Monero Ring Attack: Recreating Zero Mixin Transaction Effect

Dimaz Ankaa Wijaya[1], Joseph Liu[1], Ron Steinfeld[1], Dongxi Liu[2]
[1]Faculty of Information Technology, Monash University, Australia
{dimaz.wijaya, joseph.liu, ron.steinfeld}@monash.edu
[2]Data61, CSIRO, Australia
dongxi.liu@data61.csiro.au

Abstract— Monero is one of the privacy-preserving cryptocurrencies employing CryptoNote protocol. The privacy features in Monero are provided by cryptographic techniques called linkable ring signature and one-time public key. Recent studies show that the majority of Monero inputs are traceable prior to mandatory RingCT transaction. After the RingCT was implemented, the problem was mitigated. We propose a novel attack to reduce the anonymity of Monero transactions or even to fully deanonymise the inputs. The proposed protocol can be launched in RingCT scenario and enable multiple attackers to collaborate without trusting each other. The attack scheme can be planted in the existing Monero services without extra fees and without putting the users' money at risk. (*Abstract*)

Keywords—Monero; ring signature; anonymity; privacy; traceable (key words)

I. Introduction

Monero is one of the most valuable privacy-preserving cryptocurrencies in the world. It is built based on blockchain technology where the transaction data is visible to everyone, similar to the one implemented in Bitcoin [1]. But, unlike in Bitcoin where anyone can track the flow of money between addresses, in Monero the observers cannot do the same. Ring signature and one-time public key technologies are implemented as the default settings to improve the anonymity of the transaction data. The real senders are obfuscated by adding multiple decoys where the set of possible senders are equal and cannot be distinguished among each other. The one-time public key means for each output there will be a unique address being created, while the real address of the receiver is never revealed in the blockchain. Without any additional information, it is infeasible to determine which addresses belong to a specific user.

Despite the privacy-preserving methodologies were already implemented in Monero, there are at least 4 different analyses that have been developed to reveal hidden information in Monero environment. These analyses were successfully conducted due to the transparency of the blockchain data, liquidity problem, and identified users behavior.

We propose a novel attack against the Monero untraceability. The proposed attack can be used to reveal the real outputs being spent in Monero transactions or at least reduce the anonymity of the inputs. The attack scheme can be conducted by a single attacker or multiple attackers colluding to launch the scheme without the need of trusting each other. Each attacker will take the benefit of others' results. Our attack is effective to be conducted in RingCT environment where the transaction amount cannot be seen by an observer. Constructing a "malicious transaction" as described in the proposed attack scheme will not cost an extra fee if the attack is attached into an existing service.

II. Background

A. Monero

Monero is a cryptocurrency forked from an existing cryptocurrency called Bytecoin. Both of the coins are based on a protocol called CryptoNote, a protocol proposed by Nicolas van Saberhagen (pseudonym) in 2013 [2]. The focus of the protocol is to create a privacy-preserving cryptocurrency. In Bitcoin, there are problems related to the anonymity of the users. Previous studies were able to determine information regarding the bitcoin users and what activities they conduct by using bitcoin [3]. Moreover, a graph analysis discovers the transaction patterns by malicious actors [4].

CryptoNote offers a better anonymity feature by employing linkable ring signature to ensure the untraceability and one-time public key to bring the unlinkability into the new system. These features are implemented in protocol level which become the mandatory procedure for all users of the system.

The main feature of Monero is the existence of "plausible deniability". It is infeasible to determine which public keys are being spent in the transactions over a set of public keys (there is only 1 public keys being spent in an input constructed by several public keys). Hence, other public keys are the decoys (fake ones). Despite this feature is the main appealing feature of Monero, limitations in its implementation hinders the system to reach its full potential. Analyses have shown that a large part of Monero transactions can be traced [5, 6].

As with any other cryptocurrencies, there are at least 2 participants that make up the environment: daemons and wallets. Monero daemon is a server providing information to the clients. Monero daemon synchronizes the blockchain data to its peers and keep a complete record of all transactions in a local storage. Monero wallet is an application which helps the users to manage their wallets, detect if they receive new payments, calculate balance, and create new transactions. Monero wallet does not keep a blockchain in the local storage. Instead, it creates requests to Monero daemon for any information required by Monero wallet to keep the data updated.

There are different products of Monero wallet available in the market. The first and probably the main one is the wallet provided by the core developers, which is monero-wallet-cli. Now the wallet is equipped with a GUI version. The second wallet is an online wallet called MyMonero[1]. It is a web-based wallet which can be used to create a new wallet, create transactions, and scan the blockchain to calculate the current balance. OpenMonero[2] is the open source version of MyMonero with similar interface but better compatibility with monero-wallet-cli by employing the same 25 words seed. The third product is an Android-based wallet called Monerujo[3] which is also an open source project[4].

OpenMonero and Monerujo use the same codebase as the official monero-wallet-cli to handle the Monero computations, only they use a different interface. Another Monero wallet provided by Freewallet[5] is closed-source and not recommended by the Monero community since the users do not hold their private keys.

When creating a transaction, Monero Wallet cannot work by itself; it requires information supplied by the Monero Daemon. It is because in Monero, each real output to be spent in an input needs to be obfuscated with several other outputs (decoys). These decoys are often called as mixins. The decoys together with the real output are used to construct ring signature. The number of the decoys and the real output is called ringsize.

The decoys are real public keys already showing up in the blockchain. In other words, these decoys are outputs of other transactions. These public keys are grouped based on the amount of coins contained in the public keys and then indexed sequentially based on their appearance in the blockchain in time series.

First, the Monero Wallet requests for a “histogram data”. It is an information of the maximum index for each amount of every outputs in the blockchain. Based on this histogram data, the Monero Wallet picks multiple indexes. The number of indexes exceeds the ringsize. For RingCT transaction, the indexes will be picked from the histogram with the amount of 0 (since all amount information in RingCT transactions are encrypted, the system is unable read the information and marks the amount as 0 although it might not actually 0).

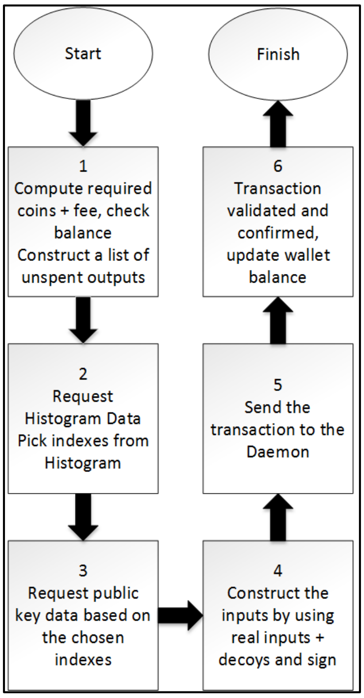
Start
Finish
1
Compute required coins + fee, check balance
Construct a list of unspent outputs
2
Request Histogram Data
Pick indexes from Histogram
3
Request public key data based on the chosen indexes
4
Construct the inputs by using real inputs + decoys and sign
5
Send the transaction to the Daemon
6
Transaction validated and confirmed, update wallet balance

Fig. 1. How Monero transaction is created

B. *Monero Anonymity*

The anonymity of Monero is broken down into 2 parts: unlinkability and untraceability [2]. Unlinkability is defined as for any 2 different transactions, it is impossible to decide whether they are sent to the same person, while untraceability is defined as for a set of inputs, it is impossible to decide which input is being spent by the transaction [5]. Based on the definition, unlinkability is about protecting the receiver, while untraceability is about protecting the sender.

Both untraceability and unlinkability are included in the CryptoNote protocol as the focus of the system. The untraceability is reached by employing ring signature. The unlinkability is ensured by using one-time public key.

1) *Ring Signature*

Rivest, Shamir, and Tauman were the first to propose ring signature to leak a secret to public [7]. The leak ensures that it comes from a reputable source (e.g. from a company’s board of director), but the person leaking the secret does not want anyone to learn that he is the one leaking the secret. The ring signature makes it possible for the information to be signed by using a private key that corresponds to a public key which is included in a set of public keys. Nobody will be able to determine which public key is the one signing the transaction.

The ring signature construction in CryptoNote was derived from previous works on linkable ring signature [8] and traceable ring signature [9]. These constructions ensure that signatures can be determined to be signed by the same public key if the public key is used to sign more than once. The characteristic is important in cryptocurrency to avoid double spending. Double spending is an event where a coin (or a balance) is spent more than once during its lifetime. If

[1] https://mymonero.com
[2] https://github.com/moneroexamples/openmonero
[3] https://monerujo.io
[4] https://github.com/m2049r/xmrwallet
[5] https://freewallet.org/currency/xmr

double spending can occur in any cryptocurrency, then the coins in the system is worth nothing and cannot be used as a medium to store value [10].

In Monero, the ring signature is constructed by combining several existing outputs (called decoys or mixins) which have the same amount of coins into a single input. These outputs must have a real output which will be spent in the transaction. A transaction might have multiple inputs and multiple outputs as well.

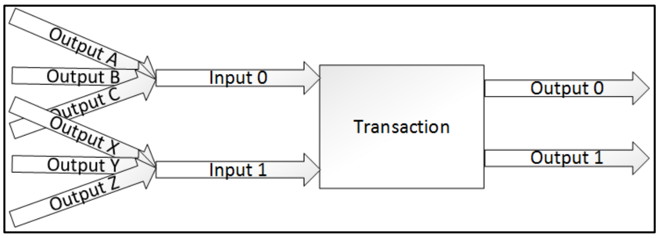

Fig. 2. The structure of a Monero Transaction. Multiple outputs in an input constructed by mixing other outputs with the real output.

The purpose of the ring signature is to reduce the possibility of an adversary to guess a real output over a number of outputs N. The probability (P) of guessing the real output being spent in an input is denoted as

$$P = \frac{1}{N}$$

2) One-Time Public Key

The one-time public key employed in the CryptoNote protocol is somewhat similar to the stealth address which was created in Bitcoin ecosystem [11, 12]. In the scenario, the receiver sends a "parent public key" to the sender. The sender then generates new "child public keys" by using secret keys, which are in1cluded in the transaction data in encrypted format [13].

The receiver scans the network for new transactions and compute the secret key of each transaction with the parent private key she holds. If the result matches the destination key, then she includes the transaction in her wallet as an incoming transaction.

By using the one-time public key, it is assured that only the sender and the receiver knows the relations between the parent public key and the child public keys involved in the transaction, while an observer cannot analyse the relationship between the public key and the child keys without any additional data, although the observer has access to the blockchain.

3) Ring Confidential Transaction

Ring Confidential Transaction (RingCT) is a feature in Monero which was first deployed in Wolfram Warptangent (version 5) and has become a mandatory in Helium Hydra (version 6). The first block containing a RingCT Transaction is block number 1,220,517 in the Monero blockchain. RingCT is a technique combining Ring Signature and Confidential Transaction [14]. It is developed to add Confidential Transaction feature into the existing Monero system which uses ring signature [15].

The purpose of the RingCT is to mitigate the problem of liquidity by hiding the amount of coins contained in the public keys. The main requirement for constructing a ring signature is that each ring member must hasve the exact same amount of coins and therefore the real one cannot be distinguished from the decoys.

There is a problem of liquidity that the users cannot construct a transaction with enough decoys that they employ zero mixin transaction [16]. The zero mixin transaction does not have any decoys. Hence, it is traceable because there is 100% chance of guessing which public key is being spent in the zero mixin transaction, unlike the ones with decoys. As RingCT transaction outputs will be marked as having 0 coin, therefore the number of decoys can be selected from a large pool of public keys.

C. k-*Anonymity*

The term *k*-anonymity is used to model data privacy where an information within a set of *K* cannot be distinguished among other *k*-1 elements of *K* [17]. In a ring signature that contains > 1 elements, the anonymity of each element depends on other elements such that if any elements *n* can be removed from the anonymity set, each remaining element has *k-n* anonymity.

In Monero, *k*-anonymity can be used to draw the anonymity level of every input containing multiple outputs as the decoys. The anonymity of the real input depends on the indistinguishability of each decoy and the number of decoy used.

D. *Threat Model*

We define the threat model in Monero as follows. Everyone has the ability to see all information stored in Monero blockchain. The security of the confirmed transactions depend on the consensus model of Monero. We also define 2 types of attacker, Attacker A and Attacker B. The Attacker A has a sufficient fund to create standard transactions and modified transactions, but does not have any access to coin exchanges or wallet services software.

There exists a group of Attacker A = [A_1, A_2, A_3, …] colluding to attack the system but they do not trust each other. The Attacker B has all ability owned by the Attacker A plus the ability to modify coin exchanges or wallet services software. There also exists a group of Attacker B = [B_1, B_2, B_3, …] colluding without trusting each other. The Attacker A and Attacker B can also collude to get the best result out of their efforts. Attackers conduct all phases in the proposed method. There also exists observers who are not interested to craft transactions but curious towards the attack result and its impact. There also exists users who are using the wallets and creating transactions but not interested to evaluate the privacy of their activities.

III. Known Attacks Against Monero Anonymity

A. *Black Marbles Attack*

This paper uses the term "Black Marbles Attack" to refer to an attack against the Monero anonymity by controlling as many outputs as possible in the Monero blockchain [16]. It is said that an attacker, tries to control more outputs in the Monero blockchain. If all the outputs of the blockchain are assumed as marbles in an urn, the black marbles are the outputs controlled by the attacker, while the white marbles are the honest outputs created by the users. The urn is the shared ledger (the blockchain) where the black marbles (bad outputs) and the white marbles (good outputs) are stored and visible to all observers.

To maximise the impact of the attack, the attacker needs to create more outputs (the black marbles) to have more number than other users' outputs (the white marbles). This is achieved by sending the coins back to her own address [18]. Since there is no information whether an output has been spent, the attacker needs to constantly add more black marbles to increase the probability of her outputs being picked up by new transactions as decoys.

B. *Zero Mixin Transaction and Cascade Effect*

In Monero, zero mixin transactions are a transaction which have at least 1 input using no decoys or mixins. Zero mixin transactions do not have any anonymity feature offered by ring signature and therefore, any observers can immediately trace the real sender of the transaction. The anonymity problem do not happen only for the zero mixin transactions, but also for every other transaction that happen to use the same outputs as their decoys which were proved to be spent by the zero mixin transactions.

The ring signature is an effective method to create a plausible deniability for untraceability under an optimum environment: there exist enough outputs that share identical characteristics such as age and amount of coins contained in the outputs). Unfortunately, this environment could not be sufficiently provided by Monero prior to the release of the mandatory RingCT usage.

Although the users were urged to split their transactions according to a specific denomination regulation, this regulation was never strictly applied. A unique amount of coins can still be confirmed in the transaction although it will create a liquidity problem where the user cannot find other outputs containing the exact same amount of coins. For all outputs that cannot be combined with any other outputs, the users create a zero-mixin transaction: an input contains only the real output without any mixin or decoys.

Although the zero mixin transactions were described as having a cascade effect towards the anonymity of other transactions [16, 18], new investigations show that the impact is greater than expected. Based on techniques presented in the previous research, the effect is reaching the rate of 87% [5] and 62% [6]. It means that at least more than half of all analysed inputs (prior to RingCT) can be distinguished between the decoys and the real outputs.

C. *Temporal Analysis*

The zero-mixin transaction analysis also reveals that in most cases, the real outputs being spent are the most recent outputs [6]. The extrapolation of the gathered data mentions that 80% of the real outputs that can be detected are the newest. The uniform mixin sampling used by the system could not hide this characteristics.

New sampling methods, triangular distribution, were introduced to tackle the problem. Triangular distribution protocol describes that at least 25% of all decoys must be taken from recently added outputs. By using this technique, it is expected that the temporal analysis is nullified because at least 1 in the mandatory minimum of 5 mixins in the recent version of Monero (Helium Hydra) is aged less than 5 days.

D. *Publishing Private Viewkeys*

The private viewkey is a feature within Monero system to provide an auditability of the coins owned by a user. Assuming that the user provides the private viewkey of her wallet to an auditor, the auditor is then able to track every coins received by the associated address. Although the private viewkey enables such thing, it is impossible for the auditor to steal the coins from the user by using the private viewkey. It is also impossible for the auditors to determine whether the coins have been spent.

The private viewkey can also be utilized to launch an attack to Monero unlinkability. It is assumed that the anonymity of a user depends on the anonymity of other users. The private viewkeys can be used to distinguish between the outputs sent to the owner of the private viewkeys and the outputs sent back to the sender (the change). Although a private viewkey can be used to determine all outputs destined to the address of that private viewkey, but it cannot determine whether the outputs have been spent by the associated private spending key.

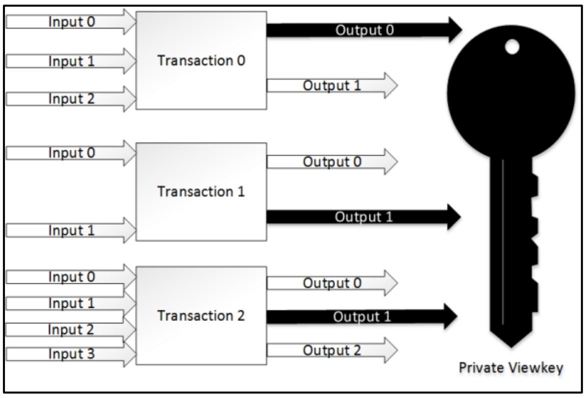

Fig. 3. A Private Viewkey Determines All Incoming Outputs (Payments). The private viewkey can determine the outputs sent to the corresponding address by scanning all transactions in the blockchain.

The private viewkey is considered as a feature rather than a weakness. It is used mainly for for cases requiring compliance, e.g. auditing and charities [19]. Although, after the audit it is not possible to regain the unlinkability feature without creating a new address and move all the balance to the new address.

E. *Our Proposed Attack Compared to Existing Attacks*

In the Black Marbles Attack, an attacker is supposed to create new outputs by creating transactions, matching the number of outputs created by other users. The more outputs the attacker has, the better chance the attacker reduces the anonymity of other users. This type of attack can only be done individually, or if the attack is a coordinated attack by several attackers, each attacker needs to trust others when determining whether an output is a product of the attack or not. The Black Marble Attack can also be combined with publishing private viewkeys and send the transactions to the attacker's address. But by doing so, the transactions are not doing any other purpose and therefore the transaction fees paid to the miners are wasted.

Compared to the Black Marble Attack, our proposed attack is better in a coordinated attack scenario of multiple attackers. Each attacker conducts the attack and the result of the attack can also be evaluated by other attackers without any additional information such as private viewkeys. Our proposed attack's transaction does not need to be sent to our own address and therefore can be easily implemented in existing online services such as exchanges or wallet services. The exchanges and the wallet services do not need to spend any extra transaction fees since they only need to implement the technique into the system and convert their regular transactions into malicious transactions where only the anonymity of the transactions are reduced.

Our proposed attack does not rely on the existence of zero mixin transaction which becomes obsolete when Monero was upgraded to have RingCT and mandatory minimum number of mixins. The setup phase recreates the similar impact of the zero mixin transaction, and the attack phase recreates the cascade effect of zero mixin transaction.

Our proposed attack is even more effective to be launched in RingCT-enabled system, because an attacker does not need to attack multiple coin denominations and only focus on one denomination. Moreover, the RingCT enables the attacker to use a small amount of coins. The system cannot detect the amount of coins sent and therefore even the attacker sends 0 coin, the system will still accept it.

Our attack has a higher precision rate compared to temporal analysis. In temporal analysis, the attack depends on how the decoys are selected among all available outputs in the system. If the selection algorithm is optimum, then temporal analysis cannot determine the real output being spent in the input. Our attack can precisely determine the real output in the input with 100% accuracy. The summary of the comparison can be found in Table I.

TABLE I. COMPARING ATTACK METHODS

Factors	BM	ZM	TA	PPV	Ours
Collaboration between attackers	X	V	X	X	V
Requires no extra fees	X	V	V	X	V
RingCT resistant	V	X	V	V	V
Minimum mixin resistant	V	X	V	V	V
Accuracy in determining real outputs	V	V	X	V	V

IV. OUR PROPOSED ATTACK

A. *Overview*

The proposed attack scheme utilises the leniency of the Monero daemon towards the transaction creation by the monero wallet. Monerod only checks for the validity of the transactions submitted to the server in which those transactions require correct balances and valid digital signatures The ring construction during ring signature creation is entirely processed by the wallet. Monero daemon helps the wallet by providing public keys information based on indexes picked by the wallet.

Based on the given information, it is possible to construct a malicious transaction to reduce the *k*-anonymity or even de-anonymize Monero transactions. The impact is similar to cascade effect from the zero mixin transaction.

B. *The Proposed Method*

The proposed attack is divided into 3 phases: preparation, setup, and attack. The attack phase has 2 different methods, the passive and active attack. Each of the phases will be explained below.

1) *Preparation Phase*

For each thread of attack, the attacker needs to have a number of unspent outputs. The number of outputs depends on the minimum ring size *r* required by Monero system. For version 6 (Helium Hydra), the minimum ring size *r* is five and therefore the minimum number of outputs required by the attacker is equal to that number. The purpose of the preparation phase is to have a set of unspent outputs in which every single output will be spent in the setup phase. If the attacker has more unspent outputs than the minimum ring size but less than multiples of *r*, then the remaining outputs can be used on attack phase.

Since the deployment of RingCT, it is not necessary to have the same amount of coins for each output. Therefore, an attacker can use a small amount of coins split across multiple outputs. It means that an attacker can only focus on paying the transaction fees and do not need to have extra reserved coins.

The number of threads to be created by an attacker depends on the type of attack that the attacker wants to use. If the attacker intends to launch a passive attack, then the attacker needs to create as many threads as possible. The success of the attacker depends on the number of threads created by the attacker, whereas in the active attack, the attacker only needs to create one thread. The outputs will be reused in the forthcoming transactions which will not reduce the effectiveness of the attack, although it might rise suspicion if a certain output is reused many times.

2) *Setup Phase*

In the phase, it is required to create exact *r* inputs for each attack thread. It means there will be *r* ring signatures created by the attacker. Each ring signature will spend a transaction output owned by the attacker. Let a set of *l* public keys L = [PK_A, PK_B, PK_C, PK_D, PK_E, …] and their secret key image pairs K = [I_A, I_B, I_C, I_D, I_E, …]. The number of public keys in L is equal to *r*. The decoys for each ring signature is chosen from *L* as shown in Fig. 4.

The inputs can be included in a transaction or multiple transactions, but it is more cost-effective to have *r* inputs all in the same transaction. The setup phase has a similar effect as the zero mixin transaction, with one difference. In the zero mixin transaction, anyone can precisely determine which input spends an output. In this setup phase, it is infeasible to determine the exact input that spends a particular output. We can only say that all inputs in the setup phase spend all members of *L* regardless of which input spends which output. The focus of the setup phase is to nullify the probability of other transactions spending any member of *L*.

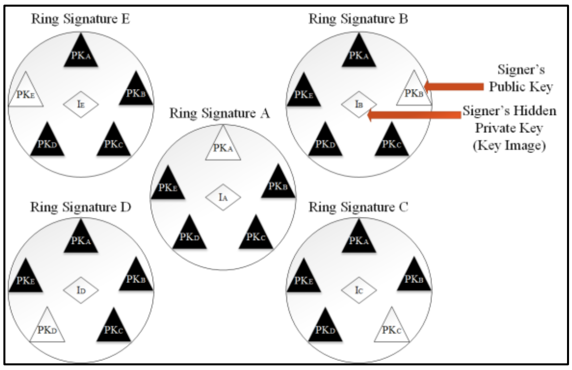

Fig. 4. The Setup Phase Where $r = 5$

If the attacker does not create *r* inputs where all of the outputs are the member of *L*, then the requirement to recreate the zero mixin transaction effect is not fulfilled. Other observers cannot detect whether the inputs have been spent and therefore the attack phase cannot be conducted.

3) *Attack Phase*

There are 2 types of attack which can be launched: passive attack and active attack. Each attack has different purposes and different methods. Both will be further described.

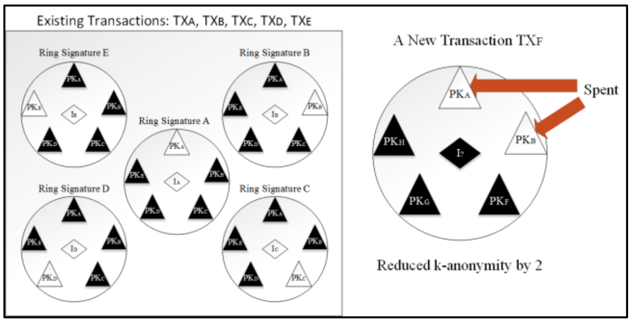

Fig. 5. The Passive Attack

Passive attack. The purpose of the passive attack is to allow the outputs spent (members of *L*) in the setup phase to be used by other users in multiple transactions. If it happens, the observer can omit the public keys as they have been spent by transactions created in the setup phase and it is not possible to re-spend the public keys in any other transactions. These transactions suffer a reduced *k*-anonymity. The degree of the reduced *k*-anonymity depends on the number of decoys coming from the transactions in the setup phase. The example in Fig. 5 depicts a case with a reduced anonymity by 2, according to the number of spent public keys used as decoys.

Active attack. Let there be a malicious Monero wallet service run by Attacker B. The purpose of the wallet is not to steal the coins owned by the users but to make the transactions traceable. The wallet knows the public keys *L* and use them as decoys in the ring signature as in Fig. 6.

The active attack is efficient when targeting others' outputs, especially when the attack protocol is implemented in a wallet. The user might not be able to determine the malicious behavior of the wallet as long as they create the transactions successfully.

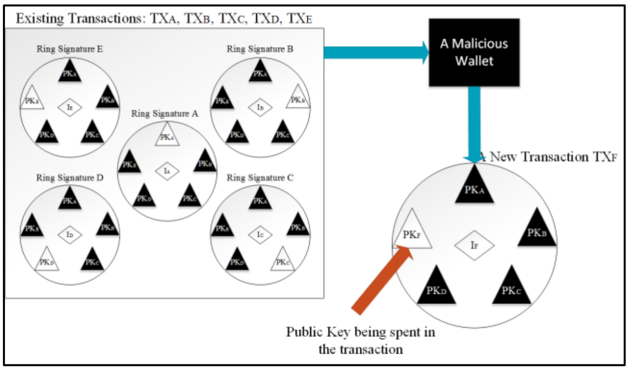

Fig. 6. The Active Attack

The difference between the passive attack and the active attack is that in the passive attack, the attacker conducts a brute force attack to reduce the anonymity of other transaction, while in the active attack, the attacker is able to completely remove the anonymity of the inputs. In the passive attack, it is not required to setup any services (coin exchanges or wallet services) to be used by the users, while in the active attack, these services are compromised and the attack only happens to the service users.

V. Evaluation

A. *Proof of Concept*

As a proof of concept of our proposed attack, we conducted the preparation phase, setup phase, and passive attack phase. We modify the source code of Monero to create a malicious wallet with the ability of creating transactions which comply with our scheme, in particular `simplewallet.cpp` and `wallet2.cpp`. The flowchart of our malicious wallet is shown in Fig. 7.

1) *Preparation Phase*

Instead of having a normal protocol when picking indexes from the histogram data, we pick the indexes from the public keys stored in our own wallet. Therefore, as the wallet also stores the global index for each outputs, it is not necessary to create any requests to the daemon for the data as the wallet itself can supply all necessary requirements.

We have successfully launched the preparation phase and setup phase into the Monero mainnet. The setup phase consists of 1 thread with *r* = 5. The transaction ID for the preparation phase is

`b6781f2a6f5608553546442b84888346fdc3f78d d8995170180ed74081c05362`

2) Setup Phase

We have executed the setup phase with transaction ID of

`8d4a0c7eccf92542eb5e1f09e72cc0d934b180b7 68bc95388d33051db83194bb`.

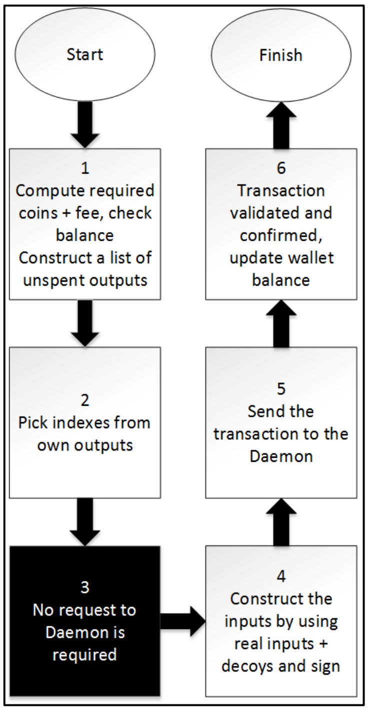
Start
Finish
1
Compute required coins + fee, check balance
Construct a list of unspent outputs
6
Transaction validated and confirmed, update wallet balance
2
Pick indexes from own outputs
5
Send the transaction to the Daemon
3
No request to Daemon is required
4
Construct the inputs by using real inputs + decoys and sign

Fig. 7. Flowchart of the attack transaction

3) Passive Attack Phase

In our setup phase, we have a set of 5 public keys which can be determined to be spent by a transaction. These public keys were picked by 12 other inputs as one of their decoys. It means by using our 5 public keys, we can reduce the anonymity of other 12 inputs by 1. The success of confirming the setup phase transaction to the Monero blockchain proves that the system does not check the ring signature construction. We do not conduct the active attack phase because the transaction creation in the active attack phase is trivial. The number of impacted transactions really depend on other users, and the spent outputs can still be picked up as decoys long after the malicious transactions were confirmed in the blockchain due to the random sampling method that is implemented in Monero.

To see whether our method has been used in the Monero system, we extracted Monero blockchain data into RDBMS format from block number 0 up to 1,470,000. We use a hash function to hash the output members of each input contained in the blocks and compare the hash values to find duplicates.

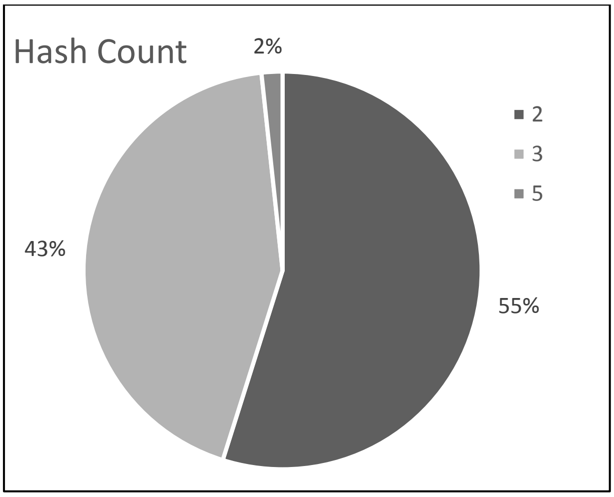

Fig. 8. Percentage of Duplicates Based on Ring Size

For a minimum ring size of two, we discovered 2,947 ring duplicates which resemble our setup phase (including our own transaction). These duplicates consist of 1,244 distinct sets and included in 885 different transactions. The first duplicate was found in block 47,410 while the last one was found in block 1,401,899. Our transaction was included in block 1,468,439.

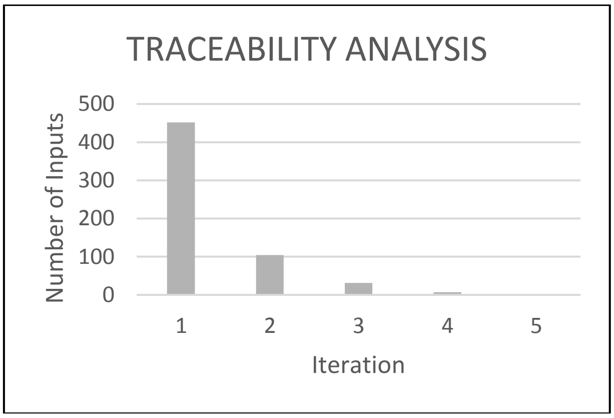

Fig. 9. Traceability Analysis by Using Passive Attack Analysis

The diagram in Fig. 8 describes the statistics of the ring duplicates based on the ring size. More than half of the duplicates have ring size of two, while 43% of them have ring size of three. A small fraction of the data (2%) have ring size of five. All of these transactions were created without RingCT.

Using the passive attack scheme, we managed to find 595 inputs. The iteration process of the passive attack scheme was done up to five iterations. We then draw all inputs we have determined to be spent in Fig. 10 based on the ring size. Out of 595 inputs, 72% of them have a ringsize of three, while the others 28% have a ringsize of two. Two inputs have a ring size of four, while only 1 input has a ring size of five.

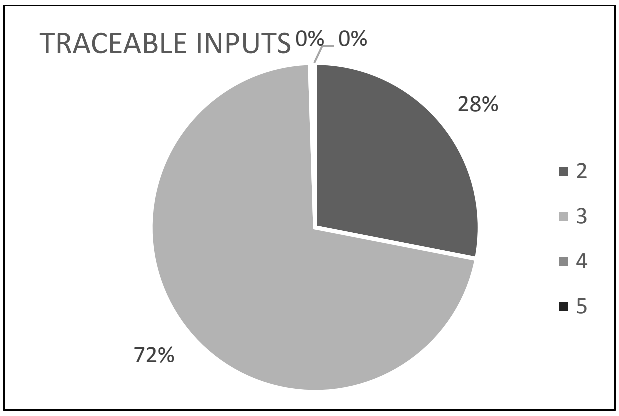

Fig. 10. Traceable Inputs Based on Ring Size

If the outputs that have been determined to be spent are used by other transactions, then these outputs can be omitted when guessing the real inputs. We found 66 other inputs to suffer a reduced anonymity by 1.

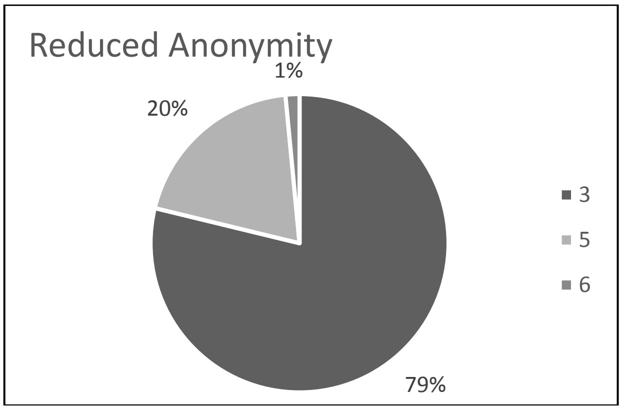

Fig. 11. Reduced Anonymity Based on Ring Size

B. *Cost Analysis*

To set up 5 public keys as the attack outputs, it is required to create 2 transactions: the first one for the preparation phase, and the second one for the setup phase. In our examples, the preparation phase requires around 0.034 XMR, while the setup phase requires 0.0135 XMR and the total cost is 0.0475 XMR under a default setting. With the current market stands at US$216.14, the value of the Monero paid to the miner was US$10.27.

C. *Comparing The Result with Zero Mixin Cascade Effect*

We reproduced the procedures described in [5, 6]. The extraction process is done by utilizing APIs provided by Onion Monero Blockchain Explorer[6]. The Onion Monero Blockchain Explorer works as the gateway to map indexes into real public keys of transaction mixins taken from Monerod fully synchronized to the network. Then we compare the results of the previous known techniques with our findings.

The result of this comparison shows that none of our findings was detected by the known techniques. It is possible that the outputs involved were having a liquidity problem and the owners of those outputs combined these outputs into a smaller number of outputs. We were unable to compare our results with the data provided by MoneroLink[7] due to differences in output indexing. The MoneroLink system does not provide any information regarding the detail of each outputs and therefore it is infeasible to determine the indexing methods being used by the system.

VI. Limitation, Conclusion And Further Work

A. *Limitation*

In Monero, it is infeasible to determine the owners of the coins, since a public key can only be spent once during its lifetime. Therefore it is also infeasible to calculate the number of transactions created by coin exchanges to calculate the impact if any regulation is enforced to the coin exchanges to craft such transactions.

B. *Conclusion*

We have proposed and demonstrated a new attack against the untraceability of Monero system. We showed that the anonymity of the system relies on the implementation of the wallet and the construction of each transaction. Our attack explores the weakness of the ring signature assumption where the sampling for the mixins is assumed to be always random.

Malicious wallets can break the users' anonymity without necessarily stealing the money owned by the users. Detecting such activities require efforts by scanning all existing combinations of the ring construction ever existed in the blockchain and it is unlikely that unaware users detect such activity as long as they do not lose their money.

In case the governments want to reveal the traceability of ring signature-based privacy-preserving cryptocurrencies, these governments can enforce regulations to companies providing coin exchange and wallet services to construct transactions as defined in our proposed attack. Although the regulations might not be able to trace every transaction happening in the blockchain, but a part of the transactions might be able to be traced. By enforcing the regulation, the companies do not need to provide extra money to construct such transactions other than an effort to modify their wallets.

Unlike the Black Marbles Attack, our attack can be launched by many attackers. Each attacker will benefit from other attackers' activities since the transaction output's untraceability is permanently damaged and can be used by anyone having access to Monero blockchain. There is no need to trust other attackers regarding the data exchanged between them, because the data confirms the correctness of the attack. Therefore, this type of attack can be deliberately done by coin exchangers or wallet providers. The governments can enforce regulations for these companies to do such action.

Based on our research, it is important to enhance the protocol to protect the anonymity of the users without trusting the wallet, since the wallet can construct a

[6] https://github.com/moneroexamples/onion-monero-blockchain-explorer

[7] http://monerolink.com

transaction which will reduce or eliminate the anonymity of the transaction. Detection and blacklisting are two alternative methods to avoid such attack.

C. Further Work

Our proposed attack method can also be implemented in other systems where ring signature scheme is used, e.g. electronic voting (e-voting). Such system suffers our attack in the following scenario. Suppose there is an election where multiple candidates compete for a position in the government. A candidate wants to “buy votes” from the voters in order to win the election. Since the e-voting is employing ring signature, the candidate buys the votes in bulk. In order to do this, a vote-seller coordinator is required. The coordinator lists all vote sellers and creates groups of these sellers based on the ring size n used by the e-voting system. Each group consists of n vote seller. Then, each group member informs her public keys to be used as one of the decoys by other members. Each group constructs n transactions having identical ring members to cast votes for the candidate. The transaction construction resembles our proposed setup phase.

REFERENCES

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

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

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

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

[5] 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.

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

[7] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *International Conference on the Theory and Application of Cryptology and Information Security*, 2001, pp. 552-565: Springer.

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

[9] E. Fujisaki and K. Suzuki, "Traceable ring signature," in *Public Key Cryptography*, 2007, vol. 4450, pp. 181-200: Springer.

[10] M. V. Alstyne, "Why Bitcoin has value," *Commun. ACM,* vol. 57, no. 5, pp. 30-32, 2014.

[11] P. Todd. (2014, October 8, 2015). *Stealth Addresses*. Available: http://sourceforge.net/p/bitcoin/mailman/message/31813471/

[12] unSYSTEM Wiki. (2014). *DarkWallet/Stealth*. Available: https://wiki.unsystem.net/en/index.php/DarkWallet/Stealth

[13] S. Noether and S. Noether, "Monero is Not That Mysterious," 2014.

[14] G. Maxwell, "Confidential Transactions," *URL: https://people.xiph.org/~greg/confidential_values.txt (Accessed 09/05/2016),* 2015.

[15] S. Noether and A. Mackenzie, "Ring confidential transactions," *Ledger,* vol. 1, pp. 1-18, 2016.

[16] S. Noether, S. Noether, and A. Mackenzie, "Mrl-0001: A note on chain reactions in traceability in cryptonote 2.0," Technical report2014.

[17] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 05, pp. 557-570, 2002.

[18] A. Mackenzie, S. Noether, and M. C. Team, "Improving Obfuscation in the CryptoNote Protocol," 2015.

[19] Moneroblocks. *Richlist*. Available: https://moneroblocks.info/richlist