Lockcoin: a secure and privacy-preserving mix service for bitcoin anonymity

Bin Wang, Zijian Bao, Yongxin Zhang, Qinghao Wang, and Wenbo Shi

Department of Computer Science and Engineering Northeastern University, China
binge1638@163.com zijianbao@gmail.com silence_yongxin@163.com
15161183163@163.com swb319@hotmail.com

Abstract. We propose Lockcoin, a secure and privacy-preserving mix service for bitcoin anonymity. We introduce mix servers to provide mix service for user to prevent attackers linking the input address with output address by using blind signature shceme, multisignature scheme. Lockcoin provides anonymity, scalability, bitcoin compatibillity, theft impossibility and accountability. We have proposed a prototype of Lockcoin based on bitcoin test network, experimental results show that our solution is efficient. Lockcoin's source codes are released on github.com/B-doublemint/LockCoin.

Keywords: blind signature · multi-singature · privacy-preserving.

1 Introduction

In 2008, a paper named 'A peer-to-peer electronic cash system' [1] was published by Nakamoto Satoshi, who opened a new era of digital currency. By sep 2018, about 17 million bitcoins had been issued with market cap reaching $1.25 billion [2].

However, as bitcoin is an open ledger, the transaction is collected and stored in the blockchain, and anyone can view the information on it. bitcoin can only provide pseudonyms, not anonymity. Dorit Ron and Adi Shamire [3] downloaded the full history of bitcoin and analyzed many statistical properties of its associated transaction graph. Regal Reid and Martin Harrigan [4] can prove that multiple pseudonymous addresses can be linked to a single user. Philip Koshy et al. [5] demonstrate the approach to map bitcoin addresses directly to IP address. Andrew Miller et al. [6] introduce AddressProbe, a technique to discover bitcoins public topology and influential nodes.

Therefore, there is an urgent need for providing bitcoin users a full anonymous service. Bonneau et al. [7] propose a protocol Mixcoin to introduce mix service for bitcoin, which wants to avoid the link between the input address and the output address. Luke Valenta and Brendan Rowan put forward the Blindcoin [8] protocol by improving the Mixcoin protocol, it uses blind signature to guarantee that mix server cannot know the link from input/output address. Gregory Maxwell propose Coinjoin [9] to shuffle the link of users address without a third party. However, it is susceptible to DoS attacks. Coinswap [10] is another proposal of

Gregory Maxwell which uses a third party to assist in trading, its anonymity depends on 2-of-2 escrow transactions. Van Saberhagen proposes to use ring signatures and stealth address technologies to build an anonymous higher e-cash program CryptoNote [11]. The verification complexity of the ring signature increases linearly with the anonymity provided. Zerocoin [12] proposes a new type of side chain, zerocoin to provide anonymity for bitcoin. It is compatible with bitcoin, but it has low system efficiency and performance. Zerocash [13] improves on Zerocoin by using zk-SNARKs Non-interactive zero-knowledge proof algorithm [14] to achieve stronger anonymity. However, its security depends on the honest destruction of the system's secret parameters.

Our contribution. In this paper, we propose Lockcoin on the basis of Mixcoin [7] and Blindcoin [8]. We propose a new mix service by using multi-signatures to prevent mix server stealing bitcoins from users. It has anonymity, scalability, bitcoin compatibility, theft impossibility and accountability. Lockcoin has the properties which an ideal solution should has. We also implemented Lockcoin in go language and performed the experiment. By using parallel strategy, experimental results show that our solution is efficient.

2 Background

2.1 Ideal property

For solving the privacy problem of bitcoin, an ideal solution should consider the following properties. See section 4.1 for comparisons.

Anonymity: The user should be the only entity that knows the link from input address to output address.

Scalability: The system should have good scalability to scale to numerous users.

Bitcoin compatibility: The system should be compatible with the current bitcoin system.

Theft impossibility: No party can steal bitcoin from other accounts.

Accountability (mix service only): When the protocol does not work properly, both parties can have a proof of the misconduct.

2.2 Blind signatures

Blind signature was introduced by David Chaum [15]. It is a form of digital signature. The content of a message is blinded before it is signed. The result of blind signature can be publicly verified against the original. Blind signature can also be used to provide unlinkability, which prevents the signer from linking the blinded message to a later un-blinded version. Blind signature schemes can be implemented using a number of common public key signing schemes, for instance RSA [16] and DSA [17].

2.3 Multi-signature

Multi-signature [18] is a digital signature scheme which allows a group of users to sign a single transaction. It needs multiple parties to work together to complete the signing process. These parties can be people, institutions or programmed scripts [19].

In this paper, a 2-of-2 multi-signature will be used, which means that the assets on the multi-signature address belongs to two users. Only if both two users sign the transaction, the bitcoin on the multi-signature address can be used.

3 Lockcoin protocol

3.1 System model

Our model proposes a protocol based on the mix service which requires mix server, the users and public log. Fig.1 shows the whole model diagram. For understanding, we only take a user's operations as an example. A = Alice (a user), M = mix server in the following. Parameter table is shown in Tab.1.

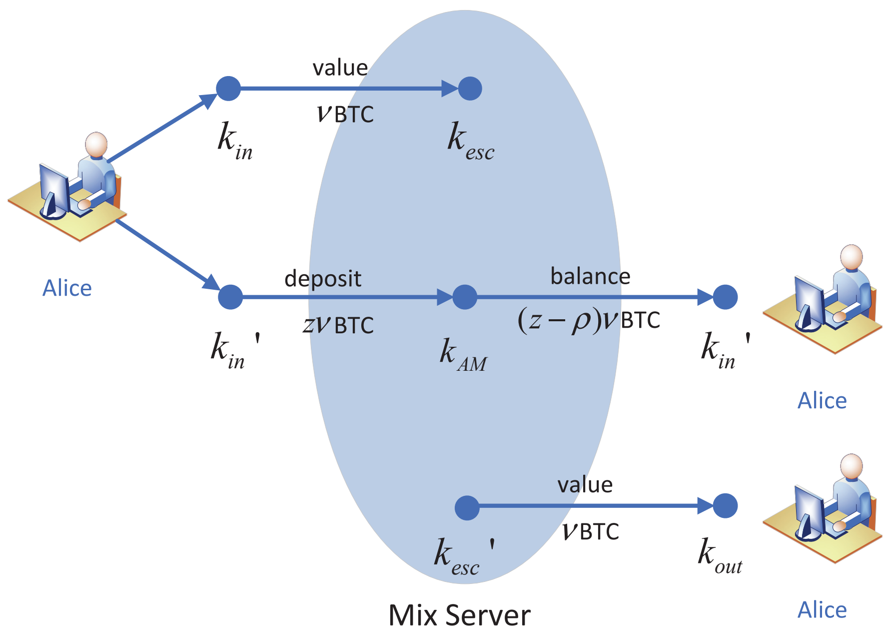

Fig. 1. model diagram

Mix server. It receives one-chunk (one-chunk means the amount of bitcoin must be the same for all users participating in the mix service each time) money from user's input address, and it puts one-chunk into user's output address, the model assumes that there are multiple competing mix servers, which has the corresponding M_{pub}, M_{priv}.

The users. The user wants to transfer funds from the input address which might associate with her real identity, and she can receive the same money in her new address. Also, user needs to pay the money laundering charges for this purpose.

Public log. Public log can be viewed as the bulletin board that can increase but can't be modified. In fact, it can be viewed as a bitcoin transaction *OP_RETURN* [20] which needs to consume little bitcoins.

Table 1. Parameter table.

Parameters	Description
k_{in}	the input address of A.
k_{out}	the output address of A.
k_{esc}	M generates a fresh escrow address.
k'_{esc}	the Ms escrow address from some previous transaction.
k_A	k_A is used to generate the address of 2-of-2 k_{AM}.
k_M	k_M is used to generate the address of 2-of-2 k_{AM}.
k_{AM}	A and M co-generate 2-of-2 signature address.
k'_{in}	A's address for payment to the A and M common address k_{AM}.
A'	an anonymous identity that A can use to post to the public log.
M_{pub}	the public key of M.
M_{priv}	the private key of M.
A_C	a secret commitment/encryption function of A.
A'_C	the inverse of A_C.
ω	the number of blocks which is required to confirm payment.
z	the deposit ratio.
v	the chunk size.
ρ	the mixing fee rate that A will pay.
D	the mix parameters, a tuple $\{$ v,t_1,t_2,t_3,t_4,t_5,t_6,t_7,ω,z,ρ,k_A $\}$.

3.2 Protocol process

The entire protocol flow is shown below.

(1) M announces its expected range of values on the bulletin board, where z is the deposit ratio, v is the chunk size to be mixed, ω is the number of blocks which is required to confirm payment. We will discuss how to set these parameters in section 3.3.

(2) A sends the protocol parameters to M, $\langle$D=(v,t_1,t_2,t_3,t_4,t_5,t_6,t_7,ω,z,ρ,k_A), [kout]$A_C\rangle$. Among them, (v,t_1,t_2,t_3,t_4,t_5,t_6,t_7,ω,z,ρ,k_A) are disclosed, and k_{out} is blinded, and the information is processed using the A_C blindness factor (only A know the inverse A'_C).

(3a) M uses k_A from A and k_M address which is generated by itself to generate 2-of-2 multi-signature address k_{AM}, and then it uses its own private key to sign $\{[k_{out}]A_C, k_{esc}, k_{AM}, D\}$.

(3b) If the mix server rejects the request sent from A, the protocol will terminate, A will delete the output address.

(4) A transfers deposit vz from k'_{in} to k_{AM} before t_1 time.

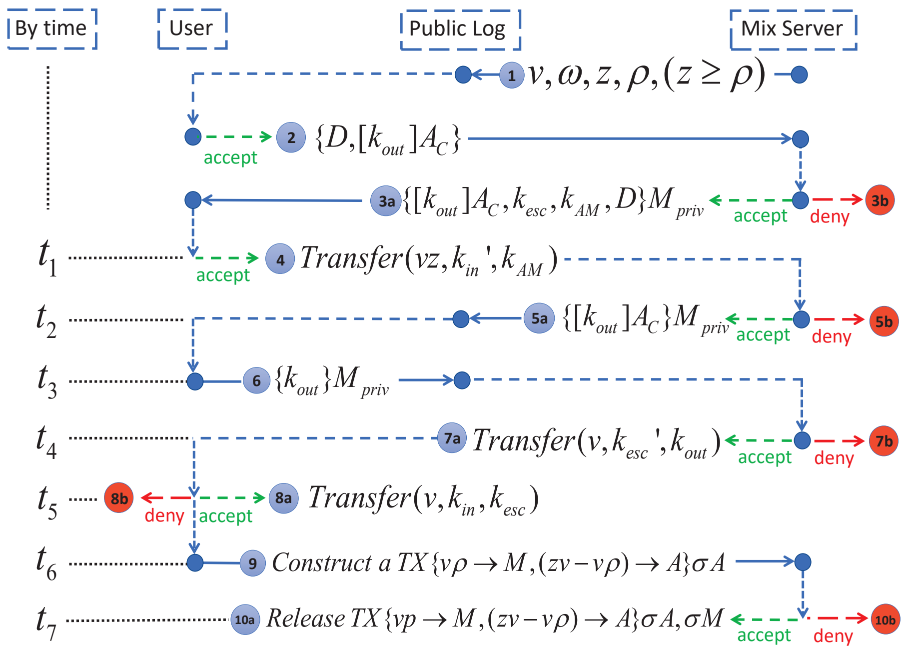

Fig. 2. protocol diagram

(5a) M checks the bitcoin record for whether the transactions in the step 4 are confirmed by ω blocks. If confirmed, M uses the private key to sign $[k_{out}]A_C$ and sends it to public log before t_2 time.

(5b) If M refuses to proceed with the protocol and doesn't enter step 5a, A will announce the evidence, which includes $\{[k_{out}]A_C, k_{esc}, k_{AM}, D\}M_{priv}$, the transaction (vz, k'_{in}, k_{AM}). Anyone can view the $[k_{out}]A_C$ that did not send a signature to public log. So it can be judged that M violates the protocol. But in our protocol, in the case of both sides are rational, almost no such situation, because if M violates the protocol it won't get any benefits.

(6) When A knows the information in the public log, A will use A'_C to unblind $\{[k_{out}]A_C\}$ M_{priv} to generate $\{k_{out}\}M_{priv}$ information, and then through anonymous network (such as Tor [21]) sent to public log by t_3 time. There-

fore, M can only view its signature on k_{out} address, but it can't know the user corresponding k_{in} address.

(7a) M finds the signature k_{out} address in step 6 on public log and transfers v bitcoins from k'_{esc} to k_{out} within t_4 time.

(7b) If M doesn't transfer v to k_{out}, A will execute the same operations as step 5b.

(8a) After ω blocks, A confirms that k_{out} address received M's v bitcoins, she will transfer v bitcoins from k_{in} to k_{esc} by t_5 time.

(8b) If A does not transfer v to k_{esc}, then mix server need to increase z to prevent such malicious attackers. For mix server, it will lose v bitcoins. For attackers, they'll lose (zv-v) bitcoins, so neither of the parties hopes it happens.

(9) A constructs a transaction that transfers vρ bitcoins from the (zv) bitcoins in the 2-of-2 multi-signature address to A's k'_{in}, and the other (zv-vρ) bitcoins is transferred to the k'_M, and A sends the transaction to M before t_6 time.

(10a) M signs the transaction constructed by step 9 to form the transaction $\{v\rho \rightarrow M, (vz - v\rho) \rightarrow A\}\sigma A,\sigma M$, and then M sends it to the bitcoin network to complete the entire protocol process.

(10b) If M does not sign, A will announce $\{[k_{out}]A_C, k_{esc}, k_{AM}, D\}M_{priv}$ and transaction$\{v\rho \rightarrow M, (vz\text{-}v\rho) \rightarrow A\}\sigma A$ and issue her own transaction hash in step 4 and 8a. Anyone can verify whether the M violates the protocol. In our protocol, in the case of M is rationality, this will not happen. Since the bitcoins in k_{AM} could not be used without the signature of both two parties.

3.3 Parameter setting

The purpose of parameter z is that we hope that the mix server will be able to adjust the size of the deposit dynamically according to the actual needs of the market. If it is too small, there may be malicious competitors who will continue to participate in the protocol by executing only to step 7a. If too large, user would like to choose a mix server which needs small deposit. v, ρ and ω have been discussed in Mixcoin [7].

4 Analysis and performance

In this section, properties, overheads and fees of Lockcoin will be discussed by comparing them with other schemes. We also have proposed a prototype of Lockcoin based on bitcoin network.

4.1 Properties

Lockcoin contains properties which an ideal solution should have, the properties comparison result is shown in Tab.2.

Anonymity. The anonymity of Lockcoin is provided by blind signature, which makes it impossible to link the input address and the output address. For an attacker, it can only obtain the bitcoin block transaction information and public log information, and it is unable to figure out the link.

Scalability. As the introduction of mix server, Lockcoin can provide services that correspond with market needs. It is efficient to add more users to mix service since users interact only with the centralized mix servers and not a single one. Further, if there is a single point of failure in a mix server, it is easily for user to turn to other different servers.

Bitcoin compatibility. The whole protocol acts as a service on bitcoin, so it is compatible with bitcoin system.

Resilience to DoS attack. Different from the Coinjoin. Due to the related fees, the DoS attack on the mix server will bring a great economic burden on the attacker, so Lockcoin can resist this attack.

Theft impossibility. Due to the security deposit by using 2-of-2 multi-signature, the server can not steal the deposit as it can not obtian the signature from user. Transfering the money to earn the mix fees is the best choice for the server.

Accountability. When one party violates the protocol, all b operations in section 3.2 can ensure the protocol is properly accountable, each one can take the proof to prove the violation party really did.

In the literature [22], each properties have been discussed between every mainstream schemes [7–10, 12, 13]. For accountability, Mixcoin [7], Blindcoin [8], Lockcoin can all present a proof of the mix servers misconduct. For scalability, because of introducing mix servers, Mixcoin [7], Blindcoin [8] and Lockcoin can easily resist to DoS attack. Coinjoin [9] will be easily broken by malicious participants. For Coinswap [10], as it is a p2p service, it doesnt have DoS attack problem. For Zerocoin [12] and Zerocash [13], their design can be resistance to DoS attack.

Table 2. Comparison of the properties with other schemes.

	Accountability	Scalability	Architecture	bitcoin compatible	Resilience to DoS attack	Theft impossible
Lockcoin	✓	✓	**service**	✓	✓	✓
Mixcoin [7]	✓	✓	service	✓	✓	×
Blindcoin [8]	✓	✓	service	✓	✓	×
Coinjoin [9]	-	weak	-	✓	×	✓
Coinswap [10]	-	×	p2p	✓	✓	partly
Zerocoin [12]	-	✓	altcoin	×	✓	✓
Zerocash [13]	-	✓	altcoin	×	✓	✓

4.2 Overheads and fees

The overheads and fees comparison result is shown in Tab.3.

Overheads. A successful Lockcoin mixing operation requires three message directly between A and M, two messages posted to the public log, three normal bitcoin transactions and one multi-signature transaction. So, 6 transaction traffic are needed in the bitcoin network in total.

Fees. Posting messages to the public log costs extra fees, in addition to the fees paid for the funds transfer. According to [11], the typical transaction fee rate is 0.0001 BTC per 1000 bytes. Although the exact message size depends on the implementation, we believe 5000 bytes is a reasonable estimate, for a total cost per message of 0.0005 BTC per message. Overall, the financial cost to a user is composed of the mixing fee and the transaction fees. For a chunk size of 0.1 BTC, a fee rate of 0.01, and a transaction fee of 0.0005 BTC per message, the total cost to the user is around 0.002 BTC or 2%, which we believe is a reasonable price to pay for the anonymity benefits.

Table 3. Comparison of the consume with other schemes.

	Traffic	Fee	Rounds	Time
Lockcoin	**6 transactions**	**vρ**	**one round**	**10$\times\omega\times$6 mins**
Mixcoin [7]	2 transactions	v$\rho\times$round	Multiple rounds	10$\times\omega\times$2$\times$round mins
Blindcoin [8]	4 transactions	vρ	one round	10$\times\omega\times$4 mins
Coinjoin [9]	1 transaction	-	one round	Negotiation+1h
Coinswap [10]	4 transactions	-	one round	10$\times\omega\times$4 mins

4.3 Performance

The experiment was performed at a Dell desktop machine having an Intel Core i5-6500 CPU at 3.20GHz and 4.00G of RAM and running 64-bit windows 10. The program is implemented by go language and runs on the bitcoin test network. We use a parallel strategy to simulate multiple users and test the runtime to mix bitcoins in different numbers of users. Without considering the bitcoin block confirmation time, the time each user spends participating in the mix server is around 0.1-0.2s. Experimental results are shown in Fig.3 and Fig.4. The relevant source code was uploaded to github[1].

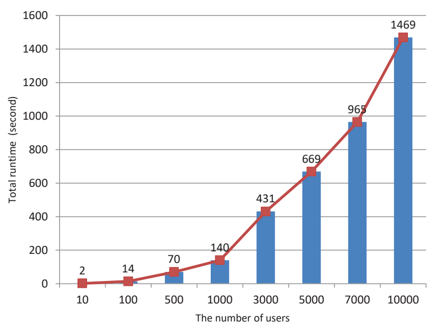

Fig. 3. The relationship between the running time of the program and the number of users.

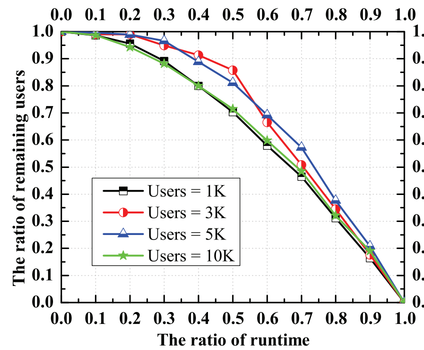

Fig. 4. The complementary cumulative distribution function of running states in different users.

5 Conclusion

We present Lockcoin, a secure and privacy-preserving mix service for bitcoin anonymity. Lockcoin uses blind signature to prevent the link between user's input address and output address. Multi-signature can be used as a deposit mechanism, which is already a funcion that bitcoin system has achieved. Lockcoin has all the proporties that the ideal scheme should have, such as anonymity, scalability, bitcoin compatibillity, theft impossibility and accountability. Meanwhile, we implemented the prototype of Lockcoin, experiments prove that it can support large-scale users and also it is very efficient.

6 Acknowledgement

This work is supported by the National Science foundation of China (No.61472074, U1708262) and the Fundamental Research Funds for the Central Universities (No.N172304023).

References

1. Nakamoto S.: bitcoin: A peer-to-peer electronic cash system[J]. 2008.

[1] https://github.com/B-doublemint/LockCoin

2. Coinmarketcap.com. https://coinmarketcap.com/currencies/bitcoin/
3. Ron D, Shamir A.: Quantitative analysis of the full bitcoin transaction graph[C]//International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2013: 6-24.
4. Reid F, Harrigan M.: An analysis of anonymity in the bitcoin system[M]//Security and privacy in social network. Springer New York, 2013: 197-223.
5. Koshy P, Koshy D, McDaniel P.: An analysis of anonymity in bitcoin using p2p network traffic[C]//International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2014: 469-485.
6. Miller A, Litton J, Pachulski A, et al.: Discovering bitcoins public topology and influential nodes[J]. et al., 2015.
7. Bonneau J, Narayanan A, Miller A, et al.: Mixcoin: Anonymity for bitcoin with accountable mixes[C]//International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2014: 486-504. 1
8. Valenta L, Rowan B.: Blindcoin: Blinded, accountable mixes for bitcoin[C]//International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2015: 112-126. 1
9. Maxwell, Gregory.: CoinJoin: bitcoin privacy for the real world, 2013.
10. Maxwell, Gregory.: CoinSwap: transaction graph disjoint trustless trading. CoinSwap: Transactiongraphdisjointtrustlesstrading, 2013.
11. van Saberhagen N.: Cryptonote v 2. 0[J]. https://cryptonote. org/whitepaper. pdf, 2013
12. Miers I, Garman C, Green M, et al.: Zerocoin: Anonymous distributed e-cash from bitcoin[C]//Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013: 397-411.
13. Sasson E B, Chiesa A, Garman C, et al.: Zerocash: Decentralized anonymous payments from bitcoin[C]//Security and Privacy (SP), 2014 IEEE Symposium on. IEEE, 2014: 459-474.
14. Ben-Sasson E, Chiesa A, Tromer E, et al.: Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture[C]//USENIX Security. 2014.
15. David Chaum.: Blind signatures for untraceable payments. In Advances in cryptology,pages 199203. Springer, 1983.
16. S. Goldwasser and M. Bellare.: This is a set of lecture notes for a summer course on cryptography, taught by the authors at the Massachusetts Institute of Technology (MIT), 1996–2008.
17. QingChun ShenTu and JianPing Yu.: A Blind-Mixing Scheme for bitcoin based on an Elliptic Curve Cryptography Blind Digital Signature Algorithm, 2015
18. Bellare M, Neven G.: Identity-Based multi-signatures from RSA[C]// Cryptographers' Track at the Rsa Conference on Topics in Cryptology. Springer-Verlag, 2007:145-162.
19. Thomas Kerin.: 'he Year of Multisig: How is it Doing So Far?'. https://www.coindesk.com/year-multisig-so-far/, 2015.
20. $OP\_RETURNwiki$. https://en.bitcoin.it/wiki/OP_RETURN
21. Roger Dingledine, Nick Mathewson, and Paul Syverson.:Tor: The secondgeneration onion router. Technical report, DTIC Document, 2004.
22. Conti M, Kumar S, Lal C, et al.: A survey on security and privacy issues of bitcoin[J]. IEEE Communications Surveys & Tutorials, 2018.