Towards making anonymous

Matthew Green
Johns Hopkins University

Joint work with
- Ian Miers, Christina Garman, Avi Rubin (Oakland '13)
- Alessandro Chiesa, Madars Virza, Ian Miers, Christina Garman, Eran Tromer, Eli Ben-Sasson (In submission)
Bitcoin privacy

• **TL;DR: Bitcoin is not very anonymous**

• Bitcoin transactions are recorded in a *public* ledger

• Parties ‘write checks’ using pseudonyms (addresses)

• If people can link you to your address, you’re screwed

• *You’re probably screwed*

```
.32, A->B
  1.03, S->J
  2.5, M->S
...

1.0, J->Z
 .23, B->C
 .9B->D
  ...
  HASH

.23, C->E
  1.2, E->J
  .2, M->J
  ...
  HASH

.23, E->F
  .9, M->B
  .9, D->Z
  ...
  HASH
```
This matters!

• Solving the privacy problem is crucial to Bitcoin’s long-term success

• Existing countermeasures don’t address the problem, and probably never will

• A real solution may yield useful new techniques
Outline of this talk

• Today I’m going to talk about two “fixes” for this problem:
  • Zerocoin - privacy for Bitcoin
  • Zerocash - efficient and deployable privacy for Bitcoin
Zerocoin

Joint work with
Ian Miers, Christina Garman, Avi Rubin (Oakland ’13)
Let’s use e-Cash for Bitcoin!

- e-Cash due to Chaum [82] (many subsequent works)

- Untraceable electronic cash

- Traditional schemes withdraw ‘coins’ from a *central bank* (using *blind signatures*)
Let’s use e-Cash for Bitcoin!

- e-Cash due to Chaum [82] (many subsequent works)
- Untraceable electronic cash
- Traditional schemes withdraw ‘coins’ from a central bank (using blind signatures)
Zerocoin

• New approach to creating electronic coins
  • Based on a technique due to Sander and Ta-shma
  • Extends Bitcoin by adding a ‘decentralized laundry’
  • **No bank:** Requires only a trusted bulletin board
  • Bitcoin block chain gives us this ‘for free’!
The high level idea

• I can take Bitcoin from my wallet

• Turn them into ‘Zerocoin’

• Where they get ‘mixed up’ with many other users’ coins

• I can redeem them to a new fresh Wallet
Minting Zerocoin

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

823848273471012983
Minting Zerocoin

• Zerocoin are just numbers

• They have value once you put them on the block chain

• This costs e.g., 1 bitcoin
You can redeem zerocoins back into bitcoins

Reveal the serial number &
Prove that it corresponds to some Zerocoin on the chain

In exchange you get one bitcoin
Spending Zerocoin

• Why is spending anonymous?

• It’s all in the way we ‘prove’ we have a Zerocoin

• This is done using a zero knowledge proof
Spending Zerocoin

• Zero knowledge [Goldwasser, Micali 1980s, and beyond]

• Prove a statement without revealing any other information

• Here we prove that:
  (a) there exists a Zerocoin 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 ZeroCoin

- Our approach

  - Use an efficient RSA one-way accumulator

  - Accumulate $C_1, C_2, \ldots, C_N$ to produce accumulator $A$

  - Then prove knowledge of a witness s.t. $C \in inputs(A)$

  - And prove knowledge that $C$ opens to the serial number

---

Requires a DDL proof (~25kb) for each spend. In the block chain.
Response from the Bitcoin community:

I’m going to need a lot of booze to handle this.

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

- Good first approach:
  - Implemented!
  - Proofs are (too?) big
  - Coins all have the same value
  - Must convert ‘zerocoins’ to ‘bitcoins’ in order to spend them
Zerocash

Joint work with
- Alessandro Chiesa, Madars Virza, Ian Miers, Christina Garman, Eran Tromer, Eli Ben-Sasson (In submission)
A better tool

• Better, smaller ‘proofs’ of knowledge:

• **S**uccinct **N**on-Interactive **A**rguments of **K**nowledge (zkSNARKs) (Parno et al., Ben-Sasson et al.)

• 288 byte proof for arbitrary-sized arithmetic circuits

• And there are C compilers!
How not to use SNARKs

• In theory this should be simple:

• We’ve already coded up Zerocoin in C++

• Let’s run our existing software through the zkSNARK compilers to get small proofs

• Surprise: This gives large, impractical circuits (proving takes a long time)
How to use SNARKs

• Start from scratch:
  • Develop an entirely new construction with small circuits
  • Modify Zerocoin to use hash functions for commitments, hash trees for an accumulator (SHA256 for all hashes)
  • Hand-optimize everything

Figure 2: Size of circuit $C_{SHA256}$ for SHA256.
Proposed Zerocash tree

\[ C_i = H(r_i || sn) \]
But wait a second...

• If the proofs are powerful & efficient, why do we need Bitcoin anymore?

• Let’s add hidden values to the coin: $C_i = H(r_i || v || sn)$

• Create transactions to split/merge coins

• Allow payments (from Alice to Bob) that don’t reveal value

  • Pay to individuals, pay to address
To split a coin:

1. “Spend” the input coin (by revealing its serial number)
2. “Mint” two new coins
3. Prove that the new coins total to the value of the first coin
To merge two coins:

1. “Spend” the input coins (by revealing their serial numbers)
2. “Mint” a new coin
3. Prove that the old coins total to the value of the new coin
To pay a coin:

1. Transfer the coin secrets to the target user
2. Embed the recipient’s ‘address’ $A = H(x)$
3. User must prove knowledge of $x$ to redeem
Result: Zerocash

• An fully untraceable, divisible electronic cash system

• Coins are anonymous starting from Coinbase transaction

• Coins can be split/joined (‘poured’), paid and revealed

• The only place where coin values need be public is when we offer transaction fees
Performance

<table>
<thead>
<tr>
<th></th>
<th>Proving time</th>
<th>Proof Size</th>
<th>Verif. time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Split</strong></td>
<td>87 sec</td>
<td>288 bytes</td>
<td>8.6 ms</td>
</tr>
<tr>
<td><strong>Merge</strong></td>
<td>178 sec</td>
<td>288 bytes</td>
<td>8.6 ms</td>
</tr>
</tbody>
</table>

128-bit security level, single core i7 @ 2.7 GHz
So what’s the catch?

- The public parameters are quite large
  - About 1.2 GB
    - In context, that’s about 7% the size of the blockchain
    - They must be generated by a trusted party
      - A party who knows a trapdoor can forge proofs
      - But cannot de-anonymize transactions
The summary

• We now have efficient and fully anonymous e-Cash
  • With practical proving times & storage costs
  • A modestly irritating set of public parameters
  • And code, which we will be releasing in May

• So what next?
Release all the things

• This is Real World Crypto, after all

• Our experience tells us that code is not enough

• We need time to work out the bugs, but...

• In May we will be launching a full alt-coin based on the Zerocoin code, mostly for testing

• Dear god do not put real money into it
A last thought

Why Bitcoin (And Other Cryptocurrencies) Will Inevitably Become Tools Of The Rich, Powerful, and Criminal

Does Zerocoin have any benefits that justify allowing these kinds of harms? I haven’t heard anyone make that case. So here’s the challenge to supporters of anonymous money transfer: Make an affirmative case for it. Give examples of where we’ll all be better off if people can make untraceable peer-to-peer money transfers. Tell us how you are going to be substantively less free in a world where financial paper trails exist.

E.J. Fagan is Deputy Communications Director at Global Financial Integrity, a research and advocacy organization based in Washington, D.C., dedicated to studying and curtailing illicit financial flows.