UNIVERSITÀ DI PISA

LAUREA MAGISTRALE IN
INFORMATICA UMANISTICA

SEMINARIO DI CUTURA DIGITALE A.A. 2016/17

CRYPTO CURRENCIES
Decentralization and privacy concerns

Valeriya Slovikovskaya
Matricola*: 503602*

Summary

Emerging technologies often present both great potential benefits as well as real risks. In case of cryptocurrencies the value of privacy comes into conflict with the values of efficiency and transparency. The topic discussed in this paper is a measure of balance between users' privacy needs and the need to enforce financial regulations. The discussion makes evident that Bitcoin owes its widespread acceptance to the malleability of its approach to privacy.

Index

1. Introduction. Bitcoin: decentralization vs. privacy conflict

Online payments systems such PayPal, Visa, and Western Union Pay, along with virtual currencies from World of Warcraft Gold to Facebook Credits to e-gold are all are part of traditional commerce that relies on financial institutions as trusted third party.

Bitcoin — invented in 2008 and rocketed to popularity with a monetary base worth over US$6 billion in early 2014 — is different.

- It is the first decentralized digital currency to achieve widespread adoption.
- It is not administered by any central authority.
- It replaces trust by cryptographic proof.
- It allows any two willing parties transact directly with each other.

Bitcoin’s key innovation is the idea of recording all transactions in the blockchain, a public, global, and immutable append-only ledger.

A cryptocurrency relies on a peer-to-peer network that keeps track of a shared append-only data structure, called a blockchain, which represents a ledger of user account balances, i.e., mappings between quantities of currency and public keys held by their current owner. Rather than having a notion of accounts and transactions among accounts, Bitcoin tracks the fractions of coins. To spend a portion of cryptocurrency, users broadcast digitally-signed messages called transactions, which are then validated and appended to the blockchain.

Each transaction in Bitcoin describes the movement of coins from one logical location to another. It contains some number of inputs and outputs; inputs consume coins, and outputs create new coins, conserving the total balance. Each input spends an unspent transaction output created in a prior transaction. Together, these form a transaction graph.

To serialize new transactions, miners aggregate transactions in a block and append the block to the ledger by solving a proof of work crypto puzzle. This process is financially rewarded by allowing the successful miner to mint new coins in a special coinbase transaction. Using a proof-of-work system, the integrity of the ledger is maintained as long as a majority of the computing power is contributed by honest participants.

Many at the Bitcoin Foundation and in the Bitcoin community are acutely aware that financial transactions in nearly every format are subject to some degree of surveillance. For good and bad, centralized payment systems always include gatekeepers and overseers. Bitcoin can facilitate unmediated transactions, which, when legal in the jurisdictions where they occur, are nobody’s business but the parties to the transactions.

This decentralization is achieved because of the traceability of transaction. That means, the decentralized design comes at a expense of user privacy.

Once transaction graph is public, it can be mined by anyone able to apply “address clustering” technique with a bunch of heuristics and link together all the pseudo-identities controlled by an individual or entity. Once peer-to-peer bitcoin network allows not encrypted communication, it becomes open for graph analysis attacks.

And what we discuss in this paper is the conflict between transparency and privacy needs.

We concentrate on techniques, developed by Bitcoin community and aimed to allow “user-defined privacy”, i. e. privacy to the extent and in terms user wants it.

2. Bitcoin: lack of anonymity and deanonymization attacks

2.1. Anonymity: formal definition

In computer science, anonymity refers to pseudonymity together with unlinkability. Unlinkability means that if a user interacts with the system repeatedly, these interactions can not be tied to each other from the point of view of the specific adversary.

For Bitcoin activity to be unlinkable:

1. It should be hard to link together different addresses of the same user.
2. It should be hard to link together different transactions made by the same user.
3. It should be hard to link the sender of a payment to its recipient.

The third property is quite hard to achieve.

Assume user pays for a product that costs a certain number of bitcoins and he sends that payment through a circuitous route of transactions. Somebody looking at the block chain reveals the fact that a certain number of bitcoins left one address and roughly the same number of bitcoins (minus transaction fees, perhaps) ended up at some other address.

He notices also that the initial sending and the ultimate receiving happen in roughly the same time period because the merchant wants to receive payment without too much of a delay.

Because of this difficulty, we usually don't try to achieve complete unlinkability among all possible transactions or addresses in the system, but rather something more limited.

2.2. Anonymity in terms of anonymity set

Given a particular adversary, the anonymity set of user transaction is the set of transactions which the adversary cannot distinguish from user transaction. Even if the adversary knows user made a transaction, he can only tell that it’s one of the transactions in the set, but not which one it is. User's level on anonymity grows with increase of the size of anonymity set where his transaction is hidden.

To determine anonymity set it's necessary to define the adversary model and reason about what adversary knows and does not know, and what is needed to be hidden from him. It requires carefully analyzing each protocol and system on a case-by-case basis.

2.3. Deanonymisation attack: Bitcoin transaction graph analysis

Suppose Alice wants to buy a teapot that costs 8 bitcoins (more likely 8 centi-bitcoins, at 2015 exchange rates). Suppose, further, that her bitcoins are in three separate unspent outputs at different addresses whose amounts are 3, 5, and 6 bitcoins respectively. Alice doesn't actually have an address with 8 bitcoins sitting in it, so she must combine two of her outputs as inputs into a single transaction that she pays to the store (Figure 1).

Heuristics 1: Joint inputs imply joint control

But this situation reveals sensitive information. When transaction gets recorded in the block chain, anyone who sees it can infer that the two inputs to the transaction are most likely under the control of the same user.

In other words, shared spending is evidence of joint control of the different input addresses. There could be exceptions, of course. Perhaps Alice and Bob are roommates and agree to jointly purchase the teapot by each supplying one transaction input. But in most real cases, joint inputs imply joint control.

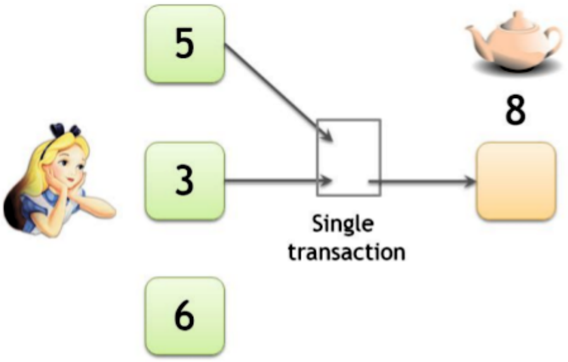

Figure 1: To pay for the teapot, Alice has to create a single transaction having inputs that are at two different address. It reveals that these two addresses are controlled by a single entity.

If another address is linked to either one of Alice’s addresses in this manner, then we know that all three addresses belong to the same entity, and we can use this observation to cluster addresses. In general, if an output at a new address is spent together with one from any of the addresses in the cluster, then this new address can also be added to the cluster.

Suppose the price of the teapot has increased from 8 bitcoins to 8.5 bitcoins. Alice can no longer find a set of unspent outputs that she can combine to produce the exact change needed for the teapot. Instead, she exploits the fact that transactions can have multiple outputs. One of the outputs is the store’s payment address and the other is a “change” address owned by herself.

In this case adversary can deduce that the two input addresses belong to the same user. They might further suspect that one of the output addresses also belongs to that same user, but has no way to know for sure which one that is.

The fact that the 0.5 output is smaller doesn’t mean that it’s the change address. Alice might have 10,000 bitcoins sitting in a transaction, and she might spend 8.5 bitcoins on the teapot and send the remaining 9,991.5 bitcoins back to herself. In that scenario the bigger output is in fact the change address.

The effectiveness of this type of heuristic depends entirely on the implementation details of commonly used wallet software.

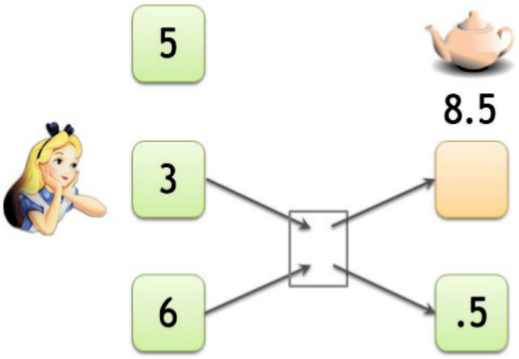

Figure 2. Change address. To pay for the teapot, Alice has to create a transaction with one output that goes to the merchant and another output that sends change back to herself

Idioms of use 2: fresh address is change address

Implementation details of this sort are called “idioms of use”. In 2013, a group of researchers found an idiom of use that was true of most online wallets, and developed a powerful heuristic for identifying change addresses. Specifically, they found that wallets typically generate a fresh address whenever a change address is required. Because of this idiom of use, change addresses are generally addresses that have never before appeared in the block chain. Non-change outputs, on the other hand, are often not new addresses and may have appeared previously in the block chain. An adversary can exploit this knowledge to distinguish change addresses and link them with the input addresses.

Relying on idioms of use is usually error prone. The fact that change addresses are fresh addresses just happens to be a feature of wallet software. It was true in 2013 when the researchers tested it. Maybe it’s still true, but maybe it’s not. Similar heuristic based assumptions can produce a lot of false positives and lead to clustering together addresses that didn’t actually belong to the same entity.

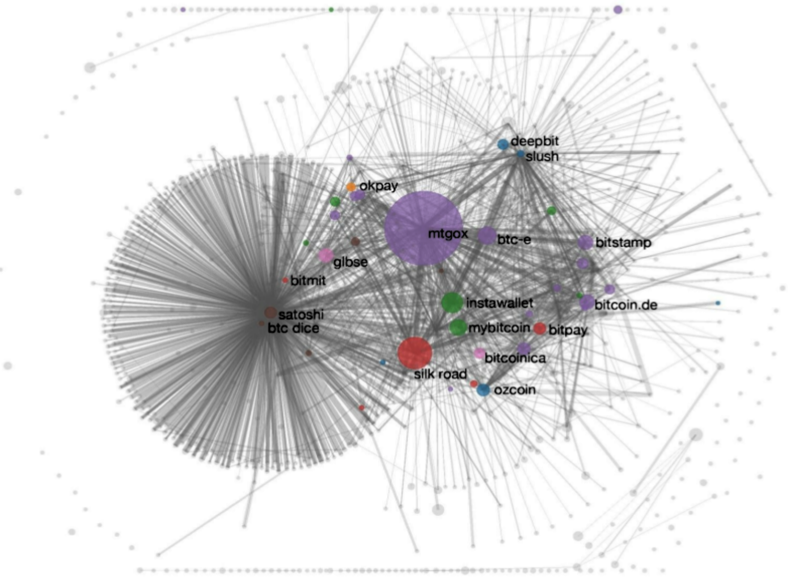

Figure 3. Clustering of addresses. In the 2013 paper A Fistful of Bitcoins: Characterizing Payments Among Men with No Names, researchers combined the shared-spending heuristic and the fresh-change-address heuristic to cluster Bitcoin addresses. The sizes of these circles represent the quantity of money flowing into those clusters, and each edge represents a transaction.

Attaching real-world identities to clusters
To label the clusters discovered in transaction graph the adversary can be able to make some educated guesses based on his knowledge of Bitcoin economy. Back in 2013, Mt. Gox was the largest Bitcoin exchange, so the adversary could have guessed that the big purple circle on Figure 3 represents addresses controlled by this company. The brown cluster on the left that has a tiny volume in Bitcoins but the largest number of transactions can be interpreted as the gambling service Satoshi Dice, because it fits the pattern of the gambling service to which people send a tiny amount of bitcoins as a wager.

Suppose the adversary visits the website for each exchange or merchant and looks up the address they advertise for receiving bitcoins? It doesn't help: most services advertise a new address for every transaction and those advertised are not yet in the block chain and most of them will never reach it.

To conduct his deanonymizing attack adversary should transact with bitcoin service provider, depositing bitcoins, purchasing the items, and so on. When he sends bitcoins to or receive them from the service provider, he discovers one of the addresses, that eventually end up in the block chain. He can then tag the entire cluster with the service provider's identity. This is is exactly what the Fistful of Bitcoins researchers [Meiklejohn et al., 2013] have done. They interacted with shops, mining pools, Bitcoin exchanges, wallet services, and gambling sites, and they guesses about Mt. Gox and Satoshi Dice were correct.

Identifying individuals
The next question if the adversary can trace individuals? The answer is yes and the ways are: directly transacting, exploiting service provider records, and, in the end, people careless.

Directly transacting: anyone who transacts with an individual — an online or offline merchant, an exchange, or a friend who splits a dinner bill using Bitcoin — knows at least one address belonging to them.

*Interacting with service providers: i*n the course of using Bitcoin over a few months or years, most users will end up interacting with an exchange or another centralized service provider. These service typically ask users for their identities (often they're legally required to do this). If law enforcement wants to identify a user, they can turn to these service providers.

Exploiting people carelessness: people often post their Bitcoin addresses in public forums. A common reason is to request donations. When someone does this it creates a link between their identity and one of their addresses. If they don't use the anonymity services, they risk having all their transactions deanonymized.

The deanonymization techniques described so far are based on analyzing the graphs of transactions in the block chain, they are known known as transaction graph analysis.

2.4. Denonymization of clients in Bitcoin P2P network

Another way to deanonymize users is network-layer deanonymization, first suggested by Dan Kaminsky at the 2011 Black Hat conference.

In networking terminology, the block chain is called the application layer and the peer-to-peer network is the network layer. In order to post a transaction to the block chain a user typically connects to many nodes and broadcasts it. If sufficiently many nodes on the network are run by the same adversary, the adversary could figure out the first node to broadcast any transaction.

With a high probability, that would be a node that’s run by the user who created the transaction. The adversary could then link the transaction to the node’s IP address. An IP address is quite close to a real world identity; there exist many ways to try to unmask the person behind an IP address. Thus, network-layer deanonymization is a serious problem for privacy.

It is known that for communications anonymity people widely use Tor, "The Onion Router", that directs Internet traffic through a free, worldwide, volunteer overlay network of more than seven thousand relays to conceal a user's location from anyone conducting network surveillance or trafice analysis. Nevertheless, Alex Birykov et al. tested an effective attacker technique that prevents Bitcoin users from connecting via Tor. They exploited the lack of authentication within the Bitcoin network, which requires the nodes to blacklist misbehaving peers by IP and they figured out that very short messages may cause a day IP ban and can be used to separate a given node or the entire network from anonymity services such as Tor.

3. Anonymization techniques: proposals

There are several mechanisms that can make transaction graph analysis less effective. The idea is to use intermediary.

3.1. Mixing

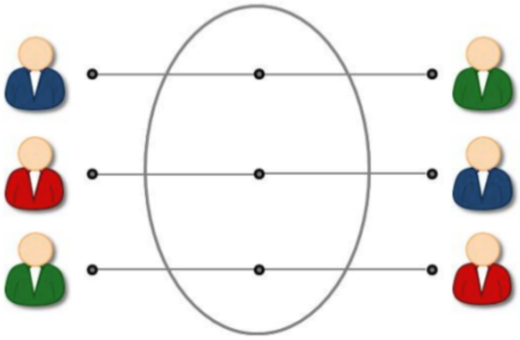

Figure 4. Mixing

3.1.1. Online wallets as mixes

As intermediaries user can use, for example, online wallets, services where user can store bitcoins online and withdraw them once they needed it. In general the coins that user gets back are not the same as the coins he deposited.

However there are several important limits to using online wallets for mixing. First, most online wallets don’t actually promise to mix users' funds, they just do it because it simplifies the engineering, and users have no guarantee that wallets won’t change their behavior.

Then, prudent wallet services will almost certainly maintain records that will allow them to link user deposit to user withdrawal. In addition to keeping logs, reputable services will also require and record user identity. User won’t be able to simply create an account with a username and password. So if the user privacy model does not accept the risk of the service provider tracking him, he can not use online wallets and look for dedicated mix service.

3.1.2. Dedicated mixing services

In contrast to online wallets, dedicated mixes promise not to keep records, nor do they require user identity. User sends his bitcoins to an address provided by the mix and he tells the mix a destination address to send bitcoins to. Hopefully the mix will soon send user other bitcoins at address he specified.

While it’s good that dedicated mixes promise not to keep records, people still have to *trust* them to keep that promise, and people have to *trust* them that they’ll send the coins back.

Since mixes are not a place where user store his bitcoins, unlike wallets, he will want his coins back quickly quickly. That means that the anonymity set, the pool of other coins that users' deposit will be mixed with, is much smaller: it contains only coins deposited at roughly the same time.

By the date researchers proposed a set of guidelines that allow to crucially improve the anonymity and security of mixing services.

Proposal: use a series of mixes

The first principle is to use a series of consecutive mixes, instead of just a single mix: the user sends the coin through various mixes, each time providing a freshly generated output address to the mix. If at least one of these mixes keeps the promise and destroys its records of the input to output address mapping, an adversary won’t be able to link the user’s original coin to their final one.

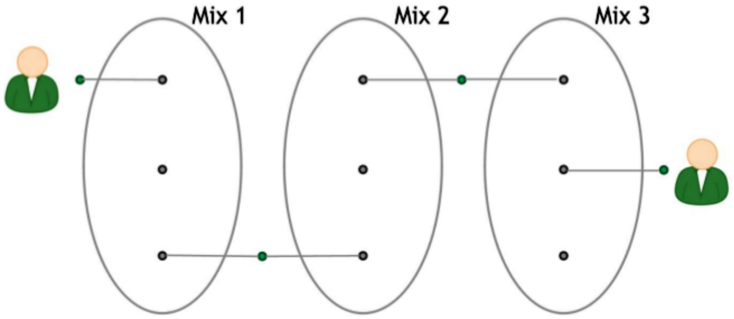

Figure 5. Series of mixes

Proposal: uniform transactions

If mix transactions by different users had different quantities of bitcoins, it will enable the attacker to link a user’s coins as they flow through the mix, or at least reduce the size of the anonymity set.

If mixed transactions are uniform in value, the linkability is minimized and the anonymity set is crucially increases. With this in mind all mixes just should agree on a standard chunk size for incoming mix transactions and all transactions going through any mix become indistinguishable based on their value. Moreover, having a uniform size across all mixes would make it easy to use a series of mixes without splitting or merging transactions.

In practice, it might be difficult to agree on a single chunk size that works for all users. Users might want to mix small as well as large amount of coins. In this perspective a reasonable trade-off between efficiency and privacy could be a series of two or three increasing chunk sizes that divide the large anonymity set in smaller ones but still of considerable size.

Proposal: automated client side
The client-side functionality for interacting with mixes should be automated and built into wallet software in a way to prevent the adversary to reveal sensitive information based on the timing of individual user transactions.

Proposal: all-or-nothing fees
The other vulnerability issue is the method users are charged for mixing service. If a mix charges fees taking a persentage of each transaction that users send in, mix transactions stop to be in standard chunk sizes.

The solution to the problem can be "all-or-nothing" probabilistic fees scheme: with a small probability (that means once in a while) the mix swallow the whole chunk. For example, if the mix wants to charge a 0.1% mixing fee, then one out every 1,000 times the mix should take the entire chunk, whereas 999 times out of 1,000 the mix should return the entire chunk without taking any mixing fee.

It not that simple as it seems, because mix, once a probabilistic decision is made, must convince the user that it does not cheat: that it does not bias its random number generator so that it has, say, a 1% probability of retaining a chunk as a fee, instead of 0.1%. Cryptography provides a way to do this and propose the various ways in which mixes can improve their trustworthiness.

3.2. Decentralized Mixing

Decentralized mixing is the idea of replacing mixing services with a peer-to-peer protocol. The protocol ensures that when users put in bitcoins to be mixed, they get bitcoins back of equal value, and thus makes the theft impossible. Users run this protocol independently of any dedicated service existence and do not need to trust a third party. This approach goes better with Bitcoin philosophy.

3.2.1. Protocol: Coinjoin

The main proposal for decentralized mixing is called Coinjoin. Group of users participate in just one round of mixing: each user supplies an input and output address and together they form a transaction with these addresses. The order of the input and output addresses is randomized so an outsider will be unable to link inputs and outputs. Participants check that their output address is included, that input, output and transaction fees sum up. When it is done, they independently sign the transaction.

Formally this protocol can be broken into 5 steps:

1. Find peers who want to mix
2. Exchange input/output addresses
3. Construct transaction
4. Send the transaction around: each peer signs after verifying their output is present.
5. Broadcast the transaction

Finding peers can be facilitated by servers acting as "watering-holes", allowing users to connect and grouping together.

To swap addresses in an unlinkable way, users should use an anonymous communication protocol (Tor network or a special-purpose anonymous routing protocol called a decryption mix-net[1]). The inputs and outputs communicated, one of the users constructs the transaction and passes it around, then each peer will validate it and sign. Any peer can assemble and broadcast the transaction.

Denial of service problem

If all peers follow the protocol, this system works. The problem is to prevent a possible denial-of-service attack, launched by one or several peers. The plan of denial-of-service attack might be as follows: a peer participates in the first phase of the protocol, providing its input and output addresses and then refuse to sign in the second phase.

There have been several proposals to exclude the DoS in Coinjoin, for example, to impose a cost of participation in the protocol, via a proof of work (analogous to mining), or by a proof of burn: a requirement to provably destroy a small quantity of bitcoins that participant owns.

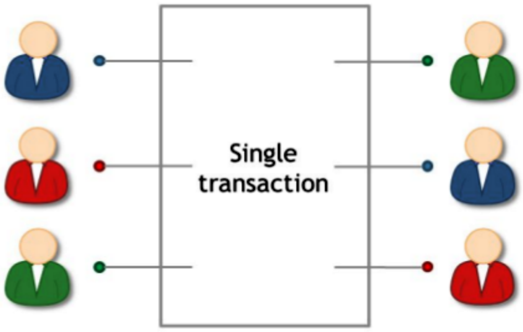

Figure 6. A Coinjoin transaction

3.2.2. Merge avoidance protocol

An alternative to coin join technique is the merge avoidance protocol, proposed by Bitcoin developer Mike Hearn.

This protocol allows the recipient of a payment to provide as many addresses to sender as necessary, the sender and receiver agree on a set of denominations to split the payment into multiple transactions (Figure 7). If the receivers (stores) run many merge-avoidance transactions, each of them can profit from increased anonymity set.

Figure 7. Merge avoidance. Alice wishes to buy a teapot for 8 BTC. The shop gives her two addresses and she pays 5 to one and 3 to the other, matching her available input funds. This prevents revealing that both addresses belong to Alice. To protect her privacy Alice avoids sending the two payments at the exact same time.

Merge avoidance can protect against address clustering techniques that rely on coins being spent jointly in a single transaction, to some extent it can soften the problem of high-level flows: an

[1] See https://en.wikipedia.org/wiki/Mix_network

adversary might not be able to discover a high-level payment pattern if it is split in many flows independent from each other.

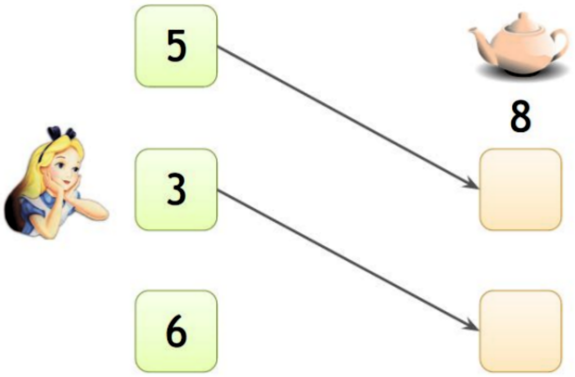

Figure 7. Merge avoidance.

3.3. Anonymity at the protocol level

All of the anonymity-improving techniques that have been seen so far add anonymity on top of the Bitcoin core protocol. But what can retain developers from trying to provide the strong anonymity guarantees on the core protocol level?

Zerocoin (2013), an extension to the bitcoin protocol, and Zcash (2016), a novel cryptocurrency, are two prominent examples of this type of solutions.

Both, incorporate protocol-level mixing, and the anonymity properties come with cryptographic guarantees, that means that promise of anonymity relies only on the adversary’s computational limits.

The drawback is the fact, that unlike centralized mixing and Coinjoin, Zerocoin as well as Zerocash are not compatible with Bitcoin. It is technically possible to deploy Zerocoin with a soft fork to Bitcoin, but the practical difficulties are serious enough to make this infeasible. With Zerocash, a fork is not even possible, and an altcoin is the only option.

3.3.1. Zerocoin

To explain Zerocoin, we’ll first introduce the concept of Basecoin. Basecoin is a Bitcoin-like altcoin, and Zerocoin is an extension of this altcoin[2]. The key feature that provides anonymity is that user can convert basecoins into zerocoins and back again, and when he does that, it breaks the link between the original basecoin and the new basecoin. In this system, Basecoin is the currency that people transact in, and Zerocoin just provides a mechanism to trade basecoins in for new ones that are unlinkable to the old ones.

User can view each zerocoin he own as a token that can be used to prove that ownership of a basecoin and made it unspendable. The proof does not reveal which exactly basecoin a person owns, it merely states that person owns a basecoin. User can later redeem this proof for a new

[2] Altcoins are cryptocurrencies other than Bitcoin. The majority of altcoins are forks of Bitcoin with small uninteresting changes.

basecoin by presenting this it to the miners. An analogy is entering a casino and exchanging some cash for poker chips: chips serve as proof that some cash was deposited.

To make this work in a cryptocurrency, these proofs have to be implemented cryptographically. It has to be assured that each proof can be used only once to redeem a basecoin.

Zero-knowledge proofs

Zerocoin is an extension of this Basecoin, that is is a Bitcoin-like altcoin[3].

Basecoins can be converted into zerocoins and back, and this convertion breaks the link between the original basecoin and the new one. So, Basecoin can be the currency that people transact in, and Zerocoin just give them a mechanism to trade their basecoins in for unlinkable ones.

Each zerocoin can be viewed as a token that proves that the owner posses a bitcoin.

This proof does not reveal which exactly basecoin users owns, it only assures that user can later redeem this proof for a new basecoin by presenting this proof to the miners. It's like a poker chips exchanged for the money.

Zerocoin relies on zero-knowledge proofs. In cryptography, a zero-knowledge proof is a method by which one party (the prover) can prove to another party (the verifier) that a given statement is true, without conveying any information apart from the fact that the statement is really true (Figure 8).

Each proof is used only once to get a basecoin, otherwise people could be able to obtain basecoins for free.

I know x such that H(x || 〈 other known inputs 〉) < 〈 target 〉

I know x such that H(x) belongs to the following set: {…}

Figure 8. Zero-proof toy examples. Suppose that Alice made a lot of work to solve a hash puzzle, she knows x and wants to convince someone of this, so makes this kind of statements, based on verifier knowledge of H(x) hash.

Minting Zerocoins

Zerocoins come in standard denominations, and anybody can mint a zerocoin.

Here is the receipt:

1. Generate serial number S and a random secret r
2. Compute Commit(S, r) , the commitment to the serial number
3. Publish the commitment onto the block chain as shown in Figure 9. This burns a basecoin, making it unspendable, and creates a Zerocoin. Keep S and r secret for now.

[3] Altcoins are cryptocurrencies other than Bitcoin.

A commitment scheme is the cryptographic analog of sealing a value in an envelope and putting it on a table in everyone's view. Zerocoin acquires value only when committed into the block chain.

In other words to put a zerocoin on the blockchain, user creates a 'mint' transaction whose output 'address' is the cryptographic commitment of the zerocoin's serial number, the input is a basecoin, which has now been spent in creating the zerocoin, the transaction does not reveal the serial number.

Figure 9. Committing to a serial number.

To spend a zerocoin and redeem a new basecoin, user has to prove that he previously minted a zerocoin. To do so he uses zero-knowledge proof of a type:

> "I know r such that Commit(S, r) is in the set {c1, c2,..., cn}",

he includes this proof along with serial number S in a special "spend" transaction, he makes miners to verify the proof and then check that serial number S has never previously been used.

Unlike a mint transaction, the spend transaction has no inputs, and no signature, a zero-knowledge proof establishes its validity instead. Once a zerocoin is spent, its serial number becomes public, since it is unique, zerocoin can be spend only once. Observe that r is kept secret: neither the mint nor the spend transaction reveals it.

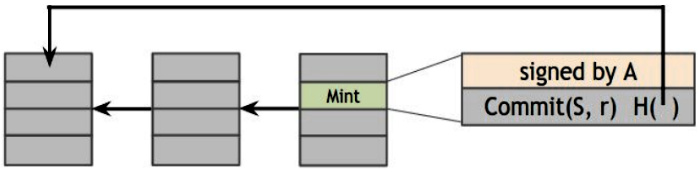

Figure 10. Putting a zerocoin on the block chain.

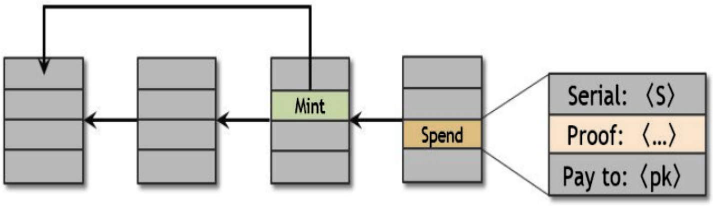

Figure 11. Spending a zerocoin.

3.3.2. Zerocash

Zerocash is an anonymous cryptocurrency. It uses a cryptographic technique called zero-knowledge SNARKs which is a way of making zero-knowledge proofs much more compact and efficient to verify.

Efficiency of Zerocash makes it possible to run the whole network without needing a basecoin, all transactions done in a zero-knowledge manner. The transaction amounts are now inside the commitments and no longer visible on the block chain. The cryptographic proofs ensure that the splitting and merging happens correctly and that users can’t create zerocash out of thin air.

The only thing that the ledger records publicly is the existence of these transactions, along with proofs that allow the miners to verify all the properties needed for the correct functioning of the system. Neither addresses nor values are revealed on the block chain at any point, the only users who need to know the amount of a transaction are the sender and the receiver of that particular transaction, the miners don't need to know it. Zerocash is immune to the side-channel attacks because the public ledger no longer contains transaction amounts.

4. Anonymization techniques: adoption criteria

Let’s now compare the anonymization techniques, both in terms of the anonymity properties that they provide and in terms of how deployable they are in practice.

Bitcoin, that is already deployed and popular, is only pseudonymous and vulnerable with respect to transaction graph analysis.

The better level of anonymity can be achieved using a single mix in a manual way, or doing a Coinjoin by finding peers manually. This obscures the link between input and output but leaves too many potential clues in the transaction graph. In addition, mixes and peers could be malicious or hacked.

4.1. Centralized mixing in practice

In practice there are many mix services out there, but they have low volumes and therefore small anonymity sets. Then, many mixes have been reported to steal bitcoins. Given the bad reputation of mixes, not many people want to use them, resulting in low transaction volumes and hence poor anonymity.

Anonymity loves company, that is, the more people using an anonymity service, the better anonymity it can provide.

Today’s mixes don’t follow any of the principles enumerated in the previous section. Each mix operates independently and typically provides a web interface, with which the user interacts manually to specify the receiving address and to choose the amount that he would like to mix. The mix takes a cut of every transaction as a mixing fee and sends the rest to the destination address. While far from perfect in terms of anonymity, mixing services exist and they are usable.

4.2. Decentralized mixing in practice

In August 2013, when mix services were fairly unattractive, Gregory Maxwell proposed Coinjoin as a trustless mixing. Since then, numerous other trustless mix services have been introduced, such as SharedCoin (sharedcoin.com). In November 2013 SharedCoin has been integrated into blockchain.info's popular wallet service.

The anonymity improvement comes from the fact that there's less reliance on any single mix or group of peers. Features like standardized chunk sizes and client-side automation minimize information leaks, but some side channels are still present. Wallets and services that implement a chain of mixes could be deployed and adopted today, but a secure mix-chain solution isn't yet readily available.

While toy implementations are not too hard to put together, robust real-world implementations with proper timeout handling, security checks, good wallet UI integration etc are a lot more effort. So far only blockchain.info has managed to create one (at sharedcoin.com), and people just have to trust that it doesn't keep logs. Otherwise anyone with the logs could unmix.

Perhaps the least discussed issue is user experience. A Coinjoin transaction requires other people to take part. The more people who take part, the better. But Bitcoin only peaks at about one transaction per second currently. That means users have to wait 10-15 seconds to get a good set of participants to mix with. That's just to start the protocol. Then those participants would all have to retrieve the candidate transaction and sign. If any time out, the whole thing has to start again. In poor conditions it could easily take a minute or more to complete this process, especially if some participants have flaky networks (say, phones) and are using Tor. Given that, performance seems seems to be a problem.

One might solve this problem by doing coinjoins in the background, unrelated to an actual spend that's taking place. That solves the problem of waiting in line at the coffee shop, but then fees must be paid on those transactions, it may be difficult to explain to people why their balance suddenly dropped overnight due to an unexpected privacy tax. That sort of nasty surprise would make Bitcoin rather unappealing to ordinary users. It also raises the question of when and how often it is done.

4.3. Zerocoin and Zerocash adoption problems

Overhead

Zerocoin bakes cryptography directly into the protocol and brings a mathematical guarantee of anonymity. However, Zerocoin would have to be launched as an altcoin. It is not compatible with Bitcoin.

The other reason Zerocoin is far from being adopted by the Bitcoin community is its performance. Bitcoin's decentralization already incurs a severe performance penalty compared to centralized payment systems such as Paypal. Achieving cryptographic privacy would further degrade performance. Recall the statement that's proved in a spend transaction:

"I know r such that Commit(S, r) is in the set {c, c2 ,...,cn }".

This sounds like it would be horribly inefficient to implement, because the size of the zeronowledge proofs would grow linearly as n increases, which is the number of zerocoins that have ever been minted. Remarkably, Zerocoin manages to make the size of these proofs only logarithmic in *n*. Note that even though the statement to be proved has a linear length, it doesn't need to be included along with the proof. The statement is implicit; it can be inferred by the miners since they know the set of all zerocoins on the block chain. The proof itself can be much shorter. Nevertheless, compared to Bitcoin, Zerocoin still adds quite a sizable overhead, with proofs about 50 kB in size.

Trusted setup requirement

The other problem that prevents adoption is a trusted setup. One of the cryptographic tools used in building Zerocoin is RSA accumulators which requires a one-time trusted setup. Specifically, a trusted party needs to choose two large primes p and q and publish $N=p\cdot q$ which is a parameter that everybody will use for the lifetime of the system. Think of N like a public key, except for all of Zerocoin as opposed to one particular entity. As long as the trusted party destroys any record of p and q, the system is believed to be secure. In particular, this rests on the widely-believed assumption that it's infeasible to factoring a number that's a product of two large primes. But if anyone knows the secret factors p and q (called the "trapdoor"), then they'd be able to create new zerocoins for themselves without being detected. So these secret inputs must be used once in generating the public parameters and then securely destroyed.

There's an interesting sociological problem here. It's not clear how an entity could choose N and convince everybody that they have securely destroyed the factors p and q that were used during the setup. There have been various proposals for how to achieve this, including "threshold cryptography" techniques that allow a set of delegates to jointly compute N in such a way that as long as any one of them deletes their secret inputs, the system will remain secure.

It's also possible to use a slightly different crystallographic construction to avoid the trusted setup. Specifically, it has been shown that simply generating a very large random value for N is secure with high probability, because the number probably cannot be completely factored. Unfortunately this carries a huge efficiency hit and is thus not considered practical.

And, finaly, Zerocash. Due to its improved efficiency, Zerocash can be run as a fully untraceable — and not just anonymous — cryptocurrency. However, like Zerocoin, Zerocash is not Bitcoin compatible, and it requires a complex setup process which the community is still figuring out how best to accomplish.

Just like Zerocoin, Zerocash requires "public parameters" to set up the zero-knowledge proof system. But unlike Zerocoin, which requires just one number N which is only a few hundred bytes, Zerocash requires an enormous set of public parameters — over a gigabyte long. Once again, to generate these public parameters, Zerocash requires random and secret inputs, and if anyone knows these secret inputs, it compromises the security of the system by enabling undetectable double-spends.

5. Conclusion

Anonymity demand for cryptocurrencies is an active area of technical innovation and debate.
This report covers a variety of privacy-enhancing technologies (Figure 12).

Despite of the fact that Bitcoin's anonymity is proved to be fragile and susceptible to to different kind of deanonymization attacks, it can be enhanced by the range of easy-to-be-implemented techniques. When "tuned", it still remains far away from the desired anonymity level.

Bitcoin community met "user-defined-privacy" challenge by developing decentralized cryptocurrencies like Zerocoin and Zerocash, unknown at the time of Bitcoin's release and approaching the anonymity problem in a revolutionary way. They wide adoption, however, faces the performance and trusted setup problems.

It is still uncertain which anonymity system for crypto currencies is going to become mainstream.

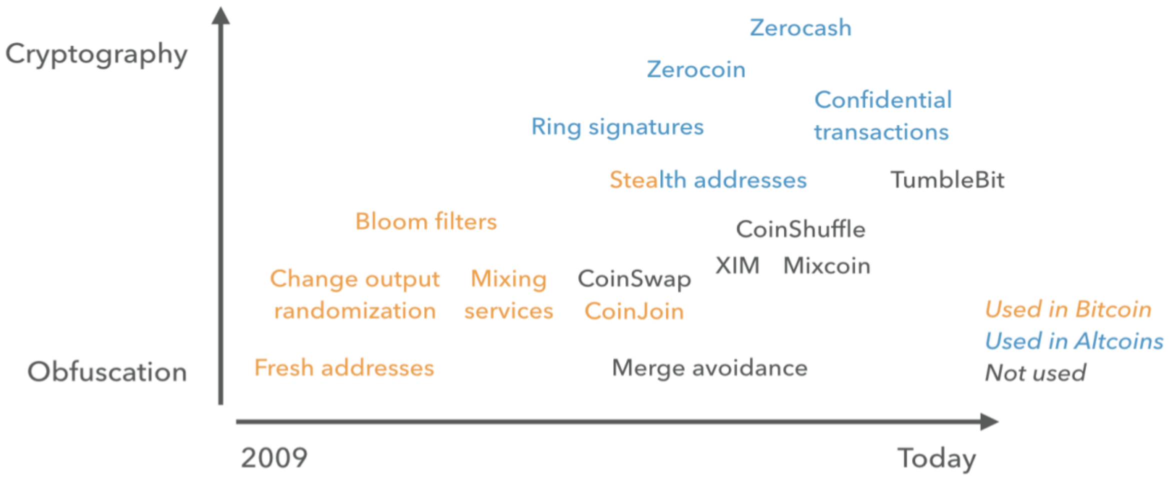

Figure 12: Privacy-Enhancing Technologies for Bitcoin.

6. Bibliography

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. http://www.bitcoin.org/bitcoin.pdf.

Garay J, Kiayias A, Leonardos N. (2015). The Bitcoin Backbone Protocol: Analysis and Applications. EUROCRYPT '15.

Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press.

Murck, P. (2013). Testimony of Patrick Murck General Counsel, the Bitcoin Foundation to the Senate Committee on Homeland Security and Governmental Affairs “Beyond Silk Road: Potential Risks, Threats, and Promises of Virtual Currencies”. Available online at https://www.hsgac.senate.gov/download/?id=4CD1FF12-312D-429F-AA41-1D77034EC5A8 (retrieved on 2017-06-02).

Brito, J. (2013). Beyond Silk Road: Potential Risks, Threats, and Promises of Virtual Currencies. Testimony to the Senate Committee on Homeland Security and Governmental Affairs. Available online at https://www.mercatus.org/system/files/Brito_BeyondSilkRoadBitcoin_testimony_111313.pdf), https://www.mercatus.org/publication/bitcoin-primer-policymakers

Maesa DdF, Ricci L. Anonimity mechanisms for digital currencies: a bitcoin perspective, (2015). Available at https://www.di.unipi.it/Documents/didattica/PhD/VerificheEsami/2016/Proposte/DiFrancescoMaesa.pdf

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., Savage, S. 2013. A fistful of Bitcoins: characterizing payments among men with no names.
In Internet Measurement Conference: 127-140;
https://www.usenix.org/system/files/login/articles/03_meiklejohn-online.pdf.

Miller A., Litton J., Pachulski A., Gupta N., Levin D., Spring N., and Bhattacharjee B. (2015). Discovering bitcoin’s public topology and influential nodes.

Biryukov, A., Khovratovich, D., Pustogarov, I. (2014). Deanonymisation of clients in Bitcoin P2P network. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014.

Narayanan, A., Möser, M. (2017). Obfuscation in Bitcoin: Techniques and Politics. Workshop on Obfuscation at NYU, April 2017. Preprint (http://randomwalker.info/publications/bitcoin-obfuscation-abstract.pdf, https://arxiv.org/pdf/1706.05432.pdf)

Bonneau, J., Narayanan, A., Miller, A., Clark, J., Kroll, J. A., & Felten, E. W. (2014). Mixcoin: Anonymity for Bitcoin with Accountable Mixes. In International Conference on Financial Cryptography and Data Security (pp. 486-504). Springer Berlin Heidelberg.

Miers, I., Garman, C., Green, M., & Rubin, A. D. (2013). Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In 2013 IEEE Symposium on Security and Privacy (S&P) (pp. 397-411). IEEE

Möser, M., & Böhme, R. (2017). Anonymous Alone? Measuring Bitcoin’s Second-Generation Anonymization Techniques. In IEEE Security & Privacy on the Blockchain (IEEE S&B). IEEE.

Ruffing, T., Moreno-Sanchez, P., & Kate, A. (2014). CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin. In European Symposium on Research in Computer Security (pp. 345-364). Springer International Publishing.

Miers, I., Garman, C., Green, M., Zerocoin, A. D. R. (2013). Zerocoin: Anonymous distributed e-cash from bitcoin. In IEEE Symposium on Security and Privacy, pp. 397-411, IEEE. 2013.

Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M. 2014.
Zerocash: decentralized anonymous payments from Bitcoin.
IEEE Symposium on Security and Privacy;
http://zerocash-project.org/media/pdf/zerocash-extended-20140518.pdf.

Ruffing T., Moreno-Sanchez P., Kate A. (2014) CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin. In: Kutyłowski M., Vaidya J. (eds) Computer Security - ESORICS 2014. ESORICS 2014. Lecture Notes in Computer Science, vol 8713. Springer, Cham