Anonymizing Bitcoin Transaction

Dimaz Ankaa Wijaya[1], Joseph K. Liu[1](✉), Ron Steinfeld[1],
Shi-Feng Sun[2], and Xinyi Huang[3]

[1] Faculty of Information Technology, Monash University, Melbourne, Australia
dawij5@student.monash.edu,
{joseph.liu,ron.steinfeld}@monash.edu
[2] Shanghai Jiao Tong University, Shanghai, China
[3] Fujian Normal University, Fuzhou, China

Abstract. Bitcoin is a new online decentralised payment system equipped by a cryptographic system which runs in a peer-to-peer network. While it denies any central authority, it can still verify and validate the transactions by its protocol. To make the transactions accountable, Bitcoin uses an open database which can be seen and checked by anyone. Despite no direct relationship between the Bitcoin transactions and the identity of the users, the information about the users can still be gathered by analysing the information contained in the transactions. We propose a protocol which minimises the relationship between the transactions to protect the information of the payer from the curious payee.

Keywords: Bitcoin · Privacy · Anonymity

1 Introduction

1.1 Bitcoin

In the world's economic system, money has an important role as a medium of exchange where people can trade between themselves by using a specified unit. Gold and precious metals had their glory before later replaced by fiat money, such as US dollars. Also, with the development of information technology, there is a new form of money called digital currency, where in the beginning this system gets its popularity within online games, as we can see in a form of World of Warcraft Gold, Linden Dollars, or even Facebook Credit [1].

Bitcoin is one of the newest inventions of digital payment system initially proposed in 2008 and has been fully operational in early 2009 [2]. It was worth nothing in the beginning, but now the market of Bitcoin has reached $6 billion, in which 1 BTC is worth $396.62 with the total of 15.13 million bitcoins in circulation. The figure shows a massive development of the Bitcoin system and more people recognise the existence of Bitcoin system. As a pioneer of cryptocurrency, it offers fresh ideas of how anonymous people can do online transactions without any central authoritative body such as bank but can still create trusted transactions among them. Thus, it utilises several mature technologies in the cryptographic field, such as digital signature, hash functions, and public key cryptosystem.

F. Bao et al. (Eds.): ISPEC 2016, LNCS 10060, pp. 271–283, 2016.
DOI: 10.1007/978-3-319-49151-6_19

1.2 Bitcoin Anonymity

Bitcoin is designed to become an anonymous payment system with no linked information between public keys (Bitcoin addresses) and the individuals controlling those public keys. But in practice, there are properties of Bitcoin transactions which can be used to analyse the characteristics of transactions and how the transactions of bitcoins are done. By using certain methods, the behaviours of the bitcoin owners can be determined and in some cases, the Bitcoin addresses can be linked to the real identity of the users. Therefore, the privacy of the Bitcoin owners can be at stake knowing that the transactions can be analysed.

Möser [3] analysed some existing bitcoin mixing services: Bitcoin Fog[1], BitLaundry[2], and SharedCoin[3]. He investigated the way these services mix his ordered transactions and then compared the performance of these three services. He downloaded the information about his transactions from the blockchain and drew graphs based on the information he gathered. He found that because there were not many people using these services, in an experiment, his coins were reused and therefore the anonymizing result was not significant. He also concluded that combination of these services may deliver a better result and decrease the risk of the services stealing the bitcoins. Another paper by the same author [4] expanded the findings by interconnecting the mixing services and money laundering. They pointed out that the effort of identifying Bitcoin users by enforcing Know Your Customer (KYC) principle over the edges of Bitcoin system such as exchange services may be disrupted by those mixing services. The mixing services may provide significant problems towards the identification of Bitcoin transactions.

1.3 Our Contribution

The existing solutions for anonymizing Bitcoin transactions do not protect the information of the payer and the payee from themselves. Moreover, the service providers of those solutions hold the full information of how the anonymizing process is done, and therefore the identity of the participants can still be disclosed by the service providers. We propose a new protocol of anonymizing Bitcoin transactions. The protocol is designed to be fully compatible with the current Bitcoin main network system and therefore it only utilises features that are already standardised by Bitcoin core developers and deployed in the Bitcoin Core version 0.11.2.

To summarise, below is the characteristics of the proposed protocol.

- The protocol protects the Bitcoin address of the payer from the payee.
- The protocol does not allow any participant to learn the whole information of the chained transactions by dividing the information into several parts.
- The protocol can be cancelled at any stage without any participant losing money in an honest majority condition.

[1] http://bitcoinfog.com.

[2] https://en.bitcoin.it/wiki/Bitcoin_Laundry.

[3] https://sharedcoin.com.

2 Related Works

2.1 Anonymous Coin Protocol

One of the first ideas of anonymous payment system can be traced back since Chaum proposed a method called blind signatures [5]. It enables users to pay others without being able to be tracked who the payers are. With the feature also comes the counterfeit-proof by applying digital signature and cryptographic techniques.

A new concept called Zerocoin was proposed [6]. Zerocoin was developed based on zero knowledge mechanism. It supports anonymous transactions without a single authority nor trusted party. The main part of this approach is to allow users to create their own coins with an assumption that they have sufficient amount of bitcoin represented in the new coins they create. The newly created coins and the original bitcoins are bounded by using digital commitment scheme which will prevent double spending of bitcoins they originally hold. Although this approach seems to be promising, it needs a major change in current Bitcoin protocol and the requirements of running such protocol will require larger storage and memory than the current Bitcoin system.

As an improvement of Zerocoin, Zerocash was introduced [7]. Zerocash is equipped with a scheme called decentralized anonymous payment. It eradicates the information of the coin receivers as in Zerocoin, thus offer a higher level of anonymity. Zerocash transaction allows its users to privately pay each other and hides information related to the transaction such as the source coins, destination, and the amount of transacted coins. However, similar to Zerocoin, the Zerocash scheme cannot be implemented in the current Bitcoin system because it requires modification of the current Bitcoin protocol.

2.2 Coin Anonymizer

Martin and Taaki [8] implemented an idea called CoinJoin [9] which is an alternative solution to the anonymity problem in the Bitcoin system. Within CoinJoin there is a special client application which communicates with a server. The CoinJoin server creates a single multi-signature transaction which combines multiple inputs and multiple outputs from multiple clients and ensures that each output receives the correct amount of coins. Then the transaction is signed by all clients if they agree with it before the server sends the transaction to the Bitcoin network. There are problems in the CoinJoin system. First, as these addresses are involved in a single transaction, they can still be traced. Second, it may be a problem to find other users who want to mix their coins together as they need to be online at the same time to sign the transaction [7].

CoinSwap is another coin anonymizer protocol. The idea of CoinSwap is proposed by Maxwell [10] in Bitcointalk forum. The operation of CoinSwap will result in hiding the relationship between the payer and the payee. CoinSwap enables those participants to create reliable transactions by providing a guarantee that each participant cannot steal the fund. In the CoinSwap protocol, a third party is needed to pose as a gateway between the payer and the payee. The protocol utilises several mechanisms to accommodate this solution: 2-of-2 escrow and hash-locked transaction. The 2-of-2 escrow is a transaction which requires at least 2 signatures to validate. The hash-locked transaction requires a secret key to secure the transaction.

3 Preliminaries

3.1 Deterministic Wallet

Deterministic Wallet is a type of Bitcoin wallet which has the ability to create an infinite number of child public keys (or child addresses) from a master public key by using an index [11]. The private keys do not need to be known before the generation of the child addresses because they can be generated from a master private key which corresponds to the master public key by applying the same index values used in the child addresses generation.

3.2 Pay to Script Hash (P2SH)

Pay to Script Hash (P2SH) is another method of Bitcoin payment [12]. P2SH is a standard under BIP 16 which describes the detail of P2SH [13]. It enables Bitcoin users to construct a script as a requirement before redeeming the fund.

3.3 Locktime

Locktime, or also called as nLockTime, is a feature in Bitcoin system which can be used to determine the earliest time the transaction can be confirmed in the system by using 4 bytes data [14].

3.4 Sequence Number

Sequence number is 4 bytes information in Bitcoin raw transaction which can be used to setup the transaction version [15]. To change the transaction, the next version of the transaction must have higher sequence number than its predecessor.

3.5 CheckLockTimeVerify (CLTV)

CheckLockTimeVerify (CLTV) is a feature proposed by Todd [16] to lock a transaction until a certain time. By using CLTV, the transaction can be immediately included in the blockchain but it freezes and cannot be redeemed until a certain time.

3.6 Multisignature

Multisignature is a type of digital signature which requires multiple participants to sign a single document [17]. In a certain case, it is useful to add more security feature by dividing the authorization right to several participants. Multisignature is used in the Bitcoin system in which the user creates a transaction requiring multiple signatures to validate [18].

The multisignature scheme in Bitcoin is denoted as `m-of-n multisignature`. The value `m` is the minimum number of signatures required to validate the transaction.

The value n is the total number of possible signatures which can be used to validate for the transaction. Multisignature feature enables the escrow scheme to be constructed within Bitcoin system and thus may increase the security of the transaction [19].

3.7 Atomic Transaction

According to Tiernan [20], an atomic transaction is a type of transaction in which the participants can cancel the transaction at any stage. If the transaction proceeds, every participant gets what the participant wants, or if the transaction is cancelled, then no participant gets the payment nor suffers loss. The standard transactions cannot be used to construct atomic transaction; it needs a non-standard transaction or a P2SH scheme [21].

3.8 Taint Analysis

In the Bitcoin system, taint is a correlation between Bitcoin addresses [22]. The correlation comes from the past transactions (received or spent). Taint analysis determines the closeness between multiple Bitcoin addresses. As the Bitcoin system can be considered as an open ledger, the taint analysis can be queried from the Bitcoin network based on the transaction history of the addresses. In term of anonymity, the addresses should not be related each other despite being analysed by using taint analysis, which can also be determined as taint proof.

3.9 Bitcoin OpCodes

Bitcoin Operation Codes (OpCodes) are commands used in the Bitcoin script to evaluate the inputs [23]. The OpCodes together with several parameters construct the script to evaluate the inputs and produce an output. The script will evaluate the inputs and the output will determine whether the fund can be executed.

3.10 Notations

In this paper, we use notations to represent Bitcoin transactions used in the proposed protocol. Let Alice (A) send money to Bob (B), then the transaction (TX) will be called as TX_AB. If the transaction happens in the first phase of the protocol, then the transaction will be written as TX_AB1, while if it is in the second phase of the protocol then it will be called as TX_AB2. We also determine other participants represented with names such as Carol, Darth, Eve, Frank, and George. The participants may also be represented by the first letter of their names.

4 Our Proposed Solution

4.1 Communication Channel

The proposed protocol of anonymizing the Bitcoin transaction without any trusted system cannot be created without a communication channel. In this paper, it is assumed there exists an anonymous communication channel e.g. Tor [24] which can be used by multiple users to exchange information without revealing any information about their identity. The communication records cannot be linked with the transactions created within the proposed protocol. It is also assumed that the participants use a secure communication channel to send raw transactions and signed transactions between participants.

After the participants agree to form the transactions, a secure anonymous communication channel must be set up between them which can only be accessed by the participants. Let it be called general channel. Another separate communication channels must also be set up for each group in the protocol. These channels are separate channels from the general channel. Let the latter channels be called group channels. Although a participant may become a member of multiple groups, the participant is assumed to not cooperate with any member of another group.

4.2 The Protocol

Let Alice act as the payer, Bob as the payee, while Carol, Darth, Eve, Frank, and George as the middlemen. The transactions can be shown in the Fig. 1.

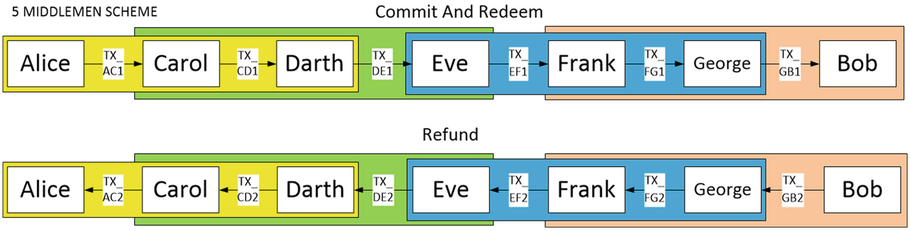

Fig. 1. The 5 middlemen scheme

The protocol requires at least 7 participants: a payer, a payee, and 5 middlemen. The participants are grouped into 4 groups, each consists of 3 members:

- Group 1: Alice, Carol, Darth
- Group 2: Carol, Darth, Eve
- Group 3: Eve, Frank, George
- Group 4: Frank, George, Bob

Each group will construct 2-of-3 multisignature over Pay to Script Hash (P2SH) scheme. By employing the 2-of-3 multisignature, if 1 of 3 members in a group cheats, then the payee of the group can still get paid. This creates a form of an escrow. The protocol also employs CheckLockTimeVerify (CLTV) to lock the fund from the payer and therefore ensures the payees that the payers already have a sufficient fund to construct a valid Bitcoin transaction. In order to make sure that the payers get a refund in case of the transaction is cancelled, the form of atomic transactions are used in the P2SH script. LockTime is also used to ensure that the transactions are done in the correct sequence. The script used in the P2SH scheme will be discussed in the appendix.

The protocol consists of several phases which can be described as below.

- Phase 0: Preparation
  - Instead of providing his address, Bob creates a new deterministic public key pair which consists of a master public key and a master private key. Bob then sends the public key to Alice in a secure channel.
  - Alice sets up an anonymous communication channel with all of the participants including Bob.
  - All participants except Bob create a new deterministic public key pair. Each of the participants publishes the public key only to the members of the group and relate the public key to the session.
  - Alice sends Bob's public key to members of Group 4 (Frank, George, and Bob) with the transaction order of paying Bob certain amount of money.
  - The sender of each transaction creates the transaction along with new addresses for the receiver and the escrow by using the deterministic public keys.
  - TX_AC is defined as a transaction between Alice and Carol, while Darth acts as an escrow. Alice creates new addresses for Carol and Darth and then informs the random value used in the address generation to the group.
  - TX_CD is defined as a transaction between Carol and Darth, while Alice acts as an escrow. Carol creates new addresses for Darth and Alice and then informs the random value used in the address generation to the group.
  - TX_DE is defined as a transaction between Darth and Eve, while Carol acts as an escrow. Darth creates new addresses for Eve and Carol and then informs the random value used in the address generation to the group.
  - TX_EF is defined as a transaction between Eve and Frank, while George acts as an escrow. Eve creates new addresses for Frank and George and then informs the random value used in the address generation to the group.
  - TX_FG is defined as a transaction between Frank and George, while Eve acts as an escrow. Frank creates new addresses for George and Eve and then informs the random value used in the address generation to the group.
  - TX_GB is defined as a transaction between George and Bob, while Frank acts as an escrow. George creates new addresses for Bob and Frank and then informs the random value used in the address generation to the group.
  - The group members can check the address generation and create the private keys which correspond to the addresses generated. The random values are shared within the group but they need to be kept secret from other groups.

The processes within phase 0 can be illustrated in Fig. 2.

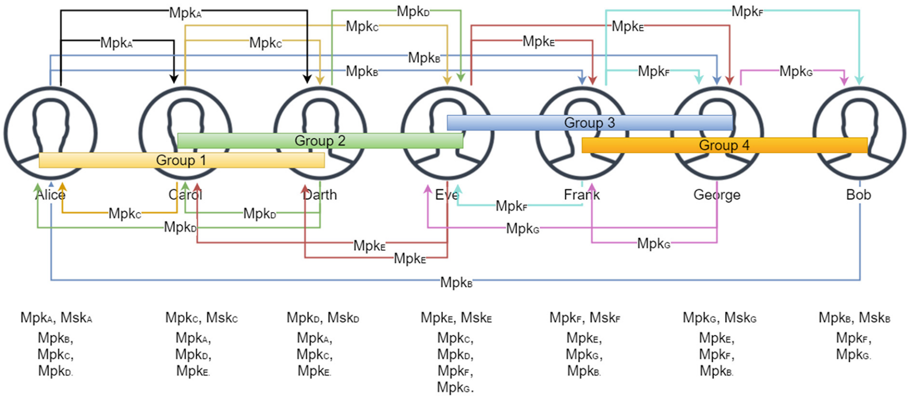

Fig. 2. Key sharing diagram.

- Phase 1: Setup the commit 2-of-3 multisignature escrow transaction.
  - Alice creates a P2SH transaction TX_AC1 which can be redeemed by 2-of-3 multisignature of Alice, Carol, and Darth or by Alice's signature after certain amount of time defined in CLTV. TX_AC1 is then published to the network. The TX_AC1 has a CLTV of C_AC1.
  - Carol creates a P2SH transaction TX_CD1 which can be redeemed by 2-of-3 multisignature of Alice, Carol, and Darth or by Carol's signature after a certain amount of time defined in CLTV. TX_CD1 is then published to the network. The TX_CD1 has a CLTV of C_CD1 < C_AC1.
  - Darth creates a P2SH transaction TX_DE1 which can be redeemed by 2-of-3 multisignature of Carol, Darth, and Eve or by Darth's signature after a certain amount of time defined in CLTV. TX_DE1 is then published to the network. The TX_DE1 has a CLTV of C_DE1 < C_CD1.
  - Eve creates a P2SH transaction TX_EF1 which can be redeemed by 2-of-3 multisignature of Eve, Frank, and Bob or by Eve's signature after a certain amount of time defined in CLTV. TX_EF1 is then published to the network. The TX_EF1 has a CLTV of C_EF1 < C_DE1.
  - Frank creates a P2SH transaction TX_FG1 which can be redeemed by 2-of-3 multisignature of Eve, Frank, and George or by Frank's signature after a certain amount of time defined in CLTV. TX_FG1 is then published to the network. The TX_FG1 has a CLTV of C_FG1 < C_EF1.
  - George creates a P2SH transaction TX_GB1 which can be redeemed by 2-of-3 multisignature of Frank, George, and Bob or by George's signature after a certain amount of time defined in CLTV. TX_GB1 is then published to the network. The TX_GB1 has a CLTV of C_GB1 < C_FG1.

- Phase 2: Redeem the transactions by using 2-of-3 multisignature
  - Bob creates TX_GB2 which redeems TX_GB1, signs it, and sends it to George. George signs the transaction and sends TX_GB2 to the network. If George does not want to sign the transaction, then Bob asks Frank to sign the transaction TX_GB2.
  - George creates TX_FG2 which redeems TX_FG1, signs it, and sends it to Frank. Frank signs the transaction and sends TX_FG2 to the network. If Frank does not want to sign the transaction, then George asks Eve to sign the transaction TX_FG2.
  - Frank creates TX_EF2 which redeems TX_EF1, signs it, and sends it to Eve. Eve signs the transaction and sends TX_EF2 to the network. If Eve does not want to sign the transaction, then Frank asks Bob to sign the transaction TX_EF2.
  - Eve creates TX_DE2 which redeems TX_DE1, signs it, and sends it to Darth. Darth signs the transaction and sends TX_DE2 to the network. If Darth does not want to sign the transaction, then Eve asks Carol to sign the transaction TX_DE2.
  - Darth creates TX_CD2 which redeems TX_CD1, signs it, and sends it to Carol. Carol signs the transaction and sends TX_CD2 to the network. If Carol does not want to sign the transaction, then Darth asks Alice to sign the transaction TX_CD2.
  - Carol creates TX_AC2 which redeems TX_AC1, signs it, and sends it to Alice. Alice signs the transaction and sends TX_AC2 to the network. If Alice does not want to sign the transaction, then Carol asks Darth to sign the transaction TX_AC2.
- Phase 3: If the transaction is cancelled and the fund is not redeemed by the receivers after CLTV time is expired, then the senders can get their money back. This is done by creating a new transaction that redeems the first transaction sent to the network by each sender.

5 Comparisons

The proposed protocol can be compared with other anonymizing solutions as described in Table 1.

Table 1. Comparison between anonymizing solutions.

No	Characteristics	Proposed protocol	Zerocash [7]	CoinJoin [9]	CoinSwap [10]
1	Atomic transaction[a]	V	V	V	X
2	No participant holds all information	V	V	X	X
3	Compatible with current Bitcoin protocol	V	X	V	V
4	Hides payer’s address from the payee	V	V	X	V
5	Taint proof[b]	V	V	X	V
6	Cheating security	V	V	V	V

[a] The concept of atomic transaction is discussed in Sect. 3.7.

[b] Taint analysis and taint proof is discussed in Sect. 3.8.

From the table above, it can be concluded that the proposed protocol can fulfil all the required characteristics of an anonymizing protocol. Zerocash in its protocol requires the Bitcoin transaction to be flagged as a Zerocash transaction and therefore requires modification to Bitcoin core system. Moreover, to create a payment, a payer needs to know the public key of the payee, despite the transaction will be encrypted and no observer will know which coin is spent.

In CoinJoin, all participants have the full information of the transaction because they need to sign the transaction, despite they may not be able to determine the identity of the participants, they can still enumerate the input addresses and the output addresses. The addresses may also be connected each other because they are used in the same transaction and therefore it is not taint proof.

CoinSwap is not atomic because it requires approval from the receiver to create refund transaction. Therefore, if the receiver does not want to sign the refund transaction, the fund owned by the sender cannot be claimed. Moreover, if one of the participants decides to reveal the secret value, then the chained transactions can be linked each other by having the same secret value. CoinSwap also only utilises a single third party and therefore creates a single point of failure in case of the third party decides to reveal the information.

All of the solutions have a mechanism of preventing the participants from cheating. Zerocash has a cryptographic mechanism to proof that the participants are honest. In CoinJoin, each of the participant can check the validity of the transaction prior to signing the transaction. In CoinSwap, the transactions are guaranteed by the hash-locked-transaction and 2-of-2 multisignature mechanisms. In the proposed protocol, the cheating security is provided by employing 2-of-3 multisignature.

6 Security Evaluation

6.1 Anonymity Model

We propose the concept of unlinkability and anonymity to measure the privacy. Unlinkability is the inability to relate different items [25]. It means that the items must not have a specific attribute to distinguish them from any other similar items.

Anonymity is the inability to identify a particular subject in a set of subjects [25]. We assume there are N number of transactions created by N number of different payers employing the same protocol in the same configuration of middlemen within a time period. A transaction sent to Bob from Alice is chosen uniformly random from N transactions within that time period. Bob then tries to identify Alice by cooperating with one of the middlemen. Our scheme has anonymity characteristic if the probability of Bob guessing Alice's address (P) is determined by the following equation.

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

6.2 Cheating Model

We define the cheating model of the protocol as follows. In the cheating scenario, one or more participants try to cheat by not paying or paying less amount of money to others despite getting a full payment from others. With the assumption that at least 1 of the sender or the escrow within each group is honest and assuming that the receiver is always honest, our scheme is secure if the probability of any participant tries to cheat is negligible.

6.3 Anonymity Evaluation

We first investigate the information gained by each participant which is shown in Table 2 below. Because the transactions within the Bitcoin system is publicly available, we also assume that everyone has the ability to access that information.

Table 2. Information gained by each participant.

Participant	Knowledge of transaction	Knowledge of deterministic public key	Group membership
Alice	TX_AC,TX_CD	Alice, Carol, Darth, Bob	1
Carol	TX_AC, TX_CD, TX_DE	Alice, Carol, Darth, Eve	1,2
Darth	TX_AC, TX_CD, TX_DE	Alice, Carol, Darth, Eve	1, 2
Eve	TX_CD, TX_DE, TX_EF, TX_FG	Carol, Darth, Eve, Frank, George	2, 3
Frank	TX_EF,TX_FB, TX_GB	Eve, Frank, George, Bob	3,4
George	TX_EF, TX_FB, TX_GB	Eve, Frank, George, Bob	3,4
Bob	TX_FG,TX_GB	Frank, George, Bob	4

In order to reveal the transaction sent from Alice to Bob, Bob must cooperate with at least 2 of the middlemen. By using the methods explained above, Bob then can construct the linked transactions which lead to the original transaction sent by Alice.

The same arguments would also apply to the unlinkability characteristic of the protocol. Bob cannot tell 2 different transactions coming from Alice assuming Bob receives multiple transactions from multiple senders each has the same amount of money.

6.4 Cheating Evaluation

The protocol utilizes 2-of-3 multisignature scheme and timing to mitigate the cheating risk. By using 2-of-3 multisignature scheme, if a payer refuses to sign the redeem transaction, then the payee can ask the escrow party to sign the transaction and the redeem transaction is valid. Despite the middlemen have the chance to cheat, they stake their reputations if they do not behave honestly.

The similar way goes to a case in which a middleman tries to pay less money to the payee, then the payee rejects the transaction and asks the escrow to sign the transaction on behalf of the payer. The middlemen can check whether they have set the correct amount of money by using the information provided within the transactions and the information provided by Alice in the beginning of the protocol.

The protocol also uses a specialised P2SH script which can be used to construct atomic transactions which can be cancelled at any stage. If the transaction is cancelled, all participants can redeem their own fund. The timing scheme is implemented by the CLTV and Locktime, and therefore the cheating scheme can be easier to detect.

In the case of one or more participants do not behave honestly, at least 2 members in each group must be honest in order to proceed the protocol.

In the case of any middleman tries to forge the transactions by faking the digital signature of Bitcoin, then the security relies on the unforgability of the 256 bit private key of ECDSA.

7 Conclusion and Further Work

The proposed protocol can be an alternative solution to hide the information of the payer's address from the payee. By implementing 2-of-3 multisignature, the escrow can take part in the transaction when a participant in the group tries to cheat by not providing the correct signature.

Despite the ability to recover the protocol up to 2 malicious participants, there are concerns regarding the protocol. If the value of N which denotes the number of transactions utilising the same protocol is small, then the effort of analysing the transactions can be smaller. The custom P2SH script can be utilized to distinguish the transactions and mark them as part of anonymizer protocol. Future works could be expanded to minimise the effect of the custom script.

Acknowledgments. This work is supported by National Natural Science Foundation of China (61472083). This work is supported by the Science & Technology Plan Projects of Shenzhen (JCYJ20150324140036830, GJHZ20160226202520268).

References

1. Piasecki, P.: Design and security analysis of Bitcoin infrastructure using application deployed on Google Apps Engine. In: Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej. University of Warsaw (2012)
2. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
3. Möser, M.: Anonymity of bitcoin transactions. In: Münster Bitcoin Conference (2013)
4. Moser, M., Böhme, R., Breuker, D.: An inquiry into money laundering tools in the Bitcoin ecosystem. In: eCrime Researchers Summit (eCRS). IEEE (2013)
5. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology, pp. 199–203. Springer, New York (1983)
6. Miers, I., et al.: Zerocoin: anonymous distributed e-cash from bitcoin. In: 2013 IEEE Symposium on Security and Privacy (SP). IEEE (2013)

7. Ben Sasson, E., et al.: Zerocash: decentralized anonymous payments from Bitcoin. In: 2014 IEEE Symposium on Security and Privacy (SP). IEEE (2014)
8. Martin, P., Taaki, A.: Anonymous Bitcoin Transactions (2013). https://sx.dyne.org/anontx/. Accessed 25 Aug 2015
9. Maxwell, G.: CoinJoin: bitcoin privacy for the real world (2013). https://bitcointalk.org/index.php?topic=279249.0. Accessed 12 Sept 2015
10. Maxwell, G.: CoinSwap: transaction graph disjoint trustless trading (2013). https://bitcointalk.org/index.php?topic=321228.0. Accessed 12 Sept 2015
11. Maxwell, G.: Deterministic Wallets (2011). https://bitcointalk.org/index.php?topic=19137.0. Accessed 12 Sept 2015
12. Bitcoin Wiki. Pay to Script Hash (2012, 27 May 2015). https://en.bitcoin.it/wiki/Pay_to_script_hash. Accessed 9 Jan 2016
13. Andresen, G.: Pay to Script Hash (2012). https://github.com/bitcoin/bips/blob/master/bip-0016.mediawiki. Accessed 9 Jan 2016
14. Harding, D.A.: Locktime, nLockTime (2015). https://bitcoin.org/en/glossary/locktime. Accessed 12 Jan 2016
15. Harding, D.A.: Sequence Number (Transactions) 2015. https://bitcoin.org/en/glossary/sequence-number. Accessed 12 Jan 2016
16. Todd, P.: OP_CHECKLOCKTIMEVERIFY (2014). https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki. Accessed 12 Jan 2016
17. Bellare, M., Neven, G.: Identity-based multi-signatures from RSA. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 145–162. Springer, Heidelberg (2006). doi:10.1007/11967668_10
18. Andresen, G.: M-of-N Standard Transactions (2011). https://github.com/bitcoin/bips/blob/master/bip-0011.mediawiki. Accessed 28 Sept 2015
19. Bitcoin Wiki. Contract (2012), 8 July 2015. https://en.bitcoin.it/wiki/Contract. Accessed 28 Sept 2015
20. Tiernan, N.: Alt Chains and Atomic Transfers, 7 May 2013. https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949. Accessed 28 Sept 2015
21. xHire. Atomic protocol #1 (2015). http://www.coincer.org/2015/01/27/atomic-protocol-1/. Accessed 11 Jan 2016
22. Piuk. What is taint? (2012). https://bitcointalk.org/index.php?topic=92416.msg1018943#msg1018943. Accessed 19 Sept 2015
23. Bitcoin Wiki. Script, 25 September 2015. https://en.bitcoin.it/wiki/Script. Accessed 28 Sept 2015
24. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router, DTIC Document (2004)
25. Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management (2010)