
# Blockchain-Based Group Key Agreement

Caifei Shen(✉)

Beijing Institute of Technology, Beijing, China
caifeishen@bit.edu.cn

**Abstract.** Blockchain can provide trusted ledgers on distributed architecture without the help of any central authority. Since all the transactions are saved in the ledgers, they can be obtained in public. By using the transactions, this paper first proposes a blockchain-based Diffie-Hellman key agreement (BDKA) protocol. Then, a blockchain-based group key agreement (BGKA) protocol is further proposed. In addition, both BDKA and BGKA protocols are implemented in the Bitcoin system. The performance of protocol execution and transaction fee are analyzed in the experiments.

**Keywords:** Blockchain · Key agreement · Group key agreement

## 1 Introduction

Blockchain technology has been widely used in many industrial applications. For instances, it enables the Internet of Vehicles to build secure communication groups in vehicular ad-hot networks. It is applied to protect video copyright in the process of video-streaming distribution. It is also praised for reliably balancing a user and her neighbors' electricity supply and demand in smart grid. Moreover, blockchain technology is innately welcome in the application of group communication since the ledger is public for all users in public blockchain and all authorized members in consortium blockchain. However, the most common blockchain applications mainly focus on maintaining data integrity and non-repudiation by using the characteristics of the publicity and immutability of the ledger. This paper tries to extend blockchain applications to support key exchange and group key exchange. The motivation is because the ledger in blockchain can be naturally regarded as a broadcasting communication channel [10]. Suppose a secret key can be exchanged by the transactions saved in the ledger, it can be used as the pre-shared key to build a covert communication channel. Messages can be secretly delivered among users when combining the pre-shared key with some blockchain-based covert encoding and decoding schemes [10]. It can be used as a forensic evidence in the consortium or even public as the ledger is nearly impossible to be tampered in reality. It can also be used to construct a secure blockchain-based communication channel for group members [10].

This paper is supported by National Natural Science Foundation of China No. 62172040, No. U1836212, No. 61872041.


S. Deng et al. (Eds.): MobiCASE 2021, LNICST 434, pp. 99–111, 2022.
https://doi.org/10.1007/978-3-030-99203-3_7

In recent years, several works designed group key agreement protocols on blockchains. For instances, McCorry et al. [11] applied Bitcoin to achieve authenticated Diffie-Hellman-based key agreement (DHKA). Specifically, the Bitcoin Core client was modified to adding the DHKA procedure as remote commands. The procedure commands are stored and executed off-chain. In [3], Thanh Bui et al. presented a two-party key exchange protocol that uses the global consistency property of blockchains. [7] used Proof of Work (POW)-based public ledger and Delegated Proof of Stake (DPoS)-based public ledger to create a key tree and store the list of authorized group members. The POW-based mechanism is different from that of Bitcoin, because the block generation is not competitive. Although blockchain-based group key exchange have been attempted, to the best of our knowledge, the existing works usually require to modify the executable software or the consensus mechanism.

The main contribution of this paper is summarized as below.

(1) A blockchain-based key agreement protocol and a blockchain-based group key agreement protocol are proposed without requiring to modify the existing blockchain system, such as the executable software or the consensus mechanism.
(2) A dynamically joining or leaving scheme is designed for the blockchain-based group key agreement protocol. New group members are unable to obtain the historical group key, while left group members cannot compute the future's group key.
(3) An address update scheme with key update is provided for users when users would like to preserve her identity private. This scheme does not require to modify the existing blockchain system as well.

The rest of the paper is organized as follows. Section 2 reviews the related works. In Sect. 3, we introduce the preliminaries. Section 4 proposes group key agreement protocols. Section 5 analyses the security and the performance of the proposed protocol in the Bitcoin system. The conclusion is drawn in Sect. 6.

## 2 Related Works

Several studies have been focusing on establishing secure key agreement protocols based on blockchain systems. Most of the protocols were essentially designed based on the integrity, non-repudiation, and global consistency of the ledger in blockchain systems. For instances, McCorry et al. proposed two blockchain-based key agreement protocols under the hard mathematical assumptions of Elliptic Curve Diffie-Hellman (ECDH) [12] and the YAK zero knowledge proof-based key agreement protocol [8]. The protocols are implemented for Bitcoin users in post-transaction scenario. Specifically, the random nonce $k_s$ used in the transaction signatures was taken to establish secret keys without requiring any Trusted Third Party (TTP). However, both protocols required to modify the Bitcoin client. Bui et al. [3] presented a family of key exchange protocols. These protocols used the global consistency of the public ledgers; therefore, every user can obtain the same messages to avoid the man-in-the-middle attack. However, these protocols need an out-of-band channel for sharing parameters, when public identities are not available.

For specific purpose, Zhang et al. [16] discussed a lightweight group key agreement protocol for the resource-limited internet of vehicles. In [2], a blockchain-based key agreement protocol was designed for smart grid to manage the group key in the Neighborhood Area Networks (NAN). However, they have not sufficient consideration about dynamically joining and leaving for the members.

## 3 Preliminaries

In this section, we will briefly recall the Bitcoin system, and the existing group key exchange protocols. They are the preliminaries when designing our protocols presented in the next sections.

### 3.1 Data Format of Bitcoin Transactions

Bitcoin was proposed by Satoshi Nakamoto in 2008 [13], which is a digital cryptocurrency system that does not contain central financial institutions. It allows everyone to access a public ledger and to agree upon recording append-only changes on the ledger.

Bitcoin transactions are basic information units for users to communicate, forming by accounts, amount, signatures and other information. There are two types of Bitcoin transaction structures currently-the Bitcoin White Paper defined transactions and new Segregated Witness transactions-shown in Table 1.

**Table 1.** Bitcoin transactions' structure [1]

| | | |
|---|---|---|
| version | 4 bytes | Transaction version number |
| *Marker | 1 byte | 0x00 |
| *Flag | 1 byte | 0x01 |
| tx_in count | varies | Number of inputs in this transaction |
| tx_in | varies | Transaction detail inputs |
| tx_out count | varies | Numbers of outputs in this transaction |
| tx_out | varies | Transaction outputs |
| *Witness | varies | SigWit transaction witness data |
| lock_time | 4 bytes | Unix epoch time or block number |

Miners will collect the most recent set of transactions from the network to form a 'block'. This block is appended to the longest Bitcoin chain of blocks, including the hash pointer pointing to the previous block. The chain-like structure makes it generally believed that Bitcoin transaction information cannot be tampered with after 6 blocks confirming.

Early back in 2013, some Bitcoin ecosystem participants were trying to include bits of information into transactions so that they could take advantages of the irreversibility of the blockchain. Even Satoshi Nakamoto writed complaint about current bank system

in Genesis Block. Doing this on the Bitcoin, someone encoded the transaction's scriptSig value to store extra information avoiding to alter the final result of running that script. Other ways were also presented, like using BTC value, outputs account, output addresses.

OP_RETURN is the Bitcoin script opcode, which also is the most direct way of placing extra data in transactions. To met this demand, the Bitcoin core-developers made Bitcoin v0.9.0 support that 40 bytes can be recorded by OP_RETURN transacitons. Then for Bitcoin v0.11.x, this ability upgraded to upto 80 bytes. So we are allowed to send less-than-80 bytes by a standard Bitcoin tansaction now, which is also one communication message in our scheme. Figure 1 highlights how transactions carry our communication information.

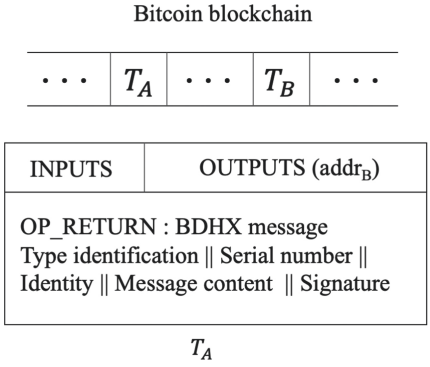


**Fig. 1.** Bitcoin blockchain and transactions' structure.

Bitcoin provides support to build standard OP_RETURN transaciton by developers making outputs to start with OP_RETURN opcode and to follow all the other consensus rules. Extra data can be appended to it, and these transacitons will be relayed and mined by Bitcoin nodes as usual. OP_RETURN transactions are not distinguished from ordinary transfer transacitons, and extra data written in will not be discarded or parsed by Bitcoin.

### 3.2 Group Key Agreement

The goal of group key agreement is to set up and maintain a shared secret key among the group members. It serves as a fundamental service for other security services. There are three kinds of methods to establish group key for user groups. One Method relies on a single entity (usually called key server) to generate and distribute keys to the group members. Another method dynamically selects group member to generate and distribute keys to other members in the group. The other method is called contributory group key agreement. This method requires each group member to contribute an equal share to the common group key. It can avoid the problems with the centralized trust and the single point of failure.

## 4 Protocols

In this section, we first propose a Blockchain-based Diffie-Hellman Key agreement (BDHA) protocol that can solve the challenge of key agreement and key update. We then extend the BDHA protocol from two-party to multiple-party, and present a blockchain-based group key agreement protocol. All of the notations and symbols are summarized in Table 2.

**Table 2.** Notations and symbols

| | |
|---|---|
| $S$ | The secret session key |
| $P$ | A prime number |
| A, B | Members in a key agreement protocol |
| $N$ | The total number of members |
| $addr_A, addr_B$ | The blockchain addresses of members |
| $addk_A, addk_B$ | The private keys of members |
| $K_{AB}$ | The pre-shared key between members A and B |
| $M_i$ | The i-th group member where $i \in \{1, \ldots, N\}$ |
| $h$ | the height of a key tree |
| $l$ | the level of a node located in a key tree |
| $\langle l, v \rangle$ | The $v$-th node at level $l$ in a key tree |
| $T_i$ | The $M_i$'s view of a key tree |
| $\hat{T}_i$ | $T_i$'s view of modified tree for member joining and leaving |
| $T_{\langle l,v \rangle}$ | A subtree rooted at node $\langle l, v \rangle$ |
| $BK_i^*$ | set of $M_i$'s blinded keys |

### 4.1 A Blockchain-Based Two-Party Key Agreement Protocol

The Diffie-Hellman(DH) key agreement (DHKA) protocol [6] is the basis of most two-party communication protocols and multi-party communication protocols [14]. It allows two users to establish a secret key between two participants for subsequent encryption without revealing the key on the communication channel. This blockchain-based key exchange protocol starts from that protocol correspondingly.

The following is a brief description of how two members negotiate a secret session key via transactions. In order to exchange a secret session key, A and B have to agree on a prime number $P$ and a generator $G$. After agreeing on $P$ and $G$, A and B randomly pick up temporarily private keys $K_A = a$, $K_B = b$, respectively. The temporarily public keys of A and B are $aG$ and $bG$, correspondingly. Coming to the blockchain part, both A and B send a transaction after embedding each temporarily public key. Then A and B records users' names with their blockchain addresses in use. After a transaction sends from A/B's blockchain address to B/A's blockchain address. B/A will download it from the ledger, checks the validity and extracts the temporarily public key. Eventually, A and B can calculate the secret session key independently.

$$S = (G^b)^a \ mod \ P = (G^a)^b \ mod \ P = G^{ab} \ mod \ P \tag{1}$$

Here, since the temporarily public keys are permanent recorded on the blockchain, any user can obtain them. But the temporarily private key generation is executed locally. Thus, it will not be broadcasted with transactions.

From the viewpoint of the adversary, she is negligible to compute $S$, because she has neither learned $a$ nor $b$ from any of the transactions. Hence, as long as the discrete logarithm problem is hard, the adversary is difficult to get the secret session key. By the blockchain, since the adversary is also difficult to generate a valid transaction for any legitimate users, the active attack is also difficult to be launched.

**The Blockchain-based Two-party Key Agreement Protocol**

step 1: A confirms parameters $g, p$, B and her Blockchain address.
step 2: A randomly generates a temporarily private key $a$ and computes the temporarily public key $aG$.
step 3: A sends a transaction to B for sending her temporarily public key $aG$ to B.

$$A \xrightarrow[A:aG]{\qquad\qquad} B$$

step 4: When B receives this transaction and calculates secret session key $S$.
step 5: B sends a transaction back to A with his temporarily public key $bG$.

$$A \xleftarrow[B:bG]{\qquad\qquad} B$$

step 6: When receiving a transaction from B, A computes the secret key $S$.

We implement the protocol in the Bitcoin system. Specifically, the Bitcoin system provides developers with the technical support for coding the Output script of each transaction with the OP_RETURN opcode. The OP_RETURN opcode is followed by the position where the data can be written without having any impact for the validity of the transaction. The written data is also known as NULL DATA. Bitcoin transactions with NULL DATA can be relayed and mined by Bitcoin miners, which are difficult to be distinguished from ordinary transfer transactions. In addition, Bitcoin network will not parse the protocol data that we write in the fragment of OP_RETURN. Therefore, we can embed the parameters and temporarily public keys in this fragment. After encoding the protocol messages into the fragment of OP_RETURN. It can be broadcasted in the ledger by the Bitcoin network, as shown in Fig. 2.

### 4.2 A Blockchain-Based Group Key Agreement Protocol

For the case of group communication, setting up and maintaining a secret session key among group members can be expanded from the two-party protocol. The Tree-based Group DH (TGDH) protocol is the first group key agreement protocol based on a tree and the DH protocol. Applying the TGDH protocol to the blockchain, each user generates her own key pair: temporarily private key and temporarily public key. The temporarily private key is known to herself, and the temporarily public key is public to other

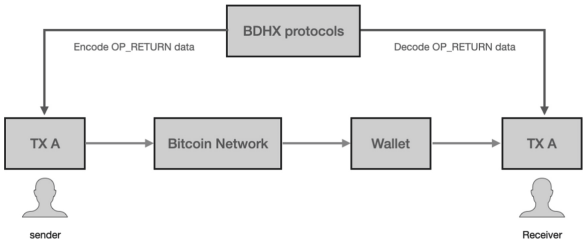


**Fig. 2.** The process of the blockchain-based two-party key agreement protocols

group members. Users can use their own blockchain addresses to send transactions carrying protocol messages to other users, while other users can extract the messages sent to themselves from the received transactions.

For each group member, they record a mapping table for member names with their blockchain addresses, so that they can identify members from their transactions. Here is a challenge that has been solved. That is, a leaved member can obtain the transaction addresses of all members of the group, and can monitor the remaining members in the group in the later period. Through information such as the number of transactions and frequency, the leaving members can learn about the changes and activities of the group members. Information such as frequency may even be used for data analysis to cluster addresses and expose the privacy of group member. Therefore, in this protocol, when a user joins or leaves, the group key is forced to be updated and the addresses have to be updated at the same time.

The protocol description is listed as below:

- The key tree structure is shown in the Fig. 3. Here, the root of the tree is located at level $l_0$, and the lowest leaf node is located at level $h$. Since each node in a binary tree is either a leaf node or the parent node of one or two nodes. Nodes can be represented by $\langle l, v\rangle$, and there are at most $2l$ nodes in layer $l$. Each node $\langle l, v\rangle$ is associated with the temporarily private key $K\langle l, v\rangle$ and the temporarily public key $BK\langle l, v\rangle$, $BK\langle l, v\rangle = f(K\langle l, v\rangle)$. The $f(\cdot)$ function can be determined due to the concrete key exchange protocol.
- Assuming that a leaf node $\langle l, v\rangle$ represents the member $M_i$, node $\langle l, v\rangle$ will have the private key $K\langle l, v\rangle$ of member $M_i$, and it can get the every key along the path from $\langle l, v\rangle$ to $\langle 0, 0\rangle$, this set is called $KEY_i^*$.
- As shown in the Fig. 3, if the key tree that can be viewed from the perspective of $M_2$ is called $T_2$, then $M_2$ can get all the keys of $KEY_2^*$ $\{K\langle 3, 1\rangle, K\langle 2, 0\rangle, K\langle 0, 0\rangle, K\langle 0, 0\rangle\}$, and all temporarily public keys $\{BK\langle 0, 0\rangle, BK\langle 1, 0\rangle, BK\langle 1, 1\rangle, \ldots, BK\langle 3, 7\rangle\}$ of $T_2$.
- $K\langle 0, 0\rangle$ is the secret group key that is negotiated by the group members, and the group session key can be derived from $K\langle 0, 0\rangle$.
- If a member joins or leaves, all remaining members independently update the key tree structure. Since all the changes are recorded on the ledger of the blockchain, all members who correctly execute the protocol can recompute identical key trees.

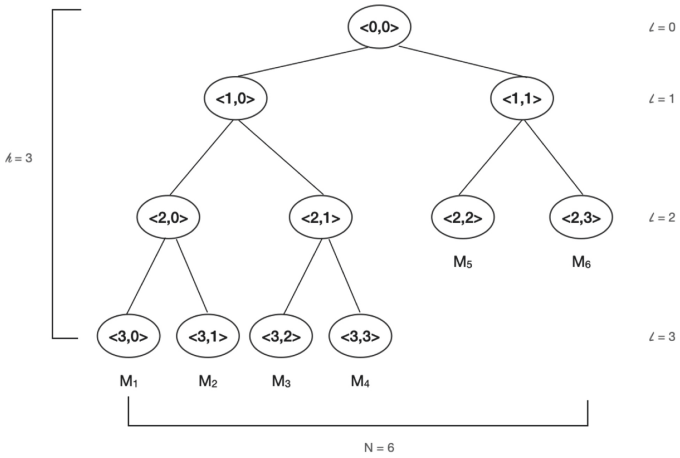


**Fig. 3.** The structure of the key tree

Any group member can be the initiator, which involves calculating intermediate keys and broadcasting to the group by the ledger. Each broadcast message contains the sender's viewpoint of the key tree that contains every key known for the sender. (The intermediate key will never be broadcasted.)

**Member Joining.** Assuming that the group has $N$ members, $\{M_1, M_2, \ldots, M_n\}$, a new member $M_{n+1}$ initiates the joining protocol by sending a transaction to group members' addresses that he knew. The transaction contains the joining request and his own temporarily public key. Each member will receive this transaction and determine the insertion point in the tree. The insertion point is the rightmost node at the lowest level to avoid increasing the height of the key tree. If the key tree is fully balanced, new members will join as the root node. The sponsor is the rightmost leaf in the subtree rooted at the inserted node, and it is the member representative of the group that negotiates the key with the joining or leaving members. Each member creates a new intermediate node and a new member node, and promotes the new intermediate node to the parent node of the insertion node and the new member node. After updating the key tree, all members except the sponsor wait. The sponsor updates its key and calculates a new group key because it knows all the temporarily public keys. Finally, the sponsor broadcasts a new tree containing all the temporarily public keys. All the members will update their tree accordingly and calculate a new group key. This group key tree will be recorded on the blockchain.

Figure 4 takes an example of the member $M_4$ joining a group where the sponsor $M_3$ performs the following actions:

1. Rename the node $\langle 1, 1\rangle$ to $\langle 2, 2\rangle$
2. Generate a new intermediate node $\langle 1, 1\rangle$ and a new member node $\langle 2, 3\rangle$
3. Promote $\langle 1, 1\rangle$ as the parent node of $\langle 2, 2\rangle$ and $\langle 2, 3\rangle$

Since all members know $BK_{\langle 2,3\rangle}$ and $BK_{\langle 1,0\rangle}$, $M_3$ performs the step 1 and 2, but cannot compute the group key in the first round. Upon receiving the broadcasted temporarily public keys, every member can compute the new group key.

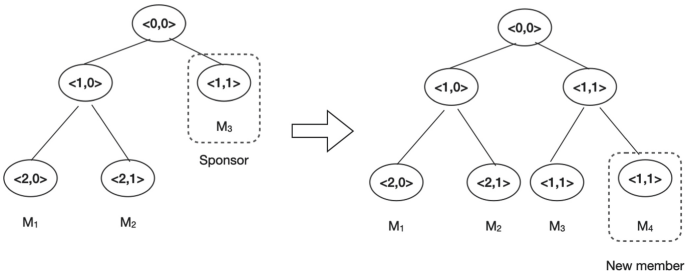


**Fig. 4.** A new member joins and key tree updates

To join a new member $M_{n+1}$, the transaction outputs do not have to include every group members, because the transactions on the blockchain are public for every member. $M_{n+1}$ can send to certain members' addresses that he knows. They will allow group members to quickly filter out the related information from a large number of blockchain transactions. Meanwhile, he can send it to a non-member address, and group members will find a request transaction that conforms to the format. After the group key is updated, all members update their transaction addresses and broadcast to other members with the new group key.

**Blockchain-based Group Key Agreement: Member Joining**

step 1: The new member sends request transactions for joining

$$M_n + 1 \xrightarrow{join,\ M_n+1,\ BK_{M_{n+1}}=G^{K_{n+1}}modP} C = \{M_1, \ldots, M_n\}$$

step 2: Every member

- update key tree by adding new member nodes and new intermediate nodes.
- removes all keys and temporarily public keys from the leaf node related to the sponsor to the root node.

The sponsor $M_s$ additionally

- generates the new share and computes all $[key, temporarilypublickey]$ pairs on the key-path.
- sends broadcast-transactions to other members which carry updated tree including only the temporarily public key.

$$CUM_n + 1 \xleftarrow{BKtree} M_s$$

step 3: Every member

- computes the group key using BKtree.
- updates and broadcasts new Bitcoin addresses.

**Member Leaving.** When a member leaves the group, the initiator is the rightmost leaf node of the subtree with the same root as the leaving member. Other members delete the leaving members from the key tree to update the key tree. The original sibling node of the leaving member replaces the parent node. The initiator updates its own key and calculates other keys on the path, and finally broadcasts a new key tree of the updated temporarily public key to other members so that all members can calculate the new group key.

Looking at the setting in Fig. 5, if a member leaves the group, every remaining member deletes $\langle 1,1\rangle$ and $\langle 2,2\rangle$. After updating the tree, the sponsor $M_5$ picks a new share $K_{\langle 2,3\rangle}$, recomputes $K_{\langle 1,1\rangle}$, $K_{\langle 0,0\rangle}$, $BK_{\langle 2,3\rangle}$ and $BK_{\langle 1,1\rangle}$. Then broadcast the updated tree. Upon receiving the broadcast message, all members compute the group key. Note that $M_3$ cannot compute the group key, though it knows all the temporarily public key, because its share is no longer part of the group key.

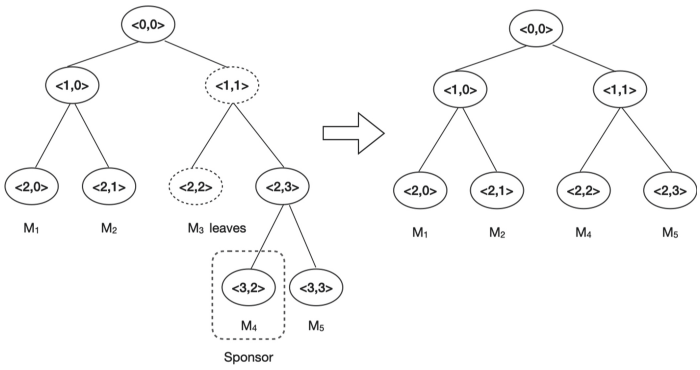


**Fig. 5.** A group member leaves

**Blockchain-based Group Key Agreement: Member Leaving**

step 1: Every member
- updates the key tree by removing the leaving member node and relevant parent node.
- removes all keys and temporarily public key from the leaf node related to the sponsor to the root node.

The sponsor $M_s$ additionally
- generates new share and computes all $[key, temporarilypublickey]$ pairs on the key-path.
- sends transactions to other members which take the updated tree including the temporarily public keys.

$$M_s \xrightarrow{BKtree} \{M_1, \ldots, M_n\} - M_d$$

step 2: Every member
- computes the group key using the BKtree.
- updates and broadcasts new Blockchain addresses.

## 5 Security Analysis

By the BGKA protocol, the communication participants can obtain the group keys to encrypt the messages and ensure the confidentiality of the group communication. Thus how to generate and update the group keys is an important factor.

### 5.1 The Informal Security Analysis for the Session Key

Because of the discrete logarithm problem and the Computational Diffie-Hellman Problem, multi-party DH protocols' security has been proven [4,5,15]. If all users in the group are honest, they will get the same group keys. Each user has the ability to verify the reliability of other group members' messages by the transactions. Meanwhile, the private keys are calculated locally and not exposed to the blockchain network.

### 5.2 The Informal Security Analysis for the Blockchain Addresses

We analyze the security of the blockchain address by the example of the Bitcoin system. In the Bitcoin system, addresses act as the identifiers without exposing the real identity. Anyone can get new Bitcoin addresses free of charge, which consist of 27–34 alphanumeric characters. According to the rules of calculating Bitcoin's public keys and addresses, there are at most $2^{160}$ different addresses for one type with 256-bits private keys and RIPEMD-160 hash. There is little chance to take collision attack successfully, which means an adversary is negligible to send Bitcoin transactions through

other's addresses if he knows nothing about the private keys. Moreover, the blockchain addresses update mechanism is designed to limit the number of blockchain transactions sent by the same addresses. The shorter time a Bitcoin address is used, the less likelihood of group members are exposed.

### 5.3 The Informal Security Analysis for Forward Secrecy and Backward Secrecy

Forward Secrecy: Any group member $M_i$ has a private key $K_i$, which is used to participate in the group key agreement protocols. When the group member $M_i$ leaves the current group, he is unable to receive any further group communication information. At the same time, after members leaving, the remaining members update their private keys and blockchain addresses. The temporarily public keys of the current group is not the same as those known by the leaving member. So leaved members cannot monitor the group's subsequent communication activity.

Backward Secrecy: New group members $M_{n+1}$ is negligible to obtain the previous group keys to decrypt messages sent before he joins the group. In the protocol, when a new member $M_{n+1}$ joins, some members update their private keys. So $M_{n+1}$ cannot get previous group keys, and can only compute others' new Bitcoin addresses.

### 5.4 Reply Attacks

The data for the key exchange protocols and group communication in the protocol contain current timestamps, meanwhile digital signatures and blockchain system guarantee that these data and timestamp cannot be modified. The attacker has to succeed under the threshold of the time period. This is difficult because if the threshold is appropriate in the context.

### 5.5 Man-in-the-Middle Attacks

The security of the blockchain also directly affects the security of the man-in-the-middle attack that ultimately exchanges the key. The Diffie-Hellman key exchange is secure if there is no adversary who can actively and covertly manipulate communication information exchanged among users. The blockchain networks are peer-to-peer networks that allow users to communicate through different nodes and access the transactions. As the Bitcoin system as an example, unless the Bitcoin system with longest-chain consensus rules can be attacked by 51%-attack [9], it is difficult to change the blockchain information by other methods. Therefore, it is hard for an adversary to have the ability to change the transaction data in the ledger of the Bitcoin system. As long as all members can verify the transaction signatures, an adversary can hardly threat the security of the blockchain-based key agreement protocol unless the attacker has the ability to launch a 51% attack.

## 6 Conclusion and Future Work

The proposed protocols in this paper include Bitcoin address update, aiming to reduce the impact of group clustering attacks for group users' privacy. And this scheme was

implemented on the real Bitcoin network without modifying the core code. Moreover, it can be used on the Ethereum, or other blockchain systems, for longer embedding space to carry information. As for the Bitcoin, the information can be transmitted in sections or slices. The scheme is compatible with different key agreement algorithms which is not exposed to the blockchain.

## References


1. Bitcoin developer. https://developer.bitcoin.org/reference/transactions.html
2. Baza, M., Fouda, M.M., Nabil, M., Eldien, A.T., Mansour, H., Mahmoud, M.: Blockchain-based distributed key management approach tailored for smart grid. In: Fadlullah, Z.M., Khan Pathan, A.-S. (eds.) Combating Security Challenges in the Age of Big Data. ASTSA, pp. 237–263. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-35642-2_11
3. Bui, T., Aura, T.: Key exchange with the help of a public ledger. In: Stajano, F., Anderson, J., Christianson, B., Matyáš, V. (eds.) Security Protocols 2017. LNCS, vol. 10476, pp. 123–136. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71075-4_15
4. Cheon, J.H.: Security analysis of the strong Diffie-Hellman problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 1–11. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_1
5. Boer, B.: Diffie-Hellman is as strong as discrete log for certain primes. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 530–539. Springer, New York (1990). https://doi.org/10.1007/0-387-34799-2_38
6. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
7. Han, S., Choi, R., Kim, K.: Adding authenticity into tree-based group key agreement by public ledger. In: 2019 Symposium on Cryptography and Information Security (SCIS 2019). IEICE Technical Committee on Information Security (2019)
8. Hao, F.: On robust key agreement based on public key authentication. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 383–390. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14577-3_33
9. Kroll, J.A., Davey, I.C., Felten, E.W.: The economics of bitcoin mining, or bitcoin in the presence of adversaries. In: Proceedings of WEIS, vol. 2013, p. 11 (2013)
10. Lin, I.C., Liao, T.C.: A survey of blockchain security issues and challenges. IJ Netw. Secur. **19**(5), 653–659 (2017)
11. McCorry, P., Shahandashti, S.F., Clarke, D., Hao, F.: Authenticated key exchange over bitcoin. In: Chen, L., Matsuo, S. (eds.) SSR 2015. LNCS, vol. 9497, pp. 3–20. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27152-1_1
12. Miller, V.S.: Advances in cryptology-crypto '85 proceedings. Use of elliptic curves in cryptography, pp. 417–426 (1986)
13. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot (2008)
14. Steiner, M., Tsudik, G., Waidner, M.: Diffie-Hellman key distribution extended to group communication. In: Proceedings of the 3rd ACM Conference on Computer and Communications Security, pp. 31–37 (1996)
15. Steiner, M., Tsudik, G., Waidner, M.: Key agreement in dynamic peer groups. IEEE Trans. Parallel Distrib. Syst. **11**(8), 769–780 (2000)
16. Zhang, Q., et al.: Blockchain-based asymmetric group key agreement protocol for internet of vehicles. Comput. Electr. Eng. **86**, 106713 (2020)