Zerocoins

Anonymous Distributed E-Cash from Bitcoin

Presentation by Ludovic Barman

Slides available at www.lbarman.ch/slides.pdf

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

LASEC

Introduction

MAIN MENU ▾ MY STORIES: 24 ▾ FORUMS SUBSCRIBE JOBS ARSCOIN STORE

LAW & DISORDER / CIVILIZATION & DISCONTENTS

Anonymous hackers uncover alleged proof of MtGox fraud from site's CEO

"It's time MTGOX got the bitcoin communities wrath instead of Bitcoin getting Goxed."

by **Nathan Mattise** - Mar 9 2014, 11:15pm WEST

ACTIVISM HACKING THE WEB 138

Following the MtGox Bitcoin exchange losing millions to a hack and filing for bankruptcy, anonymous attackers took over the personal blog and reddit account of MtGox CEO Mark Karpeles on Sunday. After seizing control, the hackers posted (Pastebin) a message to the two spaces detailing their findings and the reasoning behind the attack.

"It's time that MTGOX got the bitcoin communities [sic] wrath instead of Bitcoin Community getting Goxed," the message reads. "This release would have been sooner, but in spirit of responsible disclosure and making sure all of ducks were in a row, it took a few days longer than would have liked to

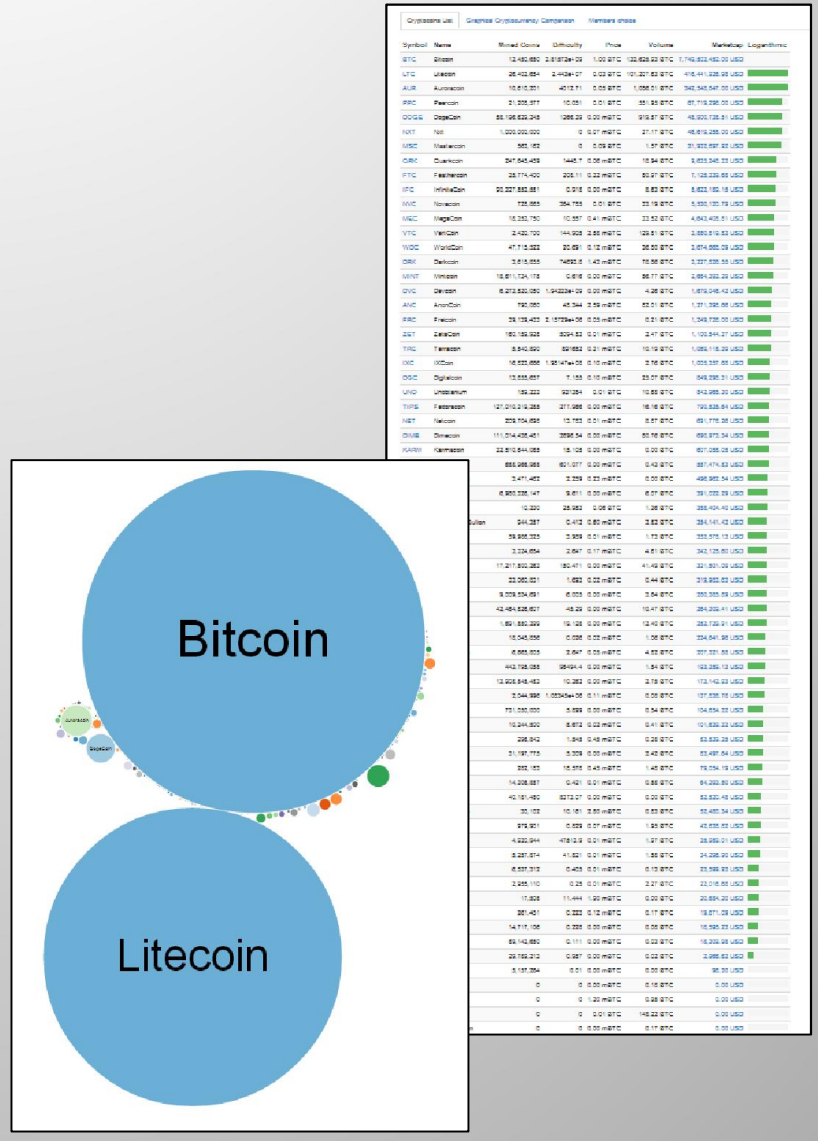
FURTHER READING

LATEST

Table of Contents

Introduction on e-cash

- Bitcoin isn't at all the first proposal, numerous attempts since 1990
- It is distinct from previous solution because of the decentralization, and the removal of a trusted party
- Currently +230 known e-moneys
- Nowadays, there's loads of "variations" of Bitcoins

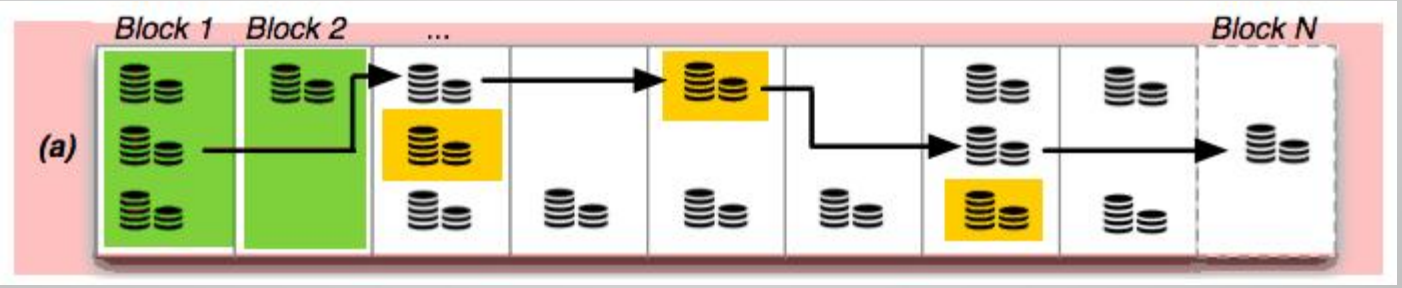


Bitcoin ฿

- P2P e-Cash System, invented by “Satoshi Nakamoto” in 2009
- Decentralized, without trusted party
- Low fees, No cost to accept them, Worldwide
- You create your own money (investing CPU power)
- Hard cap on total number of Bitcoins
  - More than 50% already created. Difficulty is increasing
- Transactions are irrevocable (safe for payee)
  - no litige management => lower processing cost
- Robust against denial of service
  - Suppose >50% of the network power is made of honest nodes

Block Chain, Accounts, Ledger

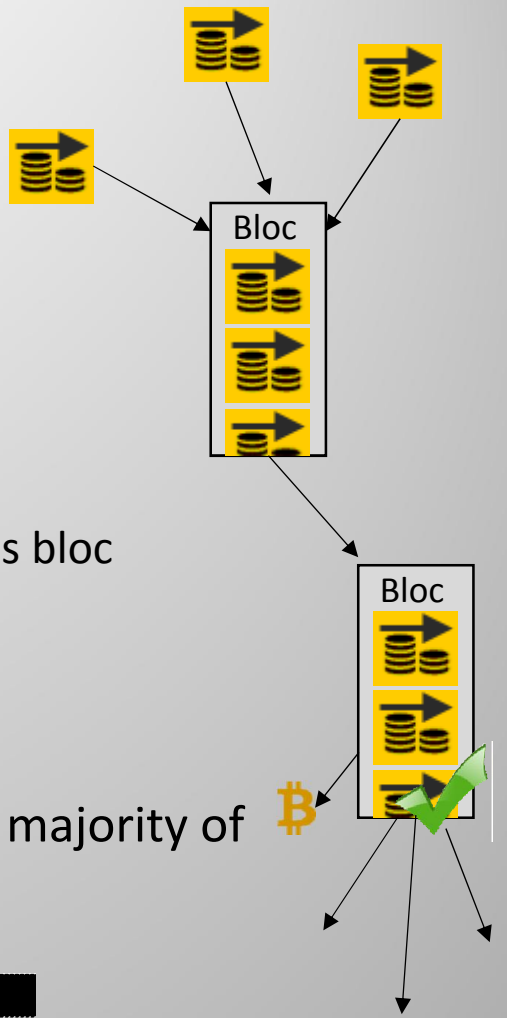
- Block Chain is a public, distributed, append-only ledger *(linked list)*
  - The *Chain* contains *Blocks* itself containing *Transactions*
  - Blocks are linked together with hash of previous block
  - Transaction : Inputs, Outputs. Script-based, allows complex transactions

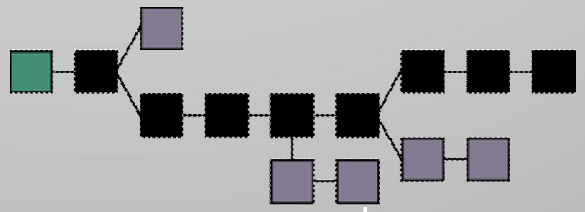


- There's no such thing as an account
  - You own a series of transactions

Verifying and Mining Bitcoins

- Transactions are broadcasted to everyone
- Two processes are combined in one
  - Mining Bitcoins (creating money)
  - Verifying transactions (supporting the system)
- The process of mining :
  1. Collecting “floating” transactions
  2. Putting them in a block, which is linked to the previous bloc
  3. Validate the block (add Proof-of-work)
  4. Send it to other nodes
  5. Get rewarded if other nodes accept it
- If/When a fork happens, decision of “truth” is taken by majority of computational power of the network





Proof-of-Work

- Function that requires loads of CPU power to compute, easy to check (DOS, Spam prevention)
- *Hashcash*, used in Bitcoin :

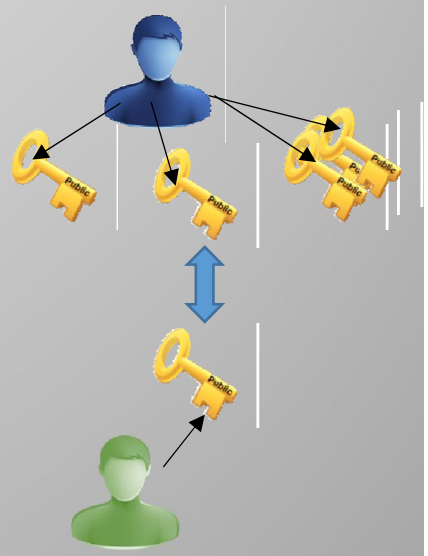
M = [Header(Sender, Recipient, Date, Merkle Head) || Counter],

valid if Hash(M) starts with k zero's

  - Checking : computing a hash = efficient
  - Mining : incrementing counter until valid. $O(2^k)$
- Publicly auditable, trapdoor-free, dynamic throttle

Anonymity in Bitcoin network

- What is public
  - Bloc chain => All transactions => All assets
  - Identified by Public Keys
  - If you know his public key(s), you can check how much your neighbor has, and what he does with his money
- Bitcoin solution :
  - Everyone can have as many public keys as wanted
  - If I want to receive a payment without revealing my assets, I create a new public key, and receive money on it.
  - => You only get a partial view on someone's assets
  - Analogy : One key is one (public) "account"
- "Linkability"
  - If you can link public keys together, you can reconstruct all transactions and assets regarding someone => Bad for privacy



Anonymity in Bitcoin network

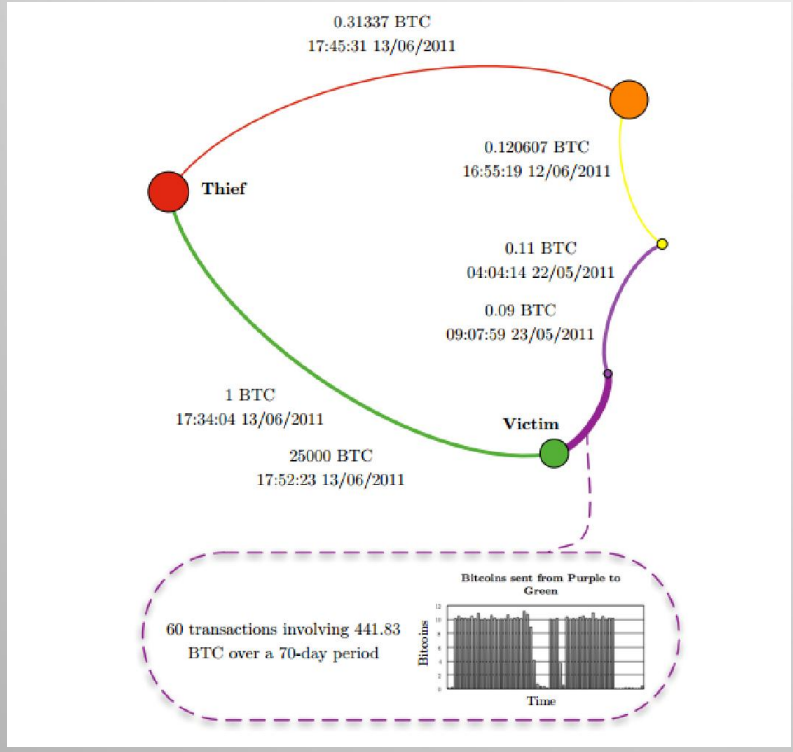
- Solution review
  - Conceptually, everything is public
  - Some tricks improve the amount of work required to trace users => “pseudonymity”
  - Not sufficient for privacy
    - Simple attack : Look for merges (likely with common users)
    - More sophisticated attack based on graph theory

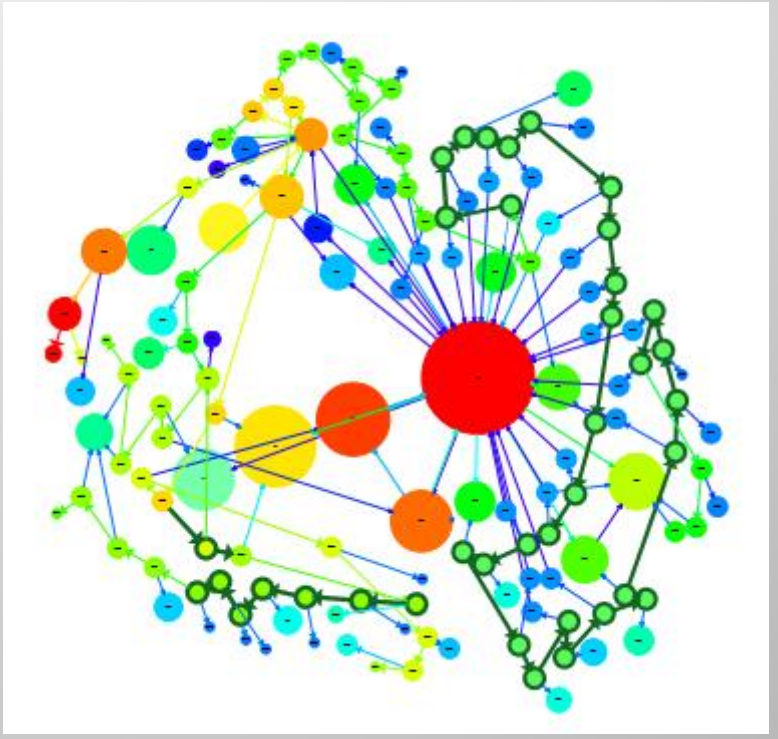
Graph-attacks : De-anonymization

- Based on [De-anonymizing Social Networks - A. Narayanan, V. Shmatikov]
- Reconstruction of the graph of Bitcoins transactions and pseudonymous identities
- Loads of information waited to be analyzed :
  - Search engines indexing public keys on Internet
  - Bitcoin Faucet (website) gives away PK, IP pairs => physical location
  - TCP/IP packets provenance (first emitter is source)
- More advanced strategies are likely, with cooperating nodes, “marked” Bitcoins [TODO: HOW TO MARK]
- Case study : theft 25K

Graph-attacks : De-anonymization

- Case study : (Amateur) tracing theft of 25K btc





- PK of final account found

Anonymity in Bitcoin network

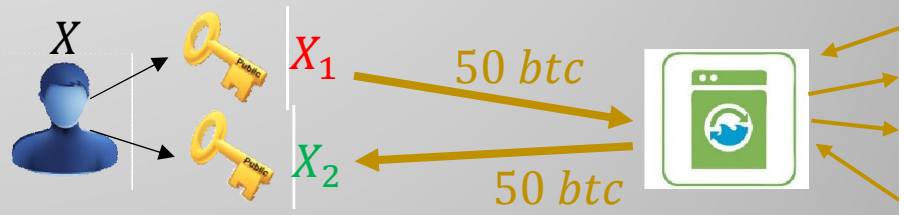
- Solution review : Bad 
  - Conceptually, everything is public
  - Some tricks improve the amount of work required to trace users => “pseudonymity”
  - Not sufficient for privacy
    - Simple attack : Look for merges (likely with common users)
    - More sophisticated attack based on graph theory :
      - *even theft aren’t able to stay anonymous*

- Reduce Linkability : Three solutions attempts

  Mixers  , Fair-Exchange protocol, Zerocoins 

Mixers and Bitcoin Laundries

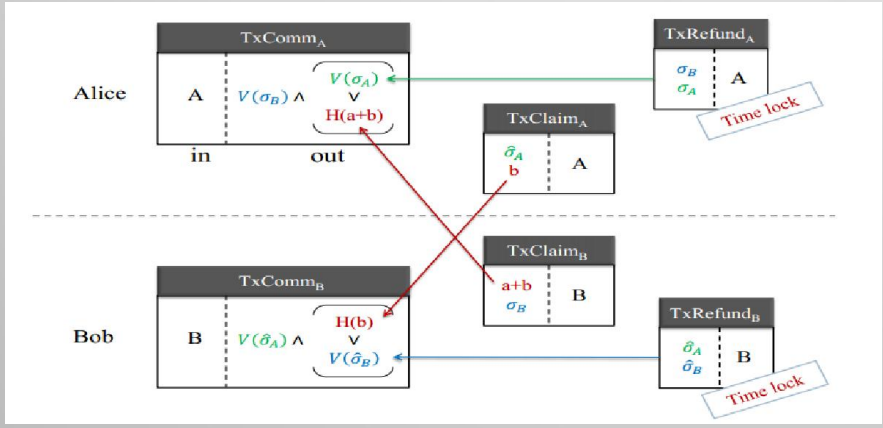
- Actors of the network (like you and me)
- Collect money from various user X, Y, Z, and merge the transaction into an account.
- Redistribute Bitcoins (hopefully) to X, Y, Z.
- You send them money with PK X_1, they send it back to PK X_2. There is no direct link (on the block chain) between X_1, X_2.

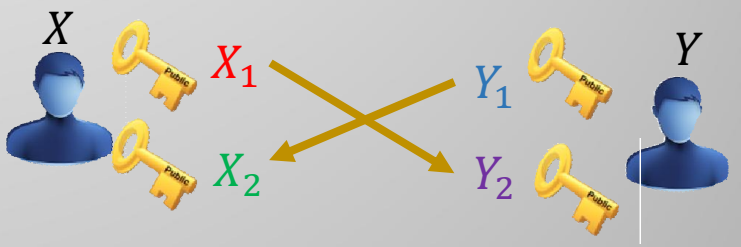


- Solution review : Terrible
  - What if the mixer is the NSA ? You introduce a Trusted Party. If malicious or compromised, worse than before.
  - You hope that you will get the coin back (cases of mixer going dark)

(Multi-User) Fair-Exchange Protocols

- Improvement of Laundries; I.E. guarantees that you will get the money back.
- Alice, Bob want to exchange A, B. Fair-Exchange ensures that afterwards, if Alice has received B then Bob has received A, and reciprocally.



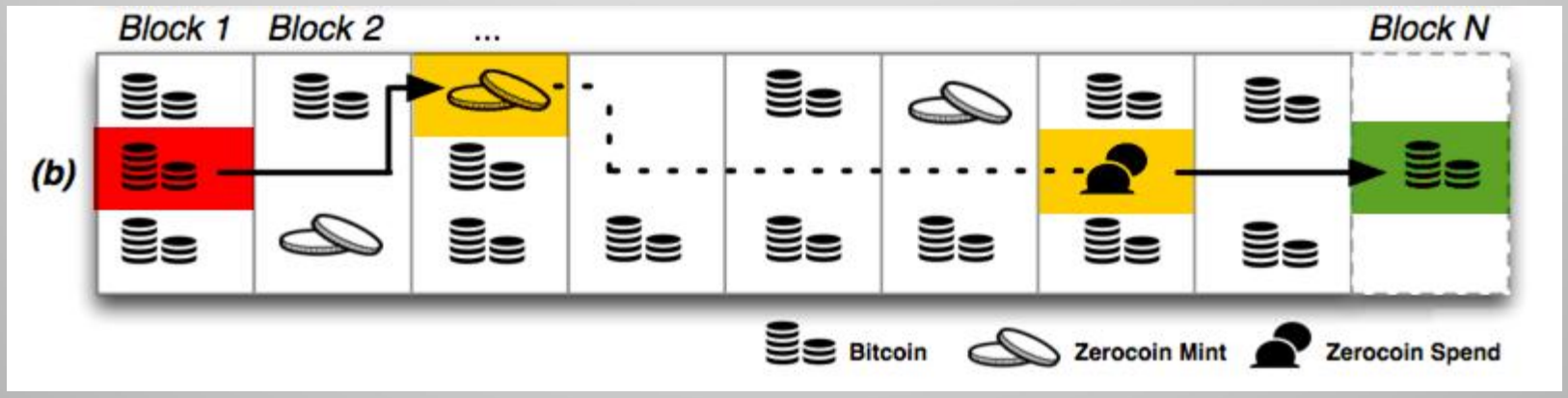


- Alice and Bob don't redeem on their account, but with another PK. X_1 and X_2 aren't linked, nor is Y_1 and Y_2. You have undesirable links, but like mixers.

Zerocoins – a new currency

- Different from Bitcoin, not an evolution. Still conceptual.
- 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"

Zerocoins – a new currency

Just a bit of formalism :

e-cash scheme $\Pi = (Setup, Mint, Spend, Verify)$

$Setup(1^\lambda) \rightarrow (params)$, with *Coin Universe* $\mathbb{C}$
$Mint(params) \rightarrow (\text{coin } c, \text{trapdoor } skc)$
$Spend(params, c, skc, R, C) \rightarrow (\text{proof } \pi, \text{serial } S)$
$Verify(params, \pi, S, R, C) \rightarrow \{true, false\}$

Zerocoins – Corkboard Analogy

- Work with fixed amount of 1$. Imagine a huge shared corkboard. 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 corkboard (with a trusted guardian).

1. Pin 1$ on the corkboard. Receive an anonymous ticket "Certificate : 1$ pinned" from the guardian.
2. Give a friend your certificate. Send him get the 1$ back.
3. The guardian will unpin a 1$ bill from the corkboard and give it to your friend

Cryptographic Accumulators

- Semi-commutative Hash functions

$h: X \times Y \to X, \quad \forall y_1, y_2, x, \quad h(h(x, y_1), y_2) = h(h(x, y_2), y_1)$

- Used to efficiently test membership without keeping full list of members
- Example (not efficient) : h(x, y) = x*y

  Easy to prove that y belongs to the set (if $y \mid x$), hard to find a member with only x (imply factorization).
- Witness v : keep state of accumulator without v $z_{\bar{v}}$, and v.

  Proof of membership : $h(z_{\bar{v}}, \mathrm{v}) = \mathrm{z}$

  (reveals v; for static accu.)

Better Cryptographic Accumulators

- Accumulator incrementally updatable (dynamic)
  - $f(u, x) = u^x \bmod N$, N made from strong primes pq
  - Based on strong RSA, resistant to collisions

- <u>Recall construct from beginning :</u>
  - Zero-Knowledge Proofs, Interactive -> Non-interactive -> Signature of Knowledge
  - Accumulator Static -> Dynamic
  - Result :
    - Dynamic Accumulator with Zero-Knowledge Signatures

Why Cryptographic Accumulators ?

- Why all of this ?
  - In ZeroCoins, a coin is effectively a Serial S, kept secret, created by investing the correct amount of Bitcoins
  - When creating a Zerocoin, you add S into the accumulator. Along with this, you “pin” the amount of Bitcoins.
    - Dynamic, collision-resistant accumulators
  - When you want to get back “your” bitcoins, you provide the Serial S, as well as the witness that, somewhere in the past, you did include S into the accumulator
    - Zero Knowledge make this check possible without giving information on which bitcoins were associated
  - You can then redeem the correct amount of Bitcoin from the “pool of bitcoins” formed by the pinned Bitcoins.

Integration in the Bitcoin Block Chain

- Not fully compatible with Bitcoins scripting language
- Math too advanced for the operations provided
  - Plus several math operation were disabled for security concerns
- But, conceptually :
  - Alice runs $Mint(params) \rightarrow (\text{coin } c, \text{trapdoor } skc)$, keep *skc* secret, embeds *c* in a classical Bitcoin transaction
  - Once accepted by the network, *c* is added to the global accumulator A
  - To redeem Bitcoins, Bob constructs a partial transaction *ptx* that references an unclaimed mint transaction as input. He runs $Spend(params, c, skc, hash(ptx), C) \rightarrow (\text{proof } \pi, \text{serial } S)$, and finishes by embedding (π, S) in *ptx*.
  - Other nodes perform $Verify(params, \pi, S, hash(ptx), C)$, if true and S didn't appear in other transaction, Bob can redeem the coins.

Drawbacks of Zerocoins

- **Fixed amount to guarantee** anonymity …
  - You always mint 1 Zerocoin (you can do it several times)
  - You spend/claim 1 Zerocoin
    - Can't be traced back to that Zerocoin since *every* Zerocoin has value "1 Zerocoin".
- … and yet, **still privacy issues regarding amounts**

  - Assume "Zerocoin network" is not active
  - You're the only one creating X zerocoins sequentially (X is a specific non-trivial number)
  - At a later point, you claim sequentially X zerocoins
  - Bitcoins are mathematically unlinkable to previous one, however you can be linked to both.
- Accumulators must be set-up by trusted party

  - Based on RSA, you must trust "someone" to compute $N = pq$ and throw away p, q. Solutions might exists though.

“ALT-Zerocoins”

Future version of Zerocoin
follow @matthew_d_green for updates

No paper on it : this is *speculation/promises/forecast* based on M. Green tweets.

- More private
  transaction amounts are hidden from anyone else except from payer/payee
- Faster (reduce proof size by 98%)
- Will be released as alternative currency
  - Not dependent from Bitcoin anymore (so no currency swap ?)

Conclusion

Bitcoin :

- Uses “scrambling” as a (too) thin layer of privacy 

Zerocoins :

- Based on well-studied crypto structures to guarantee unlinkability 
- But susceptible to “behavioral attacks” 
- Cumbersome 2nd currency 

- These problems might be solved in Zerocoin 2.0 

… Questions, anyone ?

Zerocoins

- Anonymous Distributed E-Cash from Bitcoin

Presentation by Ludovic Barman