MONERO AND ZCASH

Emmanuel Chiebuka Anyanwu
Information Systems Security
Concordia University,
Montreal, Canada
e_anyanw@live.concordia.ca

Olivia Okechukwu
Information Systems Security
Concordia University,
Montreal, Canada
olivia.okechukwu@mail.concordia.ca

Damilola Sanni
Information Systems Security
Concordia University,
Montreal, Canada
damilola.sanni@mail.concordia.ca

Ajayi Samson
Information Systems Security
Concordia University,
Montreal, Canada
samson.ajayi@mail.concordia.ca

Eunice Msheliza
Information Systems Security
Concordia University,
Montreal, Canada
mshelizaeunice@gmail.com

Benjamin k. Kolako Jr
Information Systems Security
Concordia University,
Montreal, Canada
b_kolako@live.concordia.ca

Oreoluwa Olujimi
Information Systems Security
Concordia University,
Montreal, Canada
o_oluji@live.concordia.ca

Frederick Agyemang
Information Systems Security
Concordia University,
Montreal, Canada
frederick.agyemang@mail.concordia.ca

Roseline Ayim Antwi
Information Systems Security
Concordia University,
Montreal, Canada
roseline.ayimantwi@mail.concordia.ca

***Abstract* — After discovering that Bitcoin only provides minimal privacy, several cryptocurrencies have emerged that use privacy-enhancing cryptographic techniques, such as ring signatures and zk-SNARKs. Monero and Zcash are, in turn, two of the most well-known cryptocurrencies that employ these methods. This paper will examine the ZCash and Monero protocols' working principle and thoroughly explain how they operate. We will also look at how these two methods are similar and different. Furthermore, we will analyze recent attacks on these protocols and discuss the impact these attacks had on the security and privacy of Monero and ZCash users. By the end of this paper, readers will have a comprehensive understanding of the Monero and ZCash protocols, their similarities, differences, and the challenges they face in providing their users with robust privacy and security features.**

Keywords—Attacks, Monero, Zcash, Cryptocurrency, Anonymity, Blockchain, Transactions

I. Introduction

Cryptocurrencies have been a hot topic recently, with many people seeing them as the future of money. However, one of the main criticisms of traditional cryptocurrencies like Bitcoin is their lack of privacy and anonymity. This has led to the rise of privacy-oriented cryptocurrencies, such as Monero and ZCash, which focus on providing enhanced privacy and security features to their users. Blockchain systems differ from previous cryptosystems in that they are not only intended to secure an information asset. Because a blockchain is a ledger, it serves as the asset [3]. Cryptographic techniques are used to safeguard a blockchain. Importantly, private/public key pairs are created using asymmetric encryption techniques like RSA or Elliptic Curve (EC) cryptography to secure data assets stored on blockchains. Monero and ZCash have gained significant attention in recent years. Both protocols focus on providing users with enhanced privacy, and anonymity features that traditional cryptocurrency like Bitcoin lack.

II. Overview of monero and zcash

A. Monero

The peer-to-peer digital currency Monero (XMR) prioritizes privacy and is intended to be untraceable and anonymous. Because Monero is fungible, a transaction on the blockchain cannot be traced back to a specific user or real-world identity [1]. Similar to Bitcoin, it is built on blockchain technology, in which the data from transactions are made available to the public. However, observers in Monero are unable to follow the transfer of funds between addresses, unlike in Bitcoin, where anyone can do so. The ring signature and the one-time public key are the technologies that have been employed as the default configuration to increase the privacy of the transaction data [2]. The majority of blockchains encourage the usage of

pseudonyms to maintain anonymity. Although a user's pseudonym is known to others, pseudonym identities do not grant them anonymity. Chain analysis techniques can determine who sent and received transactions, how many tokens were sent or received, how much money was in an account, and other information [3]. By employing additional cryptographic algorithms, Monero enables users to obfuscate their identities and the value of transactions.
The environment comprises at least two participants, daemons, and wallets, just like any other cryptocurrency. The Monero daemon is a service that offers data to users. The Monero daemon stores a complete record of all transactions locally and synchronizes blockchain data with its peers. Monero Wallet requires data the Monero Daemon provides to create a transaction; it cannot do this alone. This is because, in Monero, any genuine output that must be spent on an input must first be obscured by several additional outputs (decoys). Mixins is another name for this decoy. The ring signature is built using the decoys and the actual output. Ring size is defined as the sum of the decoy output and the actual output [2]. Real public keys already visible in the blockchain serve as the decoys. These decoys are, in other words, the results of earlier transactions. These public keys are organized into groups according to how many coins they own, and then they are indexed in sequential order according to when they first appeared in the blockchain [2].

How Does Monero Work

How Monero achieves transaction anonymity is a further aspect of interest. It achieves this by using three technologies:

1. Stealth addresses
2. Ring signatures
3. Ring confidential transactions.

Ring Signatures

Any member of a specific group with private keys can sign a digital signature using the general cryptography concept known as a "ring signature" [2]. Monero uses ring signatures to obfuscate the sender [4]. A ring signature on a message makes it computationally impossible to determine who signed it, but it shows that someone in the group signed it. By combining the cash sent by one user with the funds sent by other users, ring signatures enable transaction mixing. It's possible that the precise coin you sent was delivered to a different user than the one you intended. Yet, fungibility means that it is irrelevant. The value of two coins is the same [4]. To form a ring signature in Monero, multiple outputs known as "decoys" or "mixins" with an equal amount of coins are combined into a single input. These outputs should have genuine outputs that will be utilized in the transaction. A transaction can have multiple inputs and outputs simultaneously [2].

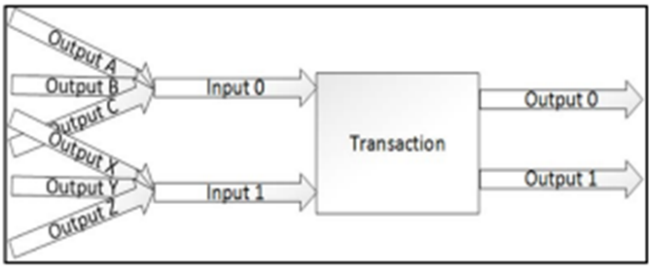

Fig. 1. A Monero Transaction's structure. an input with multiple outputs that are created by combining the real output with other outputs. Adapted from [2].

The objective of the ring signature is to minimize the possibility of an adversary accurately guessing a genuine output from a group of N outputs. The probability denoted as (P) of correctly predicting the true output that is used with an input can be calculated using the formula [2]:
$P = 1/N$

Stealth addresses

For every transaction, a sender can use a stealth address to create a one-time public address on the recipient's behalf. Monero allows the recipient to receive all payments by a single public address, just like Bitcoin does. Every user of Monero creates two distinct keys: a private view key and a private spend key. Users can access every transaction about their accounts using the private view key. While the private spend key, which functions similarly to a private key for Bitcoin, is used to authorize all types of payments [1]. If the receiver wants to initiate a transaction, they send the sender a group of "parent public keys." Next, the sender employs arbitrary numbers to produce fresh "child public keys" that are encrypted and added to the transaction information. Using the one-time public key guarantees that the relationship between these two keys, which are the parent public key and the child public keys used in the transaction, is only known to the sender and receiver. Without any additional data, observers cannot comprehend the relationship between the two keys, even with access to the blockchain [2].

Ring Confidential Transaction

This refers to a method of hiding the number of coins used in transactions using the Pedersen commitment. It is then used to create Ring Confidential Transaction in the ring signature (RingCT). The primary criteria for creating a ring signature are that all of the members of the ring must possess the same number of coins, making it impossible to tell the true ring from the decoys. By concealing how many coins are kept in the public keys, the RingCT is used to address the liquidity problems [2]. This makes Monero much less vulnerable to quantum attackers than other blockchain networks, or at the very least makes it a slightly less appealing target [3].

To summarize, due to Monero's transaction anonymization feature, its transactions are not as attractive to attackers as those on other blockchain networks, despite being highly vulnerable to quantum attacks.

B. Zcash

Zcash is a decentralized cryptocurrency that aims to give users greater privacy and anonymity when making purchases. Although it is based on the same technology as Bitcoin, it has more privacy features that let users conduct protected transactions that are not recorded in the blockchain. The digital currency prioritizes the protection of user privacy through the use of advanced scientific techniques. It enables individuals to make secure transactions with minimal fees. The shielded feature of Zcash ensures that all transactions are kept confidential, while still allowing individuals to choose to share transaction details in order to comply with regulatory requirements [5] [6].

The creation of Zcash began in 2013 under the direction of Matthew Green, a professor at Johns Hopkins, and his graduate students. Later, the for-profit Zcash Company, headed by Colorado-based computer security expert and cypherpunk Zooko Wilcox, finished the project. The Zcash Company raised more than $3 million from venture investors in Silicon Valley to pay for the development in October 2016. Zcash mining started in late October 2016, and a week later the coins were in great demand and trading for $5,000 each. For the first four years, 10% of all tokens mined were distributed to the Zcash Company, its personnel, investors, and the nonprofit Zcash Foundation [7].

Despite significant advancements in security and privacy, zcash uses zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge), a form of technology that concentrates on privacy to protect the anonymity of its users. Zcash transactions can be entirely encrypted thanks to zk-SNARKs, making it impossible for anybody to observe who sent money to whom or how much was sent. The usefulness and safety of this very advanced technology have been verified by independent security specialists.

Zcash is a cryptocurrency based on Proof of Work (PoW), this means that miners are required to first answer difficult mathematical puzzles in order to add new blocks to the network. Equihash, the mining algorithm used by Zcash, was created to be ASIC-resistant and is therefore more accessible to small-scale miners using common gear [5] [8].

Multiple Transaction Types

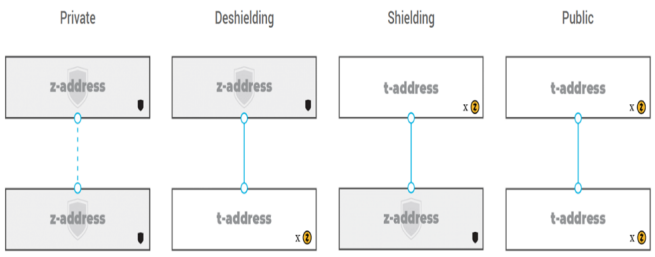

Fig. 2. Zcash multiple transaction types. Adapted from [5]

Private z-addresses and transparent t-addresses are the two distinct types of addresses in Zcash. When a Z-to-Z transaction occurs, the blockchain not only logs that the transaction happened and that the fees were paid, but it also encrypts and hides the addresses, transaction value, and memo field. Zero-knowledge algorithms are used to enable this degree of encryption. Through the use of view keys and payment transparency, users have the choice of disclosing z-address and transaction details with dependable third-party organizations, such as for audit and compliance needs [5].

Z-to-T transactions consist of a transaction with at least one public outcome but no public input. The value of the outputs as a whole must be lower than or equivalent to the value of the freshly revealed coins, with the transaction fee making up the remaining value [9].

T-to-Z transactions, on the other hand, entail a transaction with no public outputs, only public inputs. The sum of the inputs must equal to or greater than the value of the newly hidden coins, with the transaction fee being the remainder [9].

T-to-T transactions function in the same way that Bitcoin transactions do by making the sender, receiver, and transaction value publicly accessible. To increase user privacy, more wallets and exchanges are switching to shielded addresses rather than just using t-addresses solely.

The interoperability of Zcash's two address classes enables money transfers between z-addresses and t-addresses. Users must fully understand how protecting or exposing information during these interactions will affect their privacy [5].

What are zk-SNARKs?

Zk-SNARKs is a special type of zero-knowledge cryptography that was first presented through Zcash and is what enables Zcash's robust privacy protection. The term "zero-knowledge succinct non-interactive argument of knowledge," or "zk-SNARK," refers to a proof construction technique whereby a single party, the prover, will demonstrate to another party, the verifier, that they are in possession of a specific piece of information (like a secret key), without divulging any more information or engaging in communication with the verifier.

In "Proof of Knowledge" procedures based on zero-knowledge, the prover can show the verifier not only that the information is real but also that they are aware of it

without disclosing any details about the information itself. In this situation, the distinction between "Proof" and "Argument" is technical and not pertinent.

"Succinct" zero-knowledge proofs are validated quickly, typically within a few milliseconds, and are brief, even for large programmes. With non-interactive constructions, the prover can transmit the verifier a single message instead of the multiple messages required by earlier zero-knowledge protocols. Prior to the creation of Halo, the fastest method for producing non-interactive, concise zero-knowledge evidence that can be published on a blockchain and require no user interaction required an initial setup process that generated public parameters and an agreed-upon reference string between the prover and verifier.

The public parameters of the system were produced through two multi-party ceremonies for Sprout and Sapling in order to protect the integrity of Zcash's zero-knowledge proofs and stop the production of counterfeit coins. Attackers could produce fake proofs that look real to the verifier if they had access to the hidden randomness used to generate these parameters. This could result in the production of fake coins. To prevent this, the ceremonies involved multiple participants who each contributed random data, which was combined to generate the public parameters in a way that no single participant could influence or manipulate the outcome. These ceremonies were designed to be transparent and auditable, allowing anyone to verify the integrity of the resulting parameters [5].

How zk-SNARKs are built in Zcash

The function that decides whether a transaction is valid or not must return the result without revealing any of the information it used to reach its conclusion necessary for Zcash to retain zero-knowledge privacy. This is achieved by encoding some of the consensus norms of the network in zk-SNARKs. The procedure entails converting a collection of algebraic equations that represent the rules for validating a transaction into a form that can be evaluated on a potential resolution without revealing any private information to the parties looking at the equations. Essentially, this is done by first transforming the transaction validation rules into a form that is equal to knowing the answer to an algebraic equation [5].

Computation → Arithmetic Circuit → R1CS → QAP → zk-SNARK

First, it is necessary to decompose the logical operations into discrete, little-enough operations in order to describe the transaction validity function mathematically. This results in the creation of a "arithmetic circuit," which is similar to a boolean circuit and in which a programme is divided into straightforward, independent stages like AND, OR, and NOT. For an arithmetic circuit, the programme is divided into steps that involve the basic processes of addition, subtraction, multiplication, and division (however division is avoided in this specific case).

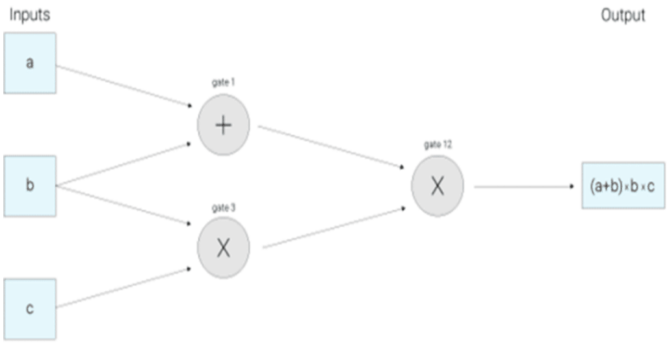

Fig. 3. Zcash multiple transaction types. Adapted from [1An example of an algebraic circuit for computing the equation (a+b)*(b*c).Adapted from [5].

When looking at an arithmetic circuit, we can see the input values moving along the wires towards the output line from left to right. A Rank 1 Constraint System (R1CS) must then be built in order to confirm that the numbers are moving as intended. When b and c are inputs to the multiplication gate, the R1CS for the example expression (a+b)(bc) would affirm that the result is b*c.

Using a Quadratic Arithmetic Program (QAP) representation, Gennaro, Parno, Gentry, and Raykova demonstrated how to combine all the constraints that the R1CS representation needs the verifier to check into a single constraint. The only constraint in the QAP that needs to be verified is one between polynomials, not two integers. Despite the size of the polynomials, an identity that breaks down between them will usually break down at some time. Therefore, the proof can be verified with a high degree of probability using just one random point.

Zk-SNARKs evaluate polynomials blindly, without knowing the point being evaluated, in order to prevent a prover from creating invalid polynomials that still fulfill the identity at a selected point. These methods include homomorphic encryption and pairings of elliptic curves. To choose which point to verify, the public parameters are used, but they are in encrypted form to protect privacy. The previous explanation primarily concentrated on how to obtain the S and N in SNARKs, which are brief, non-interactive, single message proofs. It neglected to address the "zk" (zero-knowledge) component, which enables the prover to safeguard one's confidentiality of their private inputs. The prover can simply add the "zk" part by randomly shifting the initial polynomials as long as they still satisfy the necessary identity [5].

How to create a shielded transaction using zk-SNARKs

Zk-SNARKs are employed to demonstrate the legitimacy of a transaction without disclosing private information about the parties concerned. In particular, when a sender makes a shielded transaction in Zcash, they build a proof that, with a high degree of probability, demonstrates that the input values add up to the output values and that they are authorized to spend the funds being transferred. This proof makes the entire transaction tamper-evident by cryptographically connecting the input notes' secret spending keys to a signature.

Zcash employs commitments and nullifiers rather than unspent transaction outputs (UTXOs) to achieve privacy. The shielded counterpart of UTXOs are commitments, and a commitment is spent using nullifiers. For each new note generated by a shielded payment, a commitment is released that includes a hash of the recipient address, the amount being sent, a special number called "rho," and a random nonce. A nullifier, or zero-knowledge statement that the sender has the authority to spend the money, is published by the sender when a shielded transaction is spent. The unique number rho from an extant commitment that hasn't been used, hashed, serves as the nullifier.

In addition, Zcash employs a set of proving and verifying keys for the creation and verification of proofs. These keys are created during a public parameter ceremony and distributed among all network users. The sender uses their proving key to create a proof of the validity of their inputs, and miners use their verifying key to validate the computation of the proof. Zcash's proof generation architecture offloads the bulk of the computational work to the transaction creator, which increases the prover's upfront workload but simplifies verification. This makes it possible to keep encrypted transactions on the blockchain while maintaining privacy.

Overall, Zcash's privacy is based on conventional cryptography and the addition of zk-SNARKs, which allows parties to demonstrate the validity of transactions without disclosing private data. A further layer of anonymity is added when commitments and nullifiers are used in place of UTXOs, and verification is made easier while keeping security when proving and verifying keys are used [5].

Working principle of Zcash

Created through a modification of Bitcoin's code, Zcash's primary goal is to sever the connection between the sender and recipient of a transaction. In Bitcoin, recipients receive money in specific addresses and spend them from the same addresses, creating a link between the two parties. These links can be traced as the money changes hands, allowing for monitoring of the flow of Bitcoins. In Zcash, any interaction with the shielded pool in a transaction is done through vJoinSplit. It shows the source and destination of the coins, users have the option of providing a transparent address or a shielded address in order to receive money. Money is regarded as being in the shielded pool if it is kept at the shielded address. The vJoinSplit comprises an encrypted memo field, two shielded outputs, and a list of output addresses where the money is going. The shielded outputs include an ambiguous quantity of cash as well as a secret double-spending token, which could be a fake output with no cash to conceal the absence of a shielded output. Private notes can be sent to the recipient of the shielded outputs using the encrypted memo field.

A collection of input addresses, two double-spending tokens, and a zero-knowledge proof are all included in a vJoinSplit to demonstrate where the money is coming from. A dummy value used to hide the absence of a shielded input or a distinct token from a prior shielded output is used as the double-spending token. The zero-knowledge proof guarantees that the double-spending token actually comes from a previous shielded output and ensures that the sum of the values contained in the input addresses, which comprises the double-spending tokens, equals the sum of the values allocated to the output addresses plus the sum of the values in the shielded outputs as well as the miner's fee [10].

III. Differences and Similarities of Monero and Zcash

A. The Differences between Monero and Zcash

Monero (XMR) and Zcash (ZEC) are two privacy-focused cryptocurrencies that use distinct protocols to maintain anonymity and secrecy. Although they share the same goal of providing privacy to their users, their methods of execution differ greatly.

A. Privacy protocols:

Monero - Monero offers privacy via Ring Confidential Transactions. (RingCT). RingCT achieves privacy by combining three cryptographic primitives. MLSAG (Multi-layered Linkable Spontaneous Anonymous Group) ring signatures are used to conceal the sender of the transaction, Confidential Transactions are used to obfuscate the amount sent in a transaction and stealth addresses are used to protect the privacy of the transaction's recipient [11].

Zcash - it employs a privacy technique known as Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARKs). The protocol ensures transaction verification without providing any information about the sender, receiver, or transaction amount. This technology enables transparent transactions while maintaining privacy. In Zcash, there are two sorts of addresses called z-addresses (shielded addresses) and t-addresses (transparent addresses). The input values for each transaction are split into amounts from transparent addresses and amounts from shielded addresses. Also, the output values are divided into amounts that are sent to shielded addresses and amounts that are sent to transparent addresses. The z-addresses or shielded addresses are usually hidden (i.e. not visibly), and shielded addresses transactions do not disclose the address, the amount of the transaction, or the information in the encrypted memo field. On the other hand, the transactions that take place between them can be viewable by anyone on the Zcash blockchain [11].

B. Anonymity set:

The overall anonymity provided by the cryptocurrency is made up by the anonymity set. The degree of anonymity increases with the increase in the anonymity set.

Monero - The set of anonymity in Monero is typically equal to the number of mixins used to create the ring signature. It was initially possible in the earlier versions of Monero to send transactions with zero mixins. These transactions have been discovered to be useful in de-anonymizing subsequent transactions [11].

Zcash - As opposed to Monero, where the anonymity set is much smaller, Zcash's anonymity set is equivalent to the entire pool of shielded transactions. [11].

C. Optional in privacy:

Zcash - In Zcash, various transactions between transparent and shielded addresses are supported. Which means that using the privacy-preserving features of Zcash is optional. Only a small percentage of users have been discovered to use shielded transactions, suggesting that the vast majority of Zcash transactions are transparent. Even though Zcash still offers a much bigger anonymity set than Monero, it is not as large as one might anticipate. This is due to the fact that Zcash has a much smaller amount of shielded transactions. This suggests that Zcash's anonymity is impacted by the opt-in privacy choice [11].

Monero - When making a transaction in Monero, you must use Ring Confidential Transaction. The user's transaction must generate a minimum ring size of 7. This means that every transaction is anonymous up to the specified degree of anonymity set and is equivalent to the number of mixins used. As a result, it is best for the user to add more mixins to their transactions in order to increase this anonymity set. But the vast majority of users continue to maintain the smallest ring size possible. One of the main causes of this is that users are discouraged from using more mixins as the transaction charge rises with ring size [11].

D. Transaction size:

Monero - The average transaction size in Monero is 12.5 kb. Monero transactions are way bigger than Zcash transactions because the amount of mixins chosen increases the size of the ring signature, which increases the size of the transaction. Monero's minimum mixin of 7 has already significantly raised the transaction size. This is a major flaw in Monero because users have to decide between the advantages of increased privacy and reduced transaction costs [11].

Zcash - In Zcash, its transaction size is 2 kb which is substantially less than Monero, this means that users can employ shielded transactions without paying additional fees for transactions [11].

E. Trusted setup:

Zcash - zk-SNARKs requires a trusted setup in generating the public parameters used to create and validate proofs. A multi-party generation protocol is used to implement this trusted setup. The setup was carried out by six core members of the team, each of whom held a piece of the master secret. By utilizing their respective portions of the master secret, each party completed its part of the protocol, resulting in the public parameters [11]. Each party must discard its portion of the master secret after the parameter creation is completed. If anyone keeps their part of the master secret, they will be able to generate false SNARK proofs, effectively printing money without anyone knowing because it is hidden [12].

Monero - The issue is entirely averted here because Monero doesn't require a trusted setup [11].

F. Mining algorithm:

Zcash - Equihash is the proof-of-work algorithm used by Zcash for block mining. The Equihash proof-of-work algorithm was created by Dmitry Khovratovich and Alex Biryukov and this algorithm is based on the “Generalized Birthday Problem,” which is a concept in computer science and cryptography. This algorithm was also created to be resistant to ASICs. However, ASICs for Equihash have subsequently been produced, raising worries regarding centralization in the mining process [13].

Monero - RandomX is the name of Monero's new mining algorithm. It is a Proof of Work algorithm. (PoW). RandomX is intended to generate a completely random "work zone" or scratchpad with a high memory consumption, and it operates using advanced virtualization techniques. RandomX's ASIC resistance is due to these three foundations. RandomX is not only ASIC-resistant, but also GPU-resistant, because these devices lack the instructions required to perform the complex RandomX operations in the first place. As a result, RandomX aims to be a CPU-only mining algorithm with the strongest resistance to other methods of mining. However, this is also intended to avoid the "very famous botnets" in Monero, as RandomX will make them nearly impossible to construct. Especially in low-power gadgets like Smart TVs and IoT devices [14].

G. Supply:

Monero - Monero has over 18.1 million tokens or XMR coins in circulation and has met its total supply, although there is no maximum supply logistically. Block rewards were set to be fixed at 0.6 XMR per block beginning in June 2022, and under the new software rules, rewards for new

blocks will never reach zero. This is often referred to as "Tail Emission [15]."

Zcash - Zcash has a circulating supply of 16,328,269 ZEC coins and is anticipated to have a maximum supply of 21,000,000 ZEC coins by 2032. The Zcash network receives or mines a new block every 75 seconds, and a block reward of 3.125 ZEC is distributed as a result. Every four years, the value of the block reward is reduced to half until its 21 million ZEC are circulated [16].

B. *The Similarities between Monero and Zcash*

1. Both Monero and ZCash are cryptocurrencies that prioritize anonymity and aim to provide anonymous transactions. Both protocols are intended to conceal the sender, recipient, and transaction amount [17] [18].

2. Both Monero and ZCash are decentralized cryptocurrencies that run on a peer-to-peer network without having a central authority in charge of their growth or management. They are, therefore, less susceptible to censoring and government interference [17] [18].

3. Proof-of-work mining algorithms are used by both Monero and ZCash, requiring miners to solve difficult mathematical puzzles to verify transactions and earn new coins. Proof of work is also used to guarantee the integrity of new data because the decentralized networks used by cryptocurrencies lack a central governing authority [19].

4. Both Monero and Zcash are developed by open-source communities of developers and contributors, that means their code is publicly accessible for anyone to review and make contributions. Both communities prioritize transparency and community participation in their development processes [17] [18].

5. Peer-to-peer transactions, online transactions, and value storage are just a few of the uses for both Monero and Zcash. They are also commonly used for various illegal activities like money laundering, tax evasion, and other criminal activities because of their privacy features.

IV. Attacks on Monero

A. *Anonymity Attack on Monero*

The paper investigates a new class of remote side-channel attacks against the Monero cryptocurrency, a privacy-oriented cryptocurrency that provides anonymity through its RingCT mechanism. The authors demonstrate that an attacker who has access to a victim's computer can use timing side-channel attacks to recover the victim's secret key and hence break the anonymity of the Monero transactions. The authors provided a detailed analysis of the Monero protocol and the RingCT mechanism, as well as an overview of previous attacks on Monero's anonymity. The authors then present their remote side-channel attack, which relies on timing measurements of the victim's computer. They show that these measurements can be used to recover the victim's secret key, which in turn allows the attacker to deanonymize the victim's Monero transactions. [20]

The attack takes advantage of variations in how a wallet and node communicate when the digital wallet is the recipient of a fresh, unconfirmed, or mined transaction. A network attacker (or a malicious remote node) may assume receipt of payment if the wallet establishes a connection with a remote node by idly observing the encrypted data exchanged between the two. The source of the exploit is seen as an interaction sequence between a digital wallet and a remote node. The wallet first asks the node for a record of unconfirmed transactions and gets a list of hashes in return. By reviewing the requests from the wallet, a malicious remote server can discover which transactions fund the wallet. The sheer existence of this request can potentially reveal the wallet's payee of a recent transaction The attack's validity was confirmed in the local Monero network. However, because it only considers the presence or lack of transaction signals and not timing signals, the attack is 100% successful regardless of the type of network.

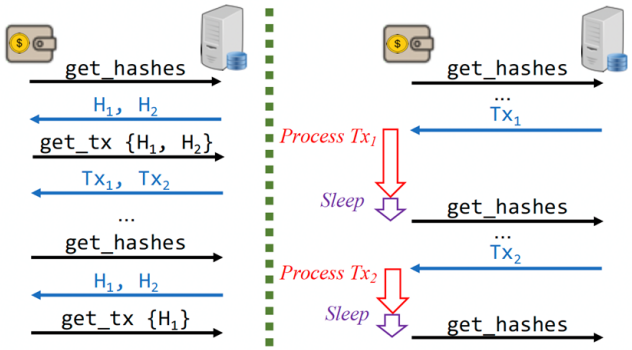

Fig 4. Side channel communication between a Monero wallet and P2P nodes. Adapted from [20]

Timing Attacks For Remote Nodes

The duration between requests also leaks if a wallet receives payment as it was previously established that the amount of network requests a wallet and node exchanged.

can lead to leaks. The wallet checks to see if it is the transaction's recipient. In the case the wallet is the payee of

the transaction, it proceeds to decrypt the obtained value which results in a longer processing time. There are two timing attacks because of the processing time difference. The processing of new blocks is the focus of the first attack. After a refresh, the wallet serially gets new blocks from the node and processes its transactions. As a result, during the period between two block requests, the time taken to process the payments in the first block leaks. Unconfirmed transactions are the target of the second attack.

Timing Attacks For Local Nodes

The data exchange patterns between the victim's node and wallet are invisible to a remote attacker. A P2P adversary can deduce exchange patterns using the attack developed, particularly when a remote wallet sends a payment request to its node.

This attack takes advantage of rough locking in P2P servers for Monero. The P2P node obtains a lock hold on its mempool while handling a payment request sent by a peer or a wallet. The lock assertion in the peer-to-peer node will delay the answer to the attacker if a peer-to-peer adversary transmits a GET message right away following the request from the victim's wallet.

As the P2P node verifies requested transactions before releasing the lock, which results in the locks remaining active for an extended period upon being requested from the wallet, the likelihood of lock assertion is high. The attacker only requests false transactions to shorten the lock period to decrease the likelihood that the adversary's request will lock out the wallet's request. The attacker can tell if the digital wallet has sent the request for a transaction to its node by observing the answer delay size. This aids the attacker in figuring out whether a specific transaction represents a transfer to the target wallet.

The findings are important because they reveal a previously unidentified weakness in Monero's privacy system. The authors show that remote side-channel attacks are a potent method for compromising the privacy of cryptocurrencies that depend on zero-knowledge proofs and ring signatures as cryptographic primitives. They also suggest several defenses. that are able to lessen the impact of the attack, such as separating wallet reload time against processing time, which thwarts our attacks because the attacker can determine when a refresh period begins but not when it ends.

B. Known attacks To Monero and Monero Anonimity

There are different types of attack on Monero and **Monero Anonymity**. However, the following highlights are discussed by the paper " Monero Ring Attack- Recreating Zero Mixin Transaction effect." The below are the attacks.[21]

Black Marbles Attack

In this section, the term "Black Marbles Attack" is one of the attacks to exploit Monero by the full control of outputs on the blockchain of Monero. The attacker would usually send coins to its own address to create outputs called black marbles.[3] As a result of the chosen address, the likelihood of the attacker's outputs being picked up as decoys for new transactions is high. The visibility of all observers to the good and bad outputs is displayed on the shared ledger.

Zero Mixin Transaction and Cascade Effect

In this attack, inputs are used and there are no decoys as compared to the Black marbles attack. As a result of this, it is easier for any threat actor to trace the sender of the transaction.[2] Before the RingCT technology, there was no strict regulation to ensure that users split transactions according to a denomination.[4] There has been recent research that proves that before the RingCT technology, there have been at least more than half input that can be separated between the decoys making the transactions to have significant impacts.

Temporal Analysis

In this type of attack, there is analysis done to show that for zero mixin transactions, the recent outputs are usually the real outputs. There is usually an 80% detection rate by the system and most sampling methods cannot hide this. To mitigate the challenge, a sampling method called triangular distribution[7]. In the triangular distribution, at least 25 out of 100 of the decoys are from the outputs to enable close identity between the user behavior and the sampling method

Publishing the Private ViewKeys

The private viewkey in Monero allows an auditor to track the coins received by a user's wallet without being able to steal the coins or determine if they have been spent. This same private key differentiates between the two outputs sent to both the sender and the user. The private viewkey is used mainly for compliance, but after an audit, the unlinkability feature can only be regained by transferring the balance and the creation of a new address. Malicious behavior by an auditor[5]who publishes private viewkeys can compromise user anonymity

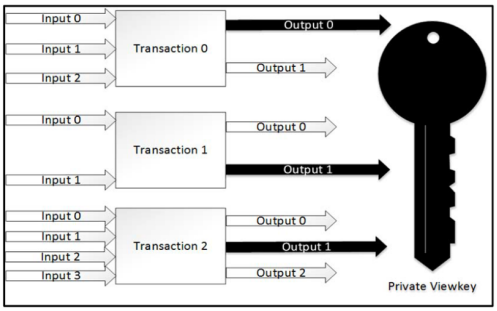

Fig. 5. A Private Viewkey Determines All Incoming Outputs (Payments). The private viewkey can determine the outputs sent to the corresponding address by scanning all transactions in the blockchain (black arrows). Adapted from [23]

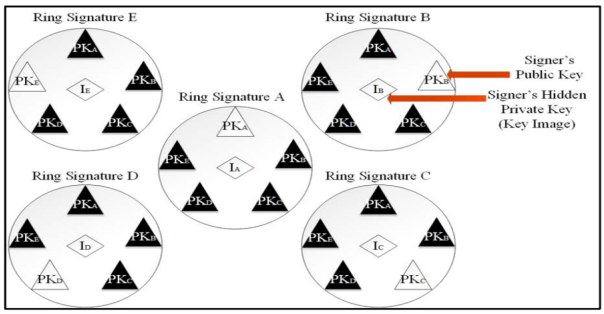

Fig. 6. The Setup Phase Where r = 5. Adapted from [26]

The Proposed Attack

The paper " " Monero Ring Attack- Recreating Zero Mixin Transaction effect." Described another attack asides the known attacks to Monero anonymity. The Monero leniency is exploited as the Monero wallet creates transactions. The wallet processes the ring leaving the daemon to confirm the validity of the transactions. The zero mixin transaction has a cascade wallet that is identical to the possibility of constructing a malicious transaction to reduce anonymity.[26]

The first stage of the proposed attack of the paper is the preparation phase, and this is described below.

The preparation phase

In this phase, the minimum ring size required by the system is important as the attacker needs several unspent outputs. Each output is usually spent in the setup phase and the preparation phase needs a lot of unspent outputs[24]. The RingCT technology enables attackers to split coins across multiple outputs removing the need for any extra coins. A passive attack requires so many outputs created by the attacker unlike the active attack. The reuse of the outputs might be suspicious; however, the effect of the attack is not reduced.[1]

The setup phase

In this phase, the creation of r inputs is required by the attacker. In the same transaction, the attacker decides to add r inputs which is very effective unlike having all inputs included in multiple transactions. The set of public keys and their secret key image pairs are chosen from a set of l public keys L.[23] The preparation phase eliminates the possibility of L member spent by any transaction, thus pinpointing the precise input that spends a certain output is impossible. The attack phase cannot be carried out if r inputs are not produced with all outputs being members of L.[25]

The Attack Phase

In this section, the paper specifically targets the two types of attack on Monero. The two attacks are Passive and Active Attack. These two attacks are very different in their purpose, and they are unique in the way they operate. We would be describing the Passive attack first.[22]

Passive Attack.

A passive attack aims to prevent the reuse of public keys by using outputs used during the setup phase of a transaction for numerous transactions. As a result, the transactions have decreased k-anonymity, which is based on how many dummy nodes were produced during setup. Due to the amount of expended public keys used as dummy keys in the example given, anonymity was lowered by two.

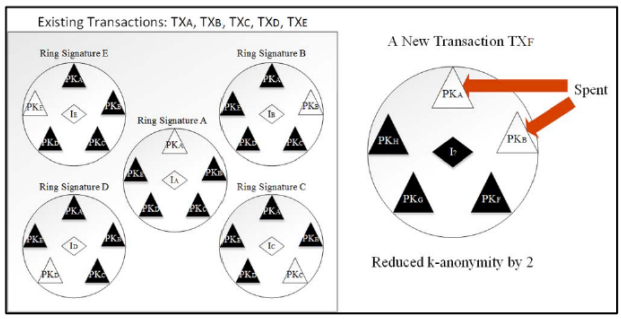

Fig. 7: The Passive Attack. Adapted from [24]

Active Attack.

Attacker B's malicious Monero wallet service does not seek to steal money from customers; instead, it aims to make transactions traceable. The wallet conducts an active attack by using public keys L as ring signature decoys. This attack is especially effective when it aims to compromise the outputs of other users. If the attack protocol is effectively deployed in a wallet, it will be challenging for consumers to recognise the illicit activity.[23][27]

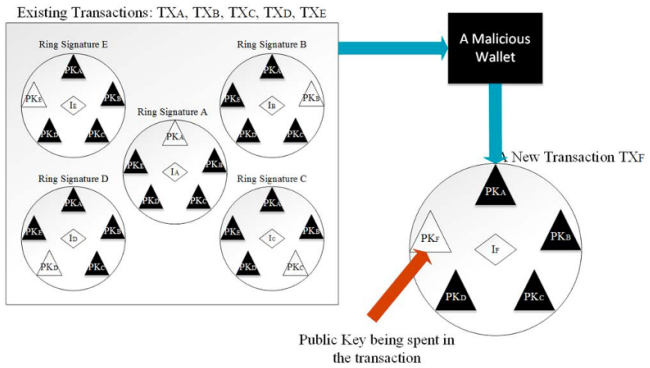

Fig. 8. The Active Attack. Adapted from [27]

The Proof of Concept

For every stage of the proposed attack, we need to show a sample of how these attacks are performed. This section describes proof for two of the phases described above. The creation of a malicious wallet by modifying Monero's source code is very importance to showcase the proposed attack[22]

The preparation phase (Proof of Concept)

The indexes from the public keys are chosen by the wallet rather than utilizing the standard protocol to select indices from histogram data. Moreover, the wallet keeps the global index for each output, thus data requests to the daemon are no longer necessary.

The Setup Phase

As part of the proof of concept in the preparation phase, we have successfully completed the setup phase using the following transaction ID.

8d4a0c7eccf92542eb5e1f09e72cc0d934b180b768bc95388d33051db83194bb

The above transaction was conducted with Xmrchain.net

An Evaluation Framework on all Attacks vs the proposed attack

In a coordinated attack scenario involving a few attackers, the proposed attack is superior to the Black Marble Attack because the outcome of the assault can be assessed using various attackers with no additional information.

In modern online services like exchanges or wallet services, it is simpler to enforce the transaction because it does not wish to be transmitted to the attacker's address.

In a RingCT-enabled system, the attack is significantly more successful and has a higher precision rate than temporal analysis because it can 100% accurately determine the genuine output from the input.

The attack may be carried out by converting ordinary transactions, so it no longer depends on the availability of 0 mixin transactions.

The Factors	BM	ZM	TA	PPV	The Proposed attack
Collaboration between attackers	×	√	×	×	√
Requires no extra fees	×	√	√	×	√
RingCT resistant	√	×	√	√	√
Minimum mixin resistant	√	×	√	√	√
Accuracy in determining real outputs	√	√	×	√	√

× - **Does not support**

√ - **Supports**

BM - **Black Marbles Attack**
ZM - **Zero Mixin Transaction and Cascade Effect**
TA - **Temporary Analysis**
PPV- **Published Private viewkeys**

Fig 9. Evaluation Framework. Adapted from [21]

The Transaction Flooding Attack

In this paper, the authors explain the transaction flooding attack in the context of Monero cryptocurrency. In this attack, the attacker tries to exploit the ring signature framework used by Monero whereby the genuine input keys are hidden by combining them with other output keys generated in older transactions [28].

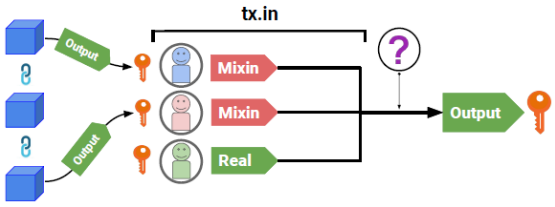

Fig 10 : Moneros' ring signature. Adapted from [28]

The attacker creates transactions in order to amass a large knowledge base where the system can choose keys to use as its mixins keys in future transactions. The attack's central idea is to use its own addresses to receive outputs of the flooded valid transactions created by him. The amount of Monero addresses that receive a payment portion determines the no. of output keys generated in a transaction. The output key that contains a certain number of coins (XMR in this case) will be given to the receiving monero addresses. The

system can later choose which of these output keys will be used for future transactions as input in the mixin set. The main difficulty in carrying out this kind of flooding attack is for an attacker to have a lot of output keys in order for the system to choose every mixins of the input key from the attacker's keys set. The Bulletproof protocol was introduced which increased the interest in and worth investigating the flooding attack which involves a weight-based transaction fee, thanks to the Bulletproof update. The number of outputs contained in a transaction determines the weight of the transaction. The paper also mentions that as the volume size of transactions increases, then the block capacity will also adjust as time passes, so users won't have to fight for block space by charging miners exorbitant fees, therefore, supporting dynamic block size changes. However, if the total amount of the transactions quickly exceeds the transaction size limit, then its size will not increase in proportion [28].

The Attacker Model

In order to trace transaction inputs, the paper describes an attack model and strategy for flooding Monero's network with transaction outputs. The assumption for the attacker to have access to blockchain data and to be willing to pay transaction fees to trace transaction inputs. The attacker is also assumed to have a minimum of two Monero addresses for the flooding attack, with one address containing the no. of coin (XMR) required to pay the attack transaction fees. The attacker can generate as many transactions as they want at any given time, but the miners must select and confirm the transactions, with no guarantee of timing. Before attempting to trace transaction inputs, A sizable amount of output keys must be generated by the attacker. As the attacking timeframe lengthens, the attack becomes more effective. Monero employs a gamma spread to choose decoy keys and then chooses an output key at random from within that set of mixin. As time passes and output keys become older, their probability of being chosen decreases because the decoy selection algorithm prefers output keys generated in the last few days [28].

The attacker's strategy is to send coins to their own Monero wallet addresses in order to generate new output keys. Each transaction output costs one piconero, so transaction fees are charged for creating transactions. To be as effective as possible, The attacker must constantly generate transaction outputs, and evenly spread the output keys across the blockchain. The paper provides findings of Monero's primary network data in order to devise a strategy for an attacker to exploit unused block space [28].

The authors describe a preliminary analysis performed to assess the viability of a transaction flooding attack in the Monero blockchain network. The analysis focuses on two scenarios: one in which the attacker generates transactions with two outputs and one in which the attacker generates transactions with sixteen outputs. It is noted that the transaction with the 16-output scenario is more efficient in the aspect of the ratio of transaction fees to outputs generated. However, because such transactions are less common in the network, an unusual increase in such transactions may raise suspicion and prompt blockchain creators to momentarily reduce the number of output keys that a transaction can generate. On the other hand, 2 output transactions are more common in the network, and even an unusual volume of such transactions may not raise too many red flags because the bulk of genuine transactions generates only 2 output keys [28].

To assess the attack's feasibility in these two scenarios, the authors conducted an initial analysis in Monero's private test network. They took into account a network scenario with 20 transactions generated by the user with two inputs and two outputs, in addition to one crypto-miner reward transaction with a single output. The Avg. size for a 2/2 transaction is 2.544 kB, while a 2/16 transaction is 3.802 kB. The no. of malicious transactions which can be added to the space available inside each block was then determined. A block can have up to 96 2/2 transactions plus user transactions, a block size of 294.43 kB, and a sum of 116 transactions. A 32 of a 2/16 transaction can be added to increase the block size to 293.95 kB and the transaction count to 52 [28].

When the number of individual transactions is included, the authors make an interesting observation, the obtained block size is much smaller than the actual block size that is displayed in the blockchain explorer. However, if the transaction weight is used in the calculation, the obtained block size is more like the actual block size shown in the explorer [28].

The Tracing Algorithm

The authors describe a tracing algorithm that a malicious actor could use to track transactions on the Monero blockchain. In a transaction input ring, the algorithm can be used to identify the true spend key, which is the key that represents the actual input used to spend a specific output. This information can be used by the attacker to compromise the privacy of other transactions and increase their chances of success in future attacks [28].

The attacker can run the algorithm on duplicate Monero blockchain data. It starts with two input keys: extracted block data from the blockchain technology of monero and the attacker's output keys set. On each iteration, the algorithm extracts from each transaction the input keys and checks the output keys stored in the input ring. If all but one of the input keys is known by the attacker, then the real transaction key is known and should be added to the

attacker's key list. When the attacker's key set expands, it indicates the discovery of new true spend keys, and as more input keys are discovered, the evaluation will be repeated on all blocks. Only when zero inputs have been traced in the previous iteration will the algorithm stop, indicating no new key and thus no new true transaction keys can be found [28].

The algorithm is effective at reducing transaction privacy while increasing the success rate of future attacks. It can be used to trace specific transactions or to run a tracking algorithm across all data from blockchain transactions generated after the flooding attack has started.

Evaluation and Result

The study's evaluation and results were based on simulations of two scenarios. In Scenario I, the attacker used transactions with two inputs and sixteen outputs to launch a flooding attack, yielding 512 malicious outputs and 32 transactions in each block with an addition to 41 outputs and 21 transactions generated by non-malicious users [28].

Scenario II examined the effect of the flooding attack in which the malicious user generates transactions with two inputs and outputs each, resulting in each block 192 malicious outputs and 96 transactions, When the 41 user-generated outputs were added, the block had a total of 233 outputs. The impact of the attack was measured for each scenario over a one-month to a twelve-month period of persistent flooding. In the evaluation, the same malicious key selection technique which was according to the gamma distribution employed in Monero's source code was used. The results demonstrated the likelihood of a “transaction input ring” created after the attack having its decoy keys made up entirely of attacker-owned outputs. According to the findings, an attacker is unlikely to have the ability to determine a minimum of three decoy keys in the network after a flooding attack has taken place for one month for the attack with 2/16 transactions. Furthermore, after one month of continuous flooding attacks, an attacker can eliminate all of the decoy keys from 41.21% of recently formed transaction input keys. After a year of continuous attack, this probability rises to 46.24%. The results of Scenario II revealed that after one month of attacking the network, In 10.72% of freshly formed transaction inputs, an attacker can determine the real spend. This probability increases to 14.47% after a year. However, the anonymity set the size of the majority of transaction inputs has been reduced by eight (8) and nine (9), respectively, giving the attacker between a 33.33% and a 50% chance of correctly predicting the true input. In both scenarios, the study found that a shortened attack time frame is less expensive [28].

Strengths of the flooding Attack:

1) The attack is effective in reducing Monero transaction anonymity because it allows the attacker to identify the real transaction input key in a significant percentage of newly created input keys.

2) The attack is relatively inexpensive and simple to carry out, requiring only a few inputs and outputs per transaction.

3) To maximize its effectiveness, the attack can be sustained for an extended period of time (up to 12 months).

Weaknesses of the flooding Attack:

1) Because the attack involves flooding the network with a high volume of transactions, it necessitates a significant amount of network resources to execute.

2) Because the attack generates a high volume of transactions with a similar pattern of inputs and outputs, it is relatively easy to detect and counter.

3) As the Monero network adjusts to the increased traceability power by increasing the size of input ring signatures, the attack becomes less effective over time.

V. Attacks on zcash

Remote Timing Side-Channel Attacks on Zcash-PING and REJECT

Introduction

Crypto currencies whose focus is privacy such as Zcash provide strong cryptographic assurances for transaction confidentiality and unlinkability. A wide range of attack types using analysis of traffic and timing side-channels, however, exploit side-channel information provided because of the implementation of the system components. This allows a remote attacker the ability to locate the payment’s receiver of any Zcash’s transactions. Any information that is leaked breaks the privacy assurances of anonymous transactions. Overview of Zcash Anonymous Transaction The attacks on the privacy of transactions in Zcash abuse communication time or pattern information revealed by various system parts. The figure below describes an anonymous transaction's life cycle as it navigates the system.

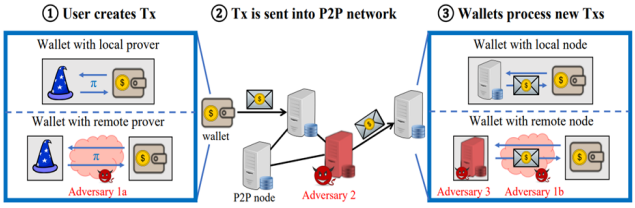

Fig. 11.Anonymous transactions in Zcash Adapted from[29].

A payer's wallet is used to initiate a transaction, and an isolated server generates a required zero knowledge proof to prove the transaction [29]. A peer-to-peer network transmits the transaction to the payee wallet, which receives it. To find the transactions for which the payee is the beneficiary, the wallet belonging to the receiver scans all the transactions in the network which are anonymous. It achieves this by attempting to decrypt the transactions using its secret key. Each of these steps contains side-channel information that an attacker can use to discover details about the transaction.

Technical Background

Step1

The diversified addresses of Zcash uses static Diffie-Hellman keys.

The key is: [PKd, Gd],

PKd = Gd. ivk [29]

Transactions made to public key (Gd, PKd) has UTXO (Note commitment) such as cm = Commit (Gd||PKd||v: rcm), [29].

Step2

On the sender side, El-Gamal encryption is used. An ephemeral secret key esk is used to compute public key EPK = esk *Gd by the sender ,

the key shared k = PKd* esk = ivk*Gd*esk.

In the transaction, the plaintext note np is encrypted by the sender. The sender uses the key k and encrypts it the np together with the ciphertext [29].

Step3

Since the receiver scans all transactions to determine which they are the recipient, the receiver does so with its private key ivk. It compute sthe following

{TRIALDECRYPT (cm, EPK, ivk, C}

i)k = EPK*ivk

ii)np = Decryptk (c)

iii) If np =.⊥ return. ⊥

iv) Parse np as np = (memo, v, Gd, rcm)

v) PKd = Gd * ivk

vi) if cm= Commit (Gd||PKd||v: rcm), .return. ⊥.

vii) .return np.

Supposing the decryption succeeds, the receiver checks if the np has an opening of the note commitment cm [29].

Attacks Overview

In the side- channel and timing attack, the vulnerability weaknesses in the system model of peer-to-peer users and wallets in Zcash is exploited. The vulnerability is based on the processes that a user's wallet uses to check if it is the receiver of new transactions made. Wallet behaviors are therefore leakable to a remote P2P adversary since there is no isolation between the wallet and the P2P node. The success of these attacks breaks the transaction unlinkability and user anonymity of Zcash. The PING attack exploits this vulnerability in the Zcash. Both the PING and REJECT attacks exploit a (weak) form of "decryption oracle" that allows the attacker to determine if a remote Zcash node was successful in the decryption of the ciphertext of the transaction. This further allows them to know who the receiver of the transaction is.

Ping Attack

As stated, the vulnerability which is exploited in the Zcash is the result of the close linking between the wallet and peer to peer elements in Zcash user. The time used to completely process a payment transaction (that is time taken for the node to determine if it is the right receiver of the transaction or not) affects the time the node processes other messages. A timing side-channel can allow a remote P2P adversary to determine if a node is the receiver of a transaction

How it Works

The attacker measures the TRIALDECRYPT duration by placing a PING request to the receiver node as soon as a transaction is received by the node. When the node decrypts a ciphertext, it verifies that the commitment note is legitimate. It achieves this by computing a Pedersen hash [30]. This increases the time taken to complete the TrialDecrypt call. Since the node processes the TrialDecrypt call first before responding the PING request, the time taken to receive a response from the PING request allows the attacker to know whether the receiver node was the receiver of the transaction.

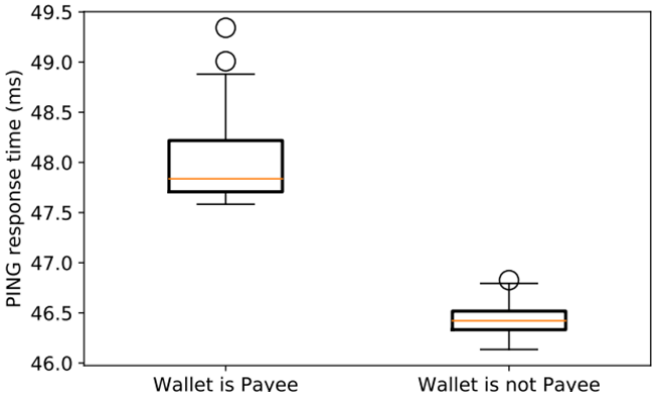

Fig. 12: Ping response time Adapted from [29].

Reject Attack

Reject attack exploits a vulnerability in how malformed transactions are managed. The REJECT attack generally is weaker when compared to the PING attack, but it is easier to implement. This attack works on transactions which were sent by the attacker to an address the attacker knows. This further enables an attacker who has a target node's public key the ability to direct a transaction to a victim node and cause the node to reply with a 'reject' message. The vulnerability exploited by the REJECT attack can further lead to a denial-of-service attack. This is because in the REJECT attack, supposed a transaction with a bad plaintext note is added to a block which is mined, this would cause the receiver's client to crash on validating the block. Furthermore, suppose an attacker is in possession of an enormous number of payment addresses of users, the attackers could implement a sturdy denial of service attack. Similarly, an attacker with the addresses of numerous miners on Zcash, implementing a denial-of-service attack could be exploited to throttle the network as seen in a 51% attack.

ITM Attack on z2z Transactions

The objective of the ITM Attack is to target transactions that involve sending "z" to "z", whereby "z" refers to a completely concealed Zcash Protocol transaction that offers the most advanced level of privacy. For an ITM attack to be successful, it depends on several key factors including the ability to determine the value of the zaddr used to send funds, the value of any zaddrs that receive funds, and the value of Shielded Inputs used in the transaction.

The ITM attack consists of predicting the range of potential values that are transmitted to a zaddr, which could fall between 0.42 to 1.7. To complete the attack, it is also necessary to make an estimation of the potential values that may be held in the zaddr responsible for transmitting the funds.

For an ITM attack to be successful, it relies on the ability to reveal any metadata.

ITM Attack makes use of zaddrs and transaction IDs as inputs, in addition to other publicly accessible information sources like GitHub, Twitter, public forums, mailing lists and similar platforms etc. By leveraging these openly available resources, the ITM Attack can transcend from being a theoretical concept to actually revealing the identity of a zaddr by linking it to email addresses, social media accounts, location data and other personal data.

The ITM Attack is not suitable for casual users or those with limited budgets, as it requires significant financial resources to carry out effectively. It is best suited for large entities such as the NSA, GCHQ, and other similar groups who likely already employ the techniques and strategies outlined in the attack. Only well-funded private blockchain analysis firms can afford the necessary infrastructure to conduct this attack, but the collected data can be given out to those with little resources. ITM Attack offers extra analysis that can complete many partial de-anonymization by utilizing timing, amount, and fee analyses. It can identify transactions involving multiple zaddrs and predict their input and output figures, providing use information that is not available. The precise figures of the transactions are not as crucial as the fact that they include substantial amounts of money, which directs the examination and inquiry. While complete de-anonymization is unfeasible and unnecessary, the ITM information can be used by special software to determine transaction outputs with figure ranges and possibly related zaddrs from publicly available sources.

Assumptions of ITM Attack.

It is assumed that an attacker has a budget of at least $100,000 to allocate towards studying a single Zcash blockchain. The majority of this cost involves purchasing a GPU/FPGA farm to analyze the blockchain data. The cost of studying a blockchain will increase for those with longer histories and larger shielded pools.

It's important to point out that this attack isn't financially feasible if done only once. Its real purpose is to analyze an entire blockchain, so that the data within can be sorted and searched for any valuable information. Companies specializing in blockchain analysis and intelligence agencies are best suited to make use of this data, as they already have the necessary infrastructure in place, which means it won't cost them much [31].

Defending ITM attacks with zk-SNARKS.

zk-SNARK (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) refers to a way of proving that someone has access to certain data, including secret key, without exposing that data and without any link between the person proving their access and the person verifying it. ITM attack can be considered as defeating zero-knowledge math only in practice, and not in theory. The utilization of zk-SNARKs in higher-level protocols like the Zcash Transaction Format Protocol and other associated rules presents an opportunity for attackers to exploit.

It is mathematically impossible for knowledge to be directly leaked from a zero-knowledge proof, as zk-SNARKs are secure and do not allow for such leakage. The leakage of knowledge occurred due to the application of these proofs in the broader system known as the Zcash Protocol. This protocol is an extension of the Bitcoin Protocol, which is notorious for its metadata leakage. Zcash was the initial extensive use case of zk-SNARKs, an innovative version of zero-knowledge cryptography. The strength of Zcash's privacy protection originates from the concept that its shielded transactions can be fully encrypted on the blockchain, while still being validated using zk-SNARK proofs according to the network's consensus rules. Zcash's

way of validating transactions is by connecting the address of the sender, address of the receiver, output and input values on the public blockchain. In order to demonstrate the validity of a transaction while safeguarding sensitive information about the involved values and addresses, Zcash utilizes zk-SNARKs. The sender of a protected transaction creates evidence demonstrating that the transaction meets the necessary criteria with a high degree of certainty:
the total value of the inputs is equivalent to the total value of the outputs. the individual who initiates the transaction proves their ownership of the private spending keys for input notes, thereby obtaining authorization to utilize them.

The private spending keys associated with the input notes are linked to a signature that covers the entire transaction through cryptography. This approach ensures that the transaction cannot be altered by anyone who does not possess knowledge of these keys.

There are additional criteria that shielded transactions must meet. Zcash employs the term "commitment" to refer to the shielded counterpart of an unspent transaction output (UTXO), and to spend a commitment, an individual must disclose a "nullifier". Lists of both commitments and nullifiers are maintained by Zcash nodes, and to prevent any revealing of information about these commitments or the relationships between nullifiers and commitments, they are stored in hash form. Whenever a shielded payment generates a new note, a commitment is released that comprises a hash containing several elements: the receiver's address, the sum being transferred, a random nonce and a special number known as "rho" (which is subsequently used to obtain the nullifier).

"Commitment" is equal to "HASH of (recipient address, amount, rho, r)"
In shielded transaction expenditure, spender employs its spending key to reveal a nullifier that corresponds to the secret special value, rho from a commitment that has not been spent yet. Along with the nullifier, they submit a zero-knowledge proof that verifies their permission to utilize the commitment. It is essential that the nullifier is not already contained in the set of nullifiers stored by each node on the blockchain, which track the spent transactions.

"Nullifier" is equal to "HASH of (spending key, rho)"
In a shielded transaction, the zero-knowledge proof not only verifies the conditions mentioned earlier but also confirms the validity of the subsequent assertions:
For every input note, a commitment that has been disclosed is present:
Note commitments and the nullifiers have been accurately computed.
It is practically impossible for an output note's nullifier to match any other note's nullifier.

Apart from spending keys that manage addresses, Zcash also employs a group of verifying and proving keys to establish and authenticate proofs. The distribution of these keys, which are established during the previously mentioned public parameter ceremony, is carried out among all individuals involved on the Zcash network. The proving key of the sender is used to create a proof for every shielded transaction, thus ensuring the legitimacy of their inputs. To guarantee that the shielded transaction meets the consensus rules, miners evaluate the computation of the prover using the verifying key. While the proof generation process of Zcash demands more initial work from the prover, it streamlines the verification process, thus transferring the main computational burden to the transaction initiator. As a result, crafting a shielded Zcash transaction may require a few seconds, while verifying the transaction's validity only takes milliseconds [18].

Quantum Attacks on Zcash

Quantum Attack on Zcash's Signature Scheme: Zcash's ECDSA (Elliptic Curve Digital Signature Algorithm) fully depends on the discrete logarithm problem (DLP) arduousness. However, using Shor's algorithm on a quantum computer, the DLP can be solved exceedingly fast [32]. Shor's algorithm explores numbers inputted and finds their prime factors, then it eventually solves the discrete logarithm problem by factorization. As a result of resolving or transforming the discrete logarithm problem to finding the period of a function, number prime factors can be more efficiently obtained with the use of Shor's algorithm period-finding steps. This enables Shor's algorithm to breach cryptographic protocols that are fully dependent on the discrete logarithm problem's hardness. Definitively, breaching the cryptographic protocols eventuates to quantum adversaries stealing transactions intended to be broadcasted to the network ere their addition to the blockchain system.

Attack on Zcash's Cryptographic Protocol: An earlier generation of a global public key is a requirement for Zcash's trusted setup. To avoid a holder of the private key from illegitimately generating tokens of Zcash uncontrollably, the network does not allow the public key to be generated using one user's associated private key. Parts of users' private keys are used to generate a small portion of their public keys involved in a consensus, subsequently concatenating those minimalistic public keys to form a greater public key. Should a user expunge his/her private key, generating the unanimous public key for that associated user's private keys seemingly becomes computationally improbable with the usage of regular computers.
Howbeit, the public key used during the formation of zk-snarks is bereft of an associated private key, even so, it is dependent on the discrete logarithm problem's difficulty. With the use of quantum computers, an attacker can solve the asserted difficult discrete logarithm problem at a

proliferated speed using Shor's algorithm, consequently resulting in the derivation of the private key. It's relevant to note that the adversary cannot use the private key to gain unauthorized access to transactions of other parties on the network, but could generate colossal tokens unrestrictedly. Considering the obfuscation of transactions in the network and the ability of the adversary to limitlessly create transactions at will from generated tokens, attacks by the adversary will be concealed from the rest of the network. This is gravely dangerous to the network as it has the propensity to illegitimately run transactions that can exhaust some of the network's resources and traffic.

Quantum Attack on Zcash's Consensus Mechanism: A quantum-based algorithm was aptly designed by Grassi et al. tailored to the birthday or k-xor problem, which ameliorates the classical algorithm initially formulated by Wagner. This quantum-based algorithm had substantially augmented time and complexity of memory $O(2^{n|(2+log2(k))})$ in comparison to that of Wagner's algorithm $O(2^{n|(1+\log 2(k))})$ [33]. This ensures a massive room for the consensus mechanism to be exploited by a quantum attack, which further culminates to a quantum attack on Zcash's network that is approximately 51% or more. With this attack, the adversary can forge signatures, thus resulting in the compromise of the Zcash network's security. However, this potential attack will require a large number of quantum computers. This is being properly designed and enhanced to cause future exploitation to Zcash's signature scheme.

Mitigation Strategies

As a countermeasure to quantum attacks on Zcash, the development department of Zcash has been designing substantive solutions, which involve the usage of signature schemes that are quantum resistant. Additionally, a strategy to transform into a world that's post-quantum has been in progress and it entails a hybrid consensus mechanism implementation that is defiant to quantum and classical attacks.

CONCLUSION

The rise of privacy-oriented cryptocurrencies like Monero and ZCash prioritizes anonymity and aims to provide enhanced privacy and security features to its users. Monero and ZCash have been subject to attacks that exploit vulnerabilities in their system model. In this paper, we have given an overview of Monero and Zcash, their protocols, their differences and similarities and recent attacks on them. These attacks include anonymity attacks, transaction flooding attacks, side-channel and timing attacks. Despite these challenges, Monero and ZCash continue gaining attention for their privacy-focused approach to cryptocurrency. Making them less susceptible to censorship and government interference. As privacy concerns continue to grow, privacy-focused cryptocurrencies like Monero and ZCash will likely increase. It will be essential for their developers to remain vigilant in maintaining the security of their protocols.

REFERENCES

[1] "What is Monero? and how does XMR work? (2023 edition)," Kriptomat,19January2023.[Online].Available:https://kriptomat.io/cryptocurrencies/monero/what-is-monero/. [Accessed 20 March 2023].

[2] L. S. L. Wijaya, "Monero Ring Attack: Recreating Zero mixin transaction effect," 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018.

[3] P.-D. Kearney, "Vulnerability of blockchain technologies to quantum attacks," Array, vol. 10, p. 100065, 2021.

[4] Estensen, "A Comparison of Monero and Zcash," 2018.

[5] Zcash, "Zcash," ELectric Coin Company, 28 October 2016. [Online]. Available: https://z.cash/technology/. [Accessed 25 February 2023].

[6] S. B. T. H. N. W. Daira Hopwood, Zcash Protocol Specification, Zerocoin Electric Coin Company, September 15, 2022.

[7] Wikimedia, "Wikimedia," Foundation, Inc., 28 October 2016. [Online]. Available: https://en.wikipedia.org/wiki/Zcash. [Accessed 25 February 2023].

[8] O. L. Benjamin Powers, "CoinDeck," CoinDeck, 26 January 2022. [Online].Available:https://www.coindesk.com/layer2/privacyweek/2022/01/26/what-is-zcash-the-privacy-coin-explained/. [Accessed 25 February 2023].

[9] D. F. Alex Biryukov, "Privacy and Linkability of Mining in Zcash," in IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 2019 IEEE Conference on Communications and Network Security (CNS).

[10] H. Y. M. M. S. M. George Kappos, "An Empirical Analysis of Anonymity in Zcash," in 2018 ACM SIGSAC Conference on Computer and Communications Security, Baltimore, MD, USA, 2018.

[11] S. Christensen, "A Comparative Study of Privacy-Preserving Cryptocurrencies: Monero and ZCash," University of Birmingham, Birmingham, 2018.

[12] D. Salazar, "LocalMonero.co," Blue Sunday Limited, 2nd December 2021.[Online].Available:https://localmonero.co/knowledge/monero-trustless-setup?language=en. [Accessed 14 March 2023].

[13] D. K. Alex Biryukov, "Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem (Full version)," University of Luxembourg, Luxembourg.

[14] G. Ayala, "Bit2me Academy," 17 August 2020. [Online]. Available: https://academy.bit2me.com/en/que-algoritmo-mineria-randomx-monero/#:~:text=RandomX%20is%20the%20name%20of,to%20its%20protocol%20and%20blockchain%20.. [Accessed 18 March 2023].

[15] C. Sephton, "Currency.com," 10 August 2022. [Online]. Available: https://currency.com/monero-vs-bitcoin-the-pros-and-cons. [Accessed 20 March 2023].

[16] S. Jahandideh, "Bourseiness.com," Bourseiness, 9 December 2018. [Online].Available:https://www.bourseiness.com/en/84/zcash#:~:text=Zcash%20(ZEC)%20has%20a%20total,through%20a%20POW%20Equihash%20algorithm.. [Accessed 20 March 2023].

[17] Monero, "Monero," [Online]. Available: https://www.getmonero.org/. [Accessed 22 March 2023].

[18] Zcash, "Zcash," [Online]. Available: https://z.cash/technology/. [Accessed 20 March 2023].

[19] J. FRANKENFIELD, "Investopedia," 9 February 2023. [Online]. Available: https://www.investopedia.com/terms/p/proof-work.asp. [Accessed 22 March 2023].

[20] D. B. K. P. Florian Tramer, "Remote side-channel attacks on anonymous transactions," 29th USENIX Security Symposium, 2020.

[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, Available: http://bitcoin.org/bitcoin.pdf.

[22] N. van Saberhagen, "Cryptonote v 2. 0," 2013.

[23] S. Meiklejohn et al., "A Fistful of Bitcoins:Characterizing Payments Among Men with No Names," USENIX ;login: 2013.

[24] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in Financial Cryptography and Data Security, 2013, pp. 6-24:Springer.

[25] A. Kumar, C. Fischer, S. Tople, and P. Saxena, "A Traceability Analysis of Monero's Blockchain," IACR Cryptology ePrint Archive, vol. 2017, p. 338, 2017.

[26] A. Miller, M. Möser, K. Lee, and A. Narayanan, "An Empirical Analysis of Linkability in the Monero Blockchain," arXiv preprintarXiv:1704.04299, 2017.

[27] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups," Australasian Conference on Information Security and Privacy, pp. 325-335, 2004.

[28] Chervinski, J.O., Kreutzz, D., and Yu, J. (2021). Analysis of Transaction Flooding Attacks against Monero. In 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-8). IEEE. DOI: 10.1109/ICBC51069.2021.9461084.

[29]. Tramèr, Florian, et al. “Remote Side-Channel Attacks on Anonymous Transactions.” *Proceedings of the 29th USENIX Security Symposium*, USENIXAssociation,1,Jan.2020,www.research-collection.ethz.ch/handle/20.500.11850/498961?show=full.

[30]. Zurich, Samuel Steffen ETH, et al. “Zapper: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security.” *ACM Conferences*, 1 Nov. 2019, dl.acm.org/doi/10.1145/3548606.3560622.

[31]. D. Leto, "ITM Attack: z2z Transaction Linkability," *Attacking Zcash Protocol For Fun And Profit,* pp. 7-9, 2020

[32]. Joseph J. Kearney, Carlos A. Perez-Delgado, “Vulnerability of blockchain
technologies to quantum attacks”, School of Computing, University of Kent, Canterbury, Kent CT2 7NF, United Kingdom.

[33]. Grassi L, Naya-Plasencia M, Schrottenloher A. Quantum algorithms for the k-xor problem. In: International conference on the theory and application of cryptology and information security. Springer; 2018. p. 527–59.