Privacy technology in blockchain applications

Sarang Noether, Ph.D.

Monero Research Lab

10 November 2018

Disclaimers

The views expressed in this presentation are solely those of the author and do not necessarily reflect those of the Monero Research Lab or Monero Project.

The author receives funding from Monero community members to conduct research for the Monero Research Lab, a workgroup of the Monero Project.

The material in this presentation is for informational purposes only and should not be construed as endorsement or financial/investment advice.

What is all this?

Monero is a cryptographic asset project. It is an open-source project developed, researched, and tested by its community. There is no company, ICO, foundation, or other bullshittery. Some projects are funded by community donations.

The **Monero Research Lab** is a Monero workgroup that conducts research and development. There are two Ph.D. researchers who request full-time funding for their work.

Working definition for ledgers

A **cryptographic asset** or **cryptocurrency** is an asset whose ledger integrity is assured using cryptography. Such an append-only ledger is often a data structure called a **blockchain**. It is usually widely distributed.

Transactions are signed messages transferring existing quantities of the asset from one account to another, and must be added to the ledger.

Some network participants are **miners**. They collect other participants' transactions, check their validity using the ledger, and add them in **blocks** to the chain. Miners use computing power (and electricity) to compete[1] for the right to add the next block.

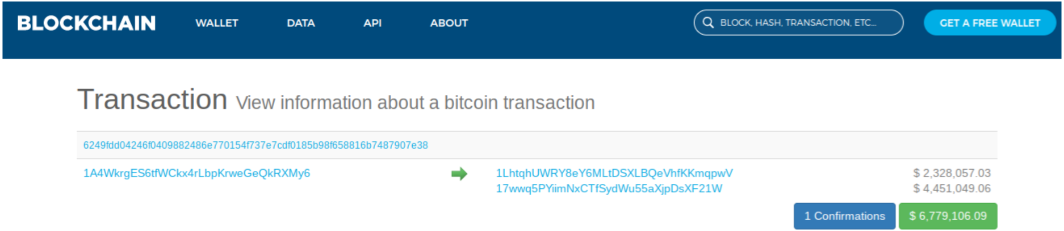
The winning miner's block is added to the blockchain, and the protocol rewards the miner by generating new assets for it.

[1] Yes, I know, PoS...

The dirty truth

In the bitcoin ecosystem, transactions are done pseudonymously, but you are not anonymous.

Transaction sender, receiver, and amount are all public.



Transaction View information about a bitcoin transaction

6249fdd04246f0409882486e770154f737e7cdf0185b98f658816b7487907e38

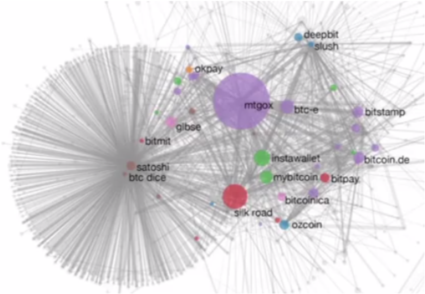
1A4WkrgES6tfWCkx4rLbpKrweGeQkRXMy6 ➡ 1LhtqhUWRY8eY6MLtDSXLBQeVhfKKmqpwV $ 2,328,057.03
17wwq5PYiimNxCTfSydWu55aXjpDsXF21W $ 4,451,049.06

1 Confirmations $ 6,779,106.09

In the real world

How does this affect actual analysis?

- Input clustering: same control
- Output clustering: change tracking
- "Cross-chain" operations: asset exchanges
- Identities: KYC/AML exchanges
- Mixers: shared responsibility?
- Tainting: it knows what you did last summer...



It is not a good idea to assume Bitcoin-style ledgers offer any anonymity.

The ledger axes

Think of a distributed blockchain ledger as having different scales.

open ↔ **closed** (availability)

public ↔ **private** (information visibility)

Bitcoin, and most of the assets you've probably heard of, is open and public. Other non-ledger applications may use closed blockchains.

The goal

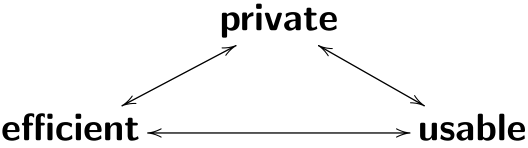
We want an **open** and **private** ledger that is easy to use.

There are several great examples of different approaches in use.

Privacy versus practicality

There are tradeoffs to practical privacy technologies.

- **Privacy:** identities, amounts, transparency
- **Efficiency:** time, space, requirements
- **Correct user behavior:** tyranny of the default



private

efficient

usable

Case study

Zerocoin

The basics

Zercoin (2013) is a Bitcoin extension that allows for a zero-knowledge asset transfer.

Mint: You burn a fixed quantity of Bitcoin by signing a minting transaction; a random value and serial number produce a Zerocoin using an obfuscating **cryptographic commitment** blob of data.

Spend: To spend a Zerocoin to recover the fixed quantity of Bitcoin, the spender provides:

- a serial number
- a proof that the serial number corresponds to some Zerocoin earlier on the chain
- a transaction message detailing the spend

Downsides already?

There are already problems with this arrangement.

- **Trusted setup**: the cryptographic accumulator (hidden coin pool) used in the proof uses an RSA number[2]
- **Proof size**: the entire spend proof is costly in terms of space
- **Verify time**: verifying the spend proof takes time
- **Fixed amounts**: arbitrary amounts don't work

[2] There are supposedly plans to move past this

RSA-2048

251959084756578934940271832400483985714292821262040320277771378360436620
207075955562640185258807844069182906412495150821892985591491761845028084
891200728449926873928072877767359714183472702618963750149718246911650776
133798590957000973304597488084284017974291006424586918171951187461215151
726546322822168699875491824224336372590851418654620435767984233871847744
479207399342365848238242811981638150106748104516603773060562016196762561
338441436038339044149526344321901146575444541784240209246165157233507787
077498171257724679629263863563732899121548314381678998850404453640235273
81951378636564391212010397122822120720357

An attack

What if the serial number isn't chosen randomly?

1. Mallory intercepts Alice's spend proof for a particular serial number
2. Mallory generates a new coin with the same serial number!
3. Mallory spends this coin to himself
4. Alice's honest spend is rejected as a double-spend attack

Alice's coin is burned!

Vulnerable: Zcoin, PIVX, SmartCash, Zoin, Hexxcoin

Case study

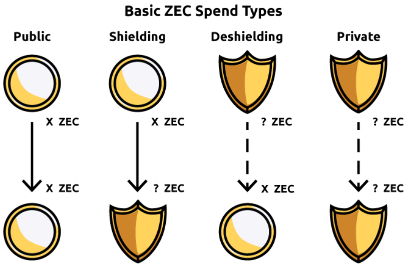
Zcash

The basics

Zcash (2014) is a protocol (and asset) that effectively uses two types of addresses: **transparent** and **shielded**.

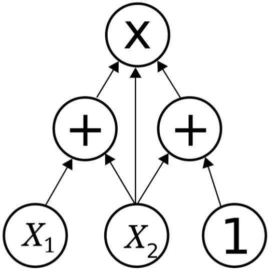
- Transparent transactions work just like in Bitcoin, with no privacy.
- Shielded transactions hide the sender and receiver (with a full anonymity set), as well as the amount.

The initialization of the Zcash system required a trusted setup process.



Tech: zk-SNARKs

What the heck is a zk-SNARK?



It's a type of proving system for arbitrary **arithmetic circuits**.

1. represent your statement as a circuit
2. transform into a polynomial representation
3. use a sampling method to show correctness

The trusted setup generates encrypted forms of specific private data to make the proof small.

This is not the only way to prove statements in zero knowledge, but it is tiny and efficient to check.

zk-WALDO

You may have heard of projects going nutso about **zero-knowledge proofs**. A zero-knowledge proof is a generated structure that lets a prover convince a verifier of the truth of a statement, without revealing any other information about the statement's details. (This is not a precise definition.)

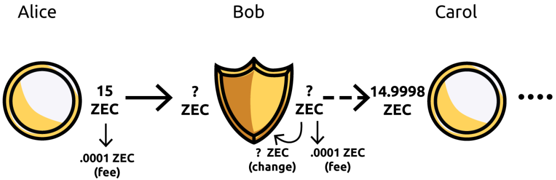
Example: "The asset spent in this transaction exists somewhere on the chain."

Example: "This hidden transaction amount represents a number that is not negative."

A zero-knowledge proof system does not give you a complete asset inherently.

Attacking optional privacy

Privacy in Zcash is optional: shielded transactions were expensive to generate (not anymore). Exchanges tend not to support shielded operations.



It's possible to link amounts leaving the shielded pool (minus fees) with amounts entering the pool. Plus, time matters!

Other attack vectors

It's also possible to use other metrics to link transactions:

- Input address clustering
- Exchange address tagging
- Founder address tagging

Motto: Optional privacy is not a good idea

Note that Zcash recently deployed a protocol upgrade (Sapling) to make shielded transactions much more efficient and encourage their use.

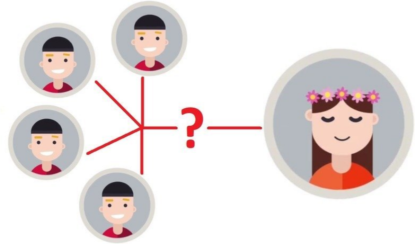
Case study

Monero

The basics

Monero is a cryptocurrency that **does not allow** transparent transactions at all.

Senders are anonymous within a small group, and receivers are fully anonymous. Transaction amounts are hidden.



Tech: Ring signatures, one-time addresses, confidential transactions

One-time addresses

In Monero, addresses *never* appear on the blockchain! Each transaction generates **one-time addresses** that cannot be linked to the recipient's true address.

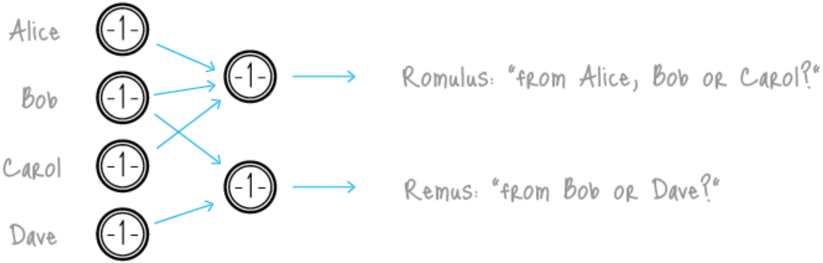
Having control of a true address provides a “spend authority” for funds directed to a one-time address.

This mitigates against some forms of transaction linking by making every transaction destination unique.

Ring signatures

A **ring signature** is a cryptographic construction proving that one of a group of one-time addresses is being spent in a transaction. Selection of decoys is non-interactive.

These are one-time addresses, not true addresses like in Bitcoin!


Alice
Bob
Carol
Dave
-1-
Romulus: "from Alice, Bob or Carol?"
Remus: "from Bob or Dave?"

Confidential transactions

Monero's **confidential transaction** model (RingCT) hides amounts in commitments. Observers gain no information about a transaction's input or outputs amounts.

Algebra on the commitments lets anyone check that the transaction balances: no funny business happens!

Result: Some output in this group spent an unknown quantity of Monero to an unknown destination.

What can go wrong?

Have a small sender anonymity set is fine as long as *one-time addresses aren't known to be (un)spent*. That is, Monero does not have a UTXO set, only a TXO set.

Not good:

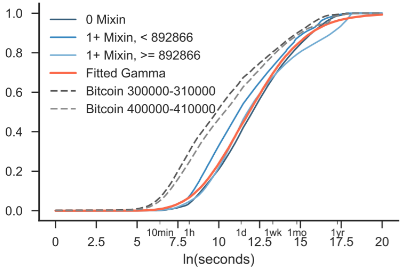
- Addresses belonging to an attacker are known to be spent/unspent
- Addresses also spent on a fork can reduce effective ring sizes
- Addresses sent to you in controlled purchases that are later spent on a compromised exchange
- Addresses that do not match typical spending patterns can be inferred as spent

Heuristics are not provable information!

The effects of ring signatures over time depend heavily on the choice of parameters.

How do we mitigate?

- **Bad:** Many old outputs are spent.
  Better: Legacy outputs aren't used as decoys.
- **Bad:** Multi-fork outputs are spent.
  Better: Ring sizes are increased.
- **Bad:** Some transactions stand out.
  Better: We enforce certain parameters.
- **Bad:** Decoy timing leads to heuristics.
  Better: We iterate on our decoy selection.



The current state of the art

Current trends tend to involve:

- Efficiency choices based on centralized or distributed trust
- Mixers in existing protocols versus newer protocols (Monero, Zcash, Zerocoin)
- Structures like accumulators or ring signatures for spend anonymity

We are working toward solutions that are:

- Trustless
- Zero-knowledge
- High anonymity
- Low metadata
- Efficient

Sarang's protips

- **Privacy needs to be defined.**
  Watch out for broad or vague claims without analysis.
- **Privacy must not be optional.**
  It is too easy to lose essential anonymity and fungibility.
- **Integration is tricky and subtle.**
  Building a cryptographic system is not like a Lego house.
- **Flaws matter, but so does response.**
  Researchers unwilling to discuss or address flaws should be treated with caution.
- **Get vetted math.**
  If it sounds too good to be true, it probably is. "We have a whitepaper" means nothing unless the technology is vetted.
- **Consider the trust model.**
  The protocol structure of an asset or blockchain application is crucially tied to the trust profile of the participants.

MONERO

MONERO RESEARCH LAB

https://getmonero.org

https://git.io/MRL

sarang@getmonero.org