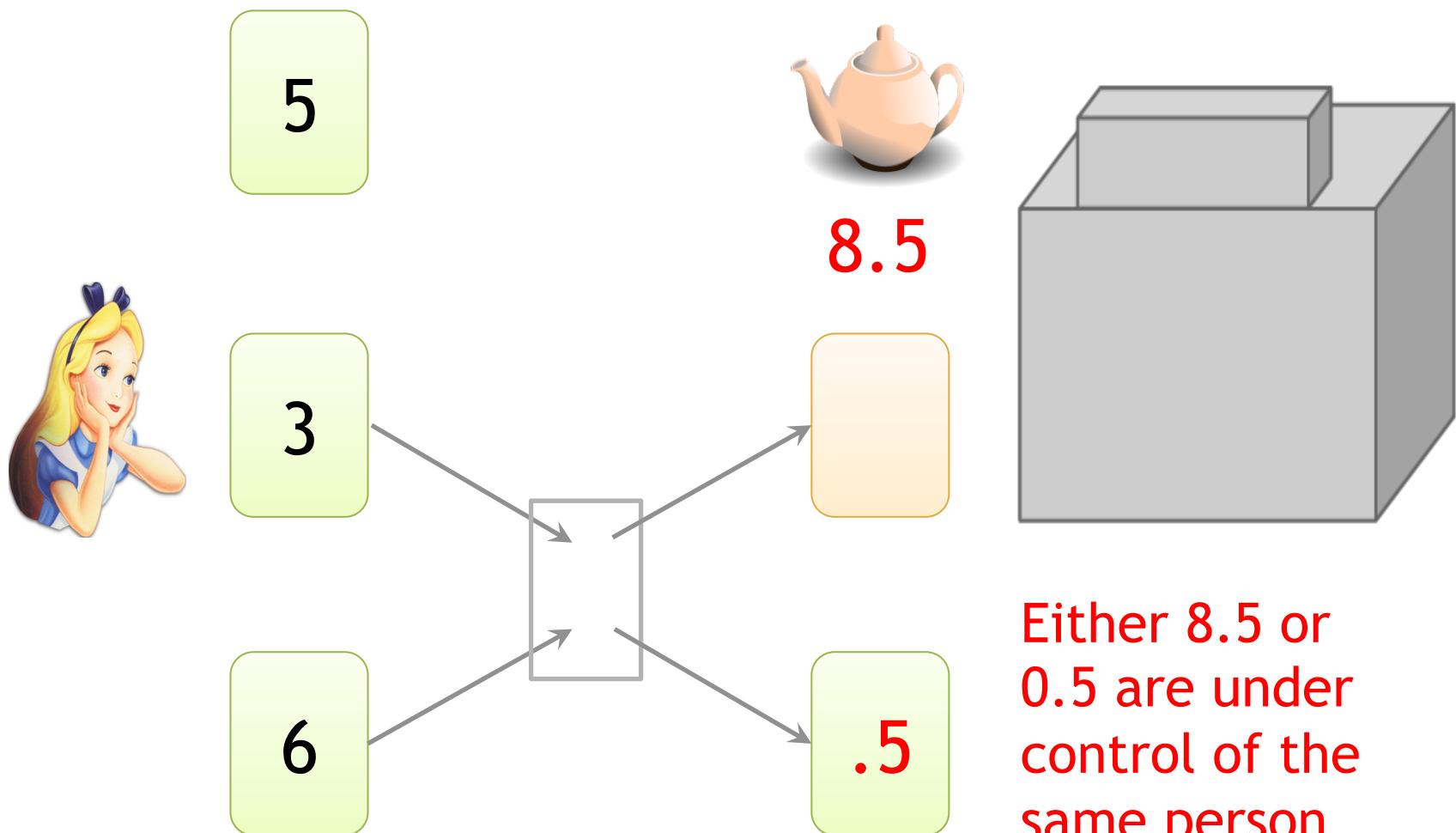
# Clustering of addresses



*An Analysis of Anonymity  
in the Bitcoin System*

F. Reid and M. Harrigan  
PASSAT 2011

# Change addresses



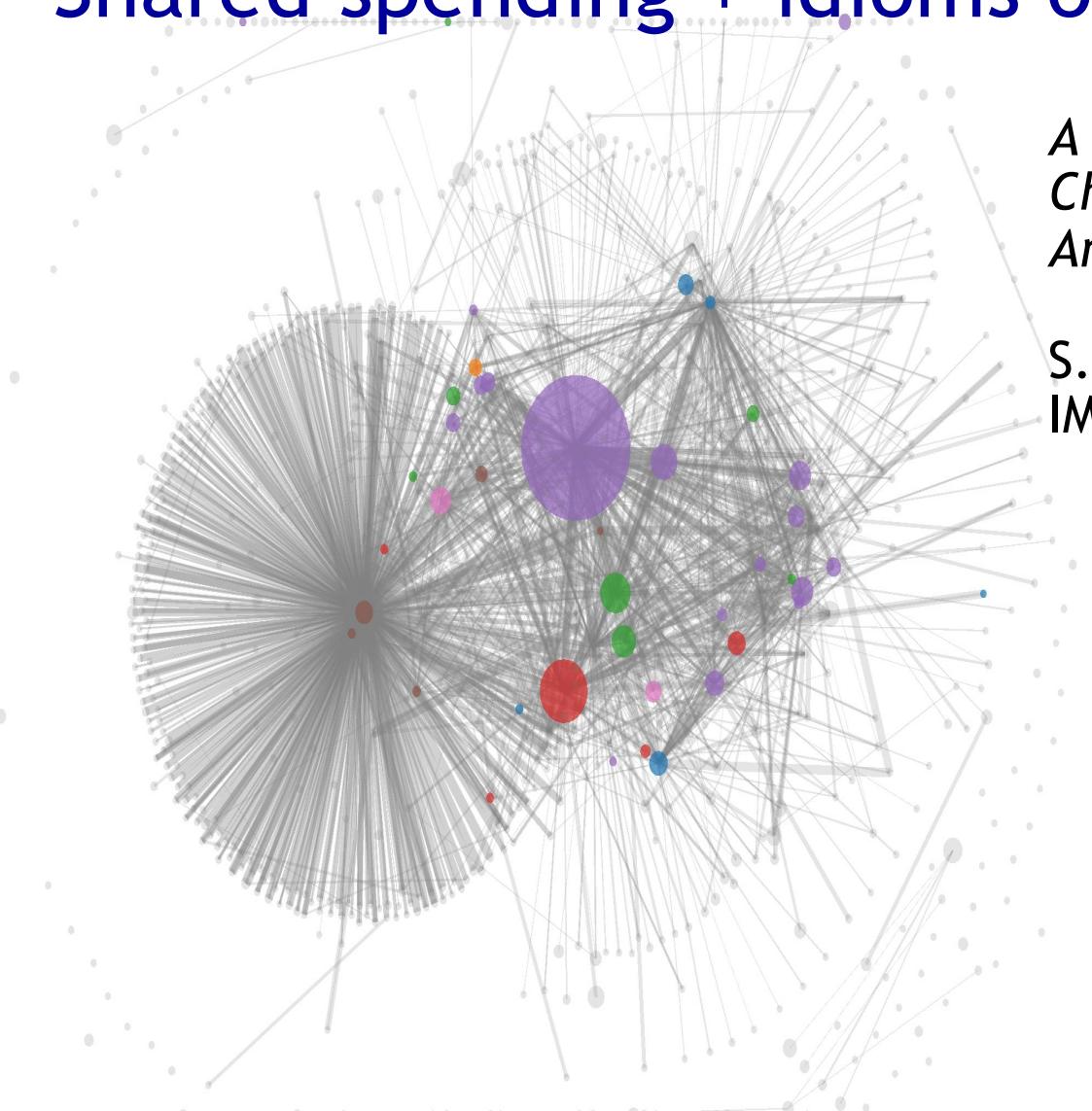
## “Idioms of use”

Idiosyncratic features of wallet software

An old version put the change address always as the first address! *Solution: random choice*

In 2013 it has been found that many wallets were using a new address as change address

# Shared spending + idioms of use



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

S. Meiklejohn et al.  
IMC 2013

# To tag service providers: transact!



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

S. Meiklejohn et al.

344 transactions

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

# From services to users

## 1. High centralization in service providers

Most flows pass through one of these – in a traceable way

## 2. Address – identity links in forums

# From links to users

Can we connect little clusters corresponding to individuals to their real-life identities?

- Directly transacting with individuals
- Via service providers: eventually a bitcoin user will interact with a service provider that must ask your identity (by the law)
- Carelessness: people are not always concerned
- Use of network layer characteristic: often “*the first node to inform you of a transaction is probably the source of it*”

# Bitcoin anonymity: conclusions

- Bitcoin is **not** anonymous
- Bitcoin is pseudonymous, but pseudonymity is not enough if your goal is to achieve privacy
- Recall that the block chain is public and anyone can look up all Bitcoin transactions that involved a given address
- If anyone is ever able to link your Bitcoin address to your real world identity, then all of your transactions — past, present, and future — will have been linked back to your identity

# Ethical concerns about anonymity

***Legitimate “goods”***: may prevent learning someone's salary information or an organization's private business dealing

***Legitimate “worries”*** : money laundering  
Bottleneck: moving large flows into and out of Bitcoin (“cashing out”)

# Legal and Regulatory Risk

- Law-abiding user might lose their funds if an exchange is shut down for criminal activity
- Uncertain tax treatment of gains/losses due to currency fluctuations
- Let's talk about Bitcoin's regulatory environment

# Regulating Bitcoin

- Bitcoin's original vision is in tension with regulation and government control
  - Strong cyber-libertarianism streak
  - The decentralized design makes it harder, but by no means impossible, to regulate

# Making the case for oversight

- Untraceable digital cash defeats capital controls
- Country can't stop Bitcoin value from flowing in or out
- Government countermeasure: disconnect BTC world from financial institutions
- Example: China

# Making the case for oversight

Untraceable digital cash facilitates some crimes:

- kidnapping and extortion
- tax evasion
- sale of illegal items

Welcome! | Silk Road

>Welcome  
messages(0) | orders(0) | account(\$0.00) | settings | log out

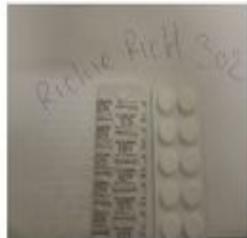
search | (0)

# Silk Road

anonymous marketplace

Shop by category:

- Drugs(1249)
- Cannabis(410)
- Ecstasy(86)
- Dissociatives(47)
- Psychedelics(142)
- Opioids(92)
- Stimulants(107)
- Other(150)
- Benzos(96)
- Lab Supplies(23)
- Digital goods(93)
- Services(107)
- Money(71)
- Weaponry(9)
- Home & Garden(4)
- Food(1)
- Electronics(11)
- Books(76)
- Drug paraphernalia(46)
- XXX(48)
- Medical(3)
- Computer equipment(19)
- Art(1)
- Apparel(8)
- Sporting goods(3)
- Tickets(1)
- Forgeries(13)
- Fireworks(2)

 1g Tangerine Kush Bubble Hash \$60.96	 -NN- DMT YELLOW CLASSIC (500mg) \$19.39	 Barcode Manipulation scam keeping... \$2.31
 3.5g OG Kush \$22.17	 MDMA and MDEA mixture 1 gram \$23.44	 Guerrilla Warfare Book's \$0.46
 co-codamol 30mg codeine / 500mg... \$4.59	 CASH BLOWOUT!! Vendors, SYG is... \$0.01	 *Super BOMB* Jolly Rancher 1/8... \$24.20

News:

- Site glitches
- Missing deposits
- Site restored
- Forum bugs addressed
- Pricing and hedging improvements
- Escrow hedging update
- New feature to help protect sellers
- Seller ranking and feedback overhaul

# SILK Road

Largest online market for illegal drugs

- ran as a Tor hidden service
- payment in Bitcoins
- site held BTC in escrow while goods shipped
- eBay-like reputation system
- run by “Dread Pirate Roberts”

operated February 2011 to October 2013

# SILK Road

- online **black market** and the first modern **darknet market** best known as a platform for selling illegal drugs.
- part of the **dark web** it was operated as a **Tor hidden service**, such that online users were able to browse it anonymously and securely without potential traffic monitoring
- Initially there were a limited number of new seller accounts available; new sellers had to purchase an account in an auction. Later, a fixed fee was charged for each new seller account.

# SILK Road



- October 2013: “Dread Pirate Roberts”, Ross Ulbricht, was arrested
- Charged with money laundering, computer hacking, conspiracy to traffic narcotics, murdering
- Convicted and sentenced to life imprisonment in 2015
- He tried to cover his tracks, but they connected the dots
- Government seized 174,000 BTC
- November 2013 (after R.Ulbricht was arrested and Silk Road was closed) Silk Road 2.0 was opened
- November 2014 responsible of Silk Road 2.0 were convicted and the site was closed
- Silk Road 3.0 ....

# Silk Road: Conclusions

- Hard to keep real life (Ross Ulbricht) and virtual life (Dread Pirate Roberts) separate: a connection between the two (e.g. Ulbricht used the same computer to access his personal accounts and Dread Pirate account)
- Hard to stay anonymous for a long time
- Police can “follow the money” ⇒  
money becomes untouchable (Ulbricht owned 174K Bitcoin = 30M US\$ but could not spend!)

# Alternate measures of Bitcoin anonymity

- . The “taint” of a Bitcoin transaction evaluates the association between an address and earlier transaction addresses. The more “taint” the stronger the link between the two addresses.
- . Give high “taint” score (to  $S\text{-}R$  pair) if Bitcoins sent by an address  $S$  always end up at an address  $R$

# Alternate measures of Bitcoin anonymity

Give high “taint” score (to  $S$ - $R$  pair) if Bitcoins sent by an address  $S$  always end up at an address  $R$

Not a good measure of Bitcoin anonymity:

- Makes implicit assumptions about the adversary’s calculations regarding linking pairs of addresses
- Adversary may use alternate techniques for linking address pairs – e.g., timing analysis or wallet software idiosyncrasies.

# Ethical concerns about anonymity

***Legitimate “goods”***: may prevent learning someone's salary information or an organization's private business dealing

***Legitimate “worries”*** : money laundering  
Bottleneck: moving large flows into and out of Bitcoin (“cashing out”)

# Can we keep only the good uses?

Common conundrum in computer security and privacy:

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

## **Similar dilemma: Tor**

Anonymous  
communication  
network

Sender and receiver of  
message unlinkable

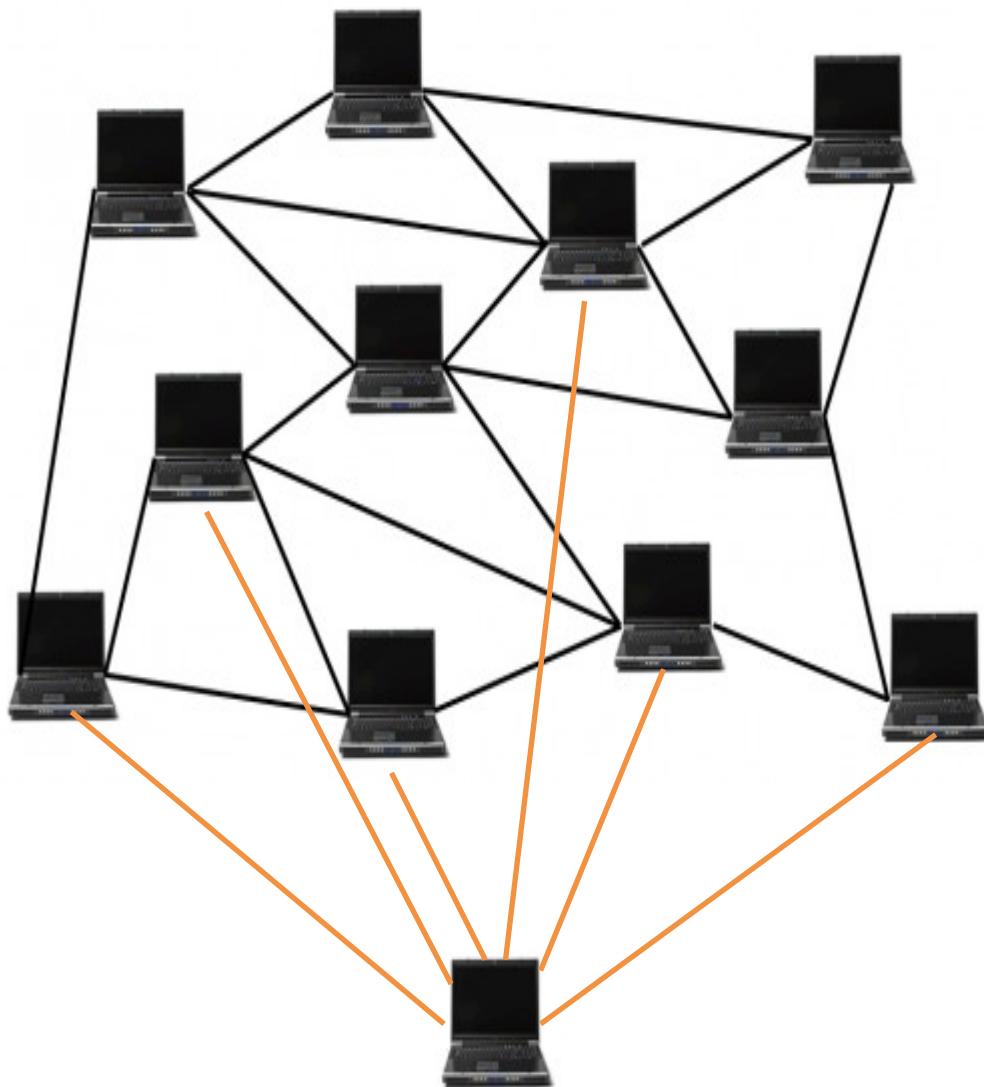
Used by:

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

Funded by (among  
others):

U.S. State Department

# Network-layer de-anonymization



“The first node to inform you of a transaction is probably the source of it”

Dan Kaminsky  
Black Hat 2011 talk

## **Solution: use Tor**

Caveat: Tor is intended for low-latency activities such as web browsing

Mix nets might provide better anonymity

BUT Tor is what's deployed and works

# How to make Bitcoin anonymous

## Mixing

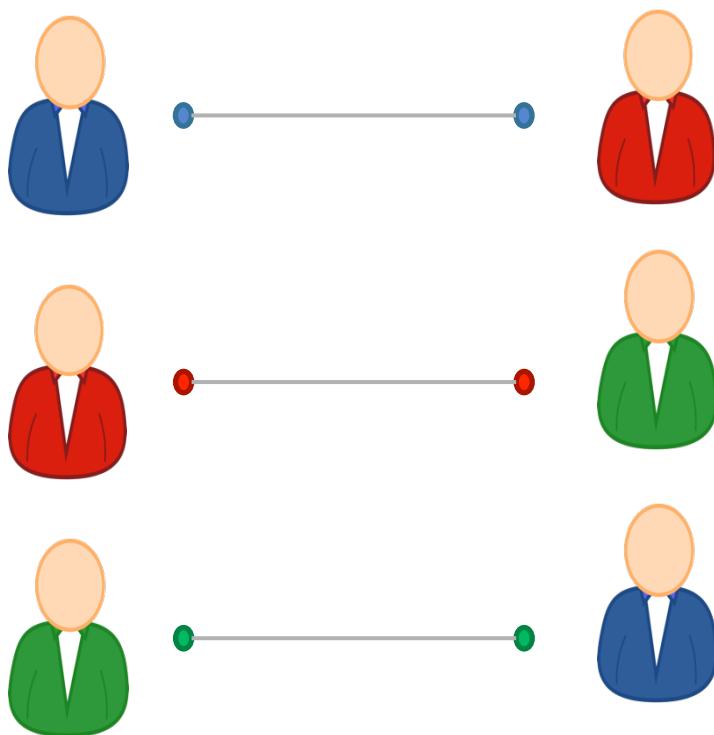
# Mixing: terminology

- Bitcoin activities are recorded and available publicly via the blockchain
- when you finally use Bitcoin to pay for goods and services, you will of course need to provide your name and address to the seller for delivery purposes.
- It means that a third party can trace your transactions and find ID information.
- To avoid this, such mixing service provide the ability to exchange your bitcoins for

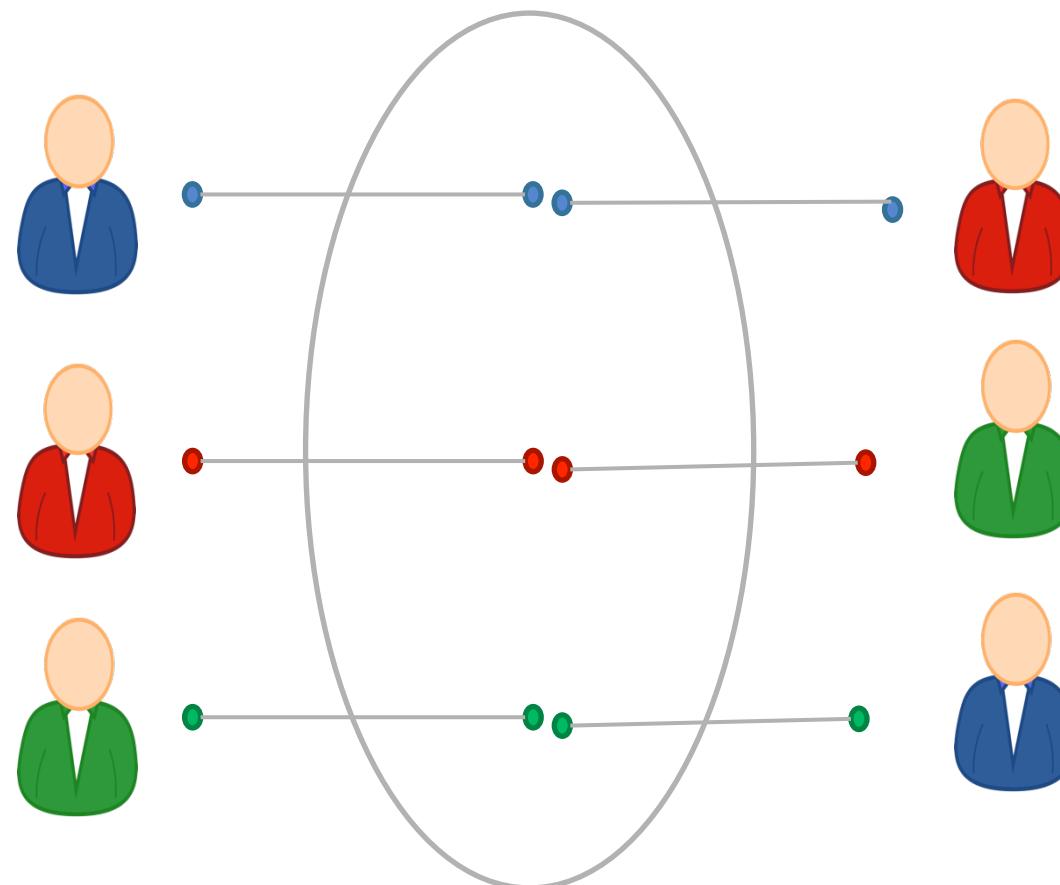
# Mixing: terminology

- **mixing services** are used to mix one's funds with other people's money, intending to confuse the trail back to the funds' original source
- In traditional financial systems, the equivalent would be moving funds through banks in countries with bank secrecy laws (Cayman islands, Bahamas etc.)
- When mixing bitcoins, you send your money to an anonymous service and they will send you someone else's coins.
- So, now, whatever those coins were used for may now be traceable back to you. Additionally, mixing large amounts of money may be illegal,

**Mixing** mix one's funds with other people's money,



# To protect anonymity, use an intermediary





---

## Study Suggests Link Between Dread Pirate Roberts and Satoshi Nakamoto

By JOHN MARKOFF NOVEMBER 23, 2013 6:13 PM □ 23 Comments



E-MAIL



FACEBOOK



TWITTER



SAVE

Two Israeli computer scientists say they may have uncovered a puzzling financial link between Ross William Ulbricht, the [recently arrested](#) operator of the Internet black market known as the Silk Road, and the secretive inventor of bitcoin, the anonymous online currency, used to make Silk Road purchases.



---

## Researchers Retract Report That Linked Bitcoin Creator and Silk Road

By JOHN MARKOFF NOVEMBER 27, 2013 12:45 PM □ 6 Comments



E-MAIL



FACEBOOK



TWITTER



SAVE



MORE

Two Israeli computer scientists who over the weekend published a paper describing a financial connection between the Bitcoin peer-to-peer transaction system and the operator of Silk Road, an Internet black market, have backed away from the claim after an independent security researcher took responsibility for the puzzling account that generated the transfer.

# Dedicated mixing services

- Promise not to keep records
- Don't ask for your identity
- Essentially swaps a user's coin (address) with other users' coins (addresses)
- Relatively small anonymity set (only comprise of those users who use the mixing service at that instant)!

## Back to online wallets

Reputable, often regulated, businesses

- Typically require identity, keep records → no anonymity w.r.t. wallet service
- Users trust them with their bitcoins → keep them for longer → bigger anonymity set w.r.t. everyone else

# Principles for mixing services

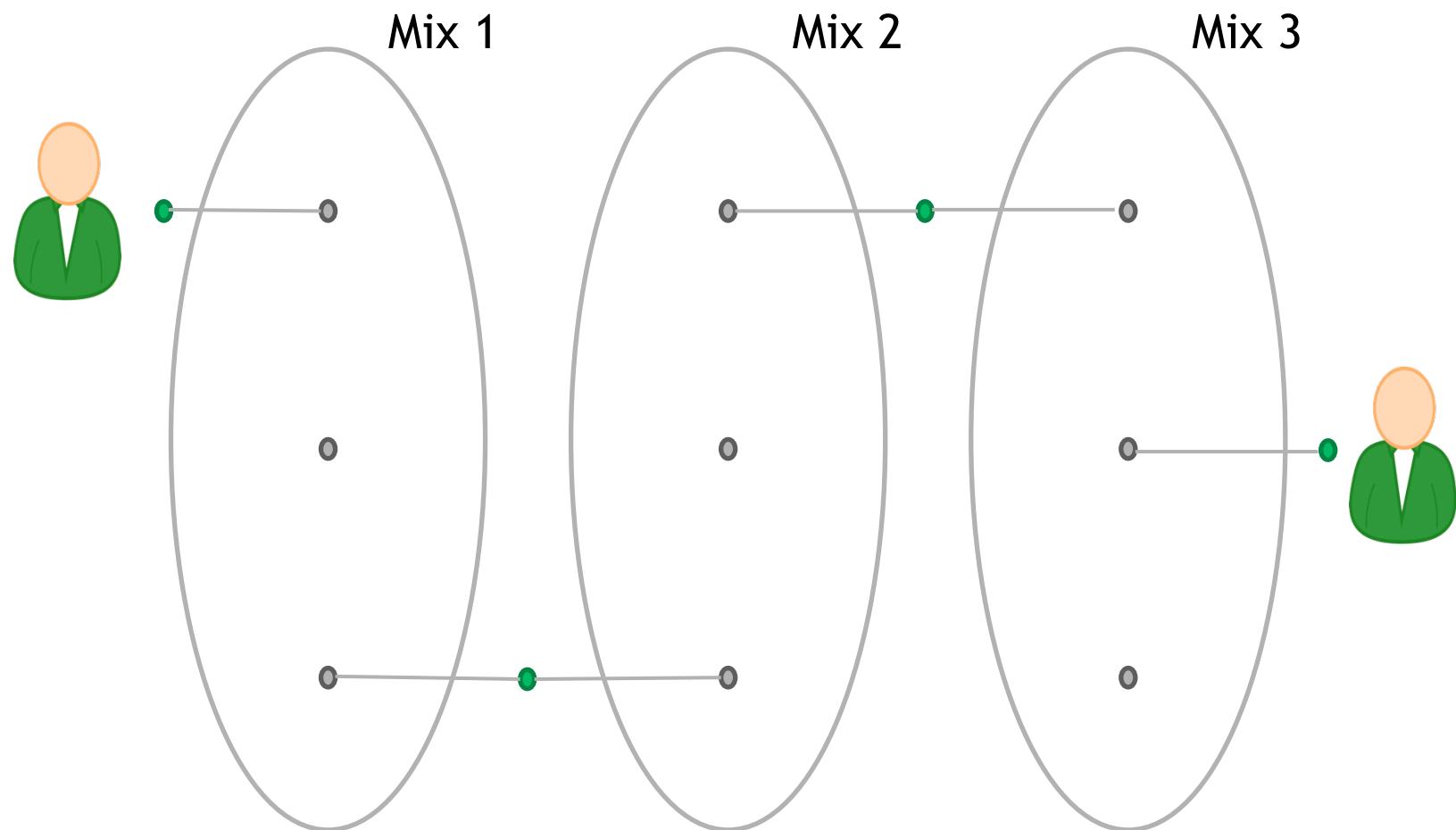
## 1. Use a series of mixes

Mixes should implement a standard API to make this easy

*Mixcoin: Anonymity for Bitcoin with accountable mixes*

J. Bonneau et al.  
Financial Cryptography  
2014

# Series of mixes



# Principles for mixing services

2. Uniform transactions

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

*Mixcoin: Anonymity for Bitcoin with accountable mixes*

J. Bonneau et al.

3. Client side must be automated

Financial Cryptography  
2014

Desktop wallet software

# Remaining problem: trusting mixes

1. Stay in business, build up reputation
2. Users can test for themselves
3. Cryptographic “warranties”

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.

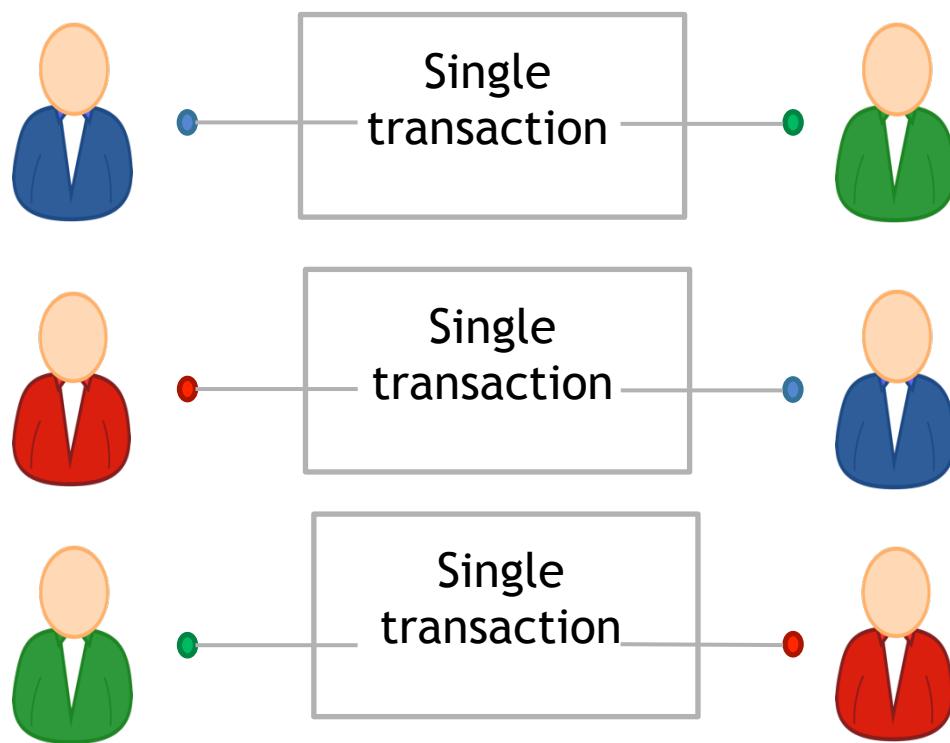
— Bitcoin Wiki

# Decentralized mixing

# Why decentralized mixing?

- No bootstrapping problem
- Theft impossible
- Possibly better anonymity
- More philosophically aligned with Bitcoin

# Coinjoin

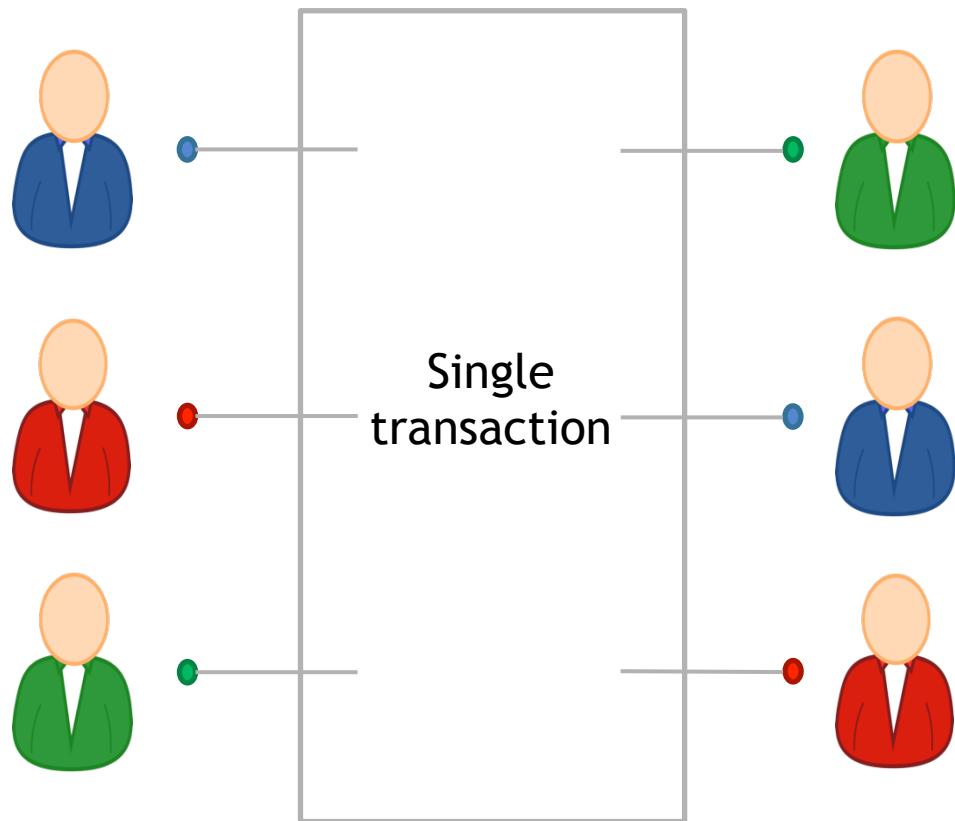


3 transactions

Each signature is entirely  
separate

Proposed by Greg Maxwell, Bitcoin core developer

# Coinjoin



Each signature is entirely separate

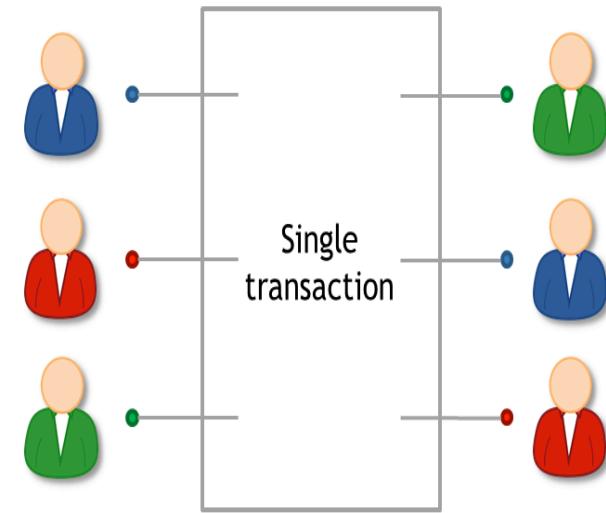
This is 1 mixing round

Mixing principles from before apply on top of basic protocol

Proposed by Greg Maxwell, Bitcoin core developer

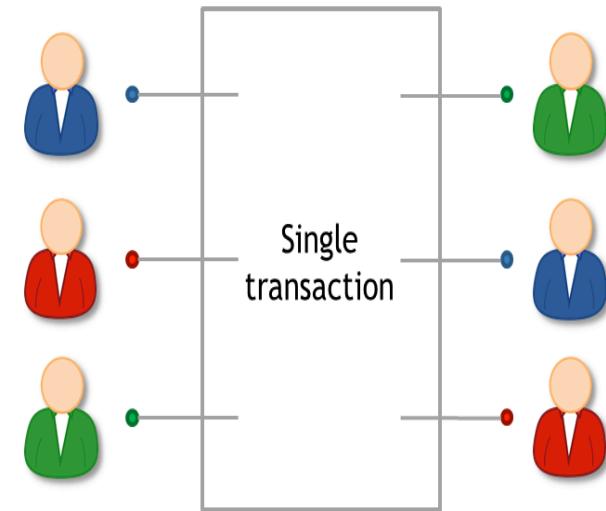
# Coinjoin algorithm

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



# Coinjoin: remaining problems

- How to find peers
- Peers know your input-output mapping  
(This is a worse problem than for centralized mixes)
- Denial of service



# Finding peers

Problem

Use an (un)trusted server

The server learns the mapping

# **Peer anonymity**

Strawman solution:

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

Better solution:

special-purpose anonymous routing mechanism

# Denial of service

Proposed solutions:

- Proof of work
- Proof of burn
- Server kicks out malicious participant
- Cryptographic “blame” protocol

*(CoinShuffle: Practical Decentralized Coin Mixing  
for Bitcoin*

T. Ruffing et al., PETS 2014)

# **frequent flows could be identifying**

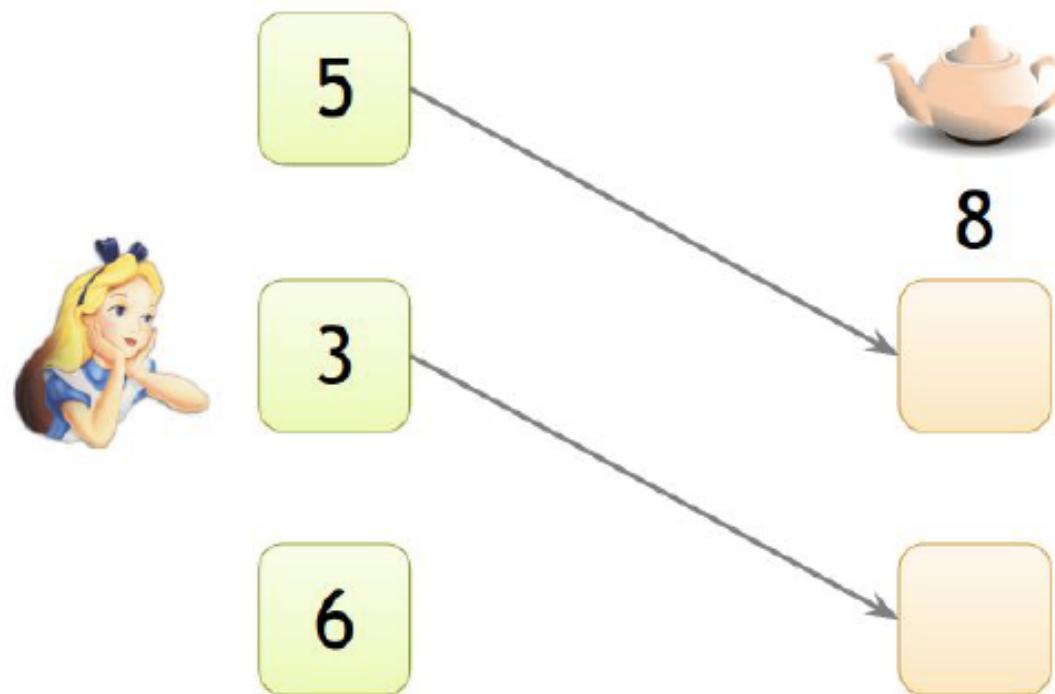
Example:

Alice receives 43.12312 BTC / week as income

Always immediately transfers 5% to retirement account

# Heuristic: merge avoidance

Instead of a single payment transaction  
receiver provides multiple output addresses  
sender avoids combining different inputs



# Main digital coins nov. 2020

Cryptocurrency	Price (USD)	Price Graph (7D)	Market Cap	24hr Volume
1  Bitcoin BTC	\$15,436.83 <span>+0.31%</span>		\$286.17 B	\$7.43 B
2  Ethereum ETH	\$459.78 <span>+1.63%</span>		\$52.13 B	\$3.70 B
3  Tether USDT	\$1.0011 <span>+0.05%</span>		\$17.48 B	\$7.70 B
4  XRP XRP	\$0.2563 <span>-1.54%</span>		\$11.62 B	\$382.81 M
5  Bitcoin Cash BCH	\$257.39 <span>-2.26%</span>		\$4.78 B	\$359.22 M
6  ChainLink LINK	\$13.11 <span>-0.52%</span>		\$4.59 B	\$429.35 M
7  Polkadot DOT	\$4.4691 <span>+0.76%</span>		\$4.16 B	\$38.00 M
8  Binance Coin BNB	\$28.13 <span>-1.30%</span>		\$4.16 B	\$118.48 M

# Hard fork and split coins

Hard fork: a state of the network or a software update? Two definitions

1. Some nodes have accepted a block, but others have rejected it. The nodes accepting the questionable block keep and extend it. The nodes rejecting the block neither keep nor extend it, but rather build their own branch. The resulting "fork" is only visible to nodes that accepted the questionable block.
2. A hardfork is a change to the bitcoin protocol that makes previously invalid blocks/transactions valid, and therefore requires all users to upgrade. In other words, a hard fork is a kind of update to Bitcoin Core (given that Bitcoin Core is the protocol)

August 2013: A block that had a larger number of total transaction inputs than previously seen was mined and broadcasted. Bitcoin 0.8 nodes were able to handle this, but some pre-0.8 Bitcoin nodes rejected it, causing an unexpected **hard fork of the blockchain**. The pre-0.8 incompatible chain at that point had around 60% of the hash power ensuring the split did not automatically resolve.

It's not hard to see how a software update could lead to persistent block chain fork as seen by some nodes. So clearly the two ideas are connected.

# Hard fork and split coins

- Soft fork and hard fork
- Split coins

# Hard fork Soft fork

A *hard fork* is a rule change such that the software validating according to the old rules will see the blocks produced according to the new rules as invalid.

- In case of a hard fork, all nodes meant to work in accordance with the new rules need to upgrade their software.
- If one group of nodes continues to use the old software while the other nodes use the new software, a permanent split can occur.

A *soft fork* or a soft-forking change is described as a fork in the blockchain which can occur when old network nodes do not follow a rule followed by the newly upgraded nodes.

- This could cause old nodes to accept data that appear invalid to the new nodes, or become out of sync without the user noticing.

# Hard fork:

On a blockchain, a fork is a technical event that occurs because diverse participants need to agree on common rules (not orphan block)

At its most basic, a fork is what happens when a blockchain diverges into two potential paths forward — either with regard to a network's transaction history or a new rule in deciding what makes a transaction valid.

As a result, those who use the blockchain have to show support for one choice over the other.

So far, we know some forks resolve on their own, but others, fueled by deep rifts in a community, can cause a network to permanently split, creating two blockchain histories — and two separate currencies.

Along with that, there has also been confusion about the various types of forks, how they get activated and the risks they pose.

**A hard fork is NOT an orphan block**

# Hard fork / soft fork

**HARD FORK:** A hard fork is a software upgrade that introduces a new rule to the network that isn't compatible with the older software. You can think of a hard fork as an expansion of the rules. (A new rule that allows block size to be 2MB instead of 1MB would require a hard fork).

**What happens?** Nodes that continue running the old version of the software will see the new transactions as invalid. So, to switch over to the new chain and to continue to mine valid blocks, all of the nodes in the network need to upgrade to the new rules.

**What can go wrong?** The problem comes when some sort of political impasse arises, and a portion of the community decides to stick by the old rules no matter what. The hash rate, or network computing power, behind the old chain is irrelevant. What matters is that its data (and ruleset) is still perceived to have value, meaning miners still want to mine a chain and developers still want to support it.

# Hard fork / soft fork

**SOFT FORK:** A soft fork is any change that's backward compatible. Say, instead of 1MB blocks, a new rule might only allow 500K blocks.

**What happens?** Non-upgraded nodes will still see the new transactions as valid (500k is less than 1MB in this example). However, if non-upgraded nodes continue to mine blocks, the blocks they mine will be rejected by the upgraded nodes. This is why soft forks need a majority of hash power in the network.

**What can go wrong?** When a soft fork is supported by only a minority of hash power in the network, it could become the shortest chain and get orphaned by the network. Or, it can act like a hard fork, and one chain can splinter off.

**Soft forks have been the most commonly used option** to upgrade the bitcoin blockchain so far. Past examples:

- Bitcoin's reference implementation currently relies on OpenSSL for signature validation, which means it is implicitly defining Bitcoin's block validity rules.
- OpenSSL is not designed for consensus-critical behaviour, and thus changes to it can - and have - affected Bitcoin software.
- One specifically critical area is the encoding of signatures. Until recently, OpenSSL's releases would accept various deviations from the DER standard and accept signatures as valid.
- SoftForK: restrict to DER standard.

# Unintended hard fork: 2013

## **What went wrong: a change of software**

- A block that had a larger number of total transaction inputs than previously seen was mined and broadcasted. Bitcoin 0.8 nodes were able to handle this, but some pre-0.8 Bitcoin nodes rejected it.
- The pre-0.8-incompatible chain at that point had around 60% of the mining hash power ensuring the split did not automatically resolve (as would have occurred if the pre-0.8 chain outpaced the 0.8 chain in total work, forcing 0.8 nodes to reorganise to the pre-0.8 chain).
- In order to restore a canonical chain as soon as possible, miners BTCGuild and Slush downgraded their Bitcoin 0.8 nodes to 0.7 so their pools would also reject the larger block. This placed majority hashpower on the chain without the larger block, thus eventually causing the 0.8 nodes to reorganise to the pre-0.8 chain.
- During this time there was one large double spend; it was done by someone experimenting to see if it was possible (not a malicious double spending)

# Worst soft fork: August 2013

## What went right

- The split was detected very quickly.
- The right people were online and available in IRC or could be contacted directly.
- Marek Palatinus (Slush) and Michael Marsee (Eleuthria of BTCGuild) quickly downgraded their nodes to restore a pre-0.8 chain as canonical, despite the fact that this caused them to sacrifice significant amounts of money.
- Deposits to the major exchanges and payments via BitPay were also suspended (and then un-suspended) very quickly.
- Fortunately, the only attack on a merchant was done by someone who was not intending to actually steal money.

# Worst soft fork: August 2013

## Root cause

- Bitcoin versions prior to 0.8 configure an insufficient number of Berkeley DB locks to process large but otherwise valid blocks. Berkeley DB locks have to be manually configured by API users depending on anticipated load.
- Because max-sized blocks had been successfully processed on the testnet, it did not occur to anyone that there could be blocks that were smaller but require more locks than were available. Prior to 0.7 unmodified mining nodes self-imposed a maximum block size of 500,000 bytes, which further prevented this case from being triggered. 0.7 made the target size configurable and miners had been encouraged to increase this target in the week prior to the incident.
- Bitcoin 0.8 did not use Berkeley DB. It switched to LevelDB instead, which did not implement the same locking limits as BDB. Therefore it was able to process the forking block successfully.

# Hard fork: August 2013

## ACTIONS

### Immediately

- Release a version 0.8.1, forked directly from 0.8.0, that, for the next two months has the new rules:
- Release a patch for older versions that implements the same rules, but also increases the maximum number of locks to 537,000
- Create a web page on bitcoin.org that will urge users to upgrade to 0.8.1, but will tell them how to set DB\_CONFIG to 537,000 locks if they absolutely cannot.
- Over the next 2 months, send a series of alerts to users of older versions, pointing to the web page.

# A new hard fork soon?

A big discussion is going on BU vs SegWit

- SegWit is the proposal favoured by many bitcoin developers and enthusiasts,
- SegWit would double the transactions per second capacity of bitcoin by simplifying transactions, stripping out some details such as signatures
- One criticism of SegWit is that it appears to be only a temporary solution. It just doubles the network's bandwidth, while BU allows miners to vote to increase the capacity when they need with no upper limit.

Who will win? There could be a hard fork?

# The Transaction Throughput Problem

- The number of transactions on the Bitcoin blockchain has also increased, up to 400 K transactions per day; about 90 K trans. are backlogged
- Every time a user sends a Bitcoin transaction the transaction is added into the **memory pool** (mempool), which is essentially a pool of all unconfirmed transactions in the Bitcoin network
- From the mempool, miners select transactions that they want to verify. Once miners validate a transaction (i.e. confirm that the sender actually has enough bitcoins to send to the receiver), they add it to a new block, which is eventually published to the blockchain
- Other nodes then iterate through this newly published block's transactions to ensure the block is valid, before accepting the block as a part of its ledger.

# The Transaction Throughput Problem

Let's calculate the throughput of transactions:

- A simple transaction (1 input 2 outputs) has size approx. 226bytes
- A block's size is limited to 1MB (1000000 bytes)
- A block is published in the blockchain once every 10 minutes on average (600 seconds).
- hence no. of transactions every 10 minutes is at most  $1.000.000 / (600 * 226) = 7.37$  per second

In practice less between 4 and 7

# The Transaction Throughput Problem

- If there are more unconfirmed transactions in the mempool, how does a miner select which transactions to verify? **Transaction fees!**
- The sender of a transaction has the option of adding a custom transaction fee for the miner, incentivizing a miner to select the transaction and have it verified faster
- Miners will select the transactions that have the highest fee attached to them to maximize profits.
- Theoretically, you can send a transaction with no fee. But if there are transactions that have fees higher than yours in the pool, why would yours ever get picked?

# The Transaction Throughput Problem

The problem: if Bitcoin does change transactions fees will increase and people will leave Bitcoin

- As Bitcoin's user base grows, so does the average transaction fee. We observed that there are only 7 transact. that are processed every second and everyone wants to get their transaction verified first.
- At the moment, the average transaction fee is approximately nov. 2020 ~6 US\$ – (but in Dec. 2017 was more than 30US\$)
- This is too much for small transactions

# Solving the Trans. throughput Problem

**Solution 1: raise the block size from 1 MB thus allowing more transactions per block**

- This can be done if all miners agree therefore requiring consensus from the Bitcoin community.
- Millions of people use Bitcoin, gaining consensus is difficult and efforts should be made to avoid it.
- However Bitcoin's user base continues to grow, there will eventually be another backlog of unconfirmed transactions, so another block size increase will be needed, and subsequently another hard fork.

# Solving the Trans. throughput Problem

Solution 2: blocks can have any size. Problems

1. To reach consensus is more difficult
2. Handling a big block is impractical and will restrict mining even more introducing an element of centralization.
3. Recall that once a block is mined, all other nodes must validate the block before accepting it. If the block size is incredibly large and somebody were to publish an invalid block, nodes would waste a large amount of time attempting to validate the block before discarding it as invalid and moving onto the next block.
4. This allows a denial of service attack by repeatedly publishing insanely large invalid blocks to the network, stopping valid blocks from being processed for a long period of time.

# The Transaction Throughput Problem

- Satoshi was able to make numerous changes to the bitcoin network at the beginning; Changes are now more difficult (many users, huge economic value)
- Today no person or group can determine when and how bitcoin should be upgraded has similarly made the process of updating the system more complex.
- There has been many proposals with the main goal of allowing more transactions (i.e. large blocks)
- If there is no agreement among all miners and users there will be a **Fork in the blockchain**

# Hard forks and soft forks

- During a **hard fork**, software implementing bitcoin and its mining procedures is upgraded; once a user upgrades his or her software, that version rejects all transactions from older software, **creating a new branch of the blockchain** (unless all miners agree)
- Those users who retain the old software continue to process transactions, meaning that there is a parallel set of transactions taking place across two different chains.
- A **soft fork** is a change to the software protocol where only previously valid blocks/transactions are made invalid (more restrictions). Since old nodes will recognize the new blocks as valid, a soft fork is backward-compatible. This kind of fork requires only a majority of the miners upgrading to enforce the new rules

# Hard forks and soft forks

Example: some miners agree on 2MB block size

- Assume that a 2MB block is validated by an updated node and added on to the blockchain.
- What if the next block is validated by a node running an older version of the protocol? It will try to add its block to the blockchain. Since the latest block is not valid it will ignore that block and **attach its new validation to the previous one**.
- Therefore **you have two blockchains, one with both older and newer version blocks, and another with only older version blocks**. Which chain grows faster will depend on which nodes get the next blocks validated, and there could end up being additional splits. It is feasible that the two (or more) chains could grow in parallel indefinitely.

# Hard forks and soft forks

Problem: efficiency of handling a block

A big discussion is going. Some forks

- Bitcoin XT (2014), blocks of 8MB (joined with Bitcoin Classic)
- Bitcoin Classic (2016), blocks of 2MB (now dead)
- Bitcoin Unlimited (2014), blocks of variable size (up to 16MB) decided by users (now 1 BTC Unlimited ~ \$ 53)

Note: when there was a fork people received 1 BTC and 1 unit in the new currency ofr each BTC they owned

**Conclusion:** The majority of hash power did not begin to use these clients.

# Hard forks and soft forks

A *hard fork* is a rule change such that the software validating according to the old rules will see the blocks produced according to the new rules as invalid.

- In case of a hard fork, all nodes meant to work in accordance with the new rules need to upgrade their software.
- If one group of nodes continues to use the old software while the other nodes use the new software, a permanent split can occur
- Hard forks splitting [bitcoin](#) (aka "split coins") are created via changes of the blockchain rules and sharing a transaction history with bitcoin up to a certain time and date
- When the hard fork happened each user of bitcoin (BTC) following got 1 unit of the split coin

# Hard forks and split coins

Main split coins

- **Bitcoin Cash** First hard fork splitting bitcoin (Aug. 2017). Forked at block 478558; motivation reduce size of transaction, value (nov. 2020) \$ 258 ,but jan 2019 \$2000
- **Bitcoin SV**: Forked at block 556766, 15 November 2018; motivation new mining algorithm that tries to limit the use of big data center value \$160(nov.2020) , max value \$438
- **Bitcoin Gold**: Forked at block 491407, 24 October 2017, value \$ \$7.50(nov.2020) , max value \$456
- **Bitcoin Private**: forked in 2018): allows zero knowledge evidence of ownership without revealing which coins are owned (1BTC Private about \$0.15 total capital. \$ 735, practically dead)

# Size of the block: finished?

**SegWit 2x (2017)**: increase size blocks in a compatible way

- It hides its increased block size by changing the definition of a block to be measured as one million "units" instead of bytes. The signature data is separated from and moved to the end (details omitted)
- The overall effect is to change the block size from 1MB to 1.8MB (on average) ***without changing the Bitcoin protocol!***

November 2017: proposers renounced

# Other digital coins

# Proof of Work (PoW)

- In a PoW system, network participants have to solve so-called “cryptographic puzzles” to be allowed to add new “blocks” to the blockchain. This puzzle-solving process is commonly referred to as “mining”.
- Because the input of each puzzle becomes larger over time (resulting in a more complex calculation), the PoW mechanism requires a vast amount of computing resources, which consume a significant amount of electricity. If a network participant (i.e. a node) solves a cryptographic puzzle, it proves that he has completed the work, and is rewarded with digital form of value (or in the case of a cryptocurrency, with a newly mined coin).
- This reward serves as an incentive to uphold the network. The cryptocurrency Bitcoin is based on a PoW consensus mechanism. Other examples include Litecoin, Bitcoin Cash, Monero, etc.

# Proof of Stake (PoS)

- In a PoS system, a transaction validator (i.e. a network node) must prove ownership of a certain asset (or in the case of cryptocurrencies, a certain amount of coins) in order to participate in the validation of transactions.
- This act of validating transactions is called “forging” instead of “mining”. For example, in the case of cryptocurrencies, a transaction validator will have to prove his “stake” (i.e. his share) of all coins in existence to be allowed to validate a transaction.
- Depending on how many coins he holds, he will have a higher chance of being the one to validate the next block (i.e. this all has to do with the fact that he has greater seniority within the network earning him a more trusted position).
- The transaction validator is paid a transaction fee for his validation services by the transacting parties. Cryptocurrencies such as Neo and Ada (Cardano) utilize a PoS consensus mechanism.

# permissionless vs permissioned blockchain

- On an *open, permissionless blockchain*, a person can join or leave the network at will, without having to be (pre-)approved by any (central) entity.
- There is no central owner of the network and software, and identical copies of the ledger are distributed to all the nodes in the network.
- The vast majority of cryptocurrencies currently in circulation is based on permissionless blockchains (e.g. Bitcoin, Bitcoin Cash, Litecoin, ...).
- On a *permissioned blockchain*, transaction validators (i.e. nodes) have to be pre-selected by a network administrator (who sets the rules for the ledger) to be able to join the network.
- This allows, amongst others, to easily verify the identity of the network participants.
- However, at the same time it also requires network participants to put trust in a central coordinating entity to select reliable network nodes.

# permissioned blockchains

- In general, permissioned blockchains can be further divided into two subcategories.
- On the one hand, there are open or public permissioned blockchains, which can be accessed and viewed by anyone, but where only authorised network participants can generate transactions and/or update the state of the ledger.
- On the other hand, there are closed or “enterprise” permissioned blockchains, where access is restricted and where only the network administrator can generate transactions and update the state of the ledger.
- What is important to note is that just like on an open permissionless blockchain, transactions on an open permissioned blockchain can be validated and executed without the intermediation of a trusted third-party. Some cryptocurrencies, like Ripple and NEO utilise public permissioned blockchains.

# Ethereum (2015)

Ethereum is a decentralized platform that runs “smart contracts”.

- Smart contracts are “self-executing” contracts or applications that run exactly as programmed without any possibility of downtime (i.e. the blockchain is always running), censorship, fraud or third-party interference.
- Ethereum has a capability that goes far beyond that of a pure P2P digital cash equivalent like Bitcoin. In simple terms, it is much like a smartphone operating system on top of which software applications can be built.
- Technically speaking, the Ethereum platform itself is not a cryptocurrency and is permissionless blockchains. However Ethereum requires a form of on-chain value to incentivise transaction validation within the network (i.e. a form of payment for the network nodes that execute the operations).
- This is where Ethereum’s native cryptocurrency “ether” (ETH) comes into play. Ether does not only allow smart contracts to be built on the Ethereum platform (i.e. it fuels them), but it also functions as a medium of exchange.

# RIPPLE (public permissioned blockchain)

- Ripple is an open-source, P2P decentralized digital payment platform that allows for fast transfers of currency (e.g. US Dollar, Yen, Bitcoin, ...) launched in 2012 by the company Ripple (Labs) responsible for the further development of the Ripple protocol
- First company to have received a “BitLicense” for an institutional use case of digital assets from New York’s Department of Financial Services. It is also getting support from a number of big players in the financial services industry, such as Bank of America Merill Lynch, Santander, etc.
- Ripple’s uses the cryptocurrency XRP. XRP was built to become a bridge currency to allow financial institutions to settle cross-border payments a lot faster and cheaper than they can using the global payment networks that are in place today, which can be slow and involve multiple middlemen (i.e. banks).
- According to Ripple, XRP can handle more than 1,500 transactions per second. While it was initially developed and intended for enterprise use, it has been adopted by a large number of cryptocurrency users.
- Ripple (XRP) is not based on a PoW or a PoS mechanism to validate transactions, but it makes use of its own specific consensus protocol.
- The total supply of XRP has been fully “pre-mined” (or better: created upon the coin’s inception) by its inventors. At present, 8 B XRP is held by Ripple (Labs), (39 B XRP has been distributed; and 52 B XRP has been placed in escrow to create certainty of XRP supply at any given time.

# Stellar (public permissionless blockchain)

- Similar to Ripple: Stellar is an open-source, distributed payments infrastructure, created in 2014 by one of Ripple's founding fathers; Stellar's development is supported by the non-profit organization Stellar.org
- Its goal is to connect people to low-cost financial services to fight poverty and develop individual potential. Stellar can also be used to build smart contracts.
- It is not based on a PoW or PoS consensus mechanism, but has its own specific consensus protocol.
- Stellar is home to the cryptocurrency Lumen (XLM) that are used to pay for transactions on the Stellar network; they contribute to the ability to move money around the world and to conduct transactions between different currencies quickly and securely.
- Similar to Ripple's cryptocurrency XRP, the total supply of Stellar Lumens is "pre-mined". It is held by Stellar.org who has been given the task to distribute Lumens for free, in the following manner<sup>213</sup>: 50% is to be given away to individuals (via a direct sign-up program); 25% is to be given away to partners (via a specific partnership program); 20% is given away to Bitcoin and XRP holders; and 5% is reserved for Stellar.org's operational expenses.

# Bitcoin Cash (BCH): open, permissionless blockchain

- Bitcoin Cash was created on the 1st of August 2017 and is based on Bitcoin's original SHA-256 PoW algorithm, yet with some changes to its code.
- Bitcoin Cash is what is known in the crypto-community as a “hard fork” of the Bitcoin blockchain. It is the result of two very different visions on the future of Bitcoin and the Bitcoin blockchain, whereby the Bitcoin blockchain diverged into two potential paths forward.
- In short, some Bitcoin developers wanted to raise the block size limit from 1MB to 8MB, to reduce transaction fees and improve confirmation times, whilst others had different plans. Because the community could not reach a consensus, the new cryptocurrency Bitcoin Cash was created. Bitcoin Cash makes use of the PoW mechanism, which means that it can be mined.
- A direct result of the hard fork, is that anyone who held Bitcoin at the time Bitcoin Cash was created also became owner of the same amount of Bitcoin Cash. Any Bitcoin acquired after that specific time follows the original path and does not include Bitcoin Cash. In principle, a “hard fork” does not change the nature of a coin's blockchain. In other words, Bitcoin Cash also runs on an open permissionless blockchain, just like Bitcoin.
- Bitcoin cash is a pseudo-anonymous coin

# Litecoin: open, permissionless blockchain

- Like Bitcoin, Litecoin (LTC) is an open-source decentralized P2P cryptocurrency launched in 2011. Scrypt PoW algorithm, which utilises Bitcoin's and is based on what is known as the original SHA-256 PoW algorithm.
- Litecoin is often described as the 'silver' to Bitcoin's gold.
- Apart from the fact that it uses a different algorithm, it is different from Bitcoin in two ways.
- Firstly, and this results from the use of the Scrypt PoW algorithm, Litecoin offers a much faster transaction speed than Bitcoin. The time needed to generate a block on the Bitcoin BC is about ten minutes, while the average block creation time on the Litecoin blockchain is approximately 2.5 minutes.
- Secondly, the total supply limit of Litecoin is with 84 million coins, much higher than the 21 million supply limit of Bitcoin.
- like Bitcoin, Litecoin is a *pseudo-anonymous coin*. Everyone can verify the chain of LTC transactions on the basis of the public ledger

# IOTA: permissionless distributed ledger

- IOTA (2016), is an open-source eco-system where people and machines can transfer value (i.e. money) and/or data **without any transaction fees** in a trustless, permissionless, and decentralized environment; focus on microtransastions; total supply of IOTA is 2779 B
- IOTA is not based on blockchain technology, but constitutes a different application of distributed ledger technology; they claim is more scalable than the technology behind most other coins, and promises faster transaction speeds.
- IOTA's distributed ledger does not consist of transactions grouped into (transaction) “blocks” and stored into sequential chains (i.e. it is not a “blockchain”), but of a stream of individual transactions entangled together.
- Instead of requiring miners to perform computational PoW and validate transaction blocks in exchange for newly “mined” coins, IOTA’s network participants create a consensus themselves by validating two previous transactions each time they wish to make a new transactio; this produces a directed acyclic graph (DAG).
- The majority of IOTA it was sold by IOTA’s inventors in a crowdsale to pay for development costs and fund the IOTA Foundation.

# Other coins and split coins

Conclusion: split coins are dying or at least suffering

This also applies to other digital coins

<b>Values</b>	<b>(US \$) Nov 2020</b>	<b>MAX</b>	<b>Total capital</b>
IOTA	0.25	5	0.7 Billion
Litecoin	59	360	3.95 Billion
Stellar	0.08	0.87	8.45 Billion

There are more than 800 dead digital coins

# ALGORAND

# The blockchain challenge

There are two different needs in a blockchain.

1. The first is making the blockchain tamper-proof. This has been solved via using the one-way hash function: the hash of the latest block is included as part of the next block. All blockchains share this approach, so they are all equal in this respect.
2. The second need is generating new blocks: how you select a new block to be appended to the chain. This is the real challenge, and different blockchains have different approaches to it. A new block should contain a set of valid transactions that do not appear in the blockchain so far. The problem is that at any point in time two users may have seen the same sequence of blocks but may see different new transactions.

To solve 2 above an agreement should be reached

# The blockchain challenge

how you select a new block to be appended to the chain.

The problem is that at any point in time two users may have seen the same sequence of blocks but may see different new transactions. This is the case because, in a distributed ledger, each transaction is not instantaneously propagated through the network.

Typically, it is transmitted from each user to a few other users, who then transmit it to still other users, until the transaction reaches all users.

Accordingly, at every point in time, the sets of new valid transactions seen by different users may be different even when they have a significant overlap.

In sum, a user U may think the new block ought to be BlockU and a user V thinks it ought to be BlockV. We need to decide which block should be appended to the chain

Bitcoin solves this with the proof of work

# The blockchain trilemma.

The blockchain trilemma.

Existing blockchains can offer at most two of the following three properties:

- Security
- Scalability
- Decentralization

In fact

- Bitcoin suffer of scalability – proof of work is becoming more and more difficult
- Permissioned blockchain are not decentralized (and we need to trust to believe it is secure)

# The blockchain trilemma.

Bitcoin has two problems

1. Proof-of-Work Does Not Scale. Proof-of-work is very slow.

Bitcoin's crypto puzzles are so hard in order to guarantee that one solution is found only every 10 minutes, no matter how many miners try to solve the crypto puzzle. 10 minutes is not compatible with an increased use of BTC (recall now about 7 transactions per second).

2. Proof-of-Work Results in De-Facto Centralization.

Proof-of-work has caused a tremendous concentration of power. This centralization is a consequence of the fact that Proof-of-Work is both expensive and wasteful. Mining today utilizes racks and racks of specialized hardware and consumes an enormous amount of electricity.

# Key Ideas: proof of stake

- Weighted users
  - Users are weighted by the money in their account
- Consensus algorithm should be a Byzantine Consensus Algorithm (BA)  
Instead of having all the nodes run BA, let a subset of nodes represent the whole group and run BA\*.
  - Called a committee
  - Rotating committee per block

# Weighted Users vs Proof of Stake (PoS)

- PoS flaw in a nutshell
  - Malicious leader (who assembles new block) can create a fork in the network
  - Can be caught (e.g., since two versions of the new block are signed with his key) -> the leader loses his money
- Algorand
  - Weights ensure that the attacker cannot amplify his power by using pseudonyms
    - As long as the attacker controls less than 1/3 of the monetary value of the system
  - Guarantee that the probability for forks is negligible

# Algorand Committee

- BA is not scalable
- BA\* uses consensus by committee
  - Randomly selects a small set of representatives from the total set of users
  - Committee members will publicly broadcast messages slowing others to learn agreed-upon block
- Concerns
  - How to randomly choose committee members?
  - How to ensure adversary cannot fake being a committee member?
  - How to ensure committee members are not targeted?

# Implementing Pure PoS

In Algorand, a new block is constructed in two phases.

1. In the first phase, a single token is randomly selected, and its owner is the user who proposes the next block.
2. In the second phase, 1000 (not a fix constant) tokens are selected among all tokens currently in the system. The owners of these 1000 tokens are selected to be part of a phase-2 ‘committee,’ which approves the block proposed by the first user.

**NOTE:**

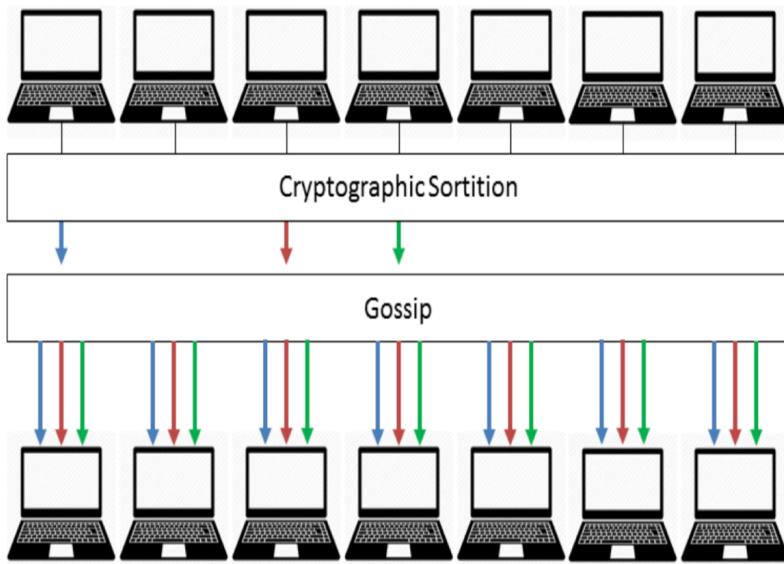
The second phase is necessary because in the first phase a dishonest token can be chosen

In the second phase if the majority of users are honest then the second phase will guarantee an agreement among honest users. (A society exists insofar as the majority of its members follow prescribed rules).

# Algorand Committee

- Problem
  - How to randomly choose committee members
  - Ensure adversary cannot fake being a committee member
- Solution: cryptographic solutions
  - Users can independently and privately determine if they are chosen
  - Sortition will choose users randomly based on their weights
    - Randomness comes from publicly known seed
  - Verifiable Random Functions (VRF)

# Algorand Committee



**Figure 2:** An overview of one step of  $BA\star$ . To simplify the figure, each user is shown twice: once at the top of the diagram and once at the bottom. Each arrow color indicates a message from a particular user.

- In the second phase the committee uses gossip to allow all users to see the exchanged messages
- **IMPORTANT:** each member of the committee sends exactly one message

# Communication

- Communicate via gossip
  - Each user collects a block of transactions they hear about
- Algorand will initiate a round starting w/block proposal
  - Create committee using Sortition
  - All committee members will propose their block
- Users will wait for a time period to receive blocks
  - Only keep highest priority block
- All users who received some block will initiate BA\* to reach majority consensus and commit a block

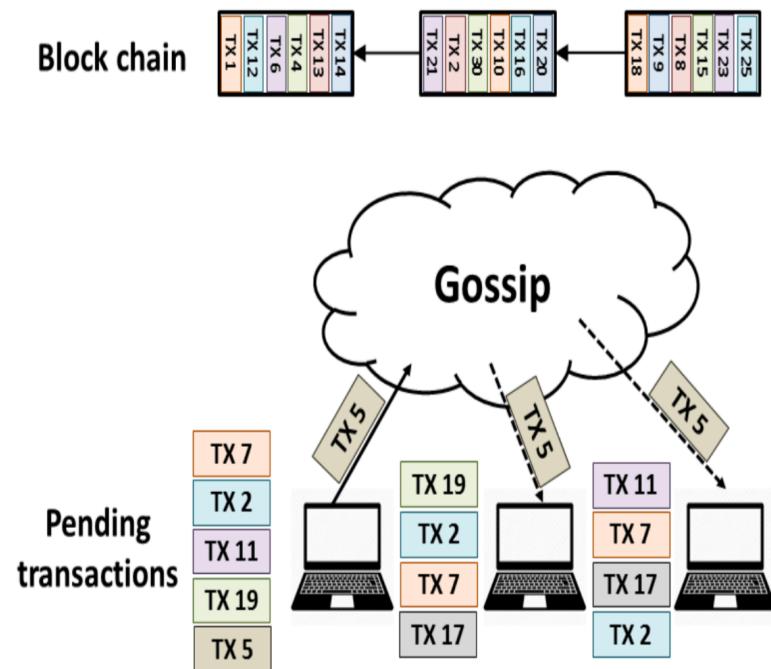


Figure 1: An overview of transaction flow in Algorand.

# How to randomly choose committee members

Problem: Ensure adversary cannot fake being a committee member

Solution: cryptographic lottery

- The lottery is done by each user privately for each token that is owned
- How to be sure nobody cheats:
  - if a token owned by you wins the lottery you send the proof that you have won (say your ticket)
  - Cryptographic lottery guarantees that nobody can cheat (too much computational effort to have a larger probability to win the lottery)

Example:

To select 1,000 random tokens among, say, 10 Billion tokens, each token is selected with probability  $1,000/10\text{ Billion}$  i.e., with probability 1 in 10 million.

If the percentage of bad users is 10% then the probability that the majority of committee is bad people is very low.

# Scalability

## Scalability

- Once selected, each member propagates to the network a single, immediately computed message. So no matter how many users are in the system, the maximum number of messages that need to be propagated is small

Example if the committee is 1000 users then there are 1000 short messages. scalable.

# Security

Assume there is a very powerful adversary, capable of corrupting users extremely quickly.

- Clearly, the adversary will try to corrupt the members of the committee,
- BUT: he/she does not know who they are.
- In fact committee members are selected by a secretly run, cryptographically fair, individual lottery. Thus, only they know who they are, up to the moment in which they propagate through the network both their winning tickets and their opinions about the block.
- Corruption is not possible: Whatever the committee members had to say, they have already said, and their winning tickets and up-or-down opinions about the block are virally propagating throughout the network.

# Conclusion

## Strength

- Scalable due to the idea of random committee
- Experimentation is great

## Weakness

- Gossip...?
- Limited to cryptocurrency
- Tradeoffs between public blockchain and private/consortium blockchain

# Zerocoins and Zerocash

# Zerocoins: protocol-level mixing

Mixing capability baked into protocol

Advantage: cryptographic guarantee of mixing

Disadvantage: not currently compatible with Bitcoin

*Zerocoins: Anonymous Distributed E-Cash from Bitcoin*

I. Miers et al.  
IEEE S&P 2013

# **Basecoin and Zerocoins**

Basecoin: Bitcoin-like Altcoin

Zerocoins: Extension of Basecoins – it is announced it will be implemented in Anoncoin (other digital coin)

Basecoins can be converted into zerocoins and back

Breaks link between original and new basecoin

# Zerocoins

A Zerocoins 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)

# Zerocoins

Work with fixed amount of 1\$.  
Imagine a huge shared board.  
Unlike Zerocoins, this is a  
centralized version (for clarity's  
sake)

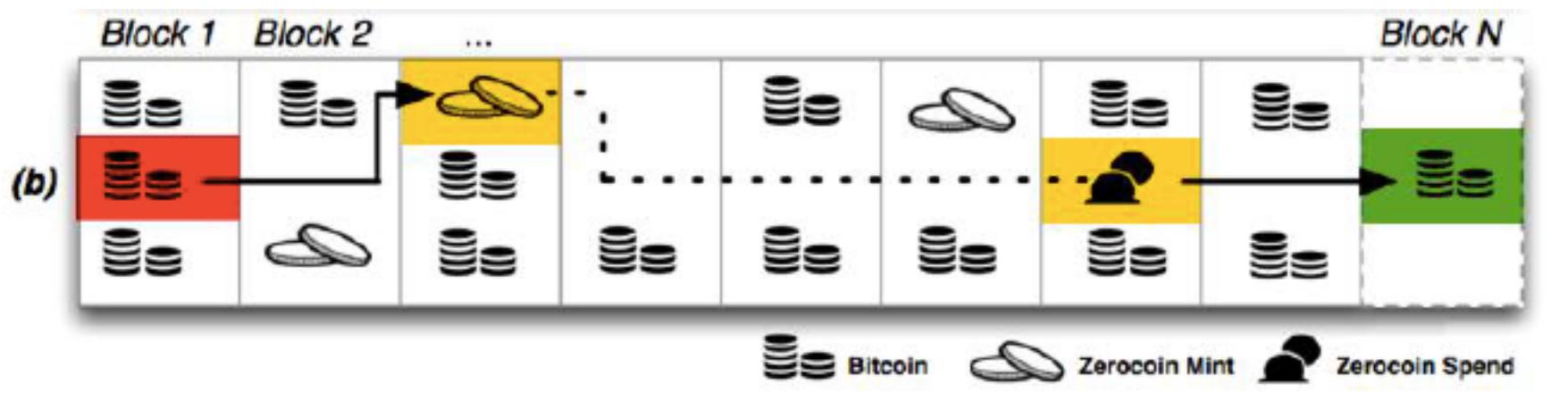


- Suppose you have 1\$ on you. You want to anonymize it, in Bitcoin terms, meaning you want to change the owner, without the transaction being public. You have access to a secure board (with a trusted guardian).
- Pin 1\$ on the corkboard. Receive an anonymous ticket “Certificate : 1\$ pinned” from the guardian
- Give a friend your certificate.
- Send him get the 1\$ back. The guardian will unpin a 1\$ bill from the corkboard and give it to your friend

# Zerocoins

Different from Bitcoin, not an evolution

- Uses Bitcoin as underlying money (Both are needed)
- Anonymity (without Trusted Party)
- Bitcoins X -> Mint -> Zerocoin -> Spend -> Bitcoins Y
- Public Key Y can't be linked to Public Key X (computationally infeasible)



- “Anonymized vouchers of Bitcoins”, “Currency Swap”

## Two challenges

Crucial observation: to mint a bitcoin transforming it in zerocoins you must have one BTC

How to construct these proofs?

1. How to make sure each proof can only be “spent” once?
2. How to be sure that these proofs do not reveal the owner of the BTC (In our example X) ?

# Commitment

Generate a message  $S$   
and a random secret  $r$

1. I commit on message  
By publishing  $H(S, r)$  ( $H$  is Hash)
2. *Then I can show my commitment  
on message  $S$  by revealing  $r$*



1. Commit: Equivalent to write a message  $S$  and putting in an envelope and close the envelope
2. Show my commitment: opening the envelope

# Zero-knowledge proofs

A way to prove a statement without revealing any other information



1. I know an input  $r$  such that  $\text{Hash}(S,r) = \text{da39a3ee5e}$  (without revealing  $r$ )

*Not sufficient*

2. I know an input  $r$  that hashes to some hash in the following set:  
 $\text{Hash}(S_1,r)$  or  $\text{Hash}(S_2,r)$  or ... $\text{Hash}(S_n,r)$

Question: why 1. above is not enough? (unlinkability!)

# Minting zerocoins

Zerocoins come in standard denominations  
(Let's assume 1 basecoin for all transactions)

Anyone can make one!

They have value once put on the block chain  
That costs 1 basecoin

# Minting a zerocoins: “commitment”

Generate serial number  $S$   
(eventually made public)

and random secret  $r$   
(never public, ensures  
unlinkability)

Compute  $H(S, r)$

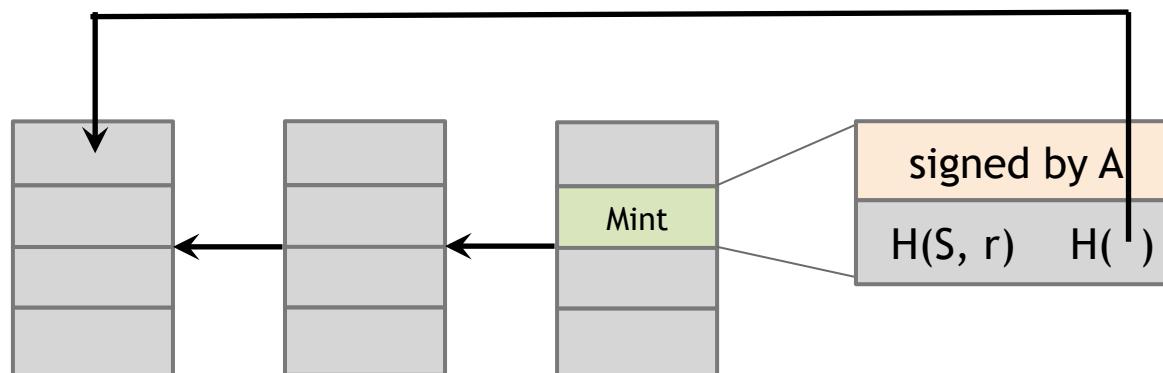


Simplification of the real protocol

# Minting a zerocoins

To put  $H(S, r)$  on block chain

Create Mint Tx with 1 bitcoin as input



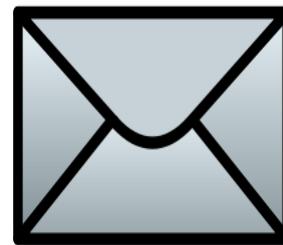
## To spend a zerocoins S:

- Reveal  $S$   
(miners will verify  $S$  hasn't been spent before)
- Create zero-knowledge proof that:  
*"I know a number  $r$  such that  $H(S, r)$  is one of the zerocoins in the block chain"*
- Pick arbitrary zerocoins in block chain & use as input to your new transaction (remember we assume all zero coins have the same value)

# Zerocoins are anonymous

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

$H(S, r)$

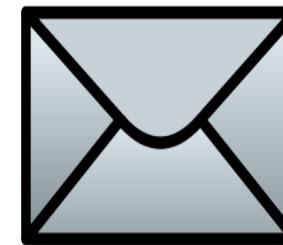


$h_1$



$h_2$

...



$h_N$

# Zerocoins is “efficient”

The proof is a giant  
disjunction over all  
zerocoins

Yet the proof is  
relatively small!

*I know  $r$  such that*

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

*OR*

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

*OR*

...

*OR*

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

# Zerocoins : conclusions

- Zerocoins supports regular transactions for when you don't need unlinkability (+)
- Zerocoins includes computationally expensive transactions that are used only for mixing (-)
- Only a fixed amount (1 BTC) can be minted; we need divisible coins
- These problems might be solved in Zerocoins 2.0

# Zerocash: Zerocoins without Basecoins

Two differences

- Different crypto for proofs  
(Much More efficient)
- More general: Proposal to  
run system without Basecoin

*Zerocash: Decentralized  
Anonymous Payments  
from Bitcoin*

E. Ben-Sasson et al.  
Usenix Security 2014

## Zerocash: untraceable e-cash

All transactions are zerocoins

Splitting and merging supported

Put transaction value inside the envelope

Ledger merely records existence of transactions

# Zerocash: the problem

Random, secret inputs are required to generate public parameters

These secret inputs must then be securely destroyed

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

# 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, tricky setup