Cross-Chain Trusted Information Match Scheme with Privacy-Preserving and Auditability

Zheng Chen, Zejun Lu, and Jiageng Chen(✉)

Central China Normal University, Wuhan 430079, Hubei, China
jiageng.chen@ccnu.edu.cn

Abstract. Recently, numerous challenges concerning personal identity security along with other security issues regarding privacy have been introduced. While the blockchain offers a solution for recording personal information and facilitating matching needs, existing researches have indicated the absence of privacy and traceability. In this paper, we introduce an information matching scheme rooted in cross-chain technology. This approach employs a public blockchain for information matching and a consortium blockchain for transmitting private data. Cross-chain consensus protocols are leveraged to accomplish privacy protection and validation for cross-chain information transmission. The scheme serves the purpose of achieving information matching between two parties within the public blockchain framework. It enhances users' identity and information privacy, while also bolstering the auditability of cross-chain information.

Keywords: Blockchain · Cross-chain · Privacy preserving

1 Introduction

With the advent of informatization and the new era, the evolving nature of information matching stands to significantly enhance global communication quality. Ensuring the privacy of individuals' information during the information exchange process remains a pivotal concern in this information matching paradigm. The following issues deserve attention: (1) Security of Identity and Information: The inevitable disclosure of both parties' identities lies at the heart of information matching, potentially compromising the privacy of their respective information. (2) Recognition of Information from Different Parties: Insufficient data sharing among distinct parties often leads to a failure in recognizing exchanged information, hindering effective communication. (3) Auditability of Published Information: The immutability of blockchains can lead to situations where malicious information is published by senders, negatively impacting the integrity of information matching. Addressing these challenges will be crucial in establishing a robust and effective information matching model that upholds privacy and communication quality.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2024
Published by Springer Nature Switzerland AG 2024. All Rights Reserved
J. Chen and Z. Xia (Eds.), BlockTEA 2023, LNICST 577, pp. 94–114, 2024.
https://doi.org/10.1007/978-3-031-60037-1_6

Blockchain is a tamper-resistant digital ledger implemented in a distributed environment and typically without a central authority. At a fundamental level, it enables a community of users to upload transactions to a shared ledger at a relatively low cost. Under the normal operation of the blockchain network, once published, no transaction can be altered, rendering the blockchain immutable. Blockchain users can transmit information and transfer their rights to another user. The blockchain publicly documents this transfer process through transactions, allowing all participants in the blockchain network to independently verify the transaction's validity.

The rise and development of blockchain technology, represented by Bitcoin [1] and PrCash [2], introduces a new scenario unlike traditional platforms. After years of in-depth development, the blockchain has witnessed a multitude of coexisting variations, each possessing distinct characteristics suitable for various application scenarios. However, a regular decentralized blockchain does not meet certain usage scenarios in information matching where all information should be stored privately, and its efficiency is also relatively low. With the rapid growth of the blockchain industry and the emergence of numerous public chains, private chains, and consortium chains, a challenge arises: how to facilitate communication and even value exchange between different blockchains. Due to the segregation of blockchains and the significant heterogeneity between different blockchains, the transmission of information and data communication among existing blockchains faces unprecedented challenges. This situation also gives rise to numerous difficulties in facilitating communication between different blockchains.

The concept of cross-chain was initially proposed by the Tendermint team in 2014. In a narrower sense, cross-chain involves asset interoperability between relatively independent blockchain ledgers. In a broader sense, it encompasses data and asset interoperability between independent blockchain ledgers. Traditionally, interoperability in the information field refers to 'the ability to exchange and use information between different systems or modules.' [3] In terms of communication between blockchains, 'Chain Interoperability' mainly concerns the ability to transfer assets, make payments, or exchange information between two blockchains. This can be accomplished by introducing third parties without altering the native chain. A single blockchain network is relatively closed and lacks active external interaction. Cross-chain technology strives to build trust bridges between chains, dismantling the notion that a blockchain is akin to an isolated island. Cross-chain technology within the blockchain domain serves as a critical method for achieving interconnection, enhancing interoperability, and bolstering scalability between blockchains. In cross-chain industry applications, such as Celer cBridge [4], MultiChain [5], Synapse Bridge [6], and Umbria Narni [7], cross-chain technologies can support the cross-chain transmission of blockchain information across various blockchain networks. Additionally, these technologies can also find applications in certain industrial scenarios. Moreover, they possess the capability to accommodate an expanding array of blockchain types,

1.1 Our Contribution

In comparison to existing research solutions, existing information matching solutions primarily operate on a single blockchain, such as public blockchains, which may not be suitable for certain scenarios. This can result in low throughput, reduced efficiency (in terms of query and record times), and a lack of privacy protection for personal information.

In this article, we present an information storage and matching scheme that leverages cross-chain technology, incorporating public blockchains, consortium education blockchains, and cross-chain consensus protocols.

(1) In our solution, personal information storage is conducted on a consortium blockchain, while information matching is performed on a public blockchain. This approach not only avoids frequent interactions with the public blockchain during information queries but also ensures efficient recording and retrieval of personal information on the consortium blockchain.
(2) The consortium blockchain facilitates the secure storage of users' private information. This information is verified through a Zero-knowledge proof provided by certified consortium blockchain members, and it can relatively reduce blockchain storage costs. Notably, an auditor is in place to unveil potentially maliciously stored information and identify the responsible member by decrypting transactions on the consortium blockchain.
(3) The public blockchain executes the matching of information between two public users. It serves to offer information matching functionality while extending security properties to both users involved in a transaction.

By employing this comprehensive approach, our scheme seeks to enhance information management, matching, and security through the synergy of cross-chain technology, consortium education blockchains, and public blockchains.

1.2 Organization

The remainder of our work is organized as follows. First, we introduce the related works in Sect. 2. The preliminaries are presented in Sect. 3. The scheme is presented in Sect. 4. Then, we describe the details of our protocol and security properties in Sect. 5. Finally, in Sect. 6, we conclude this work.

2 Related Works

In cross-chain scenarios, [8] proposes a cross-chain application called Collafab. It uses the concept of 'collect-sign' [9] to create a cross-chain consensus based on PBFT [10] on the private blockchain consensus. Additionally, it supports interoperability between both public and private users on two blockchains. Current schemes and protocols have certain weaknesses and shortcomings. There are issues with data communication and sharing among different parties, including

challenges such as unverifiable records and the risk of uncontrolled information transfer and matching.

In cross-chain application scenarios, A. Garoffolo et al. [11] proposed Zendoo, a cross-chain transfer protocol that enables decoupled and decentralized sidechain creation and communication. It extends the functionality of a blockchain system like Bitcoin as the main blockchain by utilizing smart contracts, while sidechains support different types of consensus and transactions. However, Zendoo's basic functions rely on the main blockchain, thus lacking decentralization, and the model of the sidechain provides low security and lacks user information privacy. It only adapts to a certain type of blockchain, and the relationship between the main blockchain and sidechains leads to low scalability. Yin et al. [12] proposed an open distributed secure cross-chain notary platform based on MPC (BOOL network). This platform utilizes a Ring-Verifiable random function to securely and verifiably deliver secret keys to new members in the blockchain. This approach helps in concealing the identity of the current owner while enhancing platform compatibility and user-friendliness, all the while maintaining the openness of the public blockchain. It claims that this platform supports different types of blockchains and compatibility perfectly. However, BOOL network is unable to support auditability. He et al. [13] designed a cross-chain-based computing scheme (TSQC) that deploys smart contracts in each blockchain to manage inner chain computation. This strategy reduces cross-chain information exchange, resulting in improved system efficiency. Moreover, the scheme employs the CoSi protocol and multisigncryption algorithms for safeguarding cross-chain data privacy. This scheme supports at least three blockchains and is capable of scalability; however, there is a risk of information leakage except for interchain interoperation. Yi et al. [14] proposed a cross-chain-based premium competition scheme with privacy preservation (CCUBI). This scheme employs a bridge contract and a third-party trust agent to facilitate cross-chain information transfer. User clients generate proofs of data that ensure privacy and computability in the cross-chain scenario. The framework of this scheme can be implemented on two types of blockchains. However, for a malicious node which executes aggregation verification, there is a risk of user information leakage. Kuongho Chen et al. [15] introduced a trusted reputation management scheme (TRM) that achieves cross-chain communication through a relay chain. The relay chain handles cross-chain transactions and consensus, managing the sending and receiving of requests from all nodes in the blockchain system. However, this scheme incurs a high time cost due to the deployment of smart contracts and cross-chain consensus in the blockchain it employs. Although relay chain-based blockchain perfectly supports multiple blockchains, it also lacks decentralization and scalability, and the property of anonymity is unable to provide information security.

Finally, Table 1 compares different cross-chain schemes in terms of several properties with our scheme. Here is the following explanation of the table: Decentralization denotes that information can be transferred without a central party. Info-privacy denotes that no one knows the information value except the sender,

Table 1. Comparison among cross-chain schemes.

Scheme \ Feature	Decentralization	Info-privacy	Scalability	Compatibility	Auditability
Zendoo	◑	○	◔	◑	○
BOOL network	●	●	●	●	○
TSQC	●	◑	◔	◑	○
CCUBI	●	◑	◔	◑	●
TRM	◑	○	◑	●	○
This work	●	●	◑	◑	●

●: the scheme perfectly confirms to the following feature or supports all types of blockchains.
◑: the scheme partly confirms to a feature or supports a certain type of blockchain.
◔: the scheme strictly supports a certain blockchain.
○: the scheme dose not support the feature.

receiver, and the auditor. Moreover, scalability is vital for a cross-chain scheme that supports scenarios involving more than just two blockchains. Compatibility means the scheme is capable of working with different types of blockchains. Auditability means that transactions on the blockchain can be audited by a certain trusted node.

3 Preliminaries

3.1 Pedersen Commitment [16]

A sender can commit a value to create a commitment with randomness to the receiver. Both the sender and the receiver can compute the value by evaluating the commitment. The sender can later disclose the actual value by opening the commitment using the message and the randomness [17]. In elliptic curve cryptography, we denote the Pedersen commitment as $C \leftarrow \text{Cmt}\ (v, r)$, where v is the message, and r is the randomness. Given randomness r and the commitment C, one can recover the value $v \leftarrow \text{Open}\ (C, r)$.

3.2 Boneh-Boyen Signature [18]

Let $\mathbb{G}_1, \mathbb{G}_2$ be bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_2|$ for prime p. Let public parameter g_1 be a generator of $\mathbb{G}_1$ and g_2 be a generator of $\mathbb{G}_2$. To sign a message $m \in Z_p$:

- $(pk, sk) \leftarrow \text{BBKeyGen}\ (1^\lambda)$: input a security parameter, the algorithm outputs a secret/public key pair.
- $\sigma \leftarrow \text{BBSign}\ (sk, m)$: input a secret key sk and for input a message $m \in \{0,1\}^*$, it outputs a signature σ of message m.
- $1/0 \leftarrow \text{BBVer}\ (pk, m, \sigma)$: input a public key pk, a message m and a BBS signature σ, the algorithm outputs 1 if σ is valid and outputs 0 otherwise.

3.3 Aggregate Signatures

Boneh et al. introduced BLS aggregate signatures [19] and introduced the concept of aggregate signatures [20] as follows:

Let $\mathbb{G}_1, \mathbb{G}_2$ be bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_2|$ for prime p, and g_1 be a generator of $\mathbb{G}_1$ and g_2 be a generator of $\mathbb{G}_2$. H is a full-domain collision resistance hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$.

- KeyGen (1^λ) : input a parameter 1^λ, the algorithm outputs a secret/public key pair (sk, pk).
- $\sigma \leftarrow$ Sign (sk, M) to sign a message $M \in \{0,1\}^*$ and input a secret key sk, it outputs a signature denotes as $\sigma \in \mathbb{G}_1$.
- $1/0 \leftarrow$ Ver (pk, M, σ) : the algorithm output 1 if σ is a valid signature and outputs 0 otherwise.
- $\sigma \leftarrow$ AggSig (σ_0, σ_1) Input two BLS signatures (σ_0, σ_1), the algorithm outputs an aggregate signature σ.
- $1/0 \leftarrow$ AggSigVer $(\mathbf{L}, \sigma)$. Input n tuples $\mathbf{L} : \{(pk_i, m_i)\}^n$ and a signature σ, the algorithm outputs 1 if the aggregate signature σ is valid to all messages and outputs 0 otherwise.

3.4 Zero-Knowledge Range Proof

The concept of range proof was first introduced by Boneh et al. in 2008 [21]. Then Bunz et al. [22] introduced a zero-knowledge proof scheme. To prove a secret v in range $[0, u^l]$, the secret v can be decomposed to $\varepsilon = \sum_{0 \leq j \leq l} (\varepsilon_j u^j)$. We define the zero-knowledge range proof scheme as (ZKProve, ZKVer).

- $\pi \leftarrow$ ZKProve (v, k) where v is the input secret, and k is corresponding randomnesses, it output a zeroknowledge proof $\pi : (C_s, \hat{C}_s, \pi_s)$.
- $1/0 \leftarrow$ ZKVer (π). Input a zero-knowledge range proof $\pi : (C_s, \hat{C}_s, \pi_s)$, it outputs 1 if π is a valid proof for the value v.

4 System Model

4.1 Scheme Construction

We formalize our scheme by extending the functionalities of Hyperledger Fabric [23] and Aggregate Cash system [24]. We define several types of nodes in our scheme below:

- Consortium member (CM): CM can generate response messages in response to requests from the trust agent and subsequently submit transactions to the consortium blockchain. The transaction is created for cross-chain consensus by utilizing their individual secret keys.
- Trust agent: The trust agent has the capacity to engage with both the consortium blockchain and public users. It can transfer information requests directly to the consortium blockchain, access transactions stored within the consortium blockchain, and send cross-chain information to users.

- Certificate Authority (CA): CA holds the authority to issue certificates to members of the consortium. This issuance ensures that only authorized consortium members possess the privilege to submit transactions to the consortium blockchain.
- Auditor: The auditor possesses the capability to decrypt any transaction within the consortium blockchain, thus retrieving the details of the transaction.
- Public user: Users are able to initiate information requests to the consortium blockchain through the trust agent. They can broadcast a desired target value and anticipate its reception, pre-match transactions to the designated receiver, and submit matching transactions to the public blockchain.

4.2 Security Threats

We assume that the system parameters are generated. We consider the privacy properties of the attacker model below:

The nodes of CA are honest but curious. They honestly run the protocol while recording all inputs and outputs. The auditor strictly obeys the protocol.

Attackers can act as malicious nodes in both CMs and public users.

Attackers know all parameters and public keys of public users and CMs.

Our scheme focuses on several security properties as follows:

Sybil Attacks. As mentioned above, attackers can manipulate, control, or generate malicious nodes among public users. These nodes are generated to attack the system. However, the consensus algorithm of the public blockchain (varies depending on the type of blockchain) can resist Sybil attacks.

Privacy. The identities of the two parties involved in public matching cannot be known by public blockchain users except themselves, nor can they be known by other consortium members except the one responding to the request. User private information for public matching cannot be leaked during transmission in the public blockchain; it is only known by the consortium member that generates it, the two matching parties, and the authority.

Unforgeability. Attackers cannot generate legitimate cross-chain information, consensus, or public chain matching information by forging any secret information, signatures, or proofs in an attempt to damage the system.

The details of our scheme's security are shown in Sect. 5.3.

5 Our Protocol

Our basic construction is shown in Fig. 1. In this section, we describe the process of a transaction in our scheme in Sect. 4.1, followed by the detailed construction in Sect. 4.2. The security of our scheme is discussed in Sect. 5.3.

5.1 Workflow

- For a user A, they broadcast a target value t on the public channel, expecting other users' private values to match it (they only receive values greater than the target value).
- User B initiates a query request through a trusted agent to a designated consortium member of the consortium blockchain. Their identity is indicated by the public key, and the identity of the consortium member is indicated by its public key.
- The trusted agent verifies user query requests via SPV and then posts them to the consortium blockchain. After the designated consortium members receive the request, they generate the corresponding information as a response: a public blockchain transaction and a secret key encrypted by user B's public key.
- All responses from consortium members must undergo cross-chain consensus signature, and the trusted agent only receives transactions signed by the majority of consortium members, which it then submits to the public blockchain. Similarly, user B only receives the ciphertext under the same conditions and decrypts it to obtain the key corresponding to the value in the transaction.
- If the value corresponding to the secret key received by user B is greater than the target t, a matching request message is initiated to user A, which is encrypted with A's public key and sent to A.
- User A decrypts the matching request message, verifies its correctness, and checks if m is greater than or equal to t. Then, they generate a matching transaction while setting their own secret key corresponding to the received value m, and submit the transaction to the public blockchain.
- The Auditor can reveal transactions from the consortium blockchain, recover details in the transaction, and obtain user information from the transaction.

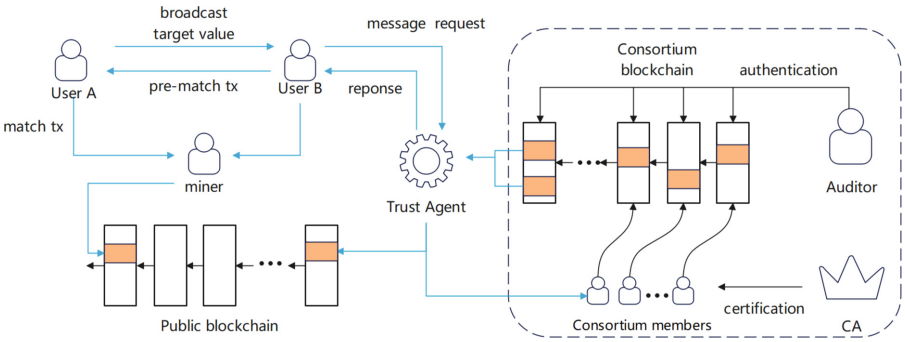

Fig. 1. Basic construction of our scheme

5.2 System Construction

System Initialization

- $(pp, R) \leftarrow$ Setup (1^λ). Input a security parameter 1^λ, and the algorithm outputs the system parameters pp, generates a tuple $(p, g, g_0, \hat{g}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ where $(g, g_0) \in G_1, \hat{g} \in g_2$, the bilinear map $e : G_1 \times G_2 \rightarrow \mathbb{G}_T$ and a range R for zero-knowledge proofs.
- A collision resistance hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$.

Consortium blockchain algorithms

Following consortium algorithms are inspired by Pachain [25]:

- $(apk, ask, capk, cask) \leftarrow$ AuKeyGen (). The algorithm generates the auditor's and the CA's public/secret key pairs.
- $(mpk, msk) \leftarrow$ CMKeyGen (1^λ). This algorithm outputs consortium members' public/secret key pairs (mpk, msk).
- The auditor generates a bitmap $\mathbb{B}$ that indicates which members signed the message for an aggregate signature. Each member who signed the message should add their bit in the proper location by $\mathbb{B}_i \leftarrow$ NewBit $(\mathbb{B})$.
- $(\Pi : (C_s, \pi_s)) \leftarrow$ EncProof (m). Input a message m, the algorithm generates a ciphertext C_s and a zero-knowledge proof π_s.
- $1 \leftarrow$ EncVer $(\Pi : (C_s, \pi_s))$.Input a ciphertext C_s and a proof π_s, the algorithm outputs 1 if Π is a valid zero-knowledge proof.
- $m \leftarrow$ EncDec $(\Pi : (C_s, \pi_s), ask)$. Input the auditor secret key ask and a ciphertext C_s and a proof π_s, the auditor can decrypt C_s to get the value m where the value is in the range of $[0, R]$.

Public CCA Encryption Scheme

- $(upk, usk) \leftarrow$ PKeyGen (pp). Input a security parameter pp, it outputs a user public/secret key pair (upk, usk).
- $C_m \leftarrow$ Enc (M, upk).Input a message M and a user public key upk, it outputs a ciphertext C_m.
- $M \leftarrow$ Dec (C_m, ask). "Input a ciphertext C_m and a user secret key usk; the user decrypts the ciphertext C_m to get the message.

Public Blockchain Algorithms

- $(tx, \hat{k}) \leftarrow$ CreateTx $((C, v, k), \hat{v})$. Input the tuple (C, v, k) and the output score $\hat{v}$. The algorithm outputs a transaction $tx = (s, C, \hat{C}, \Pi, E, \sigma)$ and an output key $\hat{k}$, where $\hat{C}$ denotes the output commitment of the transaction.
- $1/0 \leftarrow$ VerTx (tx). Input a transaction $tx = (s, \mathbf{C}, \hat{\mathbf{C}}, \Pi, E, \sigma)$, the algorithm outputs 1 if tx is a valid transaction and outputs 0 otherwise.
- $(tx) \leftarrow$ AggTx (tx_0, tx_1). Input two transactions (tx_0, tx_1), the algorithm puts them together and returns an aggregate transaction tx.

Message Transfer Algorithms

- PostPB (): This algorithm allows a public user (including the trust agent) to submit a transaction to the public blockchain.
- PostCB (): This algorithm allows a consortium member (including the trust agent) to submit a message to the consortium blockchain.
- SendtoU (): This algorithm allows a public user (including the trust agent) to send a message to other users in a public channel.
- SendtoA (): This algorithm allows a public user to send a message to the trust agent in a public channel.

Cross-Chain Consensus Construction In this section, we describe the construction as follows. The details of the construction are shown in Fig. 2.

The process begins with a consortium member CM_0 initiating cross-chain consensus on a public transaction. This member starts by creating a public transaction denoted as tx and generates the corresponding key k for the message m. Subsequently, it triggers a cross-chain message and posts it onto the consortium blockchain.

For any consortium member, denoted as CM_i where $i \in [1, N]$, that reads a cross-chain message from the consortium blockchain, submitted by another user, the member validates the message and signs it using an aggregate signature mechanism. This generates a cross-chain consensus facilitated by the secret key msk_i.

The trust agent is responsible for extracting messages from the consortium blockchain. It selectively accepts messages that satisfy the criteria of cross-chain consensus, subsequently submitting the public transaction tx to the public blockchain.

Cross-Chain Information Response to Users In this section, we describe the construction of the cross-chain response to the user as follows. The details of the construction are shown in Fig. 3.

To achieve cross-chain privacy for user information, there is a distinction from cross-chain public information (such as public blockchain transactions). In this scheme, the user's requested consortium blockchain information is encrypted by a designated consortium member and subsequently dispatched to the user.

User B initiates a message request directed at a specific consortium member denoted as CM_l, identified by the public key mpk_l, and transmits it to the trust agent. The agent verifies this request and then submits it onto the consortium blockchain.

The consortium member $\mathrm{CM^*}_0$ gets the message from the consortium blockchain and creates a cross-chain response for user B, which is subsequently submitted onto the consortium blockchain. The encrypted message creation process follows the same cross-chain consensus procedure as that of the tx.

The trust agent reviews the messages present on the consortium blockchain and only accepts messages destined for users that meet the conditions of cross-

IniReq ($\text{upk}_B, \hat{\text{mpk}}_0, \text{upk}_a$):

- $req \leftarrow \text{Enc}((\text{upk}_B, \hat{\text{mpk}}_0), \text{upk}_a)$.
- SendtoA (req) .

PostReq (req, usk_a):

- $(upk_B, \hat{mpk}_0) \leftarrow \text{Dnc}(\text{req}, \text{usk}_a)$.
- PostCB ($\text{upk}_B, \hat{\text{mpk}}_0$).

CreateCss (tx, m, msk_0):

- $tx \leftarrow \text{CreateTx}((\text{C}, \text{v}, \text{k}), \text{m}) \ : \ \text{C} \leftarrow \text{Cmt}(\text{m}, \text{k}), \pi \leftarrow \text{ZKProve}(\text{m}, \text{k})$.
- $\Pi \leftarrow \text{EncProof}(\text{m})$.
- $H_{tx} = H(tx, \Pi)$.
- $\sigma_0 \leftarrow \text{Sign}(\text{H}_{\text{tx}}, \text{msk}_0) \ : \ \sigma_0 = \text{H}(\text{tx})^{\text{msk}_0}$.
- $\mathbb{B}_0 \leftarrow \text{NewBit}()$.
- PostCB ($\text{css}_0 = (\text{tx}, \Pi, \text{H}_{\text{tx}}, \mathbb{B}_0, \sigma_0)$) .

AggCss ($\text{css}_j = (\text{tx}, \text{H}_{\text{tx}}, \Pi, \mathbb{B}_j, \sigma_j), \text{msk}_i) \ : \ (\text{i} > \text{j}; \text{j} \geq 0)$:

- if (
  $H_{tx} \neq H(tx, \Pi)$ or
  $0 \leftarrow \text{AggSigVer}(\text{mpk}_j, \text{H}_{\text{tx}}, \sigma_j)$: (j is indicated by $\mathbb{B}_j$) or
) return $\perp$.
- $\sigma' \leftarrow \text{Sign}(\text{msk}_i, \text{H}_{\text{tx}})$.
- $\sigma_i \leftarrow \text{AggSig}(\sigma', \sigma_j)$.
- $\mathbb{B}_i \leftarrow \text{NewBit}(\mathbb{B}_j)$.
- PostCB ($\text{css}_i = ((\text{tx}, \Pi, \text{H}_{\text{tx}}, \mathbb{B}_i, \sigma_i))$.

TxSubmt ($\text{ccs}_t = (\text{tx}, \Pi, \text{H}_{\text{tx}}, \mathbb{B}_t, \sigma_t)$):

- if (
  $H_{tx} \neq H(tx, \Pi)$ or
  $0 \leftarrow \text{AggSigVer}(\text{mpk}_t, \text{H}_{\text{tx}}, \sigma_t)$: (t is indicated by $\mathbb{B}_t$) or
  $0 \leftarrow \text{EncVer}(\Pi)$ or
  $|\mathbb{B}| \leq \frac{2}{3}N$ (N is the total number of CM) or
  $0 \leftarrow \text{VerTx}(\text{tx})$.
) return $\perp$.
- PostCB (tx).

- IniReq ( ) is executed by the public user B.
- PostReq ( ) is executed by the trust agent.
- CreateCss ( ) is executed by a consortium member.
- AggCss () can be executed by all consotium members.
- TxSubmt () is executed by the trust agent.

Fig. 2. Construction of public transaction cross-chain consensus

chain consensus. Subsequently, the agent sends these messages to their designated recipients.

Finally, user B receives the message, validates its authenticity, and then decrypts the content to obtain their own information represented by m.

CreateCssU (tx, m, k, $\hat{msk}_0$, $\hat{mpk}_0$):

- $\tau_k \leftarrow H(upk, \hat{mpk}_0, k, tx)$.
- $k_u \leftarrow$ Enc (k, tx, upk) and $H_k = H(k_u)$.
- $\sigma_\tau \leftarrow$ BBSign ($\hat{msk}_0, \tau_k$).
- $\hat{\sigma} \leftarrow$ Sign ($\hat{msk}_0, H_k$).
- $\mathbb{B}_\tau \leftarrow$ NewBit ().
- PostCB ($\hat{cssu}_0$: ($k_u, H_k, \tau_k, \sigma_\tau, \hat{\sigma}, \mathbb{B}_\tau$)) .
- $\hat{cssu}_i \leftarrow$ AggCss ($\hat{cssu}_j = (k_u, H_k, \tau_k, \sigma_\tau, \hat{\sigma}_j, \mathbb{B}_{\tau_j})$: (i > j; j ≥ 0).
- PostCB ($\hat{cssu}_i$).

ResVer ($\hat{cssu}_t = (k_u, H_k, \tau_k, \sigma_\tau, \hat{\sigma}_t, \mathbb{B}_{\tau_t})$):

- if (
  $H_{tx} \neq H(k_u)$ or
  $0 \leftarrow$ AggSigVer ($mpk_j, H_{tx}, \hat{\sigma}_j$) : (j is indicated by $\mathbb{B}_j$) or
  $|\mathbb{B}| \leq \frac{2}{3}N$ or
  $0 \leftarrow$ BBSVer ($\hat{mpk}_0, \tau_k, \sigma_k$).
) return ⊥.
- SendtoU ($cssu_t$).

UserRcv ($ccsu_t = (k_u, H_k, \tau_k, \sigma_\tau, \hat{\sigma}_t, \mathbb{B}_{\tau_t})$):

- if (
  $H_{tx} \neq H(k_u)$ or
  $0 \leftarrow$ AggSigVer ($mpk_t, H_{tx}, \hat{\sigma}_t$) : (j is indicated by $\mathbb{B}_t$) or
  $|\mathbb{B}| \leq \frac{2}{3}N$ or
  $0 \leftarrow$ BBSVer ($\hat{mpk}_0, \tau_k, \sigma_{\tau_k}$).
) return ⊥.
- $(k, tx) \leftarrow$ Dec (usk_B, k_u).
- $m \leftarrow$ Open (C, k).

- CreateCssU ( ) is executed by consortium members.
- ResVer ( ) is executed by the trust agent.
- UserRcv ( ) is executed by the public user B.

Fig. 3. Construction of user information cross-chain consensus

Public Information Matching In this section, we describe the construction of the public information match as follows. The details of the construction are shown in Fig. 4.

For a public user A, they generate a target value t and broadcast it to all public users, expecting to receive a value which is $\geq t$. For a public user B, who intends to send their value m satisfying the condition $m \geq t$ to user A while keeping other values denoted as v^*, initiates a Prematch transaction. They then send the encryption of the prematch transaction to user A. Upon receiving a $pmtx$ from user B, user A validates it and generates a match transaction along with a new key, effectively granting user A ownership of the information. Finally, user A submits a match transaction to the public blockchain while retaining the secret key privately.

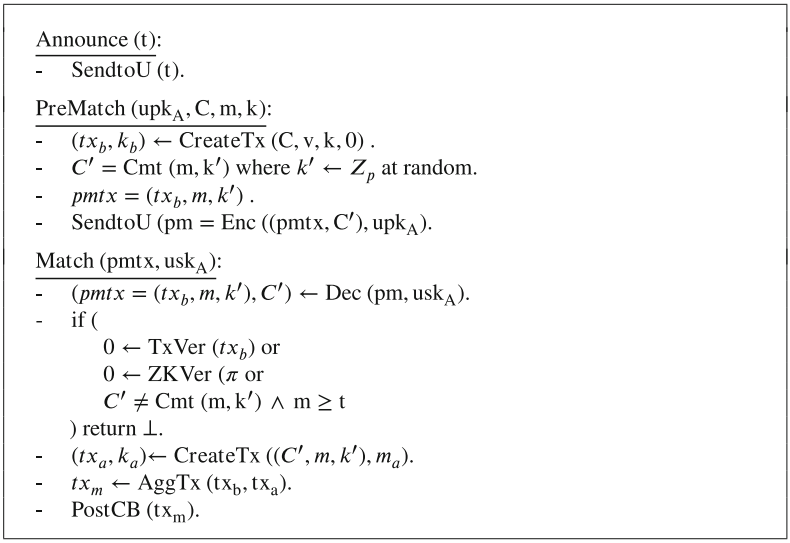

Fig. 4. Construction of public information matching

Auditability In this section, we describe the construction of the audit as follows. The details of the construction are shown in Fig. 5.

The auditor can decrypt a transaction on the consortium blockchain to recover the necessary transaction details. Notably, the auditor can particularly uncover the identity of the first member who initially submitted the transaction to the consortium blockchain. The auditor validates the message on the consortium blockchain and decrypts the ciphertext to reveal the information m in the transaction tx.

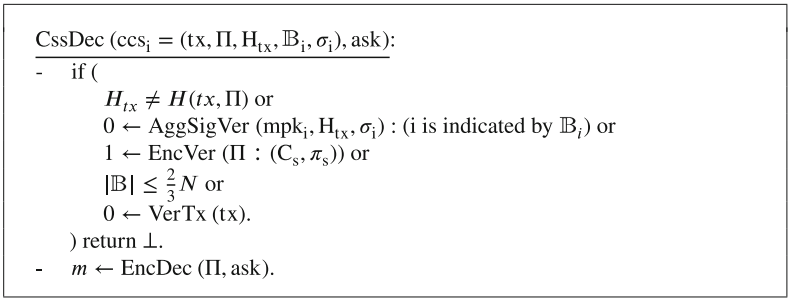

Fig. 5. Construction of auditability

5.3 System Security

In this section, we present security theorems concerning the proposed scheme and illustrate their proof by simplifying them into the security of cryptographic primitives. In our subsequent security analysis, public information (such as transactions and public keys within the public blockchain, and the public keys of consortium members) is considered to be known to potential adversaries.

We will first define the security properties of consortium blockchain algorithms in the following section.

Definition 1. *(Soundness) For any P.P.T. adversary $\mathcal{A}$:*

$$Pr[ask \leftarrow \mathcal{A} : (\Pi^* = (\pi_s^*, C_s^*)) \leftarrow \mathcal{A}(ask),$$
$$1 \leftarrow \mathrm{ZKVer}\ (\pi_{\mathrm{s}}^*) = 1, \mathrm{m} \leftarrow \mathrm{EncDec}\ (\mathrm{C}_{\mathrm{s}}^*, \mathrm{ask})]\ \textit{is negligible.}$$

Thus our consortium blockchain algorithms ensure soundness.

Definition 2. *(CPA Semantically secure indistinguishability)*

For any P.P.T. CPA adversary $\mathcal{A}$, here we define two experiments, Experiment 0 and Experiment 1. In each experiment $b = \{0, 1\}$, the adversary generates m_0, m_1 and sends to the challenger, the challenger computes $\Pi^* \leftarrow \mathrm{EncProof}\ (\mathrm{m_b})$ where $b \leftarrow \{0, 1\}$, and sends it to $\mathcal{A}$, the adversary outputs $\hat{b} \in \{0, 1\}$.

Let Exp_{A_b} be the adversary outputs 1 in experiment b, then the advantage of $\mathcal{A}$ in algorithm EncProof ():

$$Adv_{\mathcal{A}}^{\mathrm{EncProof}} = |Pr[Exp_{A_0}] - Pr[Exp_{A_1}]|\ \textit{is negligible}$$

Thus our consortium blockchain algorithms are semantically secure against a CPA adversary.

Next, we will define the security properties of public blockchain algorithms:

Definition 3. *(Security against inflation) The only way a message can be created is through cross-chain consensus transactions. This implies that for any transaction, the total value of the outputs should be equal to the sum of the total value of the inputs plus the new value of the transaction. For any P.P.T. adversary $\mathcal{A}$:*

$$Pr[(tx, v) \leftarrow \mathcal{A} : tx' \leftarrow \mathcal{A}(v'), v' > v \wedge 1 \leftarrow \mathrm{TxVer}\ (\mathrm{tx}')] = \mathrm{neg}(\lambda).$$

Definition 4. *(Security against theft) The property of theft resistance ensures that only the key owner can utilize the message for matching and aggregation purposes. For any P.P.T. adversary $\mathcal{A}$:*

$$Pr[tx \leftarrow \mathcal{A} : k^* \leftarrow \mathcal{A}(tx), (tx', \hat{k}') \leftarrow \mathrm{CreateTx}\ ((\mathrm{C}, \mathrm{v}, \mathrm{k}^*), \hat{\mathrm{v}})] = \mathrm{neg}(\lambda).$$

Definition 5. *(Transaction indistinguishable)*

The amounts involved in a transaction are concealed, ensuring that only the sender and receiver are aware of the sum of exchanged. Furthermore, a transaction comprehensively conceals the associations between inputs and outputs, making it impossible to discern which inputs funded which outputs.

For any P.P.T. adversary $\mathcal{A}$, the adversary generates $(v_0, \hat{v_0}), (v_1, \hat{v_1})$ and sends to the challenger, the challenger computes $tx^* \leftarrow \mathrm{CreateTx}\,(\mathrm{v_b}, \hat{\mathrm{v_b}})$ where $b \leftarrow \{0,1\}$, and sneds it to $\mathcal{A}$, the adversary outputs $\hat{b} \in \{0,1\}$.

Let Exp_{tx_b} be the adversary outputs 1 in experiment b, then the advantage of $\mathcal{A}$ in algorithm CreateTx ():

$$Adv_{\mathcal{A}}^{\mathrm{TxIND}} = |Pr[Exp_{tx_0}] - Pr[Exp_{tx_1}]| \leq \ negligible.$$

Definition 6. *(Correctness). Our scheme has correctness if several following conditions satisfy:*

- *User requests can be correctly decrypted and verified by the trust agent, and consortium members can generate legitimate transactions.*
- *Cross-chain consensus can be accurately executed. Once the signature count surpasses a specified threshold, the recipient can verify cross-chain information using the designated member public keys indicated by the bitmap $\mathbb{B}$.*
- *Zero-knowledge proofs generated by the consortium chain can be verified by any honest node.*
- *Public users are able to correctly decrypt cross-chain encrypted information sent by consortium members and encrypted matching information from other public users.*

Theorem 1. *Our scheme satisfies correctness if all algorithms are correct.*

Proof. The correctness can be obtained from the workflow of the scheme and can be derived from the correctness of the following cryptographic primitives:

We first focus on the abilities of the adversary $\mathcal{A}$, where $\mathcal{A}$ has the following oracles:

- O_{cu}: The adversary can corrupt any user node to obtain (upk, usk) and make queries to the consortium blockchain via the trust agent.
- The adversary can make two types of queries to the CCA public encryption simulator:
  O_{pke}: The adversary can send a message m to the simulator to obtain the encrypted ciphertext c.
  O_{pkd}: The adversary can send a ciphertext c, which was not received in previous encryption queries, to obtain the decryption of the ciphertext.
- O_{cm}: The adversary can corrupt any single consortium member to obtain (msk, mpk), read information on the consortium blockchain, and initiate a cross-chain consensus message.

– $O_{sig}(m)$: The adversary can make queries to an aggregate signature simulator to request a signature on a message signed by the secret key msk.

User query security means that attackers cannot obtain the identities of users and consortium members from queries received by the trust agent. Now, consider the experiment described below in Fig. 6. For a P.P.T. adversary $\mathcal{A}$: $\mathcal{A}$ can use oracle O_{cu} to get user secret/public key pairs, and use $(O_{pke},\ O_{pkd})$ for the CCA attack game queries. Then $\mathcal{A}$ generates $(upk_{B_0}, \hat{mpk}_{0_0}), (upk_{B_1}, \hat{mpk}_{0_1})$ sends them to the challenger. The challenger computes $req^* \leftarrow \text{Enc}\left((\text{upk}_{\text{B}_\text{b}}, \hat{\text{mpk}}_{0_\text{b}}), \text{upk}_\text{a}\right)$ where $b \leftarrow \{0,1\}$, and sends it to $\mathcal{A}$, the adversary then outputs $\hat{b} \in \{0,1\}$. We define Exp_{req_b} to be the adversary outputs 1 in experiment b, then the advantage of $\mathcal{A}$ in User query security is denoted as $Adv_{\mathcal{A}}^{\text{UQS}}$.

Exp_{req_b}:

1 : $b \leftarrow\$ \{0,1\}$

2 : $(upk_a, usk_a) \leftarrow\$ \text{PKeyGen}()$

3 : $((upk_{B_0}, \hat{mpk}_{B_0}), (upk_{B_1}, \hat{mpk}_{B_1})) \leftarrow\$ \mathcal{A}(upk_a)$

4 : $req^* \leftarrow\$ \text{Enc}(upk_a, (upk_{B_b}, \hat{mpk}_{B_b}))$

5 : $b' \leftarrow\$ \mathcal{A}((upk_a, req^*)$

6 : **return** $b = b'$

Fig. 6. Experiments of User query security

Definition 7. *(User query security). Our scheme provides user query security if for any P.P.T. adversary $\mathcal{A}$:*

$$Adv_{\mathcal{A}}^{\text{UQS}} = |Pr[Exp_{req_0}] - Pr[Exp_{req_1}]| \text{ is negligible.}$$

Theorem 2. *Our scheme provides user request security if the public encryption scheme we applied here satisfies CCA indistinguishability.*

Our scheme requires an unforgeable cross-chain consensus, where the adversary is unable to forge a cross-chain consensus with an aggregate signature that can be verified and decrypted correctly.

Consider the experiment below: For a P.P.T. adversary $\mathcal{A}$, $\mathcal{A}$ can use oracle O_{cm} to get any consortium member secret/public key pair, and use oracle $O_{sig}(m)$ to get the signature of m (We request that the oracle O_{cm} can be used at most $\frac{2}{3}N - 1$ times). The adversary is able to interact with the consortium blockchain to obtain $ccs_i, i \in [1, Q]$ (Q is the maximum number that $\mathcal{A}$ can query to the challenger). Given ccs_i, the challenger computes $m_i \leftarrow \text{EncDec}(ccs_i, ask)$

and sends m_i to $\mathcal{A}$. Finally $\mathcal{A}$ computes a valid forgery pair (css^*, m^*) which is not in the set (ccs_i, m_i), $i \in [1, Q]$ and this forgery pair can be successfully verified by the trust agent and received by the user. We denote the advantage of $\mathcal{A}$ to generate a valid pair (css^*, m^*) as $Adv_{\mathcal{A}}^{\mathrm{CCU}}$.

Definition 8. *(Cross-chain consensus unforgeability). Our scheme satisfies cross-chain consensus unforgeability if for any P.P.T. adversary $\mathcal{A}$, $Adv_{\mathcal{A}}^{\mathrm{CCU}}$ is negligible.*

Theorem 3. *Our scheme provides Cross-chain consensus unforgeability if the aggregate signature scheme we applied here satisfies unforgeability.*

Here we consider user information security, to be more precise, attackers are unable to derive m from *css* and *cssu*. For a P.P.T. adversary $\mathcal{A}_0$: the adversary generates $(tx_0, m_0), (tx_1, m_1)$ and sends them to the challenger, the challenger computes $css^* \leftarrow \mathrm{CreateCss}\left((tx_b, m_b), msk\right)$ where $b \leftarrow \{0, 1\}$, and sends it to $\mathcal{A}_0$, the adversary outputs $\hat{b} \in \{0, 1\}$. We define Exp_{css_b} to be the adversary outputs 1 in experiment b, then the advantage of $\mathcal{A}_0$ in consortium response security is denoted as:

$$Adv_{\mathcal{A}}^{\mathrm{CRS}} = |Pr[Exp_{css_0}] - Pr[Exp_{css_1}]|$$

Similarly, for a P.P.T. adversary $\mathcal{A}_1$: the adversary generates $(tx_0, m_0, k_0, \hat{mpk}_0)$, $(tx_1, m_1, k_1, \hat{mpk}_1)$ and sends to the challenger, the challenger computes $cssu^* \leftarrow \mathrm{CreateCssU}\left((tx_b, m_b, k_b, \hat{mpk}_b), msk\right)$ where $b \leftarrow \{0, 1\}$, and sends it to $\mathcal{A}_1$, the adversary outputs $\hat{b} \in \{0, 1\}$. Define Exp_{cssu_b} to be the adversary outputs 1 in experiment b, then the advantage of $\mathcal{A}_1$ in user response security is denoted as:

$$Adv_{\mathcal{A}}^{\mathrm{URS}} = |Pr[Exp_{cssu_0}] - Pr[Exp_{cssu_1}]|$$

We denote the advantage of adversary in User information security as $Adv_{\mathcal{A}}^{\mathrm{UIP}}$ (Fig. 7).

Definition 9. *(User information security). Our scheme provides user information security if for a P.P.T. adversary$\mathcal{A}_0$: $Adv_{\mathcal{A}_0}^{\mathrm{CRS}}$ is negligible, for a P.P.T. adversary$\mathcal{A}_1$: $Adv_{\mathcal{A}_1}^{\mathrm{URS}}$ is negligible, then for any P.P.T. adversary $\mathcal{A}$: $Adv_{\mathcal{A}}^{\mathrm{UIP}}$ is negligible.*

Theorem 4. *Our scheme satisfies User information security if the public encryption scheme we applied here satisfies indistinguishability, and cross-chain messages css and cssu leak no information about m.*

Proof. The property of user information privacy for *cssu* is similar to the indistinguishability of the public encryption scheme. The property of user information privacy for *css* can be directly derived from *EncProof indistinguishable* from

Exp_{css_b}:	Exp_{cssu_b}:
1 : $b \leftarrow\$ \{0,1\}$	1 : $b \leftarrow\$ \{0,1\}$
2 : $(mpk, msk) \leftarrow\$$ CMKeyGen ()	2 : $(mpk, msk) \leftarrow\$$ CMKeyGen ()
3 : $((tx_0, m_0), (tx_1, m_1)) \leftarrow\$ \mathcal{A}\ (mpk)$	3 : $((tx_0, m_0, k_0, \hat{mpk}_0), (tx_1, m_1, k_1, \hat{mpk}_1)) \leftarrow\$ \mathcal{A}\ (mpk)$
4 : $css^* \leftarrow\$$ CreateCss $((\mathrm{tx_b}, \mathrm{m_b}), \mathrm{msk})$	4 : $cssu^* \leftarrow\$$ CreateCssU $((\mathrm{tx_b}, \mathrm{m_b}, \mathrm{k_b}, \hat{\mathrm{mpk}}_\mathrm{b}), \mathrm{msk})$
5 : $b' \leftarrow\$ \mathcal{A}\ ((mpk, css^*)$	5 : $b' \leftarrow\$ \mathcal{A}\ ((mpk, cssu^*)$
6 : **return** $b = b'$	6 : **return** $b = b'$

Fig. 7. Experiments of User information security

Definition 2 and *Transaction indistinguishable* from Definition 5. If a P.P.T. adversary $\mathcal{A}_1$ can break the indistinguishability of *css*, then it is able to break *Transaction indistinguishable* as well. Thus both *css* and *cssu* leak no user information, and our scheme satisfies user information privacy.

We now focus on the ability of the auditor: All *css* that can be verified correctly on the consortium blockchain can only be decrypted by the auditor to obtain m. Anyone except the auditor is unable to decrypt *css* on the consortium blockchain. Then, the auditability requires consortium responses with cross-chain consensus to be successfully verified and decrypted. We define the advantage of the adversary to obtain m under the conditions mentioned above as $Adv_{\mathcal{A}}^{\mathrm{ADT}}$.

Definition 10. *(Auditability). Our scheme satisfies auditability if for a P.P.T. adversary$\mathcal{A}_0$: $Adv_{\mathcal{A}_0}^{\mathrm{CCU}}$ is negligible, for a P.P.T. adversary$\mathcal{A}_1$: $Adv_{\mathcal{A}_1}^{\mathrm{UIP}}$ is negligible, then for any P.P.T. adversary $\mathcal{A}$: $Adv_{\mathcal{A}}^{\mathrm{ADT}}$ is negligible.*

Theorem 5. *Our scheme satisfies auditability if it also satisfies Cross-chain consensus unforgeability and User information security.*

6 Features, Efficiency and Applications

6.1 Features and Efficiency

Our solution implements the following features:

Cross-Chain Data Transfer: We achieve cross-chain data transmission through the consortium blockchain to the public blockchain. The scheme enables the transfer of both public information and personal private information by using a trusted agent, public information and user private information can be sent from the consortium blockchain to designated public nodes.

Unforgeability: Malicious consortium members are unable to generate forgery user information and submit them to the consortium blockchain. All consensus algorithms are effective and can be read by the trust agent and submitted to the public blockchain.

Anonymity: For certain specific scenarios, both parties involved in a matching transactions unwilling to disclose their identities to the public. It is achieved by utilizing confidential transactions and compatible signatures, replacing the key corresponding to the identity with a special commitment to realize the verification of information by different parties, as well as identity privacy for both sending and receiving parties.

User Information Privacy: In certain specific scenarios, for a user the information on the public blockchain ensures that no one else can use it for matching. More simply, the user private information for public matching is only known by the consortium member which generates it, two match parties and the auditor.

Auditability: The auditor can decrypt a transaction in the consortium blockchain to recover transaction with necessary, it can reveal the detail of the transaction and each consortium member which signed the transaction to enclose a malicious attacker, the auditor can reveal the member who submit the transaction to the consortium blockchain.

Our solution helps reduce blockchain storage costs to a certain extent:

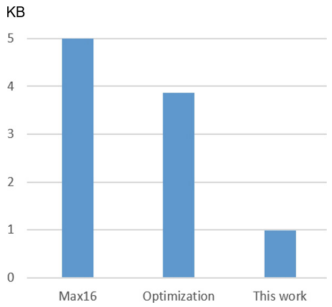

Fig. 8. Comparison of 32-bits range proofs cost

In blockchain transactions, range proofs consume a significant amount of storage in blockchain, leading to excessively large ledger information. As depicted in Fig. 8, our approach, compared with current implementations such as [17], effectively reduces the size of the range proof from $O(n)$ to $O(\log(n))$. This reduction results in a range proof cost of approximately 1 KB for 32-bit messages, a substantial improvement compared to 5.4 KB and 3.8 KB with optimizations.

6.2 Applications

Our solution facilitates data transfer from a consortium blockchain to a public blockchain using cross-chain technology. With support for vector input, our scheme finds straightforward applications in scenarios involving control signal transmission and anonymous notarization. The aggregation transaction function in our scheme has the potential to significantly reduce the expense of cross-chain

transmission. In notarization scenarios, multiple requests from users on the same node can be efficiently handled through aggregated signatures, streamlining the information transmission process.

Furthermore, our solution has the potential to be applied in the exchange and matching of educational and medical information, especially within educational settings. Educational institutions can take on the role of consortium blockchain members, and these members can release information in a quantitative format. Users' academic performance, educational information, and personal abilities can all be shared by consortium blockchain members for public chain matching.

In medical scenarios, members of the consortium blockchain can include diverse medical institutions, while patients, as users, initiate queries for medical information through trusted agents. Various parameters in the medical process can serve as inputs for transactions, and patients can utilize medical data through public blockchain matching. These methods effectively ensure the privacy protection of sensitive medical and identity information exchanged between medical institutions and patients.

7 Conclusion

In this paper, we propose an information matching scheme based on cross-chain technology. Unlike existing blockchain systems, information platforms, and traditional information matching systems, our scheme facilitates consensus between different blockchains, offering features such as unforgeability, anonymity, information privacy, and cross-chain information auditability. Our scheme functions as a distributed information transmission system capable of auditing public transactions created by consortium members to reveal the details of user information involved in an information matching process. Furthermore, our scheme demonstrates resistance within a specific threat model and satisfies various security properties under this model.

References

1. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Decentral. Bus. Rev. (2008)
2. Wüst, K., Kostiainen, K., Capkun, V., Capkun, S.: Prcash: centrally-issued digital currency with privacy and regulation. IACR Cryptology ePrint Archive 2018, p. 412 (2018)
3. Noura, M., Atiquzzaman, M., Gaedke, M.: Interoperability in internet of things: taxonomies and open challenges. Mob. Netw. Appl. **24**, 796–809 (2019)
4. Celer homepage. https://celer.network/. Accessed 10 Nov 2023
5. Multichain homepage. https://multichain.org. Accessed 10 Nov 2023
6. Synapse homepage. https://www.synapseprotocol.com. Accessed 10 Nov 2023
7. Umbri homepage. https://bridge.umbria.network/. Accessed 10 Nov 2023
8. Ghosh, B.C., Bhartia, T., Addya, S.K., Chakraborty, S.: Leveraging public-private blockchain interoperability for closed consortium interfacing. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, pp. 1–10. IEEE (2021)

9. Syta, E., et al.: Keeping authorities "honest or bust" with decentralized witness cosigning. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 526–545. IEEE (2016)
10. Castro, M., Liskov, B.: Practical byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst. (TOCS) **20**(4), 398–461 (2002)
11. Garoffolo, A., Kaidalov, D., Oliynykov, R.: Zendoo: A zk-snark verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains. In: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), pp. 1257–1262. IEEE (2020)
12. Yin, Z., Zhang, B., Xu, J., Lu, K., Ren, K.: Bool network: an open, distributed, secure cross-chain notary platform. IEEE Trans. Inf. Forensics Secur. **17**, 3465–3478 (2022)
13. He, Y., Zhang, C., Wu, B., Yang, Y., Xiao, K., Li, H.: Cross-chain trusted service quality computing scheme for multi-chain model-based 5g network slicing SLA. IEEE Internet Things J. (2021)
14. Yi, L., et al.: CCUBI: a cross-chain based premium competition scheme with privacy preservation for usage-based insurance. Int. J. Intell. Syst. **37**(12), 11522–11546 (2022)
15. Chen, K., Lee, L.F., Chiu, W., Su, C., Yeh, K.H., Chao, H.C.: A trusted reputation management scheme for cross-chain transactions. Sensors **23**(13), 6033 (2023)
16. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_9
17. Maxwell, G.: Confidential transactions (2015) (2016)
18. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_4
19. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_30
20. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_26
21. Camenisch, J., Chaabouni, R., shelat, A.: Efficient protocols for set membership and range proofs. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 234–252. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_15
22. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 315–334. IEEE (2018)
23. Androulaki, E., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, pp. 1–15 (2018)
24. Fuchsbauer, G., Orrù, M., Seurin, Y.: Aggregate cash systems: a cryptographic investigation of mimblewimble. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11476, pp. 657–689. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17653-2_22
25. Yuen, T.H.: PAChain: private, authenticated and auditable consortium blockchain. In: Mu, Y., Deng, R.H., Huang, X. (eds.) CANS 2019. LNCS, vol. 11829, pp. 214–234. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31578-8_12