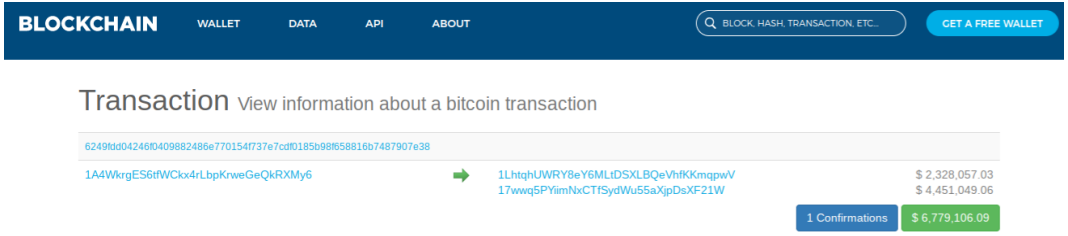
Privacy technology in blockchain applications

20 February 2019

The dirty truth

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

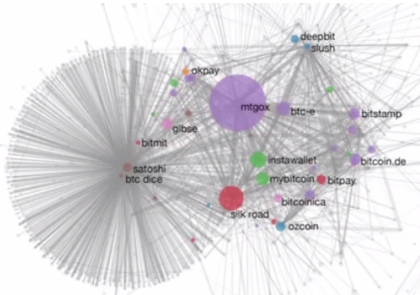
Transaction sender, receiver, and amount are all public.



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.

public $\leftrightarrow$ **private** (availability)

transparent $\leftrightarrow$ **opaque** (information visibility)

Bitcoin, and most of the assets you've probably heard of, is public and transparent. Other applications may use private blockchains.

The goal

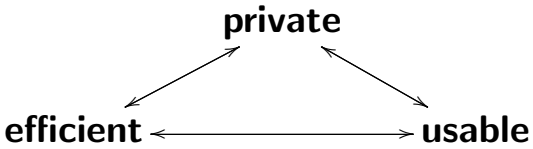
We want a **public** and **opaque** ledger that is easy to use.

There are applications (healthcare, regulated banking, business) that may require a private base layer.

Privacy versus practicality

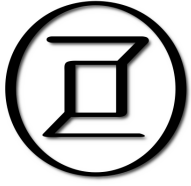
There are tradeoffs to practical privacy technologies.

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



Case study

Zerocoin



The basics

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

It provided one of the first examples of applications of cryptographic primitives toward fungible assets.

The protocol burns a fixed quantity of Bitcoin for a **zerocoin**, which can later be withdrawn to another Bitcoin address without linking to the original burn.

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[1]
- **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

Motto: With enough limitations, you don't get very far

[1] There are supposedly plans to move past this

Case study

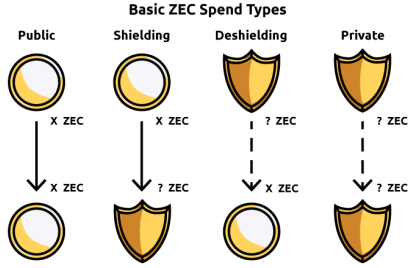
Zcash

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.

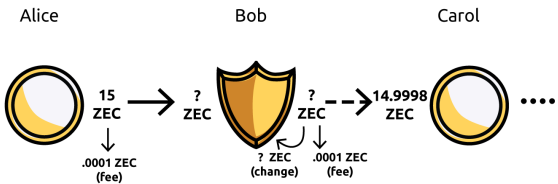
The initialization of the Zcash system required a trusted setup process.



Tech: zk-SNARKs

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 gain information:

- Input address clustering
- Exchange address tagging
- Founder address tagging
- Sprout-to-Sapling turnstile

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.

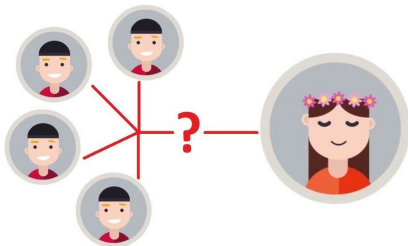
Case study

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.

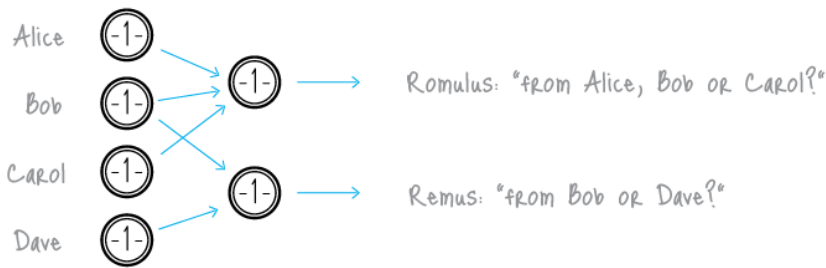
Having control of a true address provides a "spend authority" for funds directed to a one-time address.

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!



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?

Having a small sender anonymity set requires that *one-time addresses aren't known to be (un)spent*. That is, Monero should not have a UTXO set, only a TXO set.

Not good:

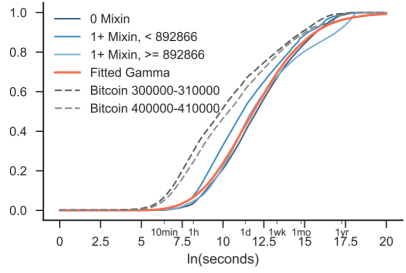
- Addresses belonging to an attacker are known to be spent/unspent
- Addresses also spent on a fork can reduce effective ring sizes
- Addresses sent to you in controlled purchases that are later spent on a compromised exchange
- Addresses that do not match typical spending patterns can be inferred as 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 non-mixing protocols
- Structures like accumulators or ring signatures for spend anonymity

We are working torward solutions that are:

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

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.