




A Cross-Chain Scheme Combining Notary with Linkable Ring Signature and Hash Time Lock Contract

Yukun Zheng^{1,a}, Yahui Guo^{1,b}, Yinyan Dou^{1,c}, Yuanzhe Liu^{1,d}, and Zhiming Cai^{2,e}

¹Faculty of Data Science, City University of Macau, Macau, China

²Faculty of Digital Science and Technology, Macau Millennium College, Macau, China

^aD22091101576@cityu.edu.mo, ^bD22091100833@cityu.edu.mo,

^cD22091100147@cityu.edu.mo, ^dD22091100672@cityu.edu.mo,

^ezmcai@mmc.edu.mo

Abstract. Cross-chain technology is essential for achieving blockchain scalability and interoperability. This paper presents a cross-chain scheme combining notary with linkable ring signatures and the hash time lock contract. The scheme uses linkable ring signatures to act in the notary election process, which prevents duplicate voting and ensures the anonymity of voting nodes to prevent user information leakage. In this paper, a notary group is employed to oversee and engage in cross-chain transactions, which reduces the centralization of notaries' power. In addition, this paper examines the election procedure of the notary and the cross-chain transaction process of this scheme and illustrates the security and effectiveness of this scheme.

Keywords: Linkable ring signature , Notary , HTLC , Blockchain

1 Introduction

As blockchain technology advances, more and more scenarios need to use the interoperability between different blockchains [1]. Cross-chain technology has developed to address the issue of transferring data and assets between several blockchains, thereby broadening the range and complexity of blockchain applications. The current mainstream cross-chain technology includes hash time lock contract (HTLC), notary, side chain, relay chain, etc., and they each have their own advantages and disadvantages as well as suitable use scenarios [2]. In the future, cross-chain technology still has ample potential for growth, and the scalability as well as efficiency of blockchain applications can be improved through continuous improvement.

However, the current blockchain technology still faces many difficulties in cross-chain. The first is the technical difficulty of cross-chain bridging, which needs to solve the problems of cross-chain asset security, efficiency of asset transfer, and cross-chain cost [3]. Secondly, due to the differences between different blockchain systems and the non-uniform consensus mechanism, it is necessary to design a common set of cross-chain bridging standards to realize the interconnection and interoperability between

different blockchain systems. Finally, due to the limitations of cross-chain application scenarios, more cross-chain applications need to be developed to expand the application scope as well as the depth of blockchain cross-chain technology [4].

As a mainstream cross-chain technology, the notary scheme is widely used due to the advantages of simple implementation and no additional modification of the underlying blockchain of both parties when it is used as a means of cross-chaining between different blockchains. However, in most of the notary schemes, there is often the problem of evil notaries, and it is difficult to ensure anonymity when electing the notary, which can easily lead to the leakage of node information. Ring signature [5], as a highly efficient and private group signature technology, can be applied in the process of anonymous voting, but the general ring signature is not able to prevent the occurrence of repeated voting due to its own limitations [6]. Linkable ring signatures (LRS) [7] can both prevent duplicate voting and protect the anonymity of the voter due to the ability to link the signatures of voting members together.

To address the issue of potential centralization and redundant voting among nodes in the notary scheme, this paper designs a cross-chain scheme combining the notary mechanism with LRS and HTLC, and the contributions made in this paper are summarized as follows:

- This paper uses a notary group in charge of the whole cross-chain process in the notary scheme, which reduces the centralization of the notary's power and will be combined with HTLC for the purpose of guaranteeing the security and atomicity of the cross-chain transaction process.
- This paper utilizes LRS to ensure privacy and efficiency in the process of electing a notary public, to guarantee the anonymity of the nodes, and to prevent the leakage of nodes' information and duplicate voting.

2 Overall Design of the Cross-Chain Scheme

2.1 Scheme Overview

The application scenario of the proposed scheme in this paper is the transfer of tokens from the source node of blockchain A to the target node of blockchain B through a notary group system. The notary group, which is elected through the LRS technique, is responsible for supervising the transaction parties on the two blockchains and serves as a link for transactions across chains. The overall framework of the cross-chain scheme is shown in Fig. 1.

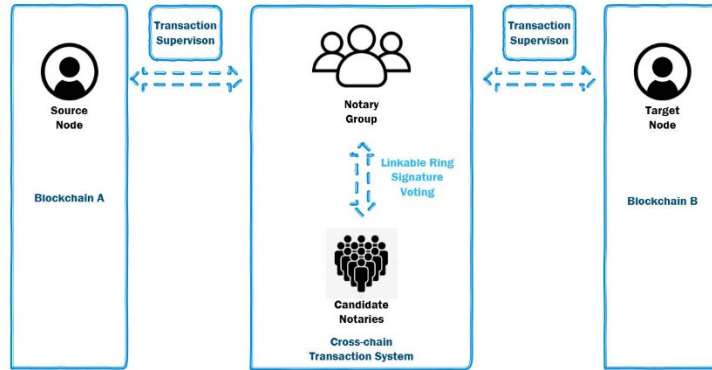


Fig. 1. The overall framework of the cross-chain scheme

2.2 Notary Election Based on Linkable Ring Signatures

This section explains the process of electing a notary group, as shown in Fig. 2. Below is a thorough description of the process.

1.Setup. The Certificate Authority (CA) determines some global parameters and generates public-private key pairs. The parameters are set as shown in the following formula:

$$H_1: \{0,1\}^* \rightarrow \mathbb{Z}_q, H_2: \{0,1\}^* \rightarrow G \quad (1)$$

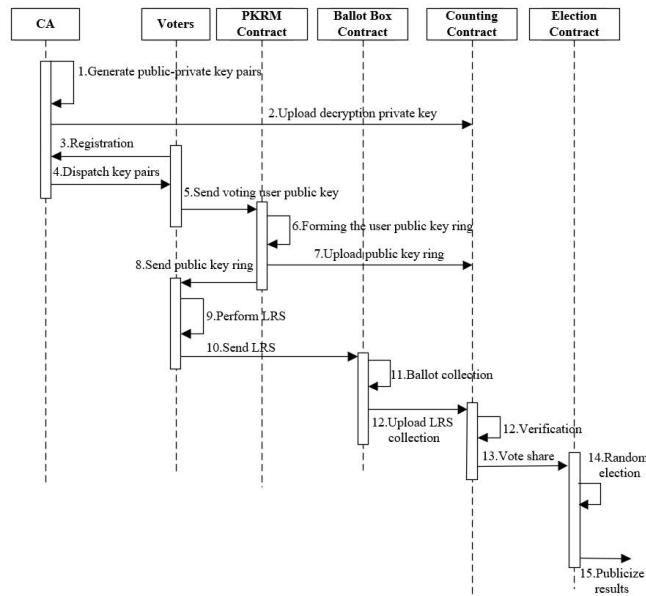


Fig. 2. Election process of the notary

where G is a base point on the elliptic curve of order q used to generate the public key; H_1 is a hash function that maps an input of arbitrary length to a finite field \mathbb{Z}_q and is used to generate random numbers during the signature process; and H_2 is a hash function that maps inputs of arbitrary length to points on an elliptic curve and is used to generate link identifiers for signatures. The formula for generating a public-private key pair is shown below:

$$sk \leftarrow \text{random} \in \mathbb{Z}_q \quad (2)$$

$$pk = sk \cdot G \quad (3)$$

where private key sk is a randomly chosen number, and the public key pk is obtained by a dot product operation on an elliptic curve. After that, the CA uploads the decrypted private key to the counting contract. This contract is used in later steps to decrypt and verify the authenticity of the vote.

2. Form a user public key ring. Nodes that want to become a notary need to initiate a registered notary application with the CA to become candidates. Subsequently, CA sends public-private key pairs to the voting users, and the voting users send their public key to the public key ring management (PKRM) contract, which will form the user public key ring as follows:

$$\text{Public Key Ring} = \{pk_1, pk_2, \dots, pk_n\} \quad (4)$$

the public key ring will be uploaded to the counting contract and sent to the voting users so that they can use it in the anonymous voting process.

3. Anonymous voting. The voting users create a LRS using their private key sk_i and add the voting information m to the signature along with their public key. The signature process produces a signature (σ, y) , where σ consists of a series of elements s_1, s_2, \dots, s_n and y is the identifier used for linking, ensuring that signatures from the same signer can be recognized. The formula for the link identifier y is shown below:

$$y = H_2(m || r \cdot G) \quad (5)$$

where r is a random number from \mathbb{Z}_q , and the formula ensures that each signature has a unique link identifier. For each member j ($j \neq i$) of the ring, randomly select $s_j \in \mathbb{Z}_q$ and $e_j \in \mathbb{Z}_q$ to compute the following equation:

$$R_j = s_j G - e_j pk_j \quad (6)$$

For the signer (indexed as i), the following calculations are required:

$$e_i = H_1(m, R_1, \dots, R_{i-1}, R_{i+1}, \dots, R_n) \quad (7)$$

$$s_i = r + sk_i \cdot e_i \text{ mod } q \quad (8)$$

This process ensures the authenticity and integrity of each vote, as only a user with the correct private key can validly sign a message. Then the users send the LRS to the ballot box contract. This LRS indicates that the voter is a member of the group but does

not reveal their identity, and even if the voting information is made public, it cannot be traced back to the identity of the specific voter.

4. Collection and validation. The ballot box contract collects the votes and uploads the LRS set to the counting contract. The counting contract is responsible for verifying the validity of the votes, by using the public key set and signature (σ, y) . For each j , the following calculations need to be performed:

$$R'_j = s_j G - e_j p k_j \quad (9)$$

For each R'_j , the next hash value e_{j+1} is computed, which is computed from the current message m and all other current R'_j by the hash function H_1 :

$$e_{j+1} = H_1(m \parallel R'_j) \quad (10)$$

Finally, the contract checks whether the last e_n (or rather the first computed e_1 , since it is a ring structure) computed through the above steps matches the value computed directly through the hash function H_1 . If it matches, the signature is valid; otherwise, it is invalid. In linkable ring signatures, linkability allows anyone to check whether two signatures were generated by the same ring member without revealing the identity of that member. This is accomplished by comparing the link identifiers y in the two signatures. If the link identifiers of the two signatures are the same, they are considered to have been generated by the same member. The mathematical representation of linkability is as follows:

$$\text{Link}((\sigma_1, y_1), (\sigma_2, y_2)) = \begin{cases} \text{linked,} & \text{if } y_1 = y_2 \\ \text{not linked,} & \text{otherwise} \end{cases} \quad (11)$$

After that, the counting contract sends the counting results to the random election contract. The random election contract randomly selects 1 node as the primary notary among the top 3 nodes with the highest number of votes to act as the link for cross-chain transactions and the other 2 nodes as the secondary notaries, and when the primary notary fails to carry out his/her duties, one of the secondary notaries will take his/her place. Finally, the results of the election are publicized.

2.3 The Process of Cross-Chain Transactions

This section outlines the process of transferring tokens from the source node in blockchain A to the account of the target node in blockchain B using this scheme, as shown in Fig. 3. Here is the specified process:

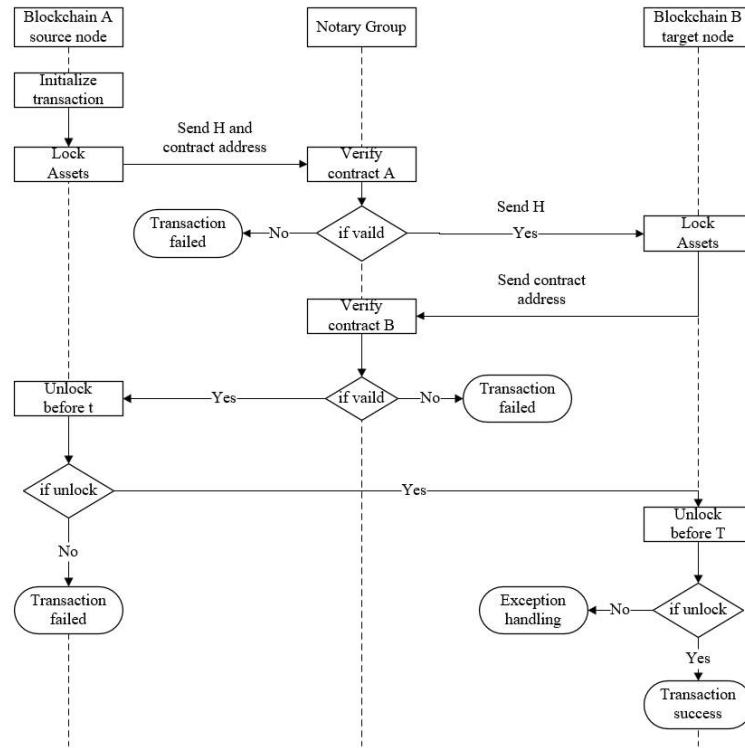


Fig. 3. Transaction process

1. The source node performs the transaction initialization algorithm, locks the asset with the quantity account1 in the smart contract on the blockchain A, and specifies that the asset will be transferred to the account of the target node in the blockchain A if the hash value H of the original image is given within time T. The source node sends the hash value H and the address of the smart contract to the notary for verification. After that, the source node sends the hash value H and the smart contract address, AddressContract1, to the notary for verification.

2. The notary verifies the validity of the smart contract address AddressSource1 in the blockchain A provided by the source node, and if it passes the verification, the hash value H will be sent to the target node, and the target node is notified of the asset lock. If the validation fails, the source node is determined to be in malicious default, the transaction fails, and the failure record of the source node is increased once.

3. The target node receives the hash value H and locks the asset in a smart contract on the blockchain B. The number of locked assets is account2, which is the same as the number of assets locked by the source node on the blockchain A, account1. If the original image with hash value H is given prior to time t (t should be significantly less than T and cannot be set too short), the asset is transmitted to the address of the source node's account (AddressSource2) in blockchain B. The target node sends the address AddressContract2 of the smart contract to the notary public for verification.

4. The notary verifies the validity of the smart contract address `AddressContract2` in the blockchain B provided by the target node, and if the verification passes, the source node will be notified to unlock the asset in time; otherwise, the transaction fails, and it is judged that the target node maliciously delinquently owes funds, and the record of the target node's transaction failure will be increased one time.

5. The source node must unlock the asset within the time set by the time lock. If the source node unlocks the asset for more than t , the transaction fails, adding one time to the source node's transaction failure record.

6. After the source node unlocks the asset, the target node will receive the original image h of the hash lock H . If the target node has unlocked the asset before time T , the transaction succeeds; otherwise, the transaction fails by adding a failure record of the target node once, and the notary intervenes to deal with the anomalies.

3 Performance Evaluation

The cross-chain scheme designed in this paper conducts simulation experiments and verifies the effectiveness and performance of the scheme in the following ways: The first set of experiments analyzes the time cost of electing the notary, and the second set of experiments compares the time required for cross-chain transactions with different numbers of nodes. The experiments in this paper use two ethereum private chains as blockchain A and blockchain B. Both blockchains are deployed on a local server with 4 processors, 4GB of RAM, and 80GB of hard disk, running Ubuntu version 20.04, with smart contracts written by Solidity. Fig. 4 illustrates the time required to elect the notary as the number of nodes increases. Fig. 5 compares the performance of the scheme when the number of nodes is 10, 20, 40, 60, 80, and 100, respectively. From Fig. 4 and Fig. 5, key findings emerge:

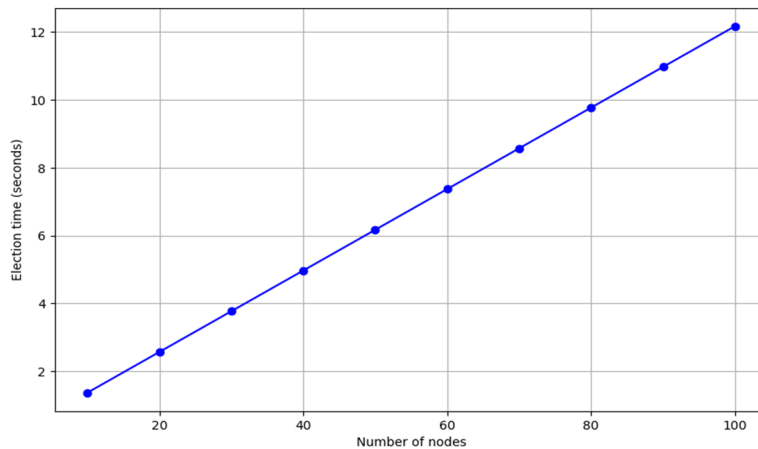


Fig. 4. Time spent on the election of the notary

- The time taken to elect a notary shows a nearly linear increase as the number of nodes increases.

- The total time to complete a transaction grows with the number of transactions, mainly linearly. The difference in the time required to complete a certain number of transactions for different numbers of nodes is mainly reflected in the different time required to elect a notary.

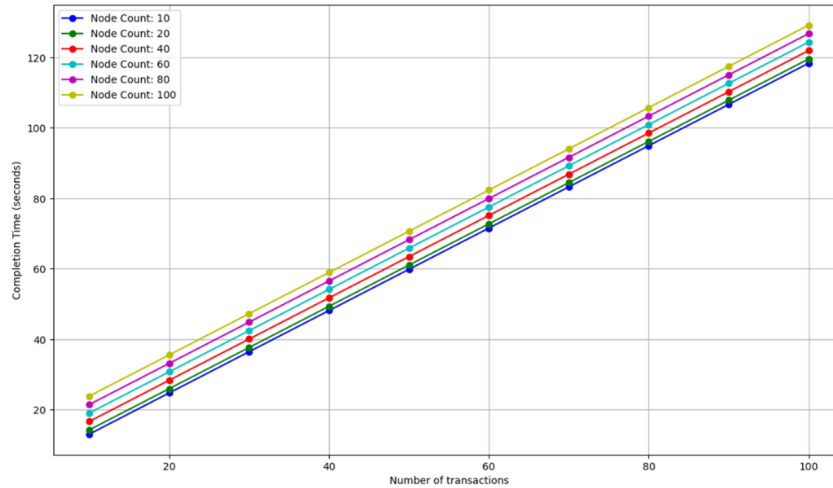


Fig. 5. The comparison of transaction time

4 Theoretical Analysis

The LRS technique allows voters to vote for a notary without revealing their identity. This is accomplished by concealing the actual identity of the signer within a group of potential signers in a ring. This ensures the anonymity of the voter since there is no way for the outside world to determine which member actually signed. And using the linkability of linkable ring signatures, two signatures made by the same signatory can be publicly linked and identified. This makes it harder for the same entity to vote for the same notary more than once, which makes the voting process more fair and safe. In addition, a notary group consisting of three notaries elected through voting can reduce the risk of centralization of power and be able to resist a single point of failure. When doing cross-chain transactions, the oversight and involvement of the notary consortium can effectively solve the timeout attack problem in the traditional scheme. The HTLC supports the atomicity of cross-chain operations, i.e., the transactions are either fully executed or not executed at all, which reduces the risk of fraud and breach of contract in cross-chain transactions.

5 Conclusion

This paper proposes a cross-chain scheme combining notary with LRS and HTLC that utilizes the anonymity and high efficiency of linkable ring signatures to improve the

security of the notary election process and prevent the duplicate voting problem that is difficult to solve with general ring signatures. At the same time, the introduction of HTLC and the use of a notary group to supervise cross-chain transactions guarantee the atomicity of the transaction and decentralize the power of the notary.

References

1. Lafourcade P, Lombard-Platet M. About blockchain interoperability[J]. *Information Processing Letters*, 2020, 161: 105976.
2. Lin S, Kong Y, Nie S, et al. Research on cross-chain technology of blockchain[C]//2021 6th International Conference on Smart Grid and Electrical Automation (ICSGEA). IEEE, 2021: 405-408.
3. Haugum T, Hoff B, Alsadi M, et al. Security and Privacy Challenges in Blockchain Interoperability-A Multivocal Literature Review[C]//Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering. 2022: 347-356.
4. Cao L, Song B. Blockchain cross-chain protocol and platform research and development[C]//2021 International Conference on Electronics, Circuits and Information Engineering (ECIE). IEEE, 2021: 264-269.
5. Rivest R L, Shamir A, Tauman Y. How to leak a secret[C]//Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7. Springer Berlin Heidelberg, 2001: 552-565.
6. Deng L, Jiang Y, Ning B. Identity-based linkable ring signature scheme[J]. *IEEE Access*, 2019, 7: 153969-153976.
7. Liu J K, Wong D S. Linkable ring signatures: Security models and new schemes[C]//Computational Science and Its Applications—ICCSA 2005: International Conference, Singapore, May 9-12, 2005, Proceedings, Part II 5. Springer Berlin Heidelberg, 2005: 614-623.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

