An Efficient Strategy to Eliminate Malleability of Bitcoin Transaction

Yi Liu, Xingtong Liu, Lei Zhang, Chaojing Tang and Hongyan Kang
National University of Defense Technology
Changsha, China
Email: louis0121@126.com

***Abstract*—Bitcoin is decentralized cryptocurrency which has recently become increasingly popular all over the world. Blockchain works as a public ledger to record all transactions in Bitcoin system. A transaction to transfer bitcoins includes a set of inputs and outputs, with its unique hash value. However, the owners of transferred bitcoins can only sign on the transaction to provide integrity for its content without guarantee for its output script. This is considered as transaction malleability in Bitcoin system, which allows an attacker to intercept, modify, and rebroadcast a transaction into the Bitcoin network. This attack deceives the transaction issuer into believing that the original transaction failed to be recorded in blockchain. The vulnerability has been utilized by attackers leading a bankruptcy of the largest Bitcoin exchange at that time. To avoid this situation happens again, we present an efficient strategy to secure user wallet in this paper. Our scheme confirms success of a transaction not only relying on its unique hash value, but also relying on the address balance it spends. This strategy doesn't add too much complexity and is easy to implement in existing wallets. It is resistant to transaction malleability and helps Bitcoin users to protect their property from malicious entities.**

***Index Terms*—Bitcoin; transaction malleability; Bitcoin wallet; cryptocurrency**

I. Introduction

Since first introduced by Nakamoto in 2008, Bitcoin has become the representative of distributed digital currency[1]. It provides a convenient service to make transactions between anybody on the globe with high speed and low fees. Recently, It has become more and more popular in the lives of ordinary people, with a market capitalization of over 60 billion dollars. Bitcoin transactions transfer from person to another through a peer-to-peer network instead of centralized banking systems.

Everyone in Bitcoin network is able to reserve the complete transaction database on their local storage and verify the legality of the transaction by themselves without trusting anyone for information integrity or authenticity. A transaction needs a signature from its legal customer to be valid, which represents the ownership of these bitcoins has been transferred from the customer to another one. Then it is going to be spread to the whole network until it is included by a valid block. The nodes who received this message send it to other nodes connecting with them until all the nodes receive it in the Bitcoin network. Adding transaction records into blockchain is considered as mining because of getting rewards. But more accurate explanation is that the node who contributes computing power to find a new block will verify each transaction included and provide security service for Bitcoin system. The node which contributes more power has a better odd to receive newly minted coins as a reward[2]. Each node tries to solve a computationally difficult puzzle for issuing newly generated coins. Mining can be used to add transactions to the blockchain and resist double-spending. Transactions get confirmed only if they have been recorded into the blockchain and later found blocks have been appended to the block which they lie in. Every block is linked together to form the blockchain. Each block includes a hash referencing to its precursor block. Every node is able to maintain a blockchain which is updated through collecting transactions and blocks which is not included in the blockchain. Fig 1 shows the linking relation among blocks through their precursor block's hash value as the sketch map of Bitcoin blockchain.

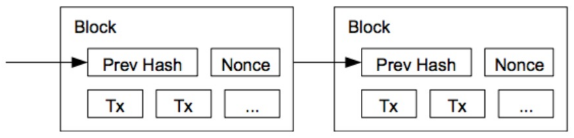


Fig. 1. Sketch map of Bitcoin blockchain

Everyone participant in Bitcoin transactions is represented by a pseudonym, called address. A Bitcoin address is hashed from the public key which is derived from elliptic curve digital signature algorithms (ECDSA)[3]. To obtain a Bitcoin address, we apply RIPMED160 and SHA256 hashing algorithm to a public key respectively. After concatenated with a checksum, it is encoded into a fixed format string through a modified Base58 encoding method[4]. A Bitcoin address is an identifier with 25-34 characters comprising upper and lower case characters with numbers. Bitcoin addresses usually start with number '1', and comprise 33 to 34 characters totally. To get better readability, there is no similar characters appearing in an address, such as upper case of 'I', lower case 'l', number 'O' and upper case 'O'.

A Bitcoin wallet generates ECDSA key pairs and Bitcoin addresses offline following the hashing and encoding rules. Since it's easy to create the addresses with many tools included in the original Bitcoin client, people is advocated to use as many addresses as he can to receive bitcoins from different transactions. A wallet preserves public and private key pairs owned by its user and records transactions related to its addresses. Addresses acting as account are used to get coins from others while private keys are used to spend coins to others. Payers request for addresses from payees to transfer bitcoins and sign on transaction messages with their private keys.

The address balances can only be modified through legal transactions. Transactions use their own SHA256 hash as their identifer. Signature can only guarantee the content in a transaction due to various form of Bitcoin script for the same funcion. The possible vulnerability can be utilized by an attacker to modify a transaction different from the original one but spending the same inputs. This will cheat users to construct another transaction for the previous payment.

To help users manage their bitcoins resistant to transaction malleability, we give a new method based on passphrase to enhance security of software wallets . It not only utilizes the hash of a transaction to confirm its completion, but also queries the address balance in case of double payment. Therefore, users have a smaller risk of losing the ownership of their bitcoins caused by transaction malleability.

The rest of this paper is structured as follows. Related work about Bitcoin wallet is outlined in Section 2. The mechanism resulting in transaction malleability in Bitcoin is detailed in Section 3. Then we presented our strategy to enhance Bitcoin wallets in Section 4. We give a conclusion of this paper in Section 5.

II. Related Work

It is the first time that transaction malleability attracted public attention in 2010[5], but is considered as a minor inconvenience instead of a vulnerability[6]. It has not attracted too much attention from public Since it is not considered as a serious vulnerability.

Andrychowicz and Dziembowski concentrate on transaction malleability in contracts and a two-party computation protocol which is built upon Bitcoin protocl[7][8]. A fair coin toss can be realized according to their schemes based on Bitcoin transactions, the same to an auction system and a decentralized voting systems[9]. They also proposed a solution for transaction malleability through eliminating the parts of the transaction that can lead to malleability. Their method removes an important script from the transaction before the hash of the transaction is computed. The drawback of their approach is the removal of this important information that is signed by the owner. At the same time, they didn't extend their observations in other fields, except for contracts and two-party computation protocols.

Decker and Wattenhofer delved into transaction malleability and the Mt. Gox issue in paper[10]. They analyzed large amount of transactions recorded in the blockchain to identify double spending attacks and tried to investigate whether attackers utilized transaction malleability to gain much profit before the Mt. Gox incident. They only presented a case study without proposing a solution to eliminate transaction malleability.

Rajput presented a modification in Bitcoin protocol which eliminates transaction malleability problem in paper [11]. They excluded the input script from a transaction to generate a hash value, which works as the identifier of the whole transaction. Even if someone modifies the input script of a transaction, the hash value still remains the same as the original transation. Then they proposed another solution to combine the hash of the transaction without the signature script and the hash of the whole transaction to form the transaction ID[12]. However, their solutions are not compatible to exiting Bitcoin protocol and will lead to a fork in Bitcoin blockchain.

Wuille presented many ad-hoc approaches to eliminate this issue, through regulation of the syntax used in Bitcoin transactions[13]. He gave some sources of malleability, like Non-DER encoded ECDSA signatures, inherent ECDSA signature malleability, and non-push operations in scriptSig. Some new rules are introduced to mitigate transaction malleability by Wuille. In order to eliminate the sources of transaction malleability mentioned above, these new rules that should be adopted enforce procedures. However, the negative effect of this solution is that it will cause a hardfork change which would make blocks of version 2 entirely invalid when blocks of version 3 in the past 1,000 blocks reach 95% or higher. There is still a lack of formal argument, even though this new approach could make sense in practice.

Eventually, malicious entities make use of transaction malleability will result in a double spending attack to get profit. There has been a related class of double spending attacks and it has attracted much attention from public. In some other cases, the malicious entity constructs two different transactions with the same input to defraud the payee. Karame first delved into the problem in fast payment, in which payee will accept non-confirmed transactions[14]. Rosenfeld revealed that coupling with computational resources can make it easier to launch an attack for double spending[15]. Bamert disclosed the message propagation mechanism to improve the security in fast payment scenario[16].

III. Transaction Malleability

It is known to us all that a small change will have a big impact on its hash value. The hash value of a transaction acts as an identifier of this transaction as mentioned above. The hash value will become completely different when someone replaces some of the details in the transaction. Then, transaction malleability can be utilized by attackers who intercept a transaction from network, tempers with its input script and rebroadcasts the changed transaction. It generates an alternative transaction with different identifier and the same content with original transaction.

A. Bitcoin Transaction

Bitcoin works on a peer-to-peer network with a large amount of nodes taken over by different owners. All nodes work together to maintain a public ledger which records all the transaction history of Bitcoin users. Each user can constructs any number of addresses arbitrarily for Bitcoin transactions.

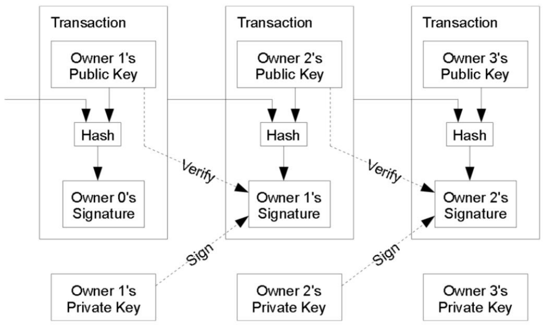


Fig. 2. Bitcoin transactions and the chain of transactions

A Bitcoin address work as an account to receive coins which is derived form a public key of ECDSA. Its private key is utilized to sign on a transaction which spend bitcoins belong to the address. Only if legal transactions has been recorded in the blockchain, the address balances can be changed.

Bitcoin addresses are used as destinations in a transaction showing where the bitcoins go to and as pockets where the bitcoins being stored. A Bitcoin address can be generated by hashing from a ECDSA public key for two times. Nobody has the right to consume the bitcoins associated with this address except for the one owning the private key corresponding to this address. A Bitcoin transaction input consists of a hash pointing to a previous transaction and an index pointing to an output address. Its output includes an address where bitcoins to transfer and an amount how much to pay. Bitcoin transactions form a transaction chain through their related inputs and outputs, as illustrated in Fig 2.

Each transaction is identified by its unique hash as identification. Every input referenced to an output address in a previous transaction which has not been used before. If a transaction includes an input pointing to an output of a previous transaction which is spent in another transaction, it will be regarded as invalid and discarded by miners. No inputs can be used twice if it has been included in existing transactions because this will be considered as a double spend attack. Since nobody is able to sign on a transaction without the private key associate with the input, Signatures can be regarded as a legal authority by its owner to transfer bitcoins.

B. Bitcoin Scripts

A transaction output consists of a claiming condition and the amount. The claiming condition is written in a script language which expresses the condition to claim coins belonging to the output. A claiming script is contained in a transaction input which is used to verify the validity of this transaction. There is also a hash and an index which specify the output it claims.

Bitcoin uses a scripting language to express the claiming conditions and its corresponding claiming scripts. The scripting language which is a stack based non-Turing complete language can transfer bitcoins in many complex situations.

Bitcoin transactions are constructed with standard claiming condition and claiming script in most cases. The public key

Listing 1: scriptPubKey

```
OP_DUP
OP_HASH160
OP_PUSHDATA*
<pubKeyHash>
OP_EQUALVERIFY
OP_CHECKSIG
```

Listing 2: scriptSig

```
OP_PUSHDATA*
<sig>
OP_PUSHDATA*
<pubKey>
```

Fig. 3. The standard form of claiming condition and claiming script in general transactions

corresponding to the input address and a legal signature made by its private key both need to be included in a claiming script. It is usually that scriptPubKey represents the standard claiming condition and scriptSig represents the standard claiming script. An instance of the standard claiming condition and claiming script is shown in Fig 3.

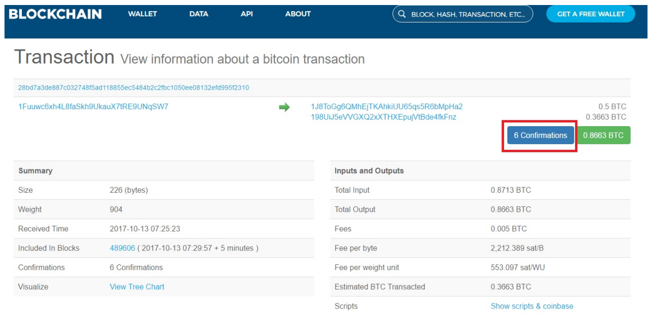
Bitcoin stores a script including the function commands on a stack upon execution. If the top stack item of a transaction returns true, the transaction is verified valid. Otherwise it will be regarded as invalid. To verify whether a transaction redeems its inputs correctly, we construct a combined script by appending the scriptPubKey of the referenced output to the scriptSig of the redeeming input. The outcome is either true or false after executing a Bitcoin script. The transaction is verified to be valid in case that there is no errors and executing successfully. Otherwise, the transaction is considered as invalid after the verification and it will be discarded from the mining pool.

C. Malleability Attack

The transaction malleability is caused by the fact that the signature can not provide integrity of the scriptSig. There are many ways to encode the claiming script with the same function. For example, OP_PUSHDATA2 can be used to push a public key on the stack instead of the opcode OP_0. 0x48 $\langle$sig$\rangle$ 0x41 $\langle$pubKey$\rangle$ can be replaced by 0x4D4800 $\langle$sig$\rangle$ 0x4D4100 $\langle$pubKey$\rangle$ in a claiming script. Both of the signature forms are valid with different hashes to identify the identical transaction.

For example, if a Bitcoin user A wants to transfer bitcoins m to another user B, she generates a transaction T, signs on it with her private key and broadcast T to other members. Anyone is able to receive transaction T, temper with its script to generate another transaction T' and rebroadcast T' instead of original transaction T. T' has the same inputs, outputs and signatures as original transaction T, but owns a definitely different hash value which is used to identified a specific transaction. These two different transactions compete in the whole network to be included in a new valid block first. The original transaction T will be discarded in case that transaction T' wins the competition. A may consider transaction has failed to transfer bitcoins m to B since T and T' has different hash, but m has been transferred by transaction T' successfully in fact.

Since Bitcoin allows more complicated transaction forms than those in the standard format mentioned above, there is another way to utilize transaction malleability. Specifically,



Transaction View information about a bitcoin transaction

28bd7a3de887c032748f5ad118855ec5484b2c2fbc1050ee08132efd995f2310

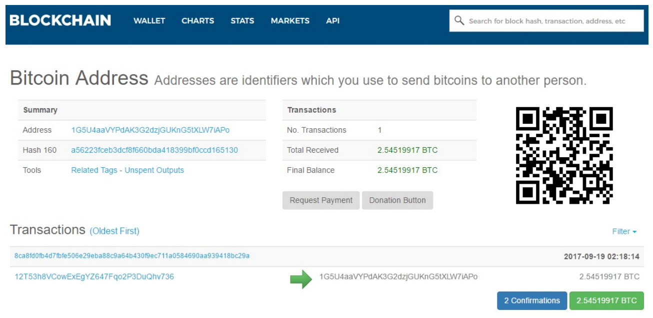
1Fuuwc6xh4L8faSkh9UkauX7tRE9UNqSW7 → 1J8ToGg6QMhEjTKAhkiUU65qs5R6bMpHa2 0.5 BTC
198UiJ5eVVGXQ2xXTHXEpujVtBde4fkFnz 0.3663 BTC

6 Confirmations 0.8663 BTC

Summary	
Size	226 (bytes)
Weight	904
Received Time	2017-10-13 07:25:23
Included In Blocks	489606 (2017-10-13 07:29:57 + 5 minutes)
Confirmations	6 Confirmations
Visualize	View Tree Chart

Inputs and Outputs	
Total Input	0.8713 BTC
Total Output	0.8663 BTC
Fees	0.005 BTC
Fee per byte	2,212.389 sat/B
Fee per weight unit	553.097 sat/WU
Estimated BTC Transacted	0.3663 BTC
Scripts	Show scripts & coinbase

Fig. 4. Transaction status queried on blockchain.info



Bitcoin Address Addresses are identifiers which you use to send bitcoins to another person.

Summary	
Address	1G5U4aaVYPdAK3G2dzjGUKnG5tXLW7iAPo
Hash 160	a56223fceb3dcf8f660bda418399bf0ccd165130
Tools	Related Tags - Unspent Outputs

Transactions	
No. Transactions	1
Total Received	2.54519917 BTC
Final Balance	2.54519917 BTC

Transactions (Oldest First)

8ca8fd0fb4d7fbfe506e29eba88c9a64b430f9ec711a0584690aa939418bc29a 2017-09-19 02:18:14

12T53h8VCowExEgYZ647Fqo2P3DuQhv736 → 1G5U4aaVYPdAK3G2dzjGUKnG5tXLW7iAPo 2.54519917 BTC

2 Confirmations 2.54519917 BTC

Fig. 5. Address balance queried on blockchain.info

assume that σ is not a standard claiming script as mentioned in last subsection. The simplest way to generate an alternative transaction is to add dummy opcodes PUSH and POP. This will also alter hash of the transaction without any changes in its content. Paper [10] and [13] detailed several different ways to utilize transaction malleability.

IV. Wallet Scheme

User wallets give a good assistance to manage private keys and confirm completion of transactions. Adversaries could easily construct two distinct transactions with identical content and different hashes through transaction malleability. It will mislead the user into thinking that the transaction is unsuccessful and do another payment. To mitigate this problem, we proposed a wallet strategy relying on both transaction hash and address balance to enhance transaction confirmation.

A. *Transaction Confirmation*

There are no notifications for users to confirm completion of transactions in Bitcoin. After authorizing a transaction T_x with a signature σ on it, the wallet broadcasts the transaction T_x into the Bitcoin network and waits for in to be included in blockchain. Data on the blockchain is public to all its users and everyone has access to query it. Until the wallet finds certain blocks appended on the block containing the transaction T_x it just broadcasted, it confirms this transaction T_x has been completed. The originator of Bitcoin recommends waiting until 6 blocks to be appended on the blockchain. Fig 4 shows a transaction status which has received 6 confirmations through the website blockchain.info[17].

In our scheme, this mechanism is still utilized in the first phase for a transaction confirmation. We use the hash of it as transaction ID to recognize whether it is included in the blockchain.

B. *Balance Query*

Besides the transaction hash, address balance is considered as an efficient way to confirm completion of a transaction in case of transaction malleability.

As usual in daily life, we spend coins in our pocket wallet to buy goods from markets. In the case that we don't remember whether we have paid for commodity or there is a dispute, what will we do? Looking into wallet to check our balance should be the prior choice for most of us.

A user wallet can obtain which address it's going to spend from the transaction input referencing to its predecessor transaction and its index. After querying the transaction hash in the blockchain, the user wallet should also query the balance of corresponding addresses to judge whether its bitcoins has been spent. If the appropriate amount is deducted from the address balance, we will check where the coins go. Otherwise, we will keep waiting for the confirmation of transactions. The address balance is available in the blockchain data on local storage or on some public website like blockchain.info[17] and blockexplorer.com[18], shown as Fig 5.

C. *Update Transaction Record*

Each transaction is stored in a fixed format by the wallet. After broadcasting a transaction to its neighbors directly, the wallet queries the public blockchain over a fixed period whether it has been recorded through its hash. Until there has been six blocks appended to the block including this transaction, the wallet confirm this transaction has completed.

In another case, the transaction indexed by the hash in wallet has not been confirmed for a long time, but the address balance it transfers has been deducted. We look up the blockchain for which transaction spent the coins belong to these addresses. Then we check the content of the latter transaction to see whether it is the same as the original one. If both of the two transactions have the same inputs and outputs, we think of them as the same transaction. The record of the original transaction is replaced by the new one and its status turns into confirmation.

Since all the transactions in the blockchain has been verified by the whole Bitcoin network, the new transaction different from the original one must be signed legally. It could be accepted by our wallets just with another identify hash. Furthermore, we can discover tempering transaction incident has happened through transaction malleability. Nevertheless, there is no impact on our wallet no matter it is on purpose or an accident.

V. Conclusion

Bitcoin is an outstanding cryptocurrency amongst other digital currencies, but has a risk to lose money caused by transaction malleability. Bitcoins worth millions of dollars was stolen due to transaction malleability. It has caused serious property loss of Bitcoin users and attracted much attention. Some different schemes to overcome this difficulty have been proposed, but each of these systems has its own deficiency. We delve into the principle of Bitcoin transactions and various attacks because of transaction malleability. In order to solve this issue, we proposed an efficient strategy to eliminate transaction malleability. Our scheme combined transaction hash and address balance to confirm completion of transactions. This manner will not only detect attacks through transaction malleability, but also resist such attacks. Our wallet scheme is compatible with existing Bitcoin network and easy to implement or upgrade existing wallet program.

ACKNOWLEDGMENT

Our research is supported by the National Natural Science Foundation of China (Project No. 61601476 and Project No. 61672527).

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] E. Voorhees, "Bitcoin: The libertarian introduction," 2012. [Online]. Available: http://evoorhees.blogspot.com/2012/04/bitcoinlibertarian-introduction.html

[3] V. S. Miller, "Use of elliptic curves in cryptography," in *Lecture Notes in Computer Sciences; 218 on Advances in cryptology—CRYPTO 85*, 1986.

[4] "Technical background of bitcoin adresses," 2012. [Online]. Available: https://en.bitcoin.it/wiki/Technical_background_of_Bitcoin_addresses

[5] "New attack vector," 2012. [Online]. Available: http-s://bitcointalk.org/index.php?topic=8392.msg122410#msg122410

[6] K. Shirriff, "Bitcoin transaction malleability: looking at the bytes," 2014. [Online]. Available: http://www.righto.com/2014/02/bitcointransaction-malleability.html

[7] M. Andrychowicz, S. Dziembowski, D. Malinowski, and u. Mazurek, "Fair two-party computations via bitcoin deposits," in *Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7*, 2014.

[8] M. Andrychowicz, S. Dziembowski, D. Malinowski, and ukasz Mazurek, "How to deal with malleability of bitcoin transactions," *Computer Science*, 2013.

[9] A. Back and I. Bentov, "Note on fair coin toss via bitcoin," *Computer Science*, 2014.

[10] C. Decker and R. Wattenhofer, "Bitcoin transaction malleability and mt-gox," in *Computer Security - ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11*, 2014.

[11] U. Rajput, F. Abbas, R. Hussain, H. Eun, and H. Oh, "A simple yet efficient approach to combat transaction malleability in bitcoin," in *Information Security Applications: 15th International Workshop, WISA 2014, Jeju Island, Korea, August 25-27*, 2014.

[12] U. Rajput, F. Abbas, and H. Oh, "A solution towards eliminating transaction malleability in bitcoin," *Journal of Information Processing Systems*, 2016.

[13] P. Wuille, "Dealing with malleability," 2014. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0062.mediawiki

[14] G. O. Karame and E. Androulaki, "Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin," 2012.

[15] M. Rosenfeld, "Analysis of hashrate-based double-spending," 2012.

[16] T. Bamert, C. Decker, L. Elsen, R. Wattenhofery, and S. Welten, "Have a snack, pay with bitcoins," in *IEEE P2P 2013 Proceedings*, ser. IEEE P2P 2013 Proceedings, 2013, pp. 1–5.

[17] "blockchain," 2017. [Online]. Available: https://blockchain.info/

[18] "blockexplorer," 2017. [Online]. Available: https://blockexplorer.com/