8271 discussion of: "Zerocoin: Anonymous Distributed E-Cash from Bitcoin"

Stephen McCamant (Original paper: Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin)

University of Minnesota (Original paper: Johns Hopkins)

Outline

Motivation

Crypto background

Zerocoin crypto

Administrative break

Application to Bitcoin

Bitcoin pseudonymity

- Block chain is radically public
- But addresses are just random crypto keys
  - Persistent and linkable pseudonyms
- How does this compare with full anonymity?

Problems of pseudonymity

- Once you know one identity, can track forward or back
  - E.g., Ron and Shamir '13 and DPR
- Analysis just from structure
  - "10 richest people on Bitcoin"
- De-anonymize via other public info?
  - Netflix prize data and IMDB

Mixing and laundering

- Standard approach: add mixing on top
  - Compare: multi-cloud, anonymous remailers, Tor, etc.
- Unsatisfactory: trusted third-party "laundry"
  - Can log permutation, or just take your money
- Existing opt-in systems all have low volume

Idea: cryptographic mixing

- Get effect of laundry without trusted party
- Put a coin into mix, later withdraw one
  - No one else can see linkage
- Use crypto to make possible without allowing cheating
  - Prove you inserted a coin without revealing which

Outline

Motivation

Crypto background

Zerocoin crypto

Administrative break

Application to Bitcoin

Cryptographic commitment

- Common building block: commit to value now, but don't reveal until later *opening*
- Compare to scratch-off lottery ticket
- Two key properties:
  - *Hiding*: can't see value until opened
  - *Binding*: can only open to one value
- One implementation: encrypt, open by revealing key

Zero-knowledge proof

- Interactive randomized protocol between Prover and Verifier
- P convinces V of a fact with high probability
- But reveals no other information
  - Afterwards, transcript could be faked by anybody

ZK example: Hamiltonian path

- P and V share a graph, P knows a path that visits each vertex once
- On each round, P commits to a shuffled version of G
- Based on a coin flipped by V, either:
  - Reveals the whole graph, shows the isomorphism; or
  - Reveals just the path

Non-interactive ZK: Fiat-Shamir

- Converts a ZK proof technique to a non-interactive signature
- Idea: replace V's random choices with the output of a hash function
  - Just as uncontrollable if the function is pseudo-random
- Security proof works only in Random Oracle Model

One-way accumulators

- Prove membership in set in constant space
- Based on function H with $H(H(x, y_1), y_2) = H(H(x, y_2), y_1)$, such as x^y mod N
- Think: represent set as product of primes: *witness* for p_i is product of all other members

Outline

Motivation

Crypto background

Zerocoin crypto

Administrative break

Application to Bitcoin

Zerocoin overview

- Zerocoin “Mint”: turn one BTC into one Zerocoin
  - Commit to serial number
- Zerocoin “Spend”: convert one Zerocoin into BTC
  - Reveal serial number, must be unique

Formal definition

- Setup$(1^\lambda) \longrightarrow$ *params*
- Mint(*params*) $\longrightarrow (c, skc)$
  - c: coin, skc: corresponding secret key
- Spend(*params*, $c, skc, R, \mathbf{C}) \longrightarrow (\pi, S)$
  - R: transaction string, $\mathbf{C}$: previous coins, π: ZK proof, S: serial number
- Verify(*params*, $\pi, S, R, \mathbf{C}) \longrightarrow \{0, 1\}$

Security def.: anonymity

- Honest party mints two valid coins c_0, c_1, adversary picks $\mathbf{C}$ and R
- Honest party picks $b \leftarrow \{0, 1\}$, spends c_b with R and $\mathbf{C} \cup \{c_0, c_1\}$
- Adversary tries to guess b, should not do much better than 50-50.

Security def.: balance

- Honest party mints N coins
- Adversary constructs m coins and $m + 1$ spends
- Adversary wins if all $m + 1$ spends verify using the $N + m$ coins, have unique serial numbers, but none are honest spends of the N coins

Core of construction

- To mint, choose S and r at random such that $c = g^S h^r \bmod p$ is prime
  - Secret key is (S, r)
- Accumulate coins $\mathbf{C}$ into A
- To spend, π proves knowledge of c, w, r such that:
  - w witnesses c is in A
  - $c = g^S h^r \bmod p$

Trusted setup issue

- Accumulator based on exponentiation mod a product of primes pq
- Knowledge of primes allows forgery (c.f. RSA)
  - Honest party deletes them, but this isn't verifiable
- Other techniques ("RSA UFO") allow constructing product without getting primes

Outline

Motivation

Crypto background

Zerocoin crypto

Administrative break

Application to Bitcoin

Project meetings

- Purpose: discuss project topics
- Email me to set up
- Thursday, Friday, or next week

Presentation choices

- Already got a volunteer for next Monday
- Expect other results soon

Presentation slides

- If you send them early, I can give suggestions
- Send final version for my grading use
- Decide whether you want them public, on Moodle, or forgotten

Outline

Motivation

Crypto background

Zerocoin crypto

Administrative break

Application to Bitcoin

Zerocoin transactions

- Simple design: only one value of Zerocoin
  - Can run multiple copies for more denominations
- Zerocoin "Mint" puts BTC in escrow
- Zerocoin "Spend" takes its BTC from any previous Mint

New state required

- Accumulator computed incrementally
  - Checkpointed in each block
- Nodes must maintain list of spent Zerocoin serial numbers
- Proofs might be kept outside the block chain

Limits of anonymity

- All Zerocoin does is obscure connection between mints and spends
- Security still limited by number of outstanding Zerocoins
  - 10 mints; 10 spends; 1 mint: no anonymity
- Also can't help if too many other users collude against you
  - E.g. other coins all created by a single adversary

Parameter sizes

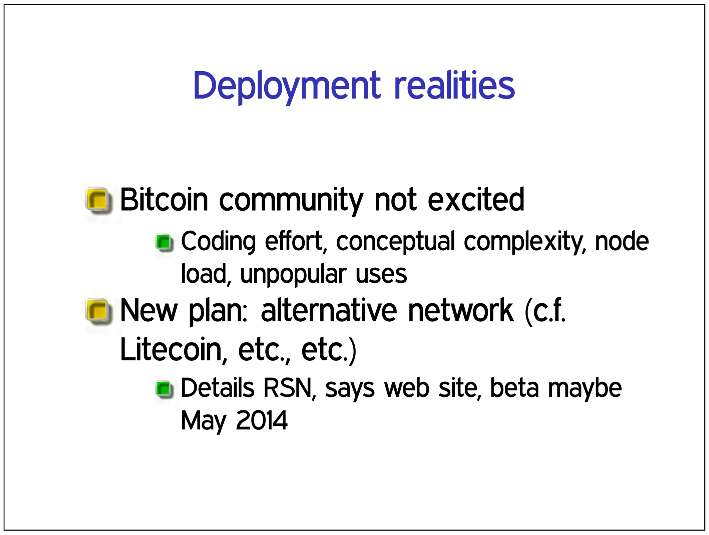
- Group used in commitments: size sensitive
  - Make 1024 bit, assume periodically regenerated
- RSA modulus used in accumulator: hard to regenerate, must last
  - At least 3072 bits proposed
- ZK # rounds: just affect a single proof
  - Proposed 2^{80} security

Performance

- Not cheap, but can scale beyond then-current Bitcoin volumes
- Proof is about 40KB
- Mint, spend, verify all less than 1 second
- Verification of blocks by nodes more problematic than by miners

Deployment: plans as of paper

- Integrate into the regular Bitcoin network
- Cleanest: add new operations in protocol, "flag day" upgrade
- Incremental alternative: build on current protocol
  - Zerocoin information is in comments
  - Signatures by a quorum of semi-trusted Zerocoin nodes



Deployment realities

- Bitcoin community not excited
  - Coding effort, conceptual complexity, node load, unpopular uses
- New plan: alternative network (c.f. Litecoin, etc., etc.)
  - Details RSN, says web site, beta maybe May 2014