Concordia Institute for Information System Engineering

(CIISE)

INSE 6120

Cryptographic Protocols and Network Security

Project Report

MONERO AND ZCASH

Submitted to:

Professor Dr. Ivan Pustogarov

Submitted By:

Student Name	Student ID
Arun Prasad Karunanithi	40220964
Raghavendran Raghunathan	40220965
Naveen Sesetti	40206610
Meetsinh Parihar	40217262
Gayatri Tangudu	40221830
Priya Meghana Raavi	40221227
Bhavya Panner Selvam	40205936

Table of Contents

ABSTRACT- While Bitcoin only provides a little amount of privacy, numerous other cryptocurrencies have developed that use zk-SNARKs and other privacy-enhancing cryptographic techniques. Two of the most well-known cryptocurrencies that employ these methods ,Monero and Zcash are referred. In order to directly compare Monero and Zcash and to determine the various cryptographic primitives employed to achieve anonymity in each, research has been done. The study demonstrates that the use of privacy-enhancing cryptographic primitives imposes new restrictions on the protocols' efficiency and usability. These restrictions are analysed and critically assessed.

1. INTRODUCTION

The introduction of Blockchain technology has opened the door to the potential of decentralised currency. A agreement based ledger may be controlled and verified by the entire network thanks to cryptocurrencies like Bitcoin. This presents the issue that user privacy is not protected because funds can be viewed and watched by anyone. By offering pseudonymous addresses, which separate the address from the actual owner of the funds, Bitcoin makes an effort to alleviate this problem. Bitcoin offers minimal anonymity because it has been demonstrated that transactions can be traced back to its users . Other cryptocurrencies have appeared in response, offering a solution to the privacy issue. Monero and Zcash are two of the most well-known cryptocurrencies that use various cryptographic primitives to maintain privacy.

It is frequently required to take into account the trade-offs between privacy, efficiency, and usability in cryptography applications. The same is valid for programmes like Monero and Zcash. In order to promote anonymity, both cryptocurrencies use various cryptographic primitives, with varying effects on their effectiveness and usability. The merits and demerits of the Monero and Zcash protocols will be evaluated in this paper. In addition to a direct comparison, modifications to the Monero protocol will be proposed and put to the test in light of the discovered drawbacks.

2. PROTOCOL DESCRIPTION

2.1. MONERO

Based on a paper by Van Saberhagen (2013) that explains the CryptoNote technology tells how to safeguard the confidentiality of the sender and recipient, respectively, CryptoNote uses ring signatures and stealth addresses. Rivest, et al. invented ring signatures first (2001). The group signature, which Chaum & Van Heyst originally developed, is a subtype of the ring signature (1991). The traceable ring signature is based on the work of Fujisaki& Suzuki, and the ring signature used by CryptoNote is a variation of it (2007).According to research, the CryptoNote system includes flaws that make it possible to de-anonymize transactions. Noether & Mackenzie (2014) illustrate how CryptoNote can be made anonymous also by tracking back already-spent mixins. In a different investigation, Miller (2017) found that 63% of Monero Cryptonote transactions can be tracked and further transactions may be turned anonymous again using zero-mixin transactions. Kumar, et al. conducted a similar investigation on the tracability of transactions in Monero (2017). A study by Wijaya (2018)concluded that there are two different kinds such attacks that can be used to lessen the level of anonymity established in the ring signature.

The RingCT protocol was developed by Monero in response to the vulnerabilities that had been examined. Niether & Mackenzie wrote the RingCT whitepaper (2016) [1] . There are two primary ways that this protocol varies from CryptoNote. RingCT uses Multi- Layered Linkable Spontaneous Anonymous Group signatures, a variation of the linkable ring signatures developed by Liu, et al., in replacement of traceable ring signatures (2004). Also, they use Confidential Transactions, which Maxwell first popularized (2015).Studies have been conducted to create ring signatures that have a constant value, meaning that their size does not rise linearly with the amount of mixins they contain. Several works outline an approach for producing short linkable ring signatures, also known as linkable ring signatures of constant size.

2.1.1. How it works ?

Monero relies on two key ideas which are stealth addresses and ring signature to offer privacy and anonymity.

2.1.1.1. Ring Singnature

Rivest, et al. (2001) first described ring signatures in their paper how to Leak a Secret. They explain a case when a cabinet member wants to disclose a secret but doesn't want to reveal his identity. Additionally, he demonstrates his membership in the cabinet to show his reliability as a source. The proposed method is for the member of cabinet to produce a signature that contains both his private key and the public keys of the other cabinet members.

Given a message m, the signer's secret key S , and the public keys P ,P ,...,P of all the ring $s12r$ members, the ring signature is computed by the signer as follows (Rivest, et al., 2001).

1. The signer computes the symmetric key k as the hash of the message m:k = h(m)
2. The signer picks an initialization value v uniformly at random from $\{0,1\}^b$
3. The signer picks random x_i for all the other ring members, uniformly and independentlyfrom $\{0,1\}^b$

$$y_i = g_i(x_i)$$

4. The signer computes the unique value for y_s and solves the following ring equation for y_s:

$$C_{k,v}(y_1,y_2,\ldots,y_r) = v$$

5. The signer obtains x_s by inverting g_s on y_s using this knowledge of trap-door:

$$X_s = g_s^{-1}(y_s)$$

6. The signature on message m is defined as:

$$(P_1,P_2,\ldots,P_r;\ v;\ x_1,x_2,\ldots\ldots,x_r)$$

The combination function C (y , y , ..., y) is a symmetric encryption function E applied to k,v12 r k the sequence (y , y , ..., y) as follows: 12r

C (y,y,...,y)=E(y⊕E(y ⊕E(y ⊕E(...⊕E(y⊕v)...))) k,v 1 2 r k r k r−1 k r−2 k k 1

The resulting ring signature can be seen in Figure 1.

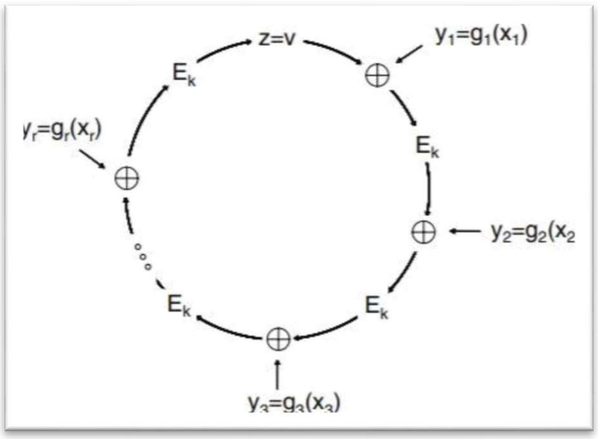


Figure 1. Ring signature generation

A verifier can verify a signature (, P , ... , P ; v; x , x , ..., x) on a message m as follows. $12r12r$

1. For i= 1,2,… .... ,r the verifier computes:

$y_i = g_i(x_i)$

2. The verifier hashes the message to obtain k:K=h(m)

3. The verifier checks that the y_i's satisfy:

$C\,(y, y, \ldots, y) = v$

4. If the ring signature is satisfied, the verifier accepts the signature as valid. Otherwise thesignature is rejected.

It's important to recognize that a group signature offered by Chaum & Van Heyst varies from a ring signature (1991). The fundamental distinction is that a group manager controls the group of signers in a group signature and distributes keys to the group members. The unit can then be defined by each member providing a signature. Furthermore, the group manager can also revoke anonymity and reveal who has signed the paperwork in real life. Another distinction to be made isthat the group of signers is specified in the group signature method and that they all agree to collaborate. The ring signature, which is on the other hand, can be created by a signer without theapproval or knowledge of the individuals who are a part of it. The ring signature may now be created dynamically, which is an important discovery for the blockchain.

2.1.1.2. Stealth Addresses

The idea that the group of signers is recognized in the group signing method and that they all consent to working together is another distinction to be made. On the other hand, a signer can create a ring signature without the knowledge or authorization of the others who are a part of it [2] . This is a major development for the blockchain since it allows for the dynamic creation of the ring signature.

1. A sender Alice wants to send a transaction to Bob, whose public key is of the form ($V = vG$, $B = bG$).
2. Alice generates a random r and computes $R = rG$
3. Alice computes one-time public key $P = H_s(rV)G + B$
4. Alice publishes R, P
5. Given R Bob computes $P' = (vR) + B$
6. Bob checks whether $P' = P$. If not, the transaction is not meant for him and he continues toscan the ledger.
7. If yes, the transaction is meant for him and Bob computes corresponding one-time privatekey $x = H_s(vR) + b$ such that $P = xG$
8. Bob can now use x to send the transaction to a new address

Always remember that Bob can give someone else access to his private viewing key so they can scan the ledger on his behalf. Unfortunately, they are unable to spend the transaction delivered to Bob without the secret spending key. For example, auditing purposes, this feature is helpful.

2.2 ZCash

Zk-SNARKs, or zero knowledge succinct arguments based on knowledge, are the technologies that Zcash uses and were first presented by Bitansky et al. (2012). The idea of non-interactive zeroknowledge proofs was introduced by Miers et al. (2013) to present a fresh idea for an addition to Bitcoin dubbed Zerocoin. The fact that the proofs of knowledge utilized by Zerocoin cannot scale,which prevents them from providing a practical answer to the privacy issue that can be used in practical apps, is a significant drawback.

In reaction, Sasson et al. (2014) introduced Zerocash, which employs zk-SNARKs, in a paper andimplemented it after making some modifications to create what is now known as Zcash. The zk- SNARKs require the trusted setup that Zcash employs in order to function. According to Bowe, etal., the multi-party protocol known as the Pinnochio protocol is used to execute the trusted setup.(2017).

The protocol's potential flaws have not received much attention due to the novelty of zk-SNARKsand the Zcash cryptocurrency. Studies by Kappos, et al. (2018) , however, demonstrate that it is feasible to differentiate between transactions made by miners, founders, and"normal" users, highlighting potential traceability problems.

2.2.1. How it Works ?

Zcash is a digital currency that protects privacy and offers cutting-edge security. It was created utilising sophisticated math and computer science. Users may transfer transactions over the network quickly, easily, and for a small price thanks to Zcash. Users can benefit from Zcash's protected transactions if they desire more anonymity. Users' private information can be protected by using shielded transactions.

2.2.1.1. Zk – SNARKs

The zk-SNARK (zero knowledge-Succinct Non-interactive Arguments of Knowledge), a category of zero knowledge proofs, is the main approach used in Zcash. Traditional zero knowledge proofs require the prover and the verifier interacting, with the verifier eventually deciding to accept or deny the prover's final proof. Because of their extreme inefficiency, these proof types are not suitable for blockchain apps like Zcash.

1. $Key(1^{\lambda}, C) \rightarrow (pk, vk)$. Given a security parameter λ and an F-arithmetic circuit C, $KeyGen$ probabilistically samples a proving key pk and a verification key vk. The keys are published as public parameters and can be used, any number of times, to prove/verify membership in L_c.
2. $Pr(pk, x, a) \rightarrow \pi$. Given a proving key pk and any $(x, a) \in$, $Prove$ outputs a non-interactive proof π for the statement $x \in L_c$.
3. $Ver(vk, x, \pi) \rightarrow b$. Given a verification key vk, an input x and a proof π, $Verify$ outputs $b = 1$ if the verifier is convinced that $x \in L_c$.

3. COMPARISION OF MONERO AND ZCASH

3.1. Anonymity set:

Zcash and Monero are two of the most popular privacy-focused cryptocurrencies, and both aim to provide high levels of anonymity to their users. However, they differ in how they achieve this goal and the size of their anonymity sets. In Monero, the anonymity set is determined by the number of mixins used to construct the ring signature, which is currently set to a minimum of 7. This means that each transaction in Monero is mixed with at least 7 other transactions to make it difficult to determine the true source of the transaction. However, if a transaction is sent with zero mixins, it can be easily de-anonymized, reducing the anonymity set of subsequent transactions that use it as a decoy.

On the other hand, Zcash has a much larger anonymity set because all shielded transactions are mixed together. This anonymity set includes all the transactions that have ever been shielded on the Zcash network, which is considerably larger than Monero’s anonymity set shown in figure 2. However, the caveat is that shielded transactions are not widely used on the Zcash network. The majority of transactions on the Zcash network are transparent, which means they do not provide any privacy or anonymity.

Overall, Zcash has the potential to provide greater anonymity than Monero due to its larger anonymity set. However, this potential is limited by the low usage of shielded transactions on theZcash network [3] . Meanwhile, Monero's anonymity set is more limited, but its ring signature construction ensures that each transaction is mixed with a minimum of 7 other transactions, providing a higher level of anonymity for each transaction.

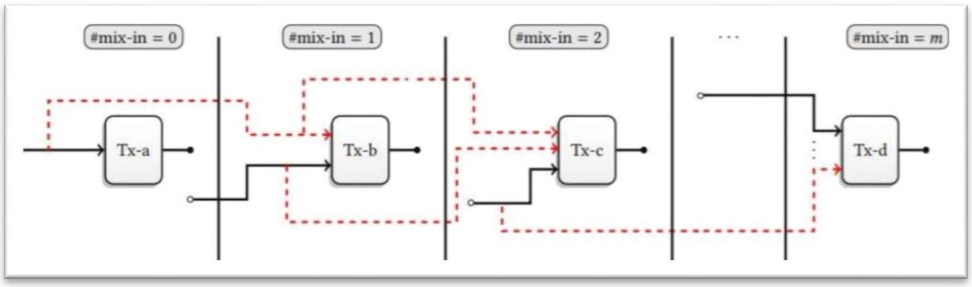


Figure 2. Illustration of the de-anonymisation attack. The dashed lines are inputs identified as decoy mixins, the bold lines are thereal input.

3.2. Opt-in Privacy

Overall, both Monero and Zcash provide privacy features that allow for anonymous transactions.However, there are some key differences between the two. In terms of anonymity set, Monero has a fixed minimum ring size of 7, which means that all transactions are anonymous to a certain degree. Zcash, on the other hand, has a larger anonymity set because it includes all shielded transactions, but the small amount of shielded transactions carried out in Zcash means that the anonymity set is not as large as expected as shown in Figure 3.

Another difference is that Monero mandates the use of RingCT and a minimum ring size, while Zcash allows for optional privacy features. This means that users in Monero have a higher degree of privacy than in Zcash, but the usability of Monero is limited due to the transaction fee increasewith the expansion of the ring size. In conclusion, both Monero and Zcash have their strengths and weaknesses when it comes to privacy features. Monero has a fixed minimum ring size and mandates the use of RingCT, which ensures a certain degree of anonymity, while Zcash has a larger anonymity set but limited usage of its privacy features. It ultimately depends on the user's preferences and needs when it comes choosing which privacy features to use.

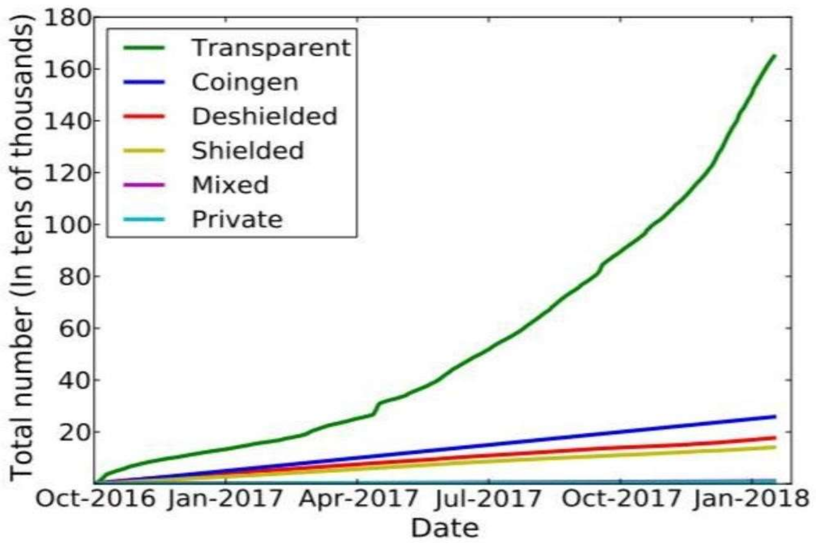


Figure 3. Total number of different types of transactions over time

3.3. Transaction Size

Another notable difference between Zcash and Monero is the cryptographic technology they use to achieve privacy. Monero uses Ring Confidential Transactions (RingCT), which obscures the transaction amount, while Zcash uses zk-SNARKs, which allows for the transaction amount to remain hidden as well as the transaction inputs and outputs. The use of zk-SNARKs in Zcash allows for greater privacy, as it completely shields the transaction inputs and outputs from publicview, whereas in Monero, while the transaction amount is obscured, the transaction inputs and outputs are still visible.

However, the use of zk-SNARKs in Zcash has some trade-offs. The use of this technology requiresa trusted setup, which is a potential point of weakness in the system. In addition, the use of zk- SNARKs can be computationally intensive, requiring more resources and time to verify transactions. Monero's RingCT, on the other hand, does not require a trusted setup and is generally considered to be more lightweight in terms of computational resources. Overall, both Zcash and Monero provide strong privacy features and are among the most private cryptocurrencies available. However, they differ in the size of the anonymity set, transaction size,and the cryptographic technology used to achieve privacy.

3.4. Trusted setup

Another notable difference between Zcash and Monero is the cryptographic technology they use to achieve privacy. Monero uses Ring Confidential Transactions (RingCT), which obscures the transaction amount, while Zcash uses zk-SNARKs, which allows for the transaction amount to remain hidden as well as the transaction inputs and outputs. The use of zk-SNARKs in Zcash allows for greater privacy, as it shields the transaction inputs and outputs from public view, whereas in Monero, while the transaction amount is obscured, the transaction inputs and outputs are still visible. However, the use of zk-SNARKs in Zcash has some trade-offs. The use of this technology requires a setup, which is a potential point of weakness in the system. In addition, the use of zk-SNARKs can be computationally intensive, requiring more resources and time to verify transactions. Monero's RingCT, onthe other hand, does not require a trusted setup and is generally considered to be more lightweight in terms of computational resources. Overall, both Zcash and Monero provide strong privacy features and are among the most private cryptocurrencies available. However, they differ in the size of the anonymity set, transaction size, and the cryptographic technology used to achieve privacy.

Comparative analysis between zcash and monero are the zk-SNARKs used in Zcash require a trusted setup to generate the public parameters that are used to create and verify proofs. Six different parties, each holding a part of the master secret, participated in the setup. Each party completed their portion of the protocol by using their part of the master secret, which finally produces the public parameters. After the parameter generation is finished it is important for each party to discard their fragment of the master secret. It is also important to note that all parts of the master secret are needed for this to happen. This means that even if only one party properly destroyed their part of the secret the reassembly of the master secret would not be possible. This is an important topic of debate within the cryptocurrency community, because a substantial amount of people does not trust that the fragments of the master secret were destroyed. This could also be one of the reasons why adoption of Zcash is so limited.

To summarize, while Zcash uses zk-SNARKs to provide privacy, which requires a trusted setup that some people in the cryptocurrency community do not trust due to the risk of collusion, Monero does not require trusted setup, thus avoiding this issue. However, Zcash allows for smaller transaction sizes and has a larger anonymity set than Monero, but the opt-in privacy feature and limited usability may reduce the anonymity provided by Zcash. Monero, on the other hand, requires mandatory RingCT usage and a minimum ring size of 7, but the transaction fees increase with more mixins, leading to a smaller anonymity set. Ultimately, the choice between Zcash and Monero depends on a user's priorities and concerns regarding pay, transaction fees, and trust in the system's setup.

3.5. Traceabiltity Analysis

The text focuses on the traceability of transactions in Monero and Zcash. While the actual cryptographic primitives used in both cryptocurrencies are not susceptible to de-anonymisation, there have been several studies into the traceability of transactions.In the case of Monero, the earlier versions of the cryptocurrency did not enforce a mixin minimum,which allowed for 0-mixin transactions that could be used to further de-anonymise other transactions that included them as mixins. However, Monero has since updated to include RingCT,which means that none of the inputs that were de-anonymised by previous research pose a threat to the anonymity of current Monero transactions. Nevertheless, Miller et al. (2017) found that thealgorithm used for sampling inputs when creating the ring signature is not representative of the spend-time distribution of Monero transactions. They were able to correctly guess that the newest input is also the real input 81% of the time as shown in figure 4.

It is presented a different heuristic, which assumes that it is possible to guess the real inputs of a transaction that takes at least two real inputs, if those inputs were both outputs in the same earlier transaction.

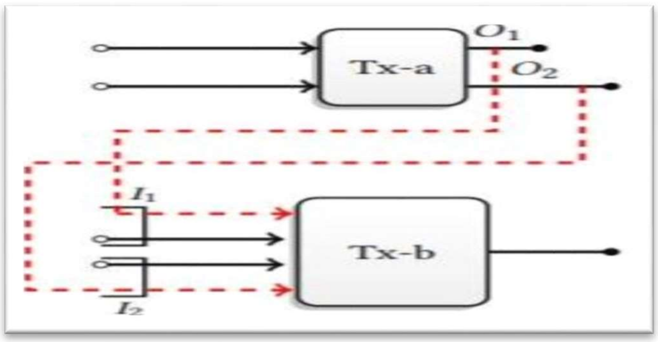


Figure 4. Heuristic Flow

Heuristic is slightly weaker because it allows for false positives, showed that it is also possible for an active attacker to reduce the anonymityof other transactions. The attacker creates a set number of inputs in the "setup phase" and spends them in a transaction. In the case of Zcash, Kappos et al. (2018) showed that there are recognisable patterns in the transactions going into and coming out of the shielded pool. Based on these patterns, they were able to identify transactions made by founders and miners and reduce the size of the overall anonymity set by 69.1%. The pattern of founder transactions showed that founders do not often put money into the pool, but when they do they transfer large amounts. It was also found that there was only ever one founder address "active" at a given moment. They also found that the deposited amount was usually always the same. The patterns of miners showed that they are the main actors transferring money to the pool.

4. Similarities between Monero and Zcash

4.1. Privacy

Both Monero and Zcash are made to offer their users a high level of privacy and anonymity. Ring signatures, stealth addresses, and confidential transactions are all used by Monero to conceal transaction data and shield user identities. Ring signatures make it impossible for outsiders to identify the sender of a message by allowing a number of users to sign it without exposing which user signed it. Stealth addresses are one-time addresses created specifically for each transaction and are intended to conceal the identity of the recipient. Confidential transactions conceal the amounts of Monero being sent using cryptographic methods.

In contrast, Zcash uses a concept called zero-knowledge proofs to let users demonstrate their knowledge of a specific piece of information without actually disclosing it. A sort of proof known as zero-knowledge proof enables one party to demonstrate to another that they are aware of a certain piece of information without disclosing what that information is. This enables Zcash users to demonstrate that they own a specific quantity of Zcash without disclosing their balance to third parties [4] . Zcash also provides transactions that are both transparent and protected, Whereas shielded transactions utilize zero-knowledge proofs to conceal the transaction details, transparent transactions are comparable to Bitcoin transactions in that the transaction details are accessible on the blockchain.

Ultimately, Monero and Zcash's privacy features aim to shield user identities and transaction data from prying eyes, making them appealing to users who value their security and privacy.

4.2. Decentralization

Decentralized cryptocurrencies like Monero and Zcash work on a peer-to-peer network without the aid of a central organization or middleman. This indicates that business can be done directly between users without a middleman handling the exchange. Both cryptocurrencies instead depend on a network of nodes to verify transactions and uphold the consistency of the blockchain. In the case of Monero, several volunteer nodes that operate the Monero software and verify transactions maintain the network. These distributed nodes are rewarded for the upkeep of the network through the mining procedure, which resolves challenging mathematical puzzles in order to add new blocks to the blockchain. The Monero network is decentralized, which makes the cryptocurrency more secure and trustworthy than conventional financial systems and makes it resistant to censorship and control.

Similar to Bitcoin, Zcash likewise runs on a node-maintained decentralized peer-to-peer network. Zcash uses a consensus algorithm called Equihash that is intended to be memory-hard and resistant to ASIC-based mining, in contrast to Monero. As a result, it is harder for powerful mining operations to control the network, ensuring that it remains decentralized and open to a larger range of users. Ultimately, Monero and Zcash's decentralization makes cryptocurrencies more safe and more dependable than conventional financial systems and increases their resistance to censorship and control. The peer-to-peer network of nodes that upholds the blockchain's integrity also contributes to ensuring that users can conduct transactions directly with one another without the need for a central authority or middleman.

4.3. Open-Source

Considering that Monero and Zcash are both open-source cryptocurrencies, anybody can see, alter, and contribute to their source code. Being able to look at the code and confirm that it is working as intended promotes a culture of accountability and transparency among the community. It also permits the development of third-party wallets and other computer programmes that can communicate with cryptocurrencies. The codebase for Monero is open-source and accessible on GitHub. The codebase must be maintained and improved by the developer community, and anyone can help with the creation of the coin. Due to Monero's open-source design, third-party wallets and other programmes that can communicate with the coin are also possible.

Similar to Bitcoin, Zcash has an open-source codebase that is accessible on GitHub. The codebase must be maintained and improved by the developer community, and anyone can help with the creation of the coin. Zcash's open-source design makes it possible to develop third-party wallets and other types of software that can communicate with the coin. Furthermore, because both Monero and Zcash are open-source, problems and vulnerabilities may be found and fixed right away. Together, the community of users and developers can find and address any potential problems, ensuring the security and dependability of cryptocurrencies. All things considered, the open-source nature of Monero and Zcash encourages a culture of responsibility and openness among the community. Anyone may inspect the code and confirm that the cryptocurrencies are operating as planned thanks to its openness. It also makes it possible to develop third-party applications that can communicate with the coins.

4.4. Community-driven development

Active and committed communities of developers, contributors, and users are responsible for the development and upkeep of both Monero and Zcash. This community-driven method of development aids in ensuring that cryptocurrencies keep developing and becoming better over time. The development process is open to community input, whether it be through code contributions, bug reports, or feature suggestions.

A vibrant community of users, developers, and contributors supports and advances the cryptocurrency known as Monero. Anybody can participate in the development process because the community is inviting and open to everybody. With new features and enhancements being regularly introduced, Monero's community-driven approach to development has helped the cryptocurrency expand and change over time. Similar to Bitcoin, Zcash has a robust community of users, developers, and contributors who collaborate to maintain and advance the cryptocurrency. Anybody can participate in the development process because the community is welcoming and open to everyone. Zcash has improved over time thanks to its community-driven approach to development, which has seen continuous additions of new features and improvements.

Both Monero and Zcash are sensitive to their users' requirements thanks to the community-driven approach to development. It is encouraged for community members to offer input on the coin, and in response to user requests, new features and enhancements are frequently implemented. Additionally, the community-driven development methodology contributes to the resilience of both Monero and Zcash against any attempts to compromise its security or decentralisation. The vibrant and committed communities of developers, contributors, and consumers contribute to the long-term security and dependability of cryptocurrencies. Ultimately, a fundamental resemblance between Monero and Zcash is the community-driven approach to development. Both cryptocurrencies have vibrant and committed user, developer, and contributor communities that collaborate to maintain and advance the coins while assuring their continued security, dependability, and adaptability to user demands.

4.5. Optional privacy

Users can choose to enable or disable the optional privacy features in both Monero and Zcash as needed. Ring signatures and secret transactions can be enabled or disabled in Monero, whereas shielded transactions can be enabled or disabled in Zcash. Depending on the circumstance, this enables users to strike a compromise between their requirements for privacy and their desires for transparency and traceability. Ring signatures, one of the optional privacy features in Monero, let users conceal their identities by grouping their transactions with those of others in a way that makes it challenging to determine who began the transaction. Also, it is challenging to ascertain how much cryptocurrency was traded because of Monero's confidential transactions feature, which conceals transaction amounts. Depending on their demands for privacy, users can opt to enable or disable these functionalities.

Shielded transactions, another of Zcash's optional privacy features, let users encrypt their transaction data so that it cannot be seen by other users on the blockchain. Users can continue to transact while maintaining their anonymity and privacy thanks to this functionality. Depending on their demands for privacy, users can decide whether to enable or disable this function. Both Monero and Zcash are aware that their users value privacy but that there are particular circumstances where greater transparency or traceability may be necessary. Users are able to find a compromise between their privacy needs and their requirements for transparency and traceability thanks to the optional privacy features.

Additionally, both cryptocurrencies optional privacy features add to the network's overall security and privacy. Users can pick their level of privacy, making the network as a whole more resistant to monitoring and censorship. In conclusion, a significant resemblance between Zcash and Monero is their optional privacy features. Both allow users to select their level of privacy, allowing them to strike a compromise between their privacy needs and their desires for openness and traceability and enhancing the network's overall security and privacy.

4.6. High Security

By guaranteeing that the blockchain can handle the necessary volume of traffic, Monero's dynamic block size limit aids in the prevention of spam assaults. With the help of this function, the network is able to adapt to fluctuations in demand, keeping the system from getting overwhelmed and potentially opening itself up to attacks. The proof-of-work consensus mechanism used by Monero guarantees network stability by making miners compete for newly created cryptocurrency by solving challenging mathematical puzzles. To keep the rate of block generation stable, the difficulty of these issues is dynamically changed. A high level of security is offered by Zcash's zk-SNARKS technology, which enables transaction verification without disclosing any of the transaction data.

The privacy and identity of the users involved are maintained while also assisting in the prevention of double spending and other types of fraud. Strong cryptography is used by both Monero and Zcash to prevent hacking and other types of attacks. Ring signatures, stealth addresses, and private transactions offered by Monero give its users extra levels of privacy and security. The adoption of cutting-edge encryption technology by Zcash also contributes to the security and legitimacy of transactions.

Ultimately, both Monero and Zcash are built with security in mind, leveraging a variety of cutting-edge tools and methods to thwart intrusions and uphold the anonymity and privacy of their users. Users that appreciate the safety and security of their financial transactions find them appealing due to their emphasis on security.

5. ATTACKS

5.1. Attacks on Monero

It has been established that the quantity of mixins used in Monero's ring signatures has a significant impact on its level of anonymity. The increased transaction fees in the present edition prevent customers from raising their ring signature size. To address this issue, short linkable ring signatures were proposed, and it was stated that since they reduce the size of the signature from O(n) to O, they provide a good improvement. To fully evaluate the significance of the change, it would be crucial to implement the short linkable ring signature and test it in subsequent work.

5.1.1. Transaction Flooding

A fresh attack that involves overwhelming the network with transactions to find the payment keys of trustworthy users in the blockchain of Monero. Although the goal is straightforward to flood the Monero network with transactions whose input and output keys belong to the attacker, we must overcome several obstacles and pay close attention to little details that have an impact on the outcome. First, because the transaction fee is inversely related to the transaction's size in bytes, the attacker must carefully select a size that is economical [5] . Also, we aim to increase the quantity of output keys every transaction. The attacker should ideally design cost-effective transactions with a large number of output keys. Third, we study chain reactions on the transaction tracing attack and how they might enhance the quantity of inputs that can be traced.

Assuming a Monero transaction tx with one input tx.in containing four input keys (tx.in = {pk1, pk2, pk3, pk4}), while one of the keys (e.g. pk4) represent the real coin being spent, the remaining three keys are being used as decoys to hide the real output key being spent in the transaction. However, if three of the public keys (e.g. pk1, pk2 and pk3) are owned by the attacker, it becomes straightforward to find out which one is the payment key. This is one of the most basic principles of the transaction flooding attack, i.e., the attacker has to own a set of valid output keys and make it as big as possible. When the key is theirs, the attacker can remove and mark as unknown a key pkx from the input of a transaction made by another user. The attacker is aware that pkx is being used as a ruse if the key has not yet been utilised to make a payment. Second, the attacker is aware that the key is a mixin if they are aware that it has already been used in a previous transaction (such as pk4). In both situations, it is safe to remove the key pkx from a transaction's input keys (tx.in).A transaction flooding attack can be used in real life to take advantage of the prior example.

The key to a successful transaction flooding attack is having enough keys to ensure that the system chooses the attacker's set of keys for all mixins of an input tx.in. The attacker must saturate the network with legitimate transactions in order to possess output keys. These transactions outputs will be forwarded to addresses that belong to the attacker. The number of addresses receiving a portion of the payment determines how many output keys there are in a transaction (used as dummy addresses in subsequent transactions). An output key containing a certain number of coins will be sent to each receiving address (XMR). The system may eventually choose to employ those output keys in the mixin set of upcoming transaction inputs. Here, we assess how an attacker might profit from a transaction flooding attack, which merely makes use of the available space within a block of Monero to carry out transaction input tracking assaults.

5.1.1.1. The Attacker Model

The attacker can obtain blockchain data since Monero's blockchain data is open to the public and available to anybody. We presume that in exchange for the capacity to track transaction inputs, the attacker is ready to pay transaction costs. In order to overwhelm the network, we also assume that the attacker has access to at least two different Monero addresses. The quantity of XMR required to reimburse the costs charged for the attack activities must be present at one of those addresses. The other address will be used to store output keys and accept transactions. Keep in mind that setting up a new Monero wallet is simple and free. The attacker is capable of generating as many deals as he wants at any given time, which is the final assumption we make.

5.1.1.2. Flooding Monero's Network

Before attempting to track transaction inputs, the attacker must first construct a large number of output keys. Tracing only applies to transactions that were made after the assault started. A block of transactions is chosen by Monero using gamma distribution to select decoy keys, and a random output key is then chosen from that block. Until 10 decoy keys are obtained for each transaction input ring, this process is repeated. The decoy selection mechanism favours blocks with recently created output keys, and as time passes on and outputs get older, their chances of being chosen decline. The output keys must be evenly dispersed throughout the blockchain in order to enhance the attack's effectiveness. So, for the transaction input tracing to be as accurate as feasible, the attacker must continuously produce transaction outputs. Based on empirical observations of the main net data for Monero from blocks 2,154,590 (1 August 2020) to 2,176,789 (31 August 2020), we developed a method for an attacker to leverage unoccupied block space to their advantage. The information gleaned from our observations is summarised in Table I.

The attacker's method is to create fresh output keys by making payments to his personal Monero wallet addresses. Since the minimum amount for a transfer is 1 piconero (1012 XMR), each transaction output costs this amount, transaction fees make up the majority of the cost of initiating transactions.

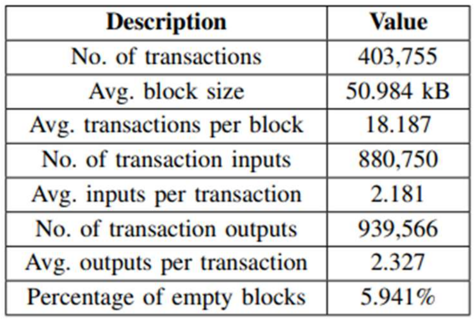


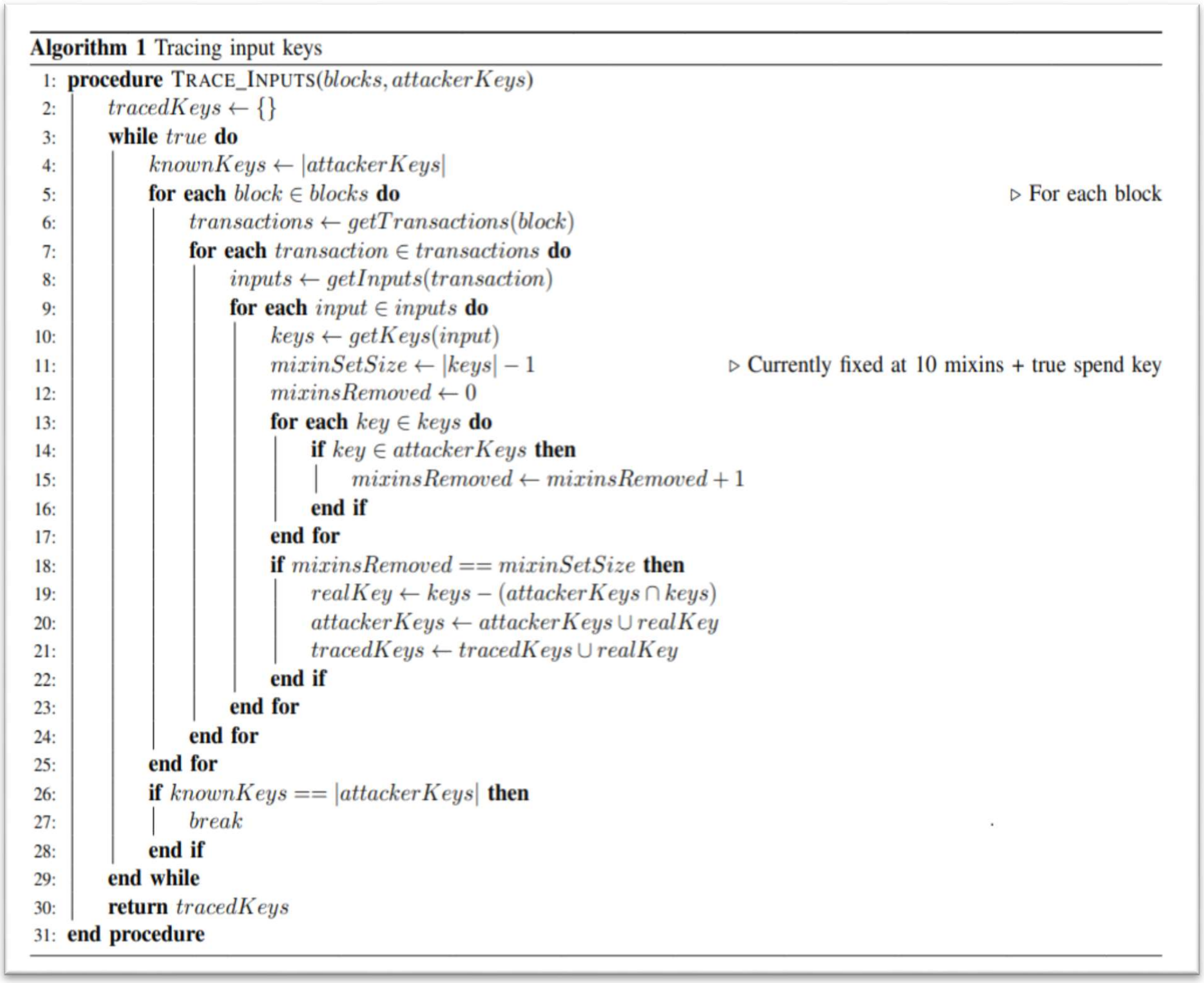
TABLE I
SUMMARY OF DATA FROM MONERO BLOCKS 2154590 TO 2176789 (AUGUST 2020).

Description	Value
No. of transactions	403,755
Avg. block size	50.984 kB
Avg. transactions per block	18.187
No. of transaction inputs	880,750
Avg. inputs per transaction	2.181
No. of transaction outputs	939,566
Avg. outputs per transaction	2.327
Percentage of empty blocks	5.941%

5.1.1.3. The Tracing Algorithm

A malicious actor can either target individual transactions and their inputs or run a tracing algorithm across all blockchain transaction data produced after the attack started in order to trace transactions. In all scenarios, it is the attacker's responsibility to maintain track of the keys produced by attack transactions during the network flooding phase. Any transaction added to the chain after the attack started can likewise undergo the input tracing technique. To accomplish this, the attacker needs obtain a copy of the blockchain data for Monero and then follow the procedures in Algorithm 1. This method is more efficient because it enables the use of traced inputs to lessen the privacy of other transactions as well. This is accomplished by include recently found true spend keys in the attacker's list of known transaction outputs since he now knows when this specific key has been spent.

Two data inputs are required to start the process: the block data taken from the Monero blockchain and the attacker's set of keys (Line 1). The inputs for each transaction are extracted throughout each iteration of Algorithm 1 (Line 8). The output keys in the ring must be checked for each input (Lines 9 to 23) and those that are in the attacker's list of keys must be marked. The remaining key is the true spend (traced key), and it should be added to the attacker's list of keys that they are aware of (Lines 19 to 21) if they are aware of all but one of the keys used in the transaction input (Lines 11 to 18). Increases in the attacker's key pool (Line 20) indicate the discovery of new true spend keys. In that instance, the analysis will be repeated over all blocks in order to potentially trace more input keys (Lines 5 to 24). The method won't terminate until zero inputs (Line 19) could be traced in the previous iteration (Line 4 and 26), indicating that no new keys and thus no new true spend keys could be found.The algorithm finishes by returning the list of identified true spend keys (Line 30), at which point the process is over.



Algorithm 1 Tracing input keys

```
 1: procedure TRACE_INPUTS($blocks, attackerKeys$)
 2:     $tracedKeys \leftarrow \{\}$
 3:     while $true$ do
 4:         $knownKeys \leftarrow |attackerKeys|$
 5:         for each $block \in blocks$ do                              ▷ For each block
 6:             $transactions \leftarrow getTransactions(block)$
 7:             for each $transaction \in transactions$ do
 8:                 $inputs \leftarrow getInputs(transaction)$
 9:                 for each $input \in inputs$ do
10:                     $keys \leftarrow getKeys(input)$
11:                     $mixinSetSize \leftarrow |keys| - 1$    ▷ Currently fixed at 10 mixins + true spend key
12:                     $mixinsRemoved \leftarrow 0$
13:                     for each $key \in keys$ do
14:                         if $key \in attackerKeys$ then
15:                             $mixinsRemoved \leftarrow mixinsRemoved + 1$
16:                         end if
17:                     end for
18:                     if $mixinsRemoved == mixinSetSize$ then
19:                         $realKey \leftarrow keys - (attackerKeys \cap keys)$
20:                         $attackerKeys \leftarrow attackerKeys \cup realKey$
21:                         $tracedKeys \leftarrow tracedKeys \cup realKey$
22:                     end if
23:                 end for
24:             end for
25:         end for
26:         if $knownKeys == |attackerKeys|$ then
27:             $break$
28:         end if
29:     end while
30:     return $tracedKeys$
31: end procedure
```

5.2. Attacks on zcash

Spam has been targeted at the privacy-focused proof-of-work blockchain Zcash. According to analysts, the blockchain has handled a sizable number of transactions intended to disrupt the network. The attacker achieved this by abusing Zcash's "shielded transactions," which were designed for anonymity.

5.2.1. PING And REJECT

5.2.1.1 Introduction

Zcash is a cryptocurrency that prioritises privacy and uses zk-SNARKs, an original type of cryptography, to support anonymous transactions. However, in 2019, researchers identified a flaw in the implementation of Zcash's zk-SNARKs that would allow attackers to produce fake money and syphon money from honest users. The "join-split" function, which transforms transparent Zcash currencies into anonymous coins, has a bug that led to the vulnerability. An attacker might use this issue to double-spend real coins, steal money from other users, or even manufacture phoney anonymous coins and utilise them [6]. The incident demonstrated the necessity for continued care and scrutiny in the creation and implementation of cryptocurrency protocols, even though the Zcash development team promptly released a patch to solve the vulnerability. The incident serves as a reminder that even the most sophisticated and advanced cryptography is susceptible to flaws and assaults.The attack on Zcash serves as a reminder of the value of continued study and development in the area of cryptocurrency security, as well as the necessity for users and developers to be watchful and proactive in spotting and patching possible flaws.

Zero-knowledge proofs are employed by the privacy-focused cryptocurrency Zcash to safeguard users' privacy. However, the researchers found a weakness in the way Zcash nodes converse with one another. They discovered that they could gather details about the transactions taking place on a Zcash node by sending specially constructed "PING" messages to that node. The PING messages take advantage of a side-channel attack, a technique for learning about a system by examining inadvertent information leakage. In this instance, the researchers were able to determine the amount of transactions being handled by the node by timing the PING answers. They were able to tell when transactions were being processed and when the node was inactive by sending many PING packets and examining the response times.

The researchers also discovered that they could use a "REJECT" message to learn more details about the transactions that were being handled. They could check whether a transaction was being handled by the node by issuing a REJECT message for a specific transaction ID [7]. The sender and recipient of the transaction may potentially be identified if the node responded with a "REJECT" message, indicating that the transaction was not being processed. Overall, the privacy of Zcash users is seriously threatened by this flaw in the way Zcash nodes communicate with one another. An attacker might possibly compromise the anonymity of such transactions by using this side-channel attack to deduce sensitive data about the transactions taking place on a node.

5.2.1.2 Sapling's In-band Secret Distribution:

- **Notation and Initials**

This article gives the terminology and preliminary results for the pairing-friendly BLS12-381 curve with modulus q and the elliptic curve over Fq of order 8r, indicated by Jubjub in Zcash, for in-band secret distribution in Sapling.

Assume that J is the collection of points on Jubjub and that J(r) stands for J's prime-order subgroup of order r. J(r)*, which is defined as J(r) O, where O denotes the point at infinity, stands for the set of points of order r.

We vary from the Zcash specification's [4] convention by using lowercase symbols to represent scalars (such as ivk) and uppercase symbols to denote curve points in order to maintain clarity (e.g., PKd)

```
GenIVK()
    $\mathsf{ivk} \xleftarrow{R} \{0, \ldots, 2^{252} - 1\}$
    return $\mathsf{ivk}$
```

```
GenDiversified(ivk)
    $\mathsf{G_d} \xleftarrow{R} \mathbb{J}^{(r)*}$
    $\mathsf{PK_d} \leftarrow \mathsf{ivk} \cdot \mathsf{G_d}$
    return $(\mathsf{G_d}, \mathsf{PK_d})$
```

- **In-band Encryption**

A sender establishes a new transaction with an Output description that includes, among other information, a Note commitment in order to transmit some value v to a shielded payment address (Gd, PKd): Commit(Gd||PKd||v;rcm) = cm. The receiver must demonstrate that she is familiar with the Note commitment cm's opening in order to spend her coins.

The sender can safely transmit an opening of the Note commitment to the receiver as part of the transaction thanks to Zcash's in-band secret distribution technique. The receiver's public parameters (Gd, PKd) and a brand-new ephemeral secret esk are used in a semi-static Diffie-Hellman key exchange for this purpose:

```
$\mathrm{KeyAgree}(\mathsf{G_d}, \mathsf{PK_d})$
$\mathsf{esk} \xleftarrow{R} \mathbb{F}_r^*$
$\mathsf{EPK} \leftarrow \mathsf{esk} \cdot \mathsf{G_d}$
$\mathsf{SS} \leftarrow 8 \cdot \mathsf{esk} \cdot \mathsf{PK_d}$
$\mathsf{k} \leftarrow \mathrm{KDF}(\mathsf{SS}, \mathsf{EPK})$
$\mathbf{return}\ (\mathsf{k}, \mathsf{EPK})$
```

The Note plaintext contains the commitment cm's introduction (np). The sender adds the ciphertext C and the ephemeral public key EPK to the transaction after encrypting np under k with a conventional symmetric AEAD technique.

- **Blockchain Scanning**

A Zcash user examines each transaction in the blockchain with her incoming viewing key ivk to recover coins sent to her. She computes an ephemeral public key EPK, a Note ciphertext C, and a Note commitment cm for a transaction.

```
$\mathrm{TrialDecrypt}(\mathsf{ivk}, \mathsf{EPK}, C, \mathsf{cm})$
$\mathsf{SS} \leftarrow 8 \cdot \mathsf{ivk} \cdot \mathsf{EPK}$
$\mathsf{k} \leftarrow \mathrm{KDF}(\mathsf{SS}, \mathsf{EPK})$
$\mathsf{np} = \mathrm{Decrypt}_{\mathsf{k}}(C)$
$\mathbf{if}\ \mathsf{np} = \perp,\ \mathbf{return}\ \perp$
$\mathrm{Parse}\ \mathsf{np}\ \mathrm{as}\ \mathsf{np} := (\mathsf{G_d}, v, \mathsf{rcm}, \mathrm{memo})$
$\mathsf{PK_d} \leftarrow \mathsf{ivk} \cdot \mathsf{G_d}$
$\mathbf{if}\ \mathsf{cm} \neq \mathrm{Commit}(\mathsf{G_d} || \mathsf{PK_d} || v; \mathsf{rcm}),\ \mathbf{return}\ \perp$
$\mathbf{return}\ \mathsf{np}$
```

- **Unlinkable Diversified Addresses**

With the use of the Zcash protocol, a user can create a variety of payment addresses that are all connected to the same private incoming viewing key. This means that a user can produce numerous pairings (Gd I PKd I that meet the discrete-log relation PKd I = ivk Gd I

This makes it possible for a user to send a unique public key to any entity that they want to accept payments from without having to open additional accounts.

It should be difficult to determine whether two different diversified payment addresses belong to the same user, according to the unlinkability criteria for diversified addresses. A game with a computationally constrained attacker defines the formal security requirement A:

```
$\mathsf{Unlinkability}_{\mathcal{A}}$
$b \xleftarrow{R} \{0, 1\}$
$\mathsf{ivk}_0 \leftarrow \mathrm{GenIVK}(), \mathsf{ivk}_1 \leftarrow \mathrm{GenIVK}()$
$(\mathsf{G_d}^{(0)}, \mathsf{PK_d}^{(0)}) \leftarrow \mathrm{GenDiversified}(\mathsf{ivk}_0), (\mathsf{G_d}^{(1)}, \mathsf{PK_d}^{(1)}) \leftarrow \mathrm{GenDiversified}(\mathsf{ivk}_1)$
$(\mathsf{G_d}^{(2)}, \mathsf{PK_d}^{(2)}) \leftarrow \mathrm{GenDiversified}(\mathsf{ivk}_b)$
$b' \leftarrow \mathcal{A}(\mathsf{G_d}^{(0)}, \mathsf{PK_d}^{(0)}, \mathsf{G_d}^{(1)}, \mathsf{PK_d}^{(1)}, \mathsf{G_d}^{(2)}, \mathsf{PK_d}^{(2)})$
$\mathbf{return}\ b = b'$
```

The unlinkability of diversified addresses holds under the DDH assumption in $\mathbb{J}^{(r)*}$ [4].

5.2.1.3. WORKING OF ATTACK

The privacy-focused cryptocurrency Zcash's Zcash client is the subject of the REJECT and PING attacks. Both methods try to figure out whether a Zcash client is able to crack the Note ciphertext C of a particular shielded transaction, which reveals whether the distant Node was the transaction's payee. The setup of the attacks and the side channel they utilise are different.

- **REJECT attack :**

A network assault called a REJECT attack, also called a reset attack, involves sending a TCP RST (reset) packet to one or both endpoints of a TCP connection. This packet is used to break a connection between two endpoints, forcing them to reconnect after losing their initial connection.

A more detailed explanation of the REJECT attack's operation is provided below:

By listening in on network traffic or running a port scan, the attacker often discovers an active TCP connection between two hosts. A faked TCP RST packet is sent by the attacker to either one or both endpoints of the connection.The legal packets of the connection's connection have the same source and destination IP addresses, port numbers, and sequence numbers as the RST packet. The endpoint considers that the other endpoint has cut off communication when it receives the faked RST message and replies by sending a TCP ACK packet back to the attacker. Both endpoints will lose their connections as a result of the connection being terminated, and the attacker will have successfully prevented communication between them.

The REJECT attack can be used for a number of things, including interfering with host communication, avoiding network detection, or launching a denial-of-service (DoS) assault. It can be lessened by employing methods like source address validation, firewalls, and intrusion detection and prevention systems. In essence, the REJECT attack was created specifically for protected transactions that the attacker had created. A Zcash client decrypts the Note ciphertext C when it receives a protected transaction [8] . The client makes an attempt to parse the Note plaintext np if the decryption is successful. A 564-bit string that contains the first byte indicating the encoding version is considered genuine plaintext. The plaintext serialisation throws a std::exception if the leading version byte is assigned an invalid value, such as 0x02. a message-containing ios base::failure exception. A "reject" message is sent back to the transaction's sender as a result of this message being intercepted by the ProcessMessages function in main.cpp. With the use of this "reject" message, the sender is given an oracle signalling the successful decryption of a Note ciphertext using a deliberately flawed plaintext.

PING attack :

The PING attack, on the other hand, targets any transaction that is shielded. A Zcash client verifies that it has received a legitimate opening of the Note commitment cm after decrypting a Note ciphertext and parsing the Note plaintext. This requires two expensive cryptographic operations—calculating a Pedersen hash function and multiplying a Jubjub point twice by a random 256-bit scalar. The attacker sends a "ping" message to a Zcash node immediately after relaying a shielded transaction. The peer-to-peer Zcash interface handles incoming messages serially. To react to the "ping" message, it first executes the transaction, including the TrialDecrypt call.

The amount of time before receiving the "ping" response reveals if the Note was successfully decrypted and, consequently, whether the node was the payee of the relayed transaction. Even transactions sent by law-abiding users are subject to the attack. To each recipient, the attacker would transmit a "ping" after relaying the transaction to additional nodes.

By taking use of a time side-channel in the TrialDecrypt function, the PING attack preys on Zcash users. When a Zcash client successfully decrypts a Note ciphertext, it verifies that the Note commitment was opened in a legitimate manner. This requires expensive cryptographic procedures, which lengthens the time of a successful TrialDecrypt call relative to an unsuccessful one. By transmitting a shielded transaction, followed by a "ping" message to a Zcash node, an attacker can establish a time side-channel. The amount of time that passed before receiving the "ping" response tells us whether the node was the payee of the relayed transaction and whether the Note decryption was successful. Even transactions sent by law-abiding users are susceptible to the assault.

Similar time side-channels also exist in the older Sprout protocol, but they are much smaller and unlikely to be reliably detectable from a distance.

5.2.1.4 Attack Practicality :

- **Payee Discovery**

On the regtest network, a payee-discovery attack was tested with the victim and attacker PCs on the same LAN. Half of the 200 transactions that the attacker communicated to the victim were payments. With the exception of five outliers who had reaction times of more than 60 ms, the victim's response time to the ping message accurately determined whether the victim was the transaction's payee. Although the payee of every shielded transaction cannot be determined with absolute certainty by the attacker in a typical WAN, the timing difference of around 2ms between a successful or unsuccessful TrialDecrypt is substantial and can lead to a potent attack.

- **Linkability:**

By transmitting ciphertexts that corresponded to incorrect plaintexts to a victim node running the original Zcash client, the REJECT variant of the node linkability attack and the diversified address linkability attack were validated in a local Zcash network. Sending one or two transactions to each peer in the network and waiting for a "reject" message to respond are all that is needed to launch the attacks. The intended victim might get the transaction, though, before the attacker's message reaches the victim, if a peer receives a faulty transaction. While the transaction is already in the victim's memory pool, the victim will in this situation send a "reject" message to the relaying peer instead of responding to the attacker's message.

- **Denial of Service:**

A local Zcash network was used to test the DoS vulnerability. A service with numerous Zcash clients, such as an exchange that supports shielded addresses, could be a feasible target. But, carrying off a large-scale DoS assault would require an attacker to be in possession of shielded payment addresses for many Zcash users. It may not be realistic to launch a DoS attack on a significant portion of Zcash miners because many of them might not have created a shielded address.

- **Remote Timing:**

It is more difficult to determine the viability of a remote timing attack on an ECDH key exchange. Although one million samples were needed for the local timing attack on the elliptic curve multiplication routine to be successful, this number would be significantly higher if the variability in network delay and transaction confirmation time were taken into account. However, it's not impossible that the timing attack might still be enhanced. One challenge with the REJECT attack type might be sending all those transactions to a target before any one of them gets mined, crashing the victim, assuming a remote timing attack was feasible with a relatively big number of samples.

- **What is Affected?**

Between version 2.0.0 (the original Sapling version) and version 2.0.7-2, the official Zcash client implementation is vulnerable to attacks. Every client user that created a shielded address and disclosed the public payment address ("zaddr") to a third party may be exposed to the aforementioned threats (the IP recovery attack can be mitigated by using Tor). Zcash release v.2.0.7-3 has mitigated every attack. All forks of Zcash that enable Sapling shielded addresses inherit these attacks, therefore they must either update to Zcash v.2.0.7-3 or develop a custom workaround. All Zcash wallets that utilise the official Zcash client and allow shielded addresses are weak points. They include Z, Zepi, and ZWallet.

5.2.2. ITM Attack

The ITM Attack is a specific kind of linkability attack that takes use of transaction graph information leaks. Because it may be used to de-anonymize transactions that were previously believed to be entirely protected, this attack is very worrying. By "squeezing" fresh information leaking from zaddrs out of areas where nobody thought to investigate, the attack is able to operate. As a result, it is significantly more challenging for users to employ shielded transactions to safeguard their anonymity. It is crucial that researchers and developers keep working on creating innovative methods to counteract the ITM Attack and other linkability assaults.

5.2.2.1. WORKING OF ATTACK

The Zcash Protocol's shielded transactions are the focus of the ITM attack, a linkability attack. By taking advantage of the data that is exposed when shielded addresses are used in Zcash transactions, this attack operates. Although shielded addresses are designed to give Zcash transactions privacy, they can potentially reveal details about the transaction, such as the amount transferred and the time it took place.

A shielded transaction can only be fully spent, which is an advantage used by the ITM attack. This means that when any one of the outputs of a transaction that has several outputs is spent, the entire transaction must also be spent. An attacker may detect transactions with numerous outputs by examining the blockchain, and based on the time and transfer amount of the transaction, they can deduce which output is being used [9] . This enables the attacker to connect the shielded address to the appropriate transparent address that was used to complete the transaction. The attacker can monitor further transactions utilising the same transparent address once the shielded address has been connected to it, possibly disclosing more details about the user's activity. Users of the protocol should be extremely concerned since the ITM attack has the potential to jeopardise the security and privacy of Zcash transactions.

5.2.2.2. ITM Attack: z2z Transaction Linkability

The ITM Attack specifically "attacks" a transaction $T : z \rightarrow z, z$, which is the most private transaction possible under the Zcash Protocol since it is totally shielded. Initially, if any of the following facts can be confirmed, we give the definition of the attack's success:

- The amount sent via the zaddr.
- The worth of any zaddrs that are getting payments.
- Any ShieldedInputs that were used throughout the transaction, if any.
- A range of potential values encoded in the transmitting zaddr, for example, between 0.42 and 1.7 (including error estimate)

The ITM Attack accepts transaction ids, zaddrs, and other OSINT that is easily accessible on Github, Twitter, Discord, Slack, open forms, mailing lists, IRC, and many other places as input. With the use of these open sources, the ITM Attack may go from being a purely theoretical attack to actually de-anonymizing a zaddr and revealing its associated social media accounts, email addresses, IP addresses, location data, and more.

5.2.2.3. ITM Attack: Assumptions

The interested blockchain research business is left to its own devices with fully functional sample code. Here they provide enough information on the assault for experts to confirm our assertions and for developers to design countermeasures.In the spirit of complete openness. They assume that an attacker has at least $100,000 USD available for the task of researching a specific Zcash blockchain. The majority of this expense goes towards buying a GPU/FPGA farm to process data. The cost of researching blockchains with more experience and larger protected pools will increase.Then point out that this assault, which is a way to examine a full blockchain and may then be indexed and searched for potentially valuable data, is not financially possible as a one-off. Since they already have extensive infrastructure to accommodate this new dataset, blockchain analysis businesses and the IC are strategically positioned to utilise this information with the least amount of expense.

5.2.2.4. ITM Attack: Defeating zk-SNARKs

Only in fact, not in principle, can we consider this attack to be a "defeat" of zero-knowledge mathematics. Many qualifications are required. By utilising the way zk-SNARKs are employed in higher level protocols, such as the Zcash Transaction Format Protocol and its accompanying consensus rules, we are not in any way "breaking" the mathematics of zk-SNARKs. Because this is mathematically impossible, we have not really leaked knowledge from a zero-knowledge proof. Instead, zk-SNARKs are sound [10] . We have divulged information on how these proofs are employed in the wider Zcash Protocol system, which is an outgrowth of the infamously data-leaky Bitcoin Protocol.

6. CONCLUSION

Researchers and attackers are engaged in a constant arms race over privacy. Privacy and bug-free software cannot be guaranteed. It is probably not a good idea to breach the law using privacy-focused cryptocurrency. It costs money to have all nodes validate every transaction, and adding privacy characteristics adds even more expense. Before consumers fully understand what public key cryptography is, it might be unrealistic to expect them to adhere to security best practises. Hence, privacy should be as unobtrusive as possible to decrease friction and free users to concentrate on spending rather than the coin itself.

There are various strategies for achieving privacy, as demonstrated by the Monero and Zcash protocols. A balance between privacy, usability, and efficiency must be struck in cryptographic applications, and they serve as a good example of how to do it. The Monero ring signatures offer a simple method for maintaining privacy. The privacy assumptions, however, are weaker than in Zcash, which in turn results in less usefulness. It has been established that the quantity of mixins used in Monero's ring signatures has a significant impact on its level of anonymity. The increased transaction fees in the present edition prevent customers from raising their ring signature size. To address this issue, short linkable ring signatures were proposed, and it was stated that since they reduce the size of the signature from O(n) to O, they provide a good improvement. To fully evaluate the significance of the change, it would be crucial to implement the short linkable ring signature and test it in subsequent work.

7. REFERENCES

1. S. Goldfeder, H. A. Kalodner, D. Reisman, and A. Narayanan, “When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies,” CoRR, vol. abs/1708.04748, 2017.
2. S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
3. “Why I Have Nothing To Hide Is the Wrong Way to Think About Surveillance.” https://www.wired.com/2013/06/why-i-have-nothing-to-hide-is-the-wrong-way-to-think-about-surveillance/. [Online; accessed June-10-2018].
4. J. Quesnelle, “On the linkability of zcash transactions,” CoRR, vol. abs/1712.01210, 2017.
5. “How Zcash Tries to Balance Privacy, Transparency in Blockchain.” https://www.americanbanker.com/news/ how-zcash-tries-to-balance-privacy-transparency-in-blockchain. [Online; accessed June-13-2018].
6. Au, M.H., Chow, S.S., Susilo, W. and Tsang, P.P., 2006, June. Short linkable ring signatures revisited. In European Public Key Infrastructure Workshop (pp. 101-115).
7. Springer, Berlin, Heidelberg. Au, M.H., Liu, J.K., Susilo, W. and Yuen, T.H., 2013. Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction.
8. Theoretical Computer Science, 469, pp.1-14. Bitansky, N., Canetti, R., Chiesa, A. and Tromer, E., 2012, January.
9. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (pp. 326- 349). ACM. Bowe, S., Gabizon, A. and Green, M.D., 2017.
10. A multi-party protocol for constructing the public parameters of the Pinocchio zk-SNARK. TR 2017/602, IACR. Bowe, S., Gabizon, A. and Miers, I., 2017.