Anonymity Reduction Attacks to Monero

Dimaz Ankaa Wijaya[1](✉), Joseph Liu[1], Ron Steinfeld[1], Dongxi Liu[2], and Tsz Hon Yuen[3]

[1] Faculty of Information Technology, Monash University, Melbourne, Australia
{dimaz.wijaya, joseph.liu, ron.steinfeld}@monash.edu
[2] Data61, CSIRO, Eveleigh, Australia
dongxi.liu@data61.csiro.au
[3] Huawei, Singapore, Singapore
yuen.tsz.hon@huawei.com

Abstract. Monero is one of the most valuable cryptocurrencies in the market, focusing on users' privacy. The built-in features in Monero help users to obfuscate the information of the senders and the receivers, hence achieve a better privacy compared to other cryptocurrencies such as Bitcoin. Previous studies discovered multiple problems within Monero systems, and based on these findings, Monero system has been improved.

Although improvements have been made, we discovered that new attacks targeting the anonymity reduction can still be conducted in Monero system. In this paper we propose two attacks. The first is an extension of a known attack called Monero Ring Attack. The second one exploits Payment ID to discover the real output of a mixin. We then propose countermeasures to these attacks.

Keywords: Monero · Anonymity · Attack · Traceability · Ring signature

1 Introduction

Bitcoin is the first cryptocurrency deployed in 2009 by Satoshi Nakamoto [1]. It allows every user to join or leave the payment system instantly. In accordance to the idea, Bitcoin relies on cryptographic methods such as public key cryptography to avoid user registration process and to provide the proof of ownership of the coins. This mechanism decouples the Bitcoin transactions with the users' real identities. Therefore, Bitcoin was supposed to be anonymous.

The term anonymity can be determined by 2 adjacent forms: **unlinkability** and **untraceability** [2]. Unlinkability is related to the privacy of the receiver, where different transactions cannot be identified to be sent to the same receiver. Untraceability, on the other hand, is related to the sender's privacy. A system is considered as untraceable when it is infeasible to determine the real sender of a transaction.

The anonymity assumption in Bitcoin environment was proven to be incorrect. Several studies have shown that information related to Bitcoin users can be revealed, either by using clustering methods [3] or extracting additional information from websites [4]. Quantitative analysis of Bitcoin transactions was able to determine malicious activities conducted by the Mt.Gox hacker(s) [5]. Consequently, Bitcoin is

F. Guo et al. (Eds.): Inscrypt 2018, LNCS 11449, pp. 86–100, 2019.
https://doi.org/10.1007/978-3-030-14234-6_5

now considered as pseudo-anonymous, since the users' identities can be revealed through some analyses.

Monero is a different type of cryptocurrency compared to Bitcoin. Monero is one of the most valuable cryptocurrencies in the world with a total market value of US $3.5billion[1]. The technology offered by Monero focuses on enhancing the anonymity of the users. The anonymity in Monero is implemented by obfuscating the information of the real senders and the real receivers, making it hard for observers to make a direct relationship of the senders and the receivers. The anonymity features of Monero are achieved by employing ring signature and stealth address in the system. The privacy-protection mechanism was further enhanced by implementing RingCT technology, which obfuscates the amount of coins transacted.

We define an anonymity reduction attack as an effort made by a malicious user (or an attacker) who tries to de-anonymise other users' transaction. This attack is done by creating transactions and breaking the attacker's own anonymity. An example of this attack is Monero Ring Attack [6]. The aforementioned attack works on both non-RingCT and RingCT environment, and does not rely on zero-mixin transactions as in [7, 8].

Contributions. We summarise our research contributions as follows.

1. We propose a mitigation strategy on a known Monero anonymity attack exploiting the users' freedom when creating mixins [9]. Our mitigation strategy uses a list of hash values of existing mixins. New transactions are not allowed to use any mixins that have existed in the system, in which their hash values are on the list. By employing the mitigation strategy, future attacks using the same method can be completely prevented.
2. We propose an extension of the attack in [6]. Our new scheme achieves the same goals of the previous attack in [6] but nullifies the mitigation strategy we designed. The new scheme provides an obfuscation method when choosing the mixins, hence identical mixins are no longer needed.
3. We propose a solution to avoid the new attack we propose. Monero developers have been working on a new blacklisting mechanism called "blackball" which will blacklist all known bad outputs to mitigate an attack over key reuse[2]. In our solution, we developed a new metric as a quantitative measurement towards the suspicious level of anonymity reduction attack. The metric is useful to complement the existing blacklisting method, either by implementing it in the Monero core software or as a separate service.
4. We propose a novel anonymity reduction attack by utilizing an extra information called Payment ID (PID) which is usually embedded in Monero transactions. The scheme enables the attacker to trace the real outputs spent in the transactions. We also suggest a possible countermeasure on the attack.

Organization. The rest of the paper is organized as follows. In Sect. 2 we present the basic knowledge about the field of the research. Section 3 describes previous studies

[1] Based on information provided by Coinmarketcap.com on 22 March 2018.

[2] https://github.com/monero-project/monero/pull/3322.

related to our research. In Sect. 4 we propose a mitigation strategy of a known attack, while in Sect. 5 we improve the known attack by removing the weaknesses and propose a stronger attack trait. Lastly, Sect. 6 presents a novel attack related to Payment ID usage.

2 Background

2.1 CryptoNote Protocol

The CryptoNote protocol was originally proposed in 2013 [2]. The purpose of the CryptoNote protocol is to create a privacy-preserving cryptocurrency with built-in features that will help users to keep anonymous, although it still preserves similarities with Bitcoin, such as transparent transaction data (inputs, outputs, and the amount of coins transacted). The main idea of the protocol is to employ a linkable ring signature [10, 11] to avoid the sender from being traced. The one-time public key (stealth address) is also implemented to make sure all users create a new address for every transaction.

By using a linkable ring signature, it is infeasible to distinguish the real output being spent by a transaction over a set of outputs. In the linkable ring signature, the user needs to construct a set of outputs as an input of the transaction, which are assumed to be picked randomly over a large set of outputs available on the network. The user needs to insert her own output to the set, which is the real output to be spent in the transaction. The other outputs are actually the decoys that help obfuscating the real output.

The linkable ring signature also protects the system from a double spending attack by allowing detection if such thing happens. Each public key is associated with a secret value; in order to spend a public key, the corresponding secret value needs to be exposed to the blockchain. If a secret key appears more than once, it means a double spending attempt occurs [11].

In the one-time public key mechanism, the receiver provides a set of master public keys to the sender, which will be used by the sender to create new public keys. Hence, the destination of the payment is created by the sender, not the receiver. When receiving payments, the receiver scans the blockchain and applies a method to determine which payments are destined to the receiver.

2.2 Monero

Monero is a CryptoNote-based cryptocurrency. As with any other cryptocurrency products, there are applications required to run the system, namely Monero daemon and Monero wallet. A Monero daemon is the node keeping a full record of all transactions happening in the network, while a Monero wallet does not need to store all transactions locally. Monero wallet can be used to create new transactions by connecting to a Monero daemon.

When creating a new transaction, the Monero wallet first sends a request to the Monero daemon. The purpose of the request is to get an information about potential public keys (outputs) to be used as decoys in the ring signature construction. Using a random sampling algorithm, the wallet selects a set of global indexes from the histogram provided by the daemon, then the daemon will need to provide the corresponding outputs. The index of the real output is also included in the request for two reasons: as a test whether the daemon sends the correct outputs and to obfuscate the final ring signature from a curious daemon.

Monero developers have adopted an additional protection on users's privacy. The Confidential Transaction (CT) is added into ring signature construction to create RingCT [12]. Confidential transaction is a Pedersen commitment to encrypt the amount of money sent from the sender to the receiver so that they cannot be visible to the world; only the respective participants can decrypt the amount [13]. RingCT was deployed in January 2017 and became mandatory since September 2017 [14].

RingCT has caused a major change on the way users create mixins. In a non-RingCT transaction, an output can only be mixed with other outputs with the same amount. Before RingCT is available, an output having a unique amount of coins cannot be mixed with other outputs, hence zero-mixin transaction occurs. Zero-mixin transaction is a transaction containing an input that does not have any mixin. By using RingCT, the amount of money will be hidden, thus an output can be mixed with any outputs.

2.3 Monero Transaction

In Monero, a transaction contains at least one input and zero or more outputs. An input contain at least one unspent output from an existing transaction confirmed in the blockchain and the input can also include several other outputs as decoys or “mixins”. The transaction can also produce outputs which are the fund sent by the sender to one or more receivers. The structure of the transaction can be seen in Fig. 1.

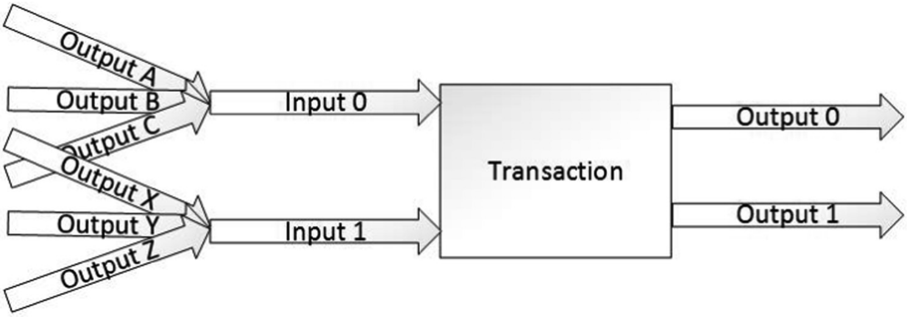

Fig. 1. In Monero, an input may contain several existing outputs. Only one output in an input to be spent, while the other outputs are decoys. [6]

2.4 Monero Payment ID

As with any other CryptoNote-based cryptocurrency, it is infeasible to distinguish a specific sender among a set of senders in Monero [2]. In the Bitcoin system, creating a new destination address for each user is practical by using a deterministic or hierarchical deterministic (HD) wallet as in [15]. In Monero, each wallet can only have one Monero address (we exclude subaddress [16] in the discussion). Therefore, a metadata called Payment ID is added into Monero transaction to help the receiver to determine the sender of the payment.

The unencrypted Payment ID is 32 bytes data inserted into the extra field on the transaction data, which is usually represented into 64 digits hexadecimal on the Monero blockchain explorer [17]. A new feature named integrated address encrypts the Payment ID into the destination address, where only the receiver can decrypt the Payment ID. The Payment ID for the integrated address is 8 bytes long or 16 digits hexadecimal [17]. In this paper, we will use the term UPID to refer the unencrypted Payment ID and EPID to refer the encrypted Payment ID.

3 Related Works

3.1 Monero Zero-Mixin Problems

In general, it is obvious that a receiver of a transaction can determine that her outputs are being used as decoys by other transactions. Therefore, it is possible that a user tries to attack the system by creating a large number of outputs in order to reduce the anonymity of other transactions [9]. In the attack, the attacker needs to pay a huge amount of transaction fees. The attacker also needs to keep creating new transactions if she wants to control a majority of the outputs available in the network [9].

Recent studies show that zero-mixin transactions have impacted the anonymity of other transactions. A research finds that at least 87% of the mixins up to a point were de-anonymised [7]. Another research proposes a change in the way the Monero wallet samples the mixins by prioritizing new outputs rather than randomly picks the mixins from all possible options [8].

3.2 Monero Ring Attack

There is a new type of attack targeting Monero anonymity called Monero Ring Attack [6]. This attack is an improvement over the previous attack presented in [9], where an attacker tries to dominate the available outputs in the system. In Monero Ring Attack, several attackers can join forces to attack the system, each by crafting special transactions [6].

If an attacker creates a “malicious transaction”, then other attackers can use the result of that transaction in addition to their own results. Therefore, in summary, each attacker will pay less transaction fees, but with bigger impact. During the attack, the attackers do not need to trust each other or communicate with each other. All they need to do is scanning the blockchain data and determine the malicious transactions created by others.

The key point on this attack is using identical mixins on the malicious transactions. Let r as the minimum number of mixins in the system; an attacker needs to have at least r outputs or sets of r outputs. Then, the attacker constructs the inputs by using r outputs as the mixins. These constructions help any attackers to determine that all the outputs used in the mixins have been spent. It is impossible that other transactions that include the outputs are double spending the outputs, hence these outputs are just the decoys.

The Monero Ring Attack has 3 phases: preparation phase, setup phase, and attack phase. In the preparation phase, the attacker prepares r outputs. In the setup phase, the attacker constructs spending transactions using identical mixins, each mixin has r outputs as the ring members. Then, in the attack phase, the attacker expands the outputs (active attack) and analyse the result of the attack (passive attack).

The purpose of the attack is to trace the real outputs spent by mixins or at least reduce the k-anonymity of the transaction mixins. The goal is achieved by creating multiple malicious transactions such that the outputs of the attack are expected to be used as mixins by honest transactions.

4 Mitigating Monero Ring Attack

4.1 Overview

The setup phase in Monero Ring Attack as described in [6] has a unique characteristic where identical mixins are used multiple times. By determining this characteristic, the attack can simply be detected. The detection is done by hashing all mixins in the blockchain then search for any hash duplicates. If any duplicates are found, then it indicates that the attack has occured.

4.2 Detection Method

We propose a method to detect whether the attack has occurred in the blockchain. Scanning and blacklisting steps of detecting the attack can be summarized as follows.

1. For all mixins in the blockchain $M = \{m_0, m_1, m_2, \ldots m_n\}$, produce a set of hash values $H = \{h_0, h_1, h_2, \ldots, h_n\}$ by using hash function FH such that $h_0 = FH(m_0)$. A mixin m_0 is a list of outputs mo_0=$\{o_v, o_w, o_x, o_y, o_z, \ldots\}$ where $v, w, x, y, z, \ldots$ are the output indexes.
2. For all mixins M, compute the corresponding ring size $R = \{r_0, r_1, r_2, \ldots, r_n\}$ such that r_0 is the ring size of mixin m_0.
3. For all hash values in H, compute the number of occurrence $U = \{u_0, u_1, u_2, \ldots, u_n\}$ such that u_0 is the number of occurrence of h_0.
4. For each mixin m_j where $0 \le j \le n$, if $r_j = u_j$ then the mixin m_j is considered as an attack. All outputs in mo_j needs to be included in a blacklist B.
5. For each mixin m_k where $0 \le k \le n$, check if mo_k contains any outputs from B. If yes, then add mo_k to the blacklist B.

The blacklist B, as the result of the detection method provided above, could then be published. All of the outputs in the blacklist B are discouraged to be used when sampling outputs for creating mixins.

4.3 Mitigation Strategy: Forbid Mixin Duplicates

The Monero Ring Attack has a characteristic of using mixin duplicates when launching the attack. In order to countermeasure this type of attack, Monero daemon can be equipped by a mechanism to reject new transactions having identical mixins. Furthermore, the daemon can maintain a list of mixin hash values that have been used in the system. New transactions need to prove that the hash values of their mixins have never existed in the blockchain. Considering that creating new mixins are easy, rejected transactions can be resubmitted after revising the duplicated mixins.

5 Extending the Monero Ring Attack

5.1 Overview

The idea of using sets of transactions and creating identical mixins in the setup phase as explained in Sect. 3.2 can simply be extended to obfuscate the attack. Instead of using identical mixins, combinations of outputs can be utilised. The result of this modification is identical as the Monero Ring Attack, but the method is harder to detect. The hashing method cannot be applied to the new attack. Below is the comparison between the Monero Ring Attack (MRA) and our proposed attack (Table 1).

Table 1. Comparison between the existing Monero Ring Attack and our proposed attack

Parameters	MRA	Ours
Attacking untraceability and anonymity	v	v
Cooperation between attackers without trust	v	v
Undetected using hash table	x	v

Both attacks are useful when launched against specific targets, such as coin exchange users, rather than targeting random Monero users. Targeting random users requires a massive amount of money to pay the transaction fee, while targeting a specific users will reduce the attack cost. The governments or regulators can enforce business entities under their jurisdictions to implement this scheme in order to secretly discover the users' activities in Monero system.

5.2 Security Model

The security model is similar to the one proposed in [6]. It is assumed that an attacker can access the public blockchain. The attacker might also have an access to wallet services or trading platforms in order to launch the attack without paying transaction fees (the transaction fees are paid by the customers). Although the attacker has an access to the public blockchain, but the attacker does not have the capability to modify any transactions that have been confirmed in the blockchain.

The goal of the attack is to define the real outputs spent by other transactions or reduce the anonymity of the transactions created by the users. The attack enables multiple attackers to analyse the work of other attackers and aggregate the result to maximize their efforts.

5.3 Attack Mode

Table 2 is an example how 6 outputs $O = \{O_1, O_2, O_3, O_4, O_5, O_6\}$ can be spent in six inputs I = $\{I_A, I_B, I_C, I_D, I_E, I_F\}$, with each input has a ring size of 5. We assume the system's mandatory ring size is 5 as in Monero fork version 6 (software version v0.11.0.0). The rasterized cells indicate the real outputs being spent in the input. The attack can be obfuscated further by employing a large set of outputs O which will be spent by a large set of inputs *I*. These inputs can be executed in one transaction or can also be spent in multiple transactions.

Table 2. Spending 6 outputs in 6 inputs. The rasterized cells are the ones being spent

Outputs	I_A	I_B	I_C	I_D	I_E	I_F
O_1	▨ v		v	v	v	v
O_2	v	▨ v		v	v	v
O_3	v	v	▨ v		v	v
O_4	v	v	v	▨ v		v
O_5	v	v	v	v	▨ v	
O_6		v	v	v	v	▨ v

We denote by ${}_nC_r$ the number of possible ways of choosing a ring of size r out of a total of n possible outputs. For example, a case where $n = 80$ and $r = 5$, we find $C = 24{,}040{,}016$ possible combinations to spend 80 outputs in 80 inputs. The calculation shows that it is infeasible to determine the attack by using a trivial mechanism.

5.4 Collaborating with Other Attackers

The proposed attack can be a collaborated attack among multiple attackers. The attackers do not need to trust each other to decide whether they have conducted the attack. Therefore, trust is not needed on the collaboration, since all of the information can always be validated by the attackers. The validation process is easy: if the number of outputs and inputs match, then the attack really occurs.

In order to make the validation process faster, the attackers still need to share a part of the information related to the attacks they conduct. The information to be shared includes:

- A list of all outputs used in the attack.
- A list of all inputs involved in the attack.

5.5 Detecting the Attack

The strategy described in Sect. 4 is not applicable to mitigate the new attack; therefore, we define a new method. It is assumed that the shared information mentioned in Sect. 5.4 is kept secret among the attackers. We simulated this attack and determined two important features that distinguish the malicious transactions and honest transactions:

- The malicious transactions repeatedly include a subset of outputs O as the mixins.
- A subset of the outputs O have a high usage value, which might be higher than the average usage value.

Precisely determining whether this type of attack has occurred in Monero and listing all related transactions might be infeasible due to the number of possible combinations. Consequently, we explore the features of this attack which will be identifiable within a set of transactions.

The diagrams in Fig. 2 shows several information regarding the average usage per output, number of transactions, and number of mixins aggregated for every 10,000 blocks. The information is useful to distinguish between regular output usages and suspected output usages. The sharp increase in the diagram A is suspected to be caused by the mandatory RingCT scheme implementation, which left the users to have limited options of outputs to be used as their mixins. At the same time, the minimum ring size was increased to five.

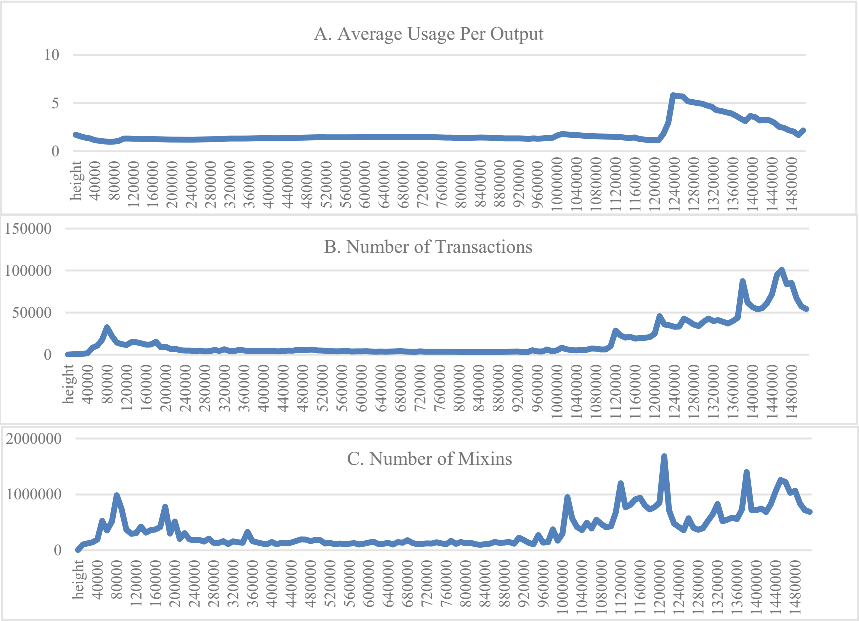

Fig. 2. Diagram A shows the average usage per output from the blocks. Diagram B shows the number of transactions on the blocks. Diagram C shows the number of mixins of the transactions on the blocks. The data is aggregated for every 10,000 blocks. The horizontal axis shows the block height, while the vertical axis shows the value.

We then evaluated the data from the Monero blockchain up to block 1,542,882 containing 24.8 million outputs, where 4.7 million outputs are from RingCT transactions. Based on the finding, we divide the transactions into non-RingCT and RingCT transactions due to their data characteristics. The diagrams are shown in Fig. 3. The average output usage for all non-RingCT transactions is 1.96, while the average output usage for all RingCT transactions is 3.68. Overall, the average output usage is 2.28.

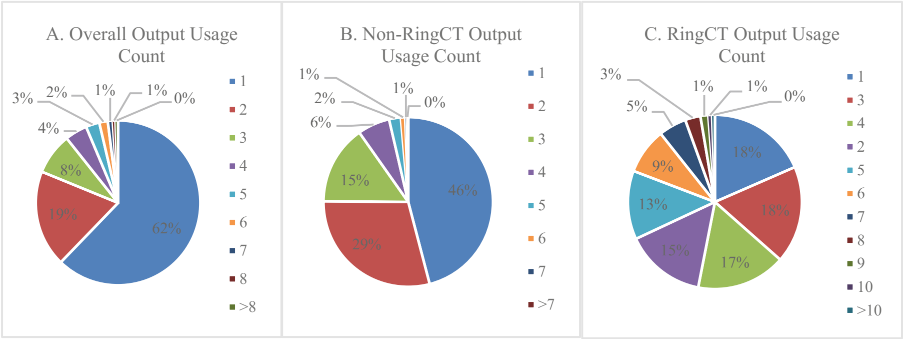

Fig. 3. The aggregate data of all output usages with the legends describe the number of output usage. Diagram A shows the aggregate output usage count on Non-RingCT and RingCT transactions. Diagram B shows the same data on Non-RingCT transactions. Diagram C shows the data on RingCT transactions. Data less than 1% is aggregated.

The difference between non-RingCT and RingCT transactions might also be affected by a change in mixin sampling method. Triangular distribution is used to replace uniform distribution to increase the data resemblance with users behaviors as suggested by [8]. The impact of the triangular distribution towards the evaluated data is ignored to simplify the case.

We define an output weight *OW* as the number of inputs where an output *O* is involved as one of the mixins. We also define an input weight *IW* as the average value of *OW* for all outputs in the input.

$$\mathrm{IW}_s = \frac{\sum_{k=0}^{r} OW_k}{r} \tag{1}$$

We scanned the blockchain and computed the value of *IW* for all inputs up to block 1,545,153 (timestamped on 5 April 2018). We found a total of 45,650,192 inputs on the blockchain. The data is then aggregated and presented on Fig. 4.

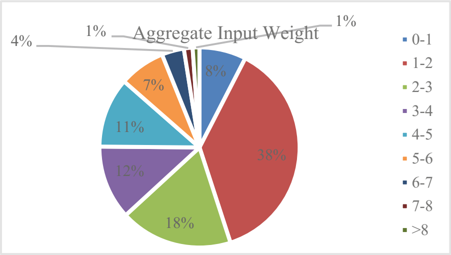

Fig. 4. Aggregated *IW* for all transactions (nonRingCT and RingCT). The data is grouped based on the *IW* range. For example, the data marked as "1-2" means the *IW* is in the range of 1 to 2.

We look for standout figures on the blockchain data. We have presented the mechanism where *IW* is used to weight all mixins. It turns out that *IW* can be used as a fingerprint where inputs that contain the same output having a high usage value can be identified. Based on our evaluation, inputs having *IW* value of at least seven are suspiciously reusing the same outputs multiple times. The total number of suspected inputs is 1,142,383 or 2% of all inputs recorded in the blockchain.

5.6 Mitigation Strategy: Input Weighting

The current triangular distribution sampling method does not guarantee that the users will use outputs that are not a part of anonymity reduction attack. Thus, there is an urgent need to improve the sampling protocol.

Our research shows that Monero outputs are not as fungible as claimed in [18]. Fungibility in Monero describes that all outputs have the same value regardless who creates the outputs. Our results show that a subset of the outputs are potentially harming the anonymity of the transactions than other outputs. Hence, the term fungibility can also be applied on the mixins, since they determine the level of anonymity gained by the users.

To increase the anonymity and mitigate the attack, we propose the use of *Input Weight* (*IW*) as one of the criterion when sampling the outputs during mixin creation. The higher the *IW* value of an output, the higher the chance of the output being a part of an attack.

Based on our evaluation, the current *IW* threshold to distinguish between "normal transactions" and "suspicious transactions" is seven. It is also possible that the threshold is changed due to changes in the system, specifically when the number of RingCT outputs increases or the mandatory ring size increases. The rule for determining the threshold is that the lower the threshold, the lower the risk would be.

6 Leveraging Monero UPID

6.1 Overview

The unencrypted Payment ID (UPID) poses an anonymity problem, where an observer can easily collect the information from the public blockchain and decode the message.

A user investigated Monero payments associated to TheShadowBroker, a hacking group that wanted to auction their secret information gained unlawfully. The investigation managed to collect email addresses of TheShadowBrokers' clients [19].

By using the similar technique, we evaluated the use of UPID in relation to the users' anonymity. The UPID is optional; hence, it is not commonly used if it is not mandatory. A user uses the same UPID to be included in multiple transactions when sending payments to the same merchant. Therefore, we assume that transactions using the same UPID are sent by the same sender to the same receiver.

Based on the above scenario, if a user uses a UPID in a transaction and creates a second transaction including same UPID which includes the outputs from the first transaction to the input mixins, then it is likely that these outputs are the outputs being spent by the latter transaction. The scheme might be possible, since the reused outputs are the change money. The change money is not transferred to the receiver and returned back to the sender's address. The scenario is described in Fig. 5.

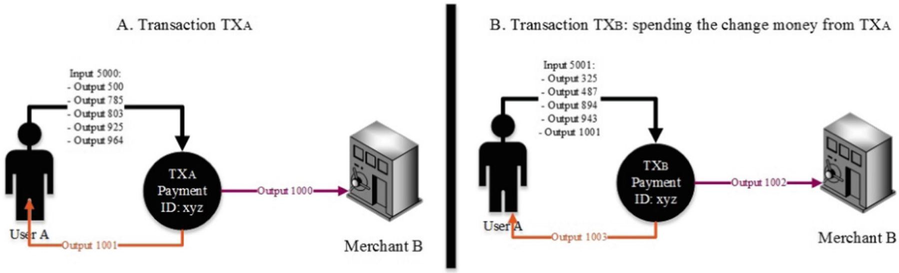

Fig. 5. User A reusing the change money from the previous transaction in a new transaction. Both transactions are sent to the same merchant. Note "Output 1001" of TX_A from diagram A is included in "Input 5001" in TX_B.

6.2 Results

We collected the transaction data from Monero blockchain and extracted the information into a relational database. The number of transactions using Payment ID is significant, that more than half of the transactions ever recorded in the Monero blockchain are using Payment IDs, as shown in Fig. 6.

From the genesis block up to block 1,535,607 (timestamped on 22 March 2018), we found 2,584,535 non-coinbase transactions (containing 23,108,911 inputs). Within the result, there are 1,033,891 transactions (containing 12,383,714 inputs) using UPIDs and 420,153 transactions using EPIDs.

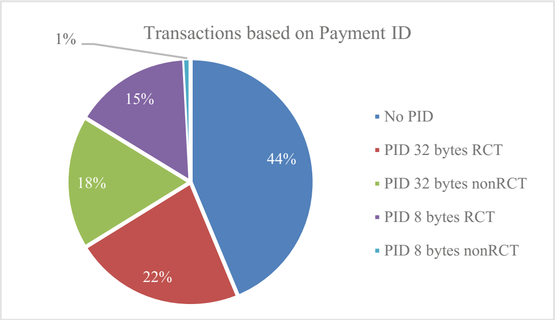

Fig. 6. The transaction percentages based on Payment ID

We further investigated the transactions using UPID and managed to cluster the data based on the UPID reuse. There are 338,318 unique UPIDs found within the 1,033,891 transactions. There are also at least 15 UPIDs used more than 1,000 times. We then cross-referenced the transaction data to see if the outputs coming from transactions with UPIDs are reused by other transactions having identical UPIDs. We discovered 332,987 inputs from 165,919 different transactions using identical UPIDs. The identified inputs are 1.6% of total inputs in transactions using 32 bytes Payment ID.

We assume that the senders reusing the same UPID are sending money to the same merchants. We also assume that a part of the outputs are the change money which are sent back to the senders' addresses. When the senders want to create other transactions to the same merchants, for example to pay different purchases, then these senders can use the coins contained in the change addresses from previous transactions. Hence, we conclude that these outputs are being spent by the new transactions.

We investigated a number of cryptocurrency exchanges supporting Monero as one of their tradeable assets, such as HitBTC, Binance, Bitfinex, Poloniex, and Kraken. Based on information in Coinmarketcap.com, these cryptocurrency exchanges hold significant Monero trading volumes among other trading platforms.

Table 3. A list of trading platforms and their Payment ID details

No.	Platform	Trading Volume[3]	Payment ID[4]	User can create a new deposit address or a new payment ID
1	HitBTC	37.68%	EPID	No
2	Binance	16.67%	UPID	No
3	Bitfinex	13.84%	UPID	Yes
4	Poloniex	6.56%	EPID	No
5	Kraken	6.23%	EPID	Yes
6	Livecoin	3.82%	UPID	No

Table 3 shows that there are three cryptocurrency exchanges using UPID, namely Binance, Bitfinex, and Poloniex. We can determine that Binance and Livecoin users will always have the same UPID for the same user, while Bitfinex is using the UPID but provides a feature where the users can regenerate the addresses and UPID by themselves.

Repeated transactions are likely to be created by the trading platforms or cryptocurrency exchange users, since the main function of cryptocurrencies such as Bitcoin nowadays are tradeable assets rather than as a payment method [20]. Repeated deposits to the trading platforms are also possible, for example sending mining rewards directly from a mining pool to the miners' accounts in cryptocurrency markets.

Cryptocurrency trading platforms rely on the Payment ID to identify the customers' deposit as it is infeasible to distinguish the correct Monero transactions belonging to different customers. As their platforms may receive thousands of Monero deposits per day, the Payment ID is useful to automate the identification process, which will credit the correct customers' accounts with the correct amount of coins they transferred.

6.3 Possible Countermeasure: Encrypted Payment ID

We have presented a case where using the same UPID can be harmful to the users' anonymity, where an attacker is able to determine the real outputs spent by the transactions. The UPID is still widely used by cryptocurrency trading platforms.

To mitigate the problem, the UPID should no longer be used, and the merchants are urged modify their system to support the EPID. By using the EPID, the users' deposits can still be determined, hence there is no change in the merchants' business process that the correct accounts can be credited based on the payments received.

7 Conclusion and Future Works

In this research, we propose a mitigation strategy of an existing attack in [6]. Then, we formulate an extension of the attack, where the improvement of the new attack makes the previous mitigation method obsolete. By using distinguishable features we found in the transactions, we propose a simple approach yet effective as one of the considerations during the mixin sampling protocols.

We also propose a second anonymity reduction attack by exploring the use of Payment ID. The Payment ID is a common method being used by Monero merchants to distinguish payments from different users. We found that transactions having the same UPID is closely linked, such that at least 1.6% of the inputs are traceable.

For future works, we plan to implement the proposed mitigation strategies in a working system. The hardened system contains all standard anonymity features such as traceable ring signature and one-time public key, including mitigation strategies as we have proposed in the paper. Then, we will analyse the impact of the newly created

[3] The information is taken from Coinmarketcap.com on 4 April 2018. The value of trading volume is calculated by summarizing all trading pair volumes.

[4] The information is taken from the platforms on 4 April 2018.

wallet into the anonymity of the users and to evaluate whether new attack methods can be developed.

References

1. Nakamoto, S.: Bitcoin: A Peer-To-Peer Electronic Cash System (2008)
2. van Saberhagen, N.: Cryptonote v 2.0 (2013)
3. Meiklejohn, S., et al.: A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. USENIX; login (2013)
4. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Altshuler, Y., Elovici, Y., Cremers, A., Aharony, N., Pentland, A. (eds.) Security and Privacy in Social Networks. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-4139-7_10
5. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 6–24. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39884-1_2
6. Wijaya, D.A., Liu, J., Steinfeld, R., Liu, D.: Monero Ring Attack: Recreating Zero Mixin Transaction Effect. Cryptology ePrint Archive (2018)
7. Kumar, A., Fischer, C., Tople, S., Saxena, P.: A traceability analysis of monero's blockchain. In: IACR Cryptology ePrint Archive 2017, p. 338 (2017)
8. Miller, A., Möser, M., Lee, K., Narayanan, A.: An Empirical Analysis of Linkability in the Monero Blockchain. arXiv preprint arXiv:1704.04299 (2017)
9. Noether, S., Noether, S., Mackenzie, A.: MRL-0001: A note on chain reactions in traceability in cryptonote 2.0. Technical report (2014)
10. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 181–200. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71677-8_13
11. Liu, J.K., Wei, V.K., Wong, D.S.: Linkable spontaneous anonymous group signature for ad hoc groups. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 325–335. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27800-9_28
12. Noether, S., Mackenzie, A.: Ring confidential transactions. Ledger **1**, 1–18 (2016)
13. Maxwell, G.: Confidential Transactions (2015)
14. Getmonero. https://getmonero.org/resources/moneropedia/ringCT.html
15. Bitcoin BIP. https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
16. Noether, S., Goodell, B.: An Efficient Implementation of Monero Subaddresses (2017)
17. Getmonero. https://getmonero.org/resources/moneropedia/paymentid.html
18. Getmonero. https://getmonero.org/resources/moneropedia/fungibility.html
19. Steemit. https://steemit.com/shadowbrokers/@wh1sks/theshadowbrokers-may-have-received-up-to-1500-monero-usd66-000-from-their-june-monthly-dump-service
20. Glaser, F., Zimmermann, K., Haferkorn, M., Weber, M.C., Siering, M.: Bitcoin - asset or currency? Revealing users' hidden intentions. In: Twenty Second European Conference on Information Systems (2014)