CSC2125: Homework #9

Due on November 26

Fan Long

Thomas Hollis

Problem 1

In Section 4.3 of the ByteCoin whitepaper, it mentions the linkable payment problem. Suppose if all users follow the practice of always creating new address when receiving payment and never reuse address. Would this problem go away?

Solution

No, the problem would not go away. It is not sufficient to always create new addresses and never reuse an address. As section 4.3 explains, Bitcoin works by users giving their address to whoever they need to receive payments from. However, even if all users follow the practice of creating a new address when receiving a payment and never reuse an address, this is still not good enough. You would also need to always transmit your address through a private channel (i.e. never tie your pseudonym to a particular address publicly). In addition, you would need to never spend those coins in large transactions. This is because as soon as you do this, all of the addresses that you control are joined together in one multi-input transaction, effectively linking all the addresses. If anyone knows your connection to either address they can conclude you controlled all the other addresses that are inputs to that transaction.

It is worth noting here that creating a new address every time is incredibly inconvenient (if not almost impossible) if you are receiving funds from many different people, many times per day (as a regular large business would). In addition, most businesses have a requirement of posting their Bitcoin address publicly (in a manner associated with their public username or company name) because they have no other way of letting new incoming users know where to pay them. This could lead to the connection of addresses simply by tracking the associated accounts used to post them online.

It is also worth noting that many 'anonymisation' schemes currently exist within the Bitcoin protocol. For example, some Bitcoin full-node users have created mixing addresses which take a large number of transactions from a variety of users, mixes them and sends these funds back out to new addresses for the input users. However, for this to truly be anonymous, users of the mixing node need to trust that the mixing node does not log transactions which defeats the entire purpose of a trust-free protocol like Bitcoin.

In fact, the above explanation is why Bitcoin is often referred to only as a pseudo-anonymous cryptocurrency rather than an anonymous one. It is this fact that has driven the development of privacy-focussed cryptocurrencies like Monero and Zcash.

Problem 2

According to Section 4.4 of the ByteCoin whitepaper, how does the ring signature algorithm prevent an adversary from tracing transactions?

Solution

Before we explain ring signatures we need to understand exactly what is required to prevent an adversary from tracing transactions. In Monero (like in ByteCoin), preventing adversaries from tracing transactions is achieved by protecting the anonymity for both the receiving party (using "stealth addresses") and the sending party (using "ring signatures").

To maintain anonymity for the receiving party, Monero uses "stealth addresses" which are implemented by using one-time public keys. These one-time public keys can only be spent by the recipient and only the recipient can detect their designated output on the blockchain. Since all outputs are unlikable, the privacy of the receiver is secured.

However, we also need to ensure privacy of the sending party to have full anonymity. This is achieved in Monero (like in ByteCoin) by using "ring signatures", as explained in section 4.4 of the whitepaper. Indeed, one-time ring signatures prevent adversaries from tracing transactions by achieving unconditional unlinkability. This is achieved by a simple protocol where a user produces a signature that is verifiable by a set of public keys (rather than just one) forming a "ring" of cryptographic signatures. This way, the identity of the signer is indistinguishable from the identities of the other users whose public keys also belong to the set. In this case, the total number n of users that are used to achieve a ring signature is known as the ring size. In Monero, $n = 5$ which means that for each ring signature, a total of 5 signatures are need (including the sending party).

From a high-level perspective, say Alice wants to send a transaction to Bob. Alice will sign the transaction she wants to send, using a ring signature with 4 other Monero users who will themselves sign decoy transactions to complete the ring. Indeed, there is no way to distinguish which 4 users signed the decoy transactions (which are actually past transactions that are already known to the network) and which single user signed the only valid transaction in the ring. Therefore, the original signer of the only valid transaction of the ring remains anonymous.

However, this introduces the issue of smuggling a double spending inside the ring-signed transaction. Since nobody knows who the author of the valid transaction is, we need a mechanism to make sure that this user is not double spending. This is problem solved by using "key images" as part of the ring signature. Key images are derived from outputs being spent for every ring signature transaction. There is only one key image for each output and since all key images cannot be mapped to a particular output, key images cannot produce linkability. However, key images can be used by miners to verify that no output is sent twice.

Therefore, we can use key images combined with ring signatures to ensure the sending party is anonymous while we can use stealth addresses to ensure the receiving party is anonymous. This is how adversaries are prevented from tracing transactions.

It is interesting to note here that, in Monero, even the amount of XMR of the transaction can be obfuscated by using Ring Confidential Transactions (a.k.a Ring CT) but this is out of the scope of this question.

Problem 3

Monero chooses a ring size of five. Do you think this size is large enough (Open question)?

Solution

Let's approach this question with the most generic example possible, by supposing that the ring size is set to a number n.

Let's say Alice is a Monero user and has just obtained some XMR from an exchange (assume she received it by trading it with another cryptocurrency). Let us also suppose that Eve, an enemy of Alice, is trying to track Alice's behaviour on the Monero blockchain from this very first transaction.

As soon as Alice takes part in her first ring signature transaction, Eve knows that there are approximately $1/n$ odds that Alice is the author of the valid transaction. This is true for all subsequent ring signatures that Alice is a part of. This creates a probability tree which splits into n branches for each ring signature participation. If n is ridiculously low (e.g. 2), Eve could try to follow all n branches of Alice's movements until Eve receives a known transaction from Alice (payment for a service for example). If this is the case Eve has better chances of reverse engineering Alices path for low values of n than if n is very high (e.g. 100). Therefore, we can state that a low n requires a larger number of transactions to provide a user with anonymity. Conversely a high n provides faster anonymity as less transactions are required before anonymity is reached by users.

However, by increasing the ring size we also increase the transaction size. A typical Monero transaction with a ring size of 5 would be approximately 13'500 B while a ring size of 7 would lead to transaction sizes of approximately 14'000 B. Monero transactions are, as it is, already larger than Bitcoin transactions. This is already controversial as it is with many people saying Monero scales less well than Bitcoin so increasing the ring size would only aggravate this. To make matters worse, a larger ring size means a longer transaction verification time. It is estimated that increasing the ring size from 5 to 7 would cause the verification time to rise by around 5% from 70ms to 74ms. Indeed, many forks have been attempted to increase the ring size to 7 or more. In addition, many forks were also attempted to allow for flexible ring sizes.

In this efficiency-privacy trade-off, I am personally of the opinion that we should opt for a more efficient and less private ring size of 5. My main rationale behind this is that privacy in Monero is still far from perfect. Indeed, Monero is not as untraceable as it may seem, as shown in a recent 2018 paper from MIT (`https://arxiv.org/pdf/1704.04299/`). This shows that traceability is still somewhat brittle in Monero and obsessing over a larger ring size is futile when other aspects of the Monero protocol still need to be refined to increase privacy. In addition, I am against the use of variable ring sizes as they reduce the level of anonymity. Indeed, a user's choice of ring size gives further metadata that can be used for fingerprinting and de-anonymisation.

Problem 4

Many people believe that zkSNARK/Zcash provides a much stronger privacy guarantee than Monero? Why? Explain a case where Monero can be traced but ZCash cannot.

Solution

For the sake of this question, let us assume both Monero's ring signatures and Zcash's zkSNARK are perfectly implemented. Many people do indeed believe that zkSNARK provides a stronger privacy guarantee than Monero. To explain this, we must first explain the overview of what a zkSNARK is and why it is considered by some to have stronger privacy than Monero.

A zkSNARK (zero knowledge SNARK) is an algorithm that allows the verification of a computation without doing the computation itself. Indeed, zkSNARK adds an extra overhead to achieve this but the hope is that this overhead will be counteracted by its improved efficiency hopefully allowing better scalability in the long term. Inside this verification, zkSNARK is particularly useful as it allows for "zero knowledge" proofs which is why it is the main technique used in Zcash to achieve privacy.

Zero knowledge is a property whereby a prover can convince a verifier that a certain statement is true without revealing *why* it is true (hence the name zero knowledge). Therefore, in Zcash we can approve transactions without concealing sending party, receiving party and amount since zero knowledge is leaked.

On the other hand, Monero's ring signature still shows that a sender is part of a group of users, leaking some information about the involved party.

One easy example to show this would be a case where Alice sends money to Bob, while Trudy tries to de-anonymise Alice in the transaction. In Zcash, zkSNARK prevents any information from leaking (since it is zero knowledge) therefore Trudy cannot de-anonymise Alice. However, in Monero, Trudy could control/know all 4 anonymising nodes that are part of the ring signature. Since she knows who controls these nodes and knows none of them are the node issuing the real transaction, Trudy could successfully conclude Alice is the signer of the real transaction. Trudy has effectively de-anonymised Alice. Therefore, this is the reason why Zcash is said to provide a much stronger privacy guarantee than Monero.

However, it is worth mentioning that Monero allows privacy by default while Zcash does not. Therefore, the debate of which cryptocurrency is "more private" is not a straightforward one, as it depends if we talk about privacy for an individual transaction (as discussed in this question) or privacy for the average user in the network.

Problem 5

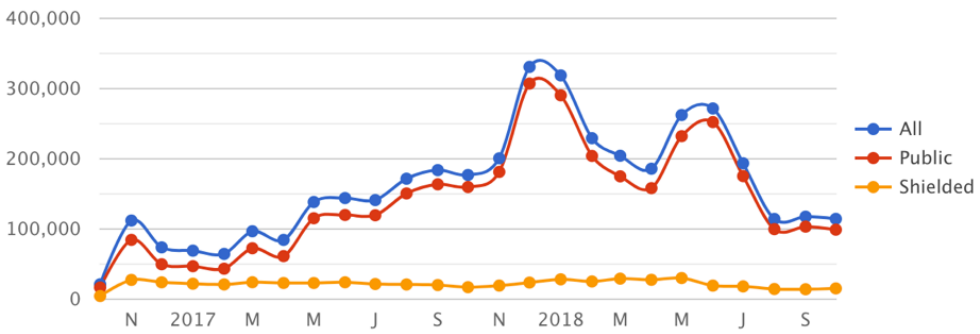
Now a significant amount of transactions in Zcash is transparent transactions. That means that although ZCash is marketed for its strong privacy, the privacy functionality of ZCash is rarely used. Why (Hint: look numbers in Section 7 in ZCash white paper)?

Solution

As section 7 of the Zcash whitepaper shows, anonymity in Zcash is optional due to its high computational cost. Indeed, while the numbers in the whitepaper are somewhat out of date, computational cost of the privacy functionality of Zcash remains high.

It is estimated that anonymising a transaction requires an extra 3GB of RAM and up to 40s on modern machines (2018). Indeed, Zcash is fundamentally a Bitcoin fork with similar supply, slightly faster block time target (2.5min) and privacy protection using zkSNARK. As a consequence, users cannot simply choose to anonymise all their transactions by default as this would not be scalable or sustainable. Only the transactions that need to be anonymised are worth the computational burden, which is why a significant amount of transactions in Zcash are transparent. Interestingly, many Zcash wallets do not even support anonymity at all for this exact reason.

However, there is some encouraging light at the end of the tunnel. In September 2017, Zcash announced a major upgrade (codenamed Sapling) which features a new elliptic curve algorithm named Jubjub (based on the BLS12-381 curve). This algorithm used inside zkSNARK ultimately helped reduce the computational burden of anonymity in Zcash. This was not enough however, as shown by the following graph of Zcash transactions by type:



Zcash transactions by type (source: `https://blockspur.com/zcash/trends/transactions_count`)