information

MDPI

Article

Cross-Chain Identity Authentication Method Based on Relay Chain

Qipeng Huang, Minsheng Tan * and Wenlong Tian

School of Computer Science, University of South China, Hengyang 421001, China;
20222008110472@stu.usc.edu.cn (Q.H.); wenlongtian@usc.edu.cn (W.T.)
* Correspondence: 1989000423@usc.edu.cn

Abstract: The cross-chain identity authentication method based on relay chains provides a promising solution to the issues brought by the centralized notary mechanism. Nonetheless, it continues to encounter numerous challenges regarding data privacy, security, and issues of heterogeneity. For example, there is a concern regarding the protection of identity information during the cross-chain authentication process, and the incompatibility of cryptographic components across different blockchains during cross-chain transactions. We design and propose a cross-chain identity privacy protection method based on relay chains to address these issues. In this method, the decentralized nature of relay chains ensures that the cross-chain authentication process is not subject to subjective manipulation, guaranteeing the authenticity and reliability of the data. Regarding the compatibility issue, we unify the user keys according to the identity manager organization, storing them on the relay chain and eliminating the need for users to configure identical key systems. Additionally, to comply with General Data Protection Regulation (GDPR) principles, we store the user keys from the relay chain in distributed servers using the InterPlanetary File System (IPFS). To address privacy concerns, we enable pseudonym updates based on the user's public key during cross-chain transactions. This method ensures full compatibility while protecting user privacy. Moreover, we introduce Zero-Knowledge Proof (ZKP) technology, ensuring that audit nodes cannot trace the user's identity information with malicious intent. Our method offers compatibility while ensuring unlinkability and anonymity through thorough security analysis. More importantly, comparative analysis and experimental results show that our proposed method achieves lower computational cost, reduced storage cost, lower latency, and higher throughput. Therefore, our method demonstrates superior security and performance in cross-chain privacy protection.

Keywords: relay chain; cross-chain authentication; privacy protection; homomorphic encryption; Zero-Knowledge Proof (ZKP); InterPlanetary File System (IPFS)

check for updates

Academic Editor: Leandros Maglaras

Received: 4 December 2024
Revised: 28 December 2024
Accepted: 31 December 2024
Published: 6 January 2025

Citation: Huang, Q.; Tan, M.; Tian, W. Cross-Chain Identity Authentication Method Based on Relay Chain. *Information* **2025**, *16*, 27. https://doi.org/10.3390/info16010027

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, with the continuous development of blockchain technology, consortium blockchains have been widely applied in fields such as healthcare [1] and finance [2], and their development is still ongoing. However, multiple independent consortium blockchains often emerge due to the varying interests and data privacy concerns of different organizations. This rapid growth has led to a significant challenge known as the "island effect", which hinders seamless value transfer and information exchange between consortium blockchains. Therefore, achieving value transfer and information integration across consortium blockchain networks has become a pressing issue. Cross-

chain technology is one key solution for addressing this challenge, which aims to solve the interoperability problems between consortium blockchain platforms.

Existing cross-chain mechanisms [3] include notary, relay, and hash-lock mechanisms, but these solutions pose potential privacy leakage risks. Notary mechanisms include single-signature notary mechanisms and multi-signature notary schemes. The single-signature notary mechanism faces the risk of a single point of failure [4]. Although the multi-signature notary mechanism alleviates the centralization risk by introducing multiple notary nodes, it still carries the risk of collusion between notary nodes, which may lead to leakage of user identities and transaction privacy [5]. In the relay mechanism, cross-chain data transfer depends on relay nodes. It requires avoiding the malicious behavior of the relay nodes to prevent privacy leakage [6]. The hash-lock mechanism focuses more on the fairness of atomic swaps, ensuring that assets are exchanged or not exchanged. However, the hash-lock mechanism fails to protect user identity and transaction privacy effectively. Additionally, the hash-lock mechanism is typically only applicable to cross-chain operations between homogeneous blockchains and lacks compatibility for cross-chain operations between heterogeneous blockchains [7]. Therefore, existing cross-chain methods generally suffer from privacy leakage risks, especially regarding transaction and identity privacy.

Identity privacy leakage is often caused by the openness and transparency of blockchain, where users' identities are not sufficiently protected, making it one of the most concerning privacy issues. Protecting identity privacy in a blockchain is crucial because transaction anonymization techniques can only offer limited privacy protection and cannot fully ensure the security of users' identities. In cross-chain access scenarios, if a user from Consortium A wishes to access data on Consortium B's blockchain, the first challenge is to verify the user's identity on Blockchain B to ensure the security of cross-chain data access. Previous studies have attempted to apply traditional identity authentication methods to cross-chain data access, but these methods still face three major challenges in practical applications.

First, identity authentication methods need to be compatible across different consortium blockchains. Some existing studies focus on cross-chain access between homogeneous blockchains and propose corresponding identity authentication schemes, but they overlook the complexity and diversity of heterogeneous blockchains. Other studies attempt to solve cross-chain authentication issues between heterogeneous blockchains but often require all users and nodes participating in the cross-chain process to reconfigure cryptographic systems to meet unified identity authentication needs [8]. This approach not only incurs significant configuration overhead but also increases the burden of key management for users and nodes. Therefore, achieving full compatibility for cross-chain authentication without altering the existing cryptographic systems has become a primary issue. Existing solutions often store users' pseudonyms and public keys directly on the blockchain. However, this approach violates the data minimization principle in the General Data Protection Regulation (GDPR) [9] and significantly strains on-chain storage in large-scale interaction scenarios.

Second, identity authentication methods must protect users' identity privacy and ensure traceability, thus achieving conditional anonymity. Although some studies have focused on privacy protection in cross-chain asset transactions, they often neglect the protection of user identity privacy [8]. While some studies propose anonymizing user identities, these solutions typically fail to ensure identity traceability [10]. Conditional anonymity requires that users' identities remain anonymous to all nodes on a blockchain during a cross-chain transaction process, with the real identity accessible only under specific conditions (such as for audit purposes) and by authorized auditors. However, if the auditor nodes have excessive privileges, there is a potential risk of abuse or unauthorized investigation of the user's identity, posing a threat to user privacy [5].

Finally, identity authentication methods must resist linkability attacks based on on-chain transaction records, thus achieving unlinkability. Due to the shared ledger feature of consortium blockchains, attackers may gain control of some nodes, read on-chain transaction records, and infer user identities by analyzing the correlation between multiple transaction records. Although some studies have proposed anonymization solutions for user identities, these methods often fail to effectively defend against linkability attacks [11]. Therefore, cross-chain identity authentication methods must possess unlinkability, ensuring the security of user identities and transaction records and preventing user identities from being linked and traced through transaction records.

To tackle the challenges mentioned earlier, we propose a new Cross-Chain Identity Authentication Method utilizing a Relay Chain. The main contributions of this work are summarized in four key aspects.

- Achieving Complete Compatibility: We store the cryptographic configurations of all consortium blockchains on a relay chain, allowing for identity verification through the relay chain during cross-chain access. This approach eliminates issues related to the target chain's inability to support identity verification from the source chain. Additionally, by utilizing InterPlanetary File System (IPFS) for distributed storage, we store user tokens on IPFS, reducing on-chain storage pressure and enhancing data security, thereby resolving compatibility issues between heterogeneous chains while satisfying GDPR data minimization requirements.
- Ensuring Conditional Anonymity: By incorporating Paillier homomorphic encryption and pseudonym generation mechanisms, auditor nodes can anonymize user identities while retaining the capability to trace true identities when necessary. Furthermore, introducing zero-knowledge proof technology guarantees that user privacy remains intact under non-malicious conditions.
- Achieving Unlinkability: We leverage the properties of homomorphic encryption to enable low-cost dynamic updates of user pseudonyms. This ensures that users utilize different pseudonyms during cross-chain access, effectively mitigating linkability attacks and safeguarding user identity.
- Validation and Evaluation of the Method: Through security analysis and performance evaluation, we demonstrate the effectiveness of our method in terms of compatibility, conditional anonymity, and unlinkability. Experimental results indicate that our approach excels in operational overhead and throughput, surpassing existing mainstream methods.

The structure of this paper is as follows: Section 2 presents related work, Section 3 discusses preliminary research, Section 4 provides a detailed description of the proposed method, Section 5 conducts a security analysis, and Section 6 introduces the performance evaluation. Finally, Section 7 concludes the paper.

2. Related Work

In cross-chain scenarios, privacy protection and authentication mechanisms are central issues in technical research. Cross-chain transactions typically require the disclosure of authentication information for verification, which enhances the system's transparency but imposes greater demands on its anonymity and unlinkability. Furthermore, the variability in cryptographic algorithms, consensus mechanisms, and data formats across different blockchain systems adds complexity to privacy protection. As a result, finding ways to balance privacy protection with efficient authentication has become a crucial development area in cross-chain technology.

2.1. Complete Cross-Domain Authentication Scheme

BTCAS [12] first introduced the concept of "complete cross-domain" by storing the cryptographic algorithm of each domain on the blockchain and automatically completing the authentication via smart contracts, although anonymity cannot be achieved. The research in [13] addresses the incompatibility between cryptographic algorithms but focuses only on the public parameters, limiting the cross-domain functionality. The Pseudonym Authentication Scheme integrates Zero-Knowledge Proof (ZKP) to link the pseudonym with the public key and store it on the blockchain, enhancing privacy protection. In [14], users generate public, private, and pseudonym keys using the original encryption settings. However, the system imposes a high storage burden in high-frequency interaction scenarios. In [15], the user's pseudonym and public key are directly stored on the blockchain, simplifying the authentication process but failing to meet privacy compliance requirements such as GDPR. This approach also requires significant storage space in many user interaction scenarios.

2.2. Unlinkability Technology

Unlinkability represents a higher degree of anonymity. Existing anonymity authentication technologies still need improvements in performance, flexibility, and auditing capabilities, and they struggle to meet the comprehensive privacy protection and regulatory needs of cross-chain systems. There are three main methods to achieve unlinkability and identity unlinkability: Anonymous certificates are increasingly less adopted due to the high cost of certificate issuance and management. Group signatures [16] and ring signatures [17] conceal the signer's identity but do not support member revocation and incur high computational overhead; although ring signatures provide privacy, they lack traceability. Pseudonym-based schemes [18,19] generate pseudonyms using homomorphic encryption or hash chains and combine them with zero-knowledge proofs to verify the user's identity. However, the hash chain scheme [18] relies on a trusted third party for frequent updates of the pseudonym, thus increasing security risks. The credential-based pseudonym generation in [19] requires periodic issuance of credentials by an authoritative organization, which adds system complexity and results in higher maintenance costs. Moreover, resource consumption must be reduced. However, none of the above approaches are fully cross-domain.

2.3. Analysis of Cross-Chain Privacy Protection Technologies

In cross-chain transactions, multi-chain heterogeneity introduces new challenges for privacy protection, particularly in the case of public transactions. Ensuring the privacy of user identities and transaction unlinkability remains a complex issue. Current research mainly focuses on asset cross-chain transfer [20], with relatively few studies addressing privacy protection in cross-chain communication. Typical characteristics of existing methods include the following: The group signature-based privacy protection method [16] relies on a central Certificate Authority (CA) to issue and maintain certificates, leading to increased authentication overhead and greater system complexity. The study referenced in [7] achieves user anonymity but is challenging to implement in federated chain scenarios that require traceability. The research presented in [21] examines encryption of communication content but does not consider the protection of user identity privacy.

3. Preliminary Work

3.1. System Model

The cross-chain system is built upon a relay chain architecture, as depicted in Figure 1. The entities involved in a cross-chain scenario are outlined as follows:

- Parachain: Parachains are consortium blockchains that manage business operations within their respective domains.
- Relay Chain: The relay chain is the central hub for cross-chain operations, validating cross-chain transactions. It aggregates transactions from the source blockchain's exit point and, once validated, transfers them to the entry point of the target blockchain.
- Identity Management Authority (IMA): Each parachain is associated with an IMA, which can be either a Certificate Authority (CA) in a Public Key Infrastructure (PKI) system or a Key Generation Center (KGC) in an Identity-Based Encryption (IBE) system. The IMA is responsible for managing the identities of all users on its corresponding parachain and is trusted by that parachain.
- CA in PKI Systems: A PKI system's Certificate Authority (CA) is responsible for issuing and managing digital certificates to authenticate user identities.
- KGC in IBE Systems: The Key Generation Center (KGC) in IBE systems generates and manages the cryptographic keys needed for identity-based encryption.
- User: Each legitimate user is registered with the IMA of their respective parachain and holds identity credentials issued by the IMA. Only authenticated users are permitted to initiate transactions.
- Collator: Collators are full nodes on the parachain, which aggregate cross-chain transactions initiated within the parachain and relay them to the relay chain, or vice versa. Collators are assumed to be honest but curious, meaning they perform their tasks diligently but may observe the input and output data.
- Validator: Validators are nodes on the relay chain responsible for verifying the validity of cross-chain transactions. Similar to collators, they are considered honest but curious.
- ZK-SNARK: Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (ZK-SNARK) generates non-interactive zero-knowledge proofs, allowing auditors to verify user actions and determine if users have the authority to disclose their identity in coordination with validators.
- Auditor (AD): Auditors are entities on the relay chain, and they can reveal the true identity of a transaction sender when necessary. If an auditor is compromised, the security of the entire cross-chain system is jeopardized.
- IPFS: The InterPlanetary File System (IPFS) stores the hash values of user tokens on the relay chain, ensuring compliance with GDPR and enhancing the security of token storage.
- Global Trusted Authority (GTA): The GTA is responsible solely for registering parachains and cross-chain users without participating in on-chain transactions. It is assumed that an adversary could access transactions on parachains and the relay chain. The attacker may also gather information from collators and validators, including their private keys.

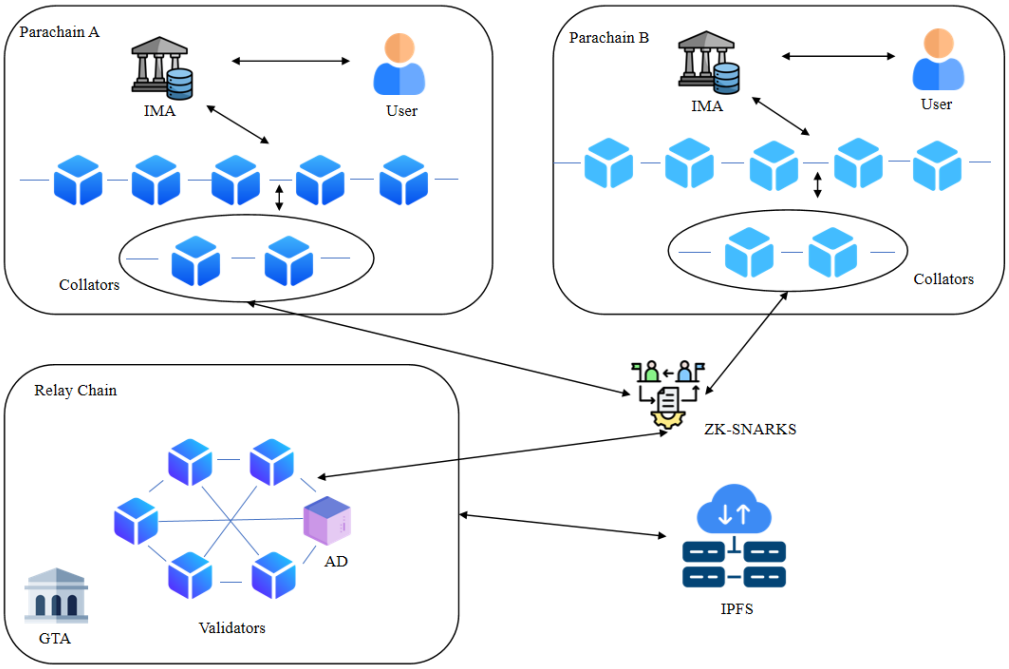

Figure 1. The cross-chain architecture.

3.2. Design Goals

To ensure the security and efficiency of the cross-chain system, the following design goals are established:

- Full Compatibility: Parachains joining the cross-chain system should not require reconfiguring their cryptographic components. Users should be able to generate keys and participate in transactions using their existing cryptographic systems.
- Conditional Anonymity: The identity of users should remain anonymous in cross-chain transactions. Adversaries must be unable to infer the sender's identity from intercepted transactions. Only auditors should be able to trace the user's identity when necessary.
- Unlinkability: Given that the consortium blockchain ledger is accessible to its members, adversaries may be able to collect and analyze multiple transactions. Therefore, preventing attackers from linking two transactions initiated by the same user is crucial.
- Confidentiality: The data exchanged in cross-chain transactions must remain confidential. Only authorized parties should be able to access the transaction data, ensuring that sensitive information is protected from unauthorized access.
- Integrity: The integrity of transaction data must be maintained. Any modification or tampering with transaction data should be detectable, and the system must ensure that the data remain consistent and trustworthy throughout its lifecycle.
- Authentication: Users' identities in cross-chain transactions must be authenticated. Only legitimate users should be permitted to initiate transactions, thereby preventing unauthorized access and impersonation.

3.3. Background Technology

3.3.1. Paillier Cryptosystems

Our work utilizes the Paillier cryptosystem [22], one of the most widely used homomorphic encryption algorithms due to its high efficiency and comprehensive security proofs.

(1) Key Generation: Let $n = pq$, where p and q are large prime numbers. Compute $\lambda = \text{lcm}(p-1, q-1)$, and randomly select a base $g \in \mathbb{Z}_{n^2}^*$ such that $\gcd(L(g^{\lambda}$

mod $n^2), n) = 1$. Then, compute $\mu = (L(g^\lambda \mod n^2) - 1) \mod n$, where $L(u) = \frac{u-1}{n}$. The public key is $pk = (n, g)$ and the private key is $sk = (\lambda, \mu)$.

(2) Encryption: To encrypt a message $m \in \mathbb{Z}_n$, randomly select $r \in \mathbb{Z}_n^*$ and compute $c = E_{pk}(m, r) = g^m \cdot r^n \mod n^2$.

(3) Decryption: To decrypt a ciphertext c, compute $m = D_{sk}(c) = \mu \cdot L(c^\lambda \mod n^2) \mod n$. The Paillier encryption has the following properties:

- Addition: $E(m_1, r_1) \cdot E(m_2, r_2) = E(m_1 + m_2, r_1 \cdot r_2)$
- Multiplication: $E(m_1, r_1)^{r_2} = E(m_1 \cdot r_2, r_1^{r_2})$

3.3.2. Zero-Knowledge Proof

To enhance privacy protection, zero-knowledge proofs [23] allow a prover to demonstrate the validity of a statement to a verifier without disclosing any specific data.

(1) Identity Verification: In cross-chain transactions, users must prove to the verifier that they are legitimate identity holders. Using ZKP, a user can generate a proof $ZKProof$ that verifies the validity of their identity without disclosing their actual identity. Given an identity ID, the user generates a $ZK(ID)$ so that

$$\text{Verify}(ZK(ID)) \Rightarrow \text{True}$$

The verifier can use this proof to validate the correctness of ID without accessing the underlying identity data.

(2) Transaction Verification: The verifier needs to confirm that a cross-chain transaction complies with predefined rules without accessing detailed transaction data. The user generates a $ZK(T)$ for each transaction T, allowing the verifier to assess the transaction's legality through this proof. Given transaction T, the user generates a $ZK(T)$ such that the following holds:

$$\text{Verify}(ZK(T)) \Rightarrow \text{Legitimate}(T) \quad \text{(if valid)}$$

The verifier can validate the legality of T without requiring transaction details.

3.3.3. InterPlanetary File System

(1) Token Storage: Users' tokens and identity information are stored in the IPFS [13] network to achieve decentralization and rapid access. Each file (data block) in IPFS has a unique hash identifier, ensuring data uniqueness and integrity. The user's token generates a hash $H(Token)$ defined as follows:

$$H(Token) = \text{Hash}(\text{Token Data})$$

This hash value serves as the file identifier stored in IPFS.

(2) Transaction Data Storage: Detailed data from cross-chain transactions (e.g., transaction amount, participants) are stored in IPFS, while metadata and transaction hashes are stored on the blockchain. This approach alleviates the storage burden on the blockchain while ensuring data security and scalability. The detailed data of transaction T generate a hash $H(T)$ defined as follows:

$$H(T) = \text{Hash}(\text{Transaction Data})$$

The metadata $Metadata(T)$ and hash $H(T)$ are stored on the blockchain, ensuring a secure and efficient transaction record.

$$\text{Store} \rightarrow \{Metadata(T), H(T)\} \quad \text{on Blockchain}$$

4. System Design

This section elaborates on the proposed solution and its underlying mechanisms. The notation used throughout this work is summarized in Table 1.

Table 1. Notation and description.

Notation	Description
pk_x, sk_x	public key and private key of entity x
$cert_x$	certificate of entity x
ID_x	identity of entity x
CID_A	identity of parachain A
$Info_x$	registration information of entity x
PS_x	pseudonym of entity x
PS_x^0	pseudonym generator of entity x
U_x^A	the xth user in parachain A
IMA_A	IMA in parachain A
$CCTX$	cross-chain transaction
σ	signature
tst	timestamp
$H_A(\cdot)$	hash algorithm adopted by parachain A
$Sig_A(\cdot)$	signature algorithm adopted by parachain A
$E_k(\cdot)$	Paillier encryption function which uses key k
$D_k(\cdot)$	Paillier decryption function which uses key k
$Sign(Sig_A, msg, sk_x)$	sign the message msg using sk_x and Sig_A
$Ver(Sig_A, \sigma, msg, pk_x)$	verify whether σ is the signature of msg using pk_x and Sig_A or not
$H_0(\cdot)$	a hash function $\{0,1\}^* \rightarrow \mathbb{Z}_n^*$
$NIZK$	Non-interactive zero-knowledge proof
$IPFS$	InterPlanetary File System, a distributed file storage

4.1. System Initialization

4.1.1. Common Configuration

Initially, the auditor generates a Paillier key pair, denoted as $\langle pk_{AD} = (n, g), sk_{AD} = (\lambda, \mu)\rangle$, and selects a hash function $H_0 : \{0,1\}^* \rightarrow \mathbb{Z}_n^*$. The auditor then derives the public parameters for the Non-Interactive Zero-Knowledge Proof (NIZK), which are represented as follows:

$$PP_{\text{NIZK}} = \{n, g, H_0(\cdot), crs\}$$

where crs denotes the Common Reference String used in the generation and verification of NIZK proofs. These system public parameters are published as

$$PP = \{n, g, H_0(\cdot), crs\}$$

4.1.2. Parachain Registration

When a new parachain A intends to join the cross-chain system, its administrator submits a registration request to the Global Trusted Authority (GTA) via an offline channel. The request includes the public key of the Identity Management Authority (IMA) pk_{IMA_A}, along with the hash algorithm $HA(\cdot)$ and signature algorithm $Sig_A(\cdot)$. The registration information $Info_A$ consists of essential details for parachain A, such as the number of nodes and the IP addresses of collators, etc.

If the registration is approved, the GTA generates a unique identifier $CIDA$ and stores the following on the relay chain:

$$\{CIDA : pk_{IMA_A}, HA(\cdot), Sig_A(\cdot)\}$$

To optimize data storage and maintain data integrity, the GTA stores the registration details in the IPFS network, retaining only the metadata and hash values on the relay chain. Subsequently, the GTA allocates and configures several validators to facilitate communication with the collators of parachain A. The GTA then returns the $CIDA$ along with the IP addresses of the assigned validators. The administrator of parachain A publishes the $CIDA$ and configures the collators for cross-chain communication. A bidirectional authentication is established between the IMA and the GTA, with subsequent communication security ensured by a shared key.

4.2. On-Chain Authentication

This method does not interfere with on-chain authentication. Parachains continue to use their original identity management systems, allowing users to retain their existing keys and credentials without reconfiguring other cryptographic components. In the on-chain authentication process, Zero-Knowledge Proofs (ZKPs) are leveraged, enabling users to prove the legitimacy of their identity without revealing any personal information, significantly enhancing the system's privacy protection.

4.3. Cross-Chain Authentication

The cross-chain authentication process consists of three stages: cross-chain registration, cross-chain transaction, and pseudonym update, as shown in Figure 2. We assume parachain A uses a Public Key Infrastructure (PKI) authentication system for this process. Before the process begins, parachain A has completed its registration and successfully joined the cross-chain system. Simultaneously, a legitimate user U_{uA} possesses a certificate $cert_u$ issued by the IMA.

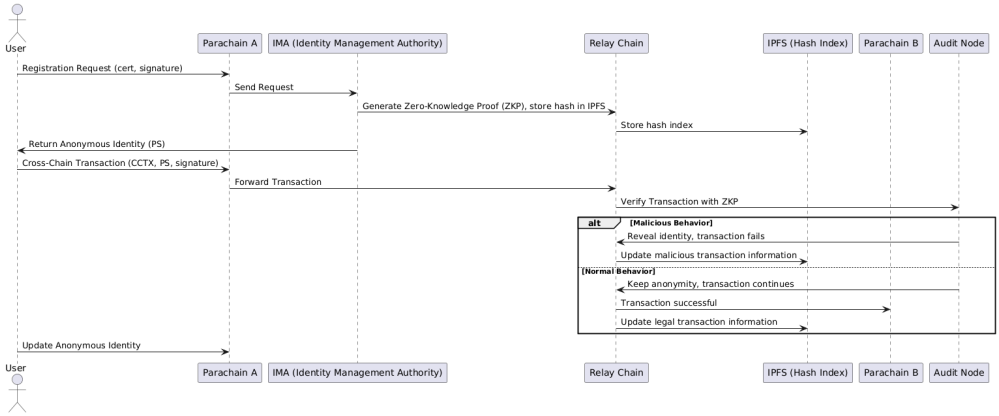

Figure 2. The workflow of cross-chain authentication.

4.3.1. Cross-Chain Registration

When a user U_{uA} requests cross-chain access, the user sends a cross-chain registration request encrypted with pk_{IMA_A}. The plaintext request includes U_{uA}'s certificate, a timestamp, and a signature.

The IMA extracts p_{ku} from $cert_u$, verifies the validity of the signature and certificate, and checks the freshness of the request. Upon successful verification, the IMA generates a unique identity:

$$ID_u = H_0(cert_u||CIDA)$$

The IMA then sends a user registration request to the GTA. Upon receiving the request, the GTA verifies the identity. If the equality holds and $Info_U$ is valid, the GTA generates

multiple tokens for U_{uA}. The GTA first generates k masks for ID_u by randomly selecting $m_{iu} \in \mathbb{Z}_n^*$ for $i = 1, 2, \ldots, k$ and computing the following:

$$M_{iu} = (m_{iu} \cdot ID_u)^{-1} \mod n$$

Then, the GTA randomly selects an integer $\alpha_u \in \mathbb{Z}_n^*$ and computes the tokens.

To ensure the security of token storage and efficient access, the GTA stores these tokens in IPFS and retains the corresponding hash values and metadata on the relay chain, thus acquiring the block addresses.

$$BID_u = \{BID_{u1}, BID_{u2}, \ldots, BID_{uk}\}$$

Subsequently, the GTA generates a pseudonym generator

$$PS_{u0} = (\alpha_u)$$

and returns it to the IMA. The IMA receives and stores PS_{u0} in its local database, then computes the pseudonym:

$$PS_u = f(PS_{u0}, ID_u) \quad (\text{for some function } f)$$

The IMA returns $\{PS_u, E_{pku}(BID_u)\}$ through a secure channel. User U_u receives the message and locally stores $\{PS_u, BID_u\}$.

When a user U_{Au} requests cross-chain access, the user must generate a zero-knowledge proof (ZKP) to demonstrate possession of a legitimate identity credential without revealing the credential's details. In the registration request sent to their respective IMA, the user not only provides $cert_u$ (the user credential) and a signature but also generates a non-interactive zero-knowledge proof π to validate the legitimacy of their credential. The registration request format becomes the following:

$$\{registration, cert_u, \pi, tst, Sign(Sig_A, tst, sku)\}$$

where π is the non-interactive zero-knowledge proof that the verifier (either the IMA or the GTA) can validate using the system's public parameters PP_{NIZK}. The proof is used to verify the following conditions:

(1) The user possesses a valid credential.
(2) The corresponding IMA indeed issued the credential.

This approach ensures the user can meet the verification requirements without disclosing their identity.

4.3.2. Cross-Chain Transaction

When a user U_{uA} initiates a cross-chain transaction, the user creates a transaction request formatted as $\{CCTX, \sigma, PSu, pku, BID_{ui}\}$, where $\sigma = \text{Sign}(Sig_A, HA(CCTX), sku)$. The collator captures this transaction and forwards it to the validator in the format $\{CCTX, \sigma, PSu, pku, BID_{ui}, CIDA\}$.

Upon receiving the transaction, the validator retrieves $Token_{iu}$ using BID_{ui} and the hash function $< HA(\cdot) >$ and obtains $Sig_A(\cdot)$ via $CIDA$. The validator then verifies the following conditions:

(1) Validity of *pku*: The validator checks the legality of *pku* using homomorphic properties.
(2) Validity of σ: The validator verifies the validity of σ against the corresponding signature system of Parachain A.

The transaction is valid and forwarded to the target parachain's verifier if all conditions are satisfied.

When U_{uA} initiates a cross-chain transaction, the user generates a Non-Interactive Zero-Knowledge (NIZK) proof π_{CCTX} to substantiate the transaction's legitimacy. This proof ensures the user has sufficient assets or authorization to perform the transaction without revealing sensitive details like account balances.

The transaction request now has the following format:

$$\{CCTX, \sigma, PSu, pku, BID_{ui}, \pi_{CCTX}\}$$

In this context, π_{CCTX} is the NIZK proof. The validator (e.g., a verification node on the relay chain) uses the system's public parameters PP_{NIZK} to validate the transaction. The NIZK proof ensures the following:

(1) A legitimate user initiated the transaction.
(2) The assets involved in the transaction or other requirements meet the system's specifications.

With the NIZK proof, the validator can verify the transaction's validity without accessing sensitive data.

4.3.3. Pseudonym Update

When a user U_{uA} needs to update their pseudonym, the user first generates a new key pair $\langle pku', sku' \rangle$ using the original cryptographic system and then sends a pseudonym update request to $IMAA$.

The $IMAA$ begins by verifying the validity of *certu* and checking whether it has been revoked. If the certificate is valid, $IMAA$ computes ID_u and searches for $PSu0$ in the local database. A new pseudonym is then generated using the following equation:

$$PSu' = (PSu0 \cdot g^{H_0(pku')} \cdot ID_u) \mod q$$

In this context, $g^{H_0(pku')}$ represents the result of applying the hash-to-point function to the public key pku'. The term ID_u refers to the unique identifier assigned to the user, while q denotes the order of the group, which ensures that the resulting values remain manageable in size. By applying the modular operation $\mod q$, the generated pseudonym PSu' constrained within the group's limits, thus minimizing the risk of issues related to excessive size. The function $IMAA$ subsequently returns PSu' to U_u, enabling U_u to use the pair $\langle PSu', pku' \rangle$ for future authentication.

To ensure the unlinkability of user identities, the pseudonym is updated with each cross-chain transaction. During this update, the Non-Interactive Zero-Knowledge (NIZK) mechanism verifies that the new pseudonym is correctly linked to the user's real identity without revealing sensitive information.

To request a pseudonym update, the user generates a NIZK proof π_{update} to confirm that the new pseudonym is correctly associated with their real identity. The format of the pseudonym update request is as follows:

$$\{update, pk'u, certu, tst', \pi_{update}, \text{Sign}(Sig_A, pk'u||tst', sku)\}$$

In this context, π_{update} serves as a non-interactive zero-knowledge (NIZK) proof that ensures the newly generated pseudonym corresponds correctly to the user's identity credentials. The validator uses the public parameters PP_{NIZK} to authenticate this proof, thereby ensuring the security and anonymity of the pseudonym update process.

5. Security Analysis

This section examines the potential threats to the proposed system and discusses the measures taken to mitigate these risks. Specifically, we analyze four common types of attacks:

(1) Replay Attack: A replay attack occurs when an adversary intercepts and retransmits a legitimate authentication message to gain unauthorized access [24]. To counter this threat, our system incorporates unique session identifiers, such as timestamps or nonces, ensuring that each authentication request is distinct and can be verified as non-repetitive.
(2) Impersonation Attack: In an impersonation attack, an attacker falsifies their identity to gain unauthorized access to the system [25]. By employing robust identity authentication mechanisms, such as Zero-Knowledge Proofs (ZKPs), our system guarantees that only authorized users can authenticate successfully, preventing impersonation.
(3) Reflection Attack: Reflection attacks exploit the symmetry of a protocol by reflecting a received message to the sender, thereby deceiving them [26]. Our protocol design includes asymmetrical message flows and challenge-response mechanisms, which thwart attempts to reflect messages to the originator.
(4) Man-in-the-Middle Attack: A man-in-the-middle attack occurs when an adversary intercepts and alters the communication between two legitimate parties [27]. To defend against such attacks, our system ensures the confidentiality and integrity of communication by employing secure encryption and robust key exchange protocols, effectively preventing unauthorized message interception or modification.

This section will further discuss the proposed method's security through formal and informal analyses.

5.1. Correctness

The authentication process described in Section 5 consists of two main parts: the first part verifies the legitimacy of pku, while the second part verifies the validity of σ. The correctness of the signature algorithm ensures the second part. In this section, we focus on proving the correctness of the first part.

The legitimacy of pku is verified using Zero-Knowledge Proofs (ZKP). ZKPs enable the verifier to confirm the authenticity of pku without revealing any underlying information. This approach enhances the system's privacy and security, as even if an adversary intercepts the communication, they cannot deduce the user's private key or identity information. This ensures that the authentication process remains both secure and correct.

5.2. Formal Analysis with Scyther

To assess the security of the proposed method, we use Scyther, a widely recognized security protocol verification tool. Scyther uses the Security Protocol Description Language (SPDL) and operates in Python 2.7. For authentication, Scyther provides four key assertions: Alive, Weakagree, Niagree, and Nisync. These assertions help detect vulnerabilities such as replay attacks, impersonation attacks, reflection attacks, and man-in-the-middle attacks. By formally analyzing the protocol using these assertions, we ensure that the proposed system is robust against these common attack vectors.

Our protocol primarily involves five entities: Uu, $IMAA$, $CollatorsA$, GTA, and V (Validator), which are modeled as roles U, I, C, GTA, and V respectively. We adopt the Dolev–Yao threat model to verify the four assertions above and the confidentiality of each role's user identity uid. As illustrated in Table 2, the verification results indicate that our scheme can withstand replay, impersonation, reflection, and man-in-the-middle attacks while achieving anonymity and unlinkability.

Table 2. Scyther analysis results.

Role	Alive	Weakagree	Niagree	Nisynch
Uu	✓	✓	✓	✓
$IMAA$	✓	✓	✓	✓
$CollatorsA$	✓	✓	✓	✓
GTA	✓	✓	✓	✓
V	✓	✓	✓	✓

✓ denotes a positive result for the corresponding analysis.

Furthermore, our protocol enhances the protection of identity information throughout the protocol by introducing ZKP. Notably, ZKP ensures the legitimacy of transactions during the verification process without exposing the user's identity information. Additionally, incorporating IPFS guarantees the security of stored user tokens and transaction data, preventing tampering or leakage during the on-chain storage process.

5.3. Informal Analysis of Design Goals

5.3.1. Complete Compatibility

User UuA generates a key pair $\langle pku, sku \rangle$ using the original cryptographic system and signs the initiated transaction with the original signature algorithm $SigA(\cdot)$ and hash function $HA(\cdot)$. The user does not need to reconfigure a new cryptographic system throughout the authentication process. Our protocol supports different encryption systems and ensures compatibility and efficiency in cross-chain scenarios by storing relevant data on IPFS.

5.3.2. Anonymity

When initiating a transaction, the user presents σ, PSu, pku, and $BIDui$. The user generates both σ and pku and do not contain any identity information. The identity information in PSu is protected by Paillier homomorphic encryption using $pkAD$. Since attackers cannot obtain $skAD$, they are unable to decrypt PSu to retrieve IDu. Moreover, the identity information in $BIDui$ undergoes randomization, preventing attackers from inferring IDu. Through Zero-Knowledge Proofs (ZKP), our system provides robust anonymity protection without exposing identity.

5.3.3. Traceability

Given $\langle PSu, pku, BIDui \rangle$, an auditor AD can first utilize NIZK zero-knowledge proofs, then use $BIDui$ to obtain $\langle miu \cdot Mui, EpkAD(\alpha_u, \alpha_u \cdot Mui \cdot miu) \rangle$. The auditor decrypts $\alpha_u = DskAD(EpkAD(\alpha_u, \alpha_u \cdot miu \cdot Mui))$ and computes $x = (\alpha_u + H0(pku))^{-1}$ to track IDu. This process ensures conditional traceability, where user identities can be revealed when necessary without compromising privacy during normal transactions.

5.3.4. Unlinkability

Different PSu, pku, and $BIDui$ are used for various authentication rounds, making it impossible for attackers to find identical elements in the authentication information. Consequently, attackers cannot link transactions sent by the same user. By combining zero-knowledge proofs and IPFS, our system ensures the independence of each transaction, further enhancing unlinkability and preventing attackers from deducing user identities through behavioral analysis of multiple transactions.

6. Performance Evaluation

This section will compare our approach with other cross-domain and cross-chain solutions regarding functionality, computational overhead, and communication overhead.

6.1. Functionality

We compare the functionalities of our method with Ridra [18], PEPA [19], BPCDA [13], CDAS [1], CCAP [14], BCIOT [7], and PPSC [8]. The comparison results are shown in Table 3, indicating that the security properties are not fully implemented. Some security properties are not considered in our work; for instance, identity validity is publicly verifiable in blockchain scenarios, so mutual authentication is not addressed.

Among these solutions, BCIOT does not achieve anonymity and unlinkability because user identity must be provided during the authentication process. BPCDA implements anonymity but uses the same anonymous address across different rounds, failing to ensure unlinkability. In Ridra, CDAS, and PPSC, authentication messages cannot be linked to observers but can be connected to verifiers. BPCDA, CDAS, and CCAP achieve complete compatibility, but BPCDA and CDAS only apply to different cryptographic parameters and hash functions, not different cryptographic systems. Only PPSC and BCIOT are designed for cross-chain scenarios.

By introducing Zero-Knowledge Proofs (ZKP), our system achieves a high level of anonymity and unlinkability and provides conditional traceability. This enables users to conduct secure cross-chain transactions without exposing their real identities. Furthermore, integrating IPFS significantly improves the system's data storage and access efficiency, ensuring high compatibility in cross-chain scenarios. In summary, our approach is the solution that simultaneously satisfies unlinkability, conditional anonymity, and complete compatibility, and it supports cross-chain scenarios.

Table 3. Comparison of security properties.

Scheme	Unlinkability	Anonymity	Traceability	Complete Compatibility	Cross-Chain
Ridra [18]	✓	×	×	×	×
PEPA [19]	×	×	×	×	×
BPCDA [13]	×	✓	×	✓	×
CDAS [1]	✓	✓	✓	×	×
CCAP [14]	×	✓	✓	✓	×
BCIOT [7]	×	×	×	✓	×
PPSC [8]	✓	✓	✓	×	✓
Ours	✓	✓	✓	✓	✓

✓ indicates the feature is supported, and × indicates the feature is not supported.

6.2. Computational Overhead

Since initialization and registration are one-time operations, we do not consider them here. For fairness, we compare the computational overhead of single authentication, including pseudonym updates and identity verification.

In our method, introducing Zero-Knowledge Proofs increases the computational complexity during the authentication process. However, through optimized algorithms, the system can provide a high level of security with relatively low computational overhead. We estimate the computational overhead of cryptographic operations using the C++ language and the PBC library on Ubuntu 22.04. Table 4 presents the computational costs, denoted by Tme, Tsm, Tbp, $Thtp$, Trs, Trv, $Tnizk$, and $Tifps$, which correspond to big integer modular exponentiation, elliptic curve scalar multiplication, bilinear pairing, hash-to-point functions, RSA signature generation and verification, Zero-Knowledge Proof verification, and IPFS hash storage, respectively.

The cryptographic algorithms, schemes, and protocols used in this analysis rely on widely accepted security parameters to ensure robustness. Specifically, there is the following:

(1) RSA Algorithm: We use a key size of 2048-bit for both signature generation and verification (*Trs* and *Trv*), aligning with industry standards for a security level of 112 bits.
(2) Elliptic Curve Operations: Scalar multiplications (*Tsm*) and bilinear pairings (*Tbp*) are implemented over a 256-bit elliptic curve, providing a security level equivalent to 128 bits.
(3) Zero-Knowledge Proofs: The security parameters for the ZKP generation and verification (*Tnizk*) correspond to the same 128-bit security level, ensuring secure proof construction.
(4) Hash-to-Point Function: Hashing to points on the elliptic curve (*Thtp*) uses a secure cryptographic hash function, such as SHA-256, to prevent collision attacks.
(5) IPFS Hashing: IPFS hash storage (*Tifps*) uses the default IPFS cryptographic hash function, typically using a SHA-256 equivalent, for reliable and secure distributed storage.

For comparison with CDAS, CCAP, PPSC, and our proposed method, we assume the use of the widely adopted RSA algorithm for unspecified signature and encryption operations. In the case of PPSC, asymmetric encryption and decryption operations are denoted by *Tre* and *Trd*, where $Tre \approx Trv$ and $Trd \approx Trs$. Other operations, such as big integer modular multiplication and basic hash functions, are excluded from the comparison due to their minimal impact [28].

Table 4. Computational overhead of cryptographic operations.

Cryptography Operation	*Tme*	*Tsm*	*Tbp*	*Thtp*	*Trs*	*Trv*	*Tnizk*	*Tifps*
Execution Time	0.32 ms	1.765 ms	2.178 ms	8.947 ms	0.327 ms	0.006 ms	0.026 ms	0.173 ms

The computational overhead is presented in Table 5. Despite the additional processing steps introduced by the zero-knowledge proofs, the overall computational cost remains low, approximately 1.531 ms, with the verification overhead being the smallest at about 0.525 ms. The system, with its comprehensive functionality, ensures usability and efficiency in large-scale interactive environments through effective resource management and the efficient implementation of ZKPs.

Table 5. Comparison of time for cryptographic operations.

Scheme	User	IMA	Verifier
Ridra [18]	$3Tme = 0.960$ ms	-	$3Tme = 0.960$ ms
PEPA [19]	$5Tme = 1.600$ ms	-	$5Tme = 1.600$ ms
BPCDA [13]	$4Tme = 1.280$ ms	-	$3Tme = 0.960$ ms
CDAS [1]	$4Tsm = 7.060$ ms	$Trv + Tre = 0.012$ ms	$5Tsm + Trd = 9.152$ ms
CCAP [14]	$2Trs = 0.654$ ms	$(3t + 45)Tme + 6Tbp = (0.960t + 27.468)$ ms	$(3t + 33)Tme + 9Tbp + Trv = (0.960t + 30.168)$ ms
BCIOT [7]	$Tsm + Tbp = 3.943$ ms	-	$2Tsm + Tbp = 5.708$ ms
PPSC [8]	$Tme + 2Tre + 2Trs = 0.986$ ms	$2Tme + 2Trs + Trd + Trv = 1.627$ ms	$2Trd + 3Trv + Tme + Trs + Tre = 1.325$ ms
Ours	$2Trs + Tnizk = 0.680$ ms	$Tme + Trv = 0.326$ ms	$Tme + Trv + Tnizk + Tifps = 0.525$ ms

6.3. Communication Overhead

To achieve an 80-bit security level, we select p, q, n as 1024-bit values in big integer-based cryptography, resulting in $|Z_n^*| = 128$ bytes and $|Z_n^{*2}| = 256$ bytes, where $|Z_n^*|$ represents the length of an element in $|Z_n^*|$. For elliptic curve-based cryptography, $|Z_q^*| = 20$ bytes. For the bilinear map $e : G_1 \times G_1 \rightarrow G_T$, we use Type e pairing with parameters set as $r = 160, q = 1024$ [29]. Thus, $|G_1| = 256$ bytes and $|G_T| = 128$ bytes. Strings like pseudonyms and timestamps are defined as $|S| = 4$ bytes. The unspecified

encryption and signing algorithms are assumed to be RSA, so the key and signature lengths are $|Z_n^*|$. The unspecified hash function is assumed to be SHA-256, with $|H| = 32$ bytes.

Assuming the use of the Groth16 protocol, each NIZK proof is approximately 192 bytes. Whenever a user submits an NIZK proof, it is accompanied by additional information (such as a user ID, signature, and timestamp). Assuming the extra information occupies 128 bytes, the total communication overhead for the user is the following:

$$|NIZK| = 192 \text{ bytes} + 128 \text{ bytes} = 320 \text{ bytes}$$

Once the user generates the NIZK proof, it can be uploaded to IPFS, and the data are relayed on the blockchain, resulting in a CID of 46 bytes. The user then only needs to submit the CID to the auditing node without transmitting the entire proof file.

The communication overhead for the user when submitting to the auditing node is as follows:

$$|IFPS| = 46 \text{ bytes} + 128 \text{ bytes} = 174 \text{ bytes}$$

For fairness, we compare the communication overhead for unidirectional authentication, focusing solely on authentication information, excluding message content. Table 6 compares the communication overhead of different schemes. It can be observed that the communication overhead of our method is moderate compared to the other schemes. Despite integrating IPFS, our method's one-time authentication communication overhead is approximately 2KB, which is acceptable in cross-chain scenarios. Thanks to IPFS, our approach not only optimizes communication overhead but also reduces the redundant storage and transmission requirements for on-chain data, thereby enhancing the overall communication efficiency of the system.

Table 6. Comparison of communication overhead.

Scheme	Communication Overhead
Ridra [18]	$7\|Z_n^*\| + 5\|Z_{n^2}^*\| = 2176$ B
PEPA [19]	$4\|Z_n^*\| + 5\|Z_{n^2}^*\| + 2\|S\| = 1800$ B
BPCDA [13]	$3\|Z_n^*\| + 2\|H\| = 448$ B
CDAS [1]	$9\|G_1\| + 3\|H\| + 92\|S\| = 2436$ B
CCAP [14]	$(3t+14)\|Z_n^*\| + (3t+9)\|Z_{n^2}^*\| + 12\|G_T\| + 2\|G_T\| + 2\|S\| = (1152t+7432)$ B
BCIOT [7]	$2\|S\| + \|G_T\| = 136$ B
PPSC [8]	$13\|Z_n^*\| + \|S\| = 1668$ B
Ours	$6\|Z_n^*\| + 3\|Z_{n^2}^*\| + 5\|S\| + 2\|NIZK\| + \|IFPS\| = 1710$ B

We implemented our proposed system on a Ubuntu 22.10 platform with an Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz and 4GB of RAM. To realize the cross-chain functionality, we deployed three consortium chains using Hyperledger Fabric v2.4.9, which included two side chains and one relay chain. The cross-chain interactions were facilitated by smart contracts written in Golang 1.18.10. We evaluated the performance of our system using Hyperledger Caliper, comparing its throughput and latency with that of PPSC [8] and BCIOT [7]. For cryptographic operations, we leveraged the built-in `math/big` package for large integer arithmetic, while bilinear pairings and elliptic curve computations used in BCIOT were handled by the `bn256` package.

6.3.1. Throughput

Throughput was evaluated under varying levels of concurrent transactions, where concurrency is defined as the number of simultaneous transactions initiated by clients. The throughput was measured in transactions per second (TPS). We tested different levels of concurrency ranging from 10 to 200 transactions, calculating the average TPS over 10 runs at each level. The results are depicted in Figure 3. Our method achieved a peak TPS of 145, maintaining stability above 140 TPS when concurrent transactions exceeded 40. As shown in Figure 3, our method outperforms both PPSC and BCIOT in terms of throughput, demonstrating superior adaptability to high-concurrency environments. The integration of IPFS further accelerates data access speed, significantly improving the system's ability to handle concurrent transactions.

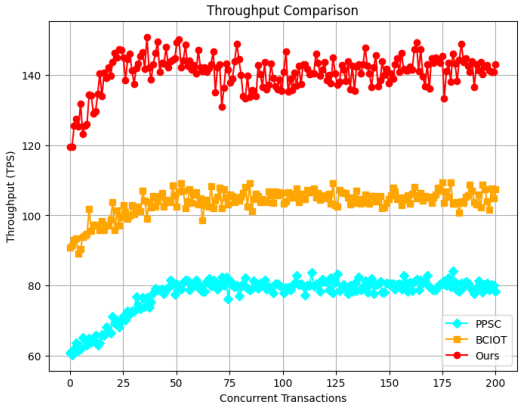

Figure 3. Comparison of throughput.

6.3.2. Latency

Latency was tested by varying the number of peer nodes. A total of 10 peer nodes were deployed on the relay chain, and we sequentially selected between 2 and 10 peer nodes to participate in each transaction. In each test round, we initiated 100 concurrent transactions and measured the time taken to complete these transactions. As illustrated in Figure 4, latency increases linearly with the number of nodes. In comparison to PPSC and BCIOT, our method consistently demonstrates lower latency and superior scalability, making it more suitable for large-scale environments. The use of IPFS helps reduce data transmission latency, while the efficient implementation of Zero-Knowledge Proofs (ZKPs) ensures that additional computational overhead during the verification process remains minimal, further optimizing latency performance.

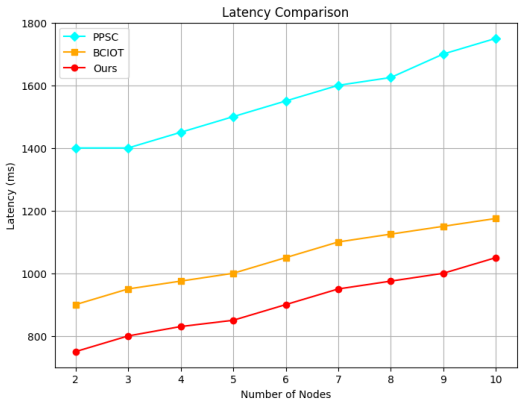

Figure 4. Comparison of latency.

7. Conclusions

In this paper, we proposed a privacy-preserving, complete cross-chain authentication scheme tailored for consortium blockchains. Our approach ensures full compatibility, conditional anonymity, and unlinkability while resisting common attacks. We introduced Paillier homomorphic encryption, pseudonymization techniques, and user-generated non-malice audit Zero-Knowledge Proofs (ZKPs) to preserve the conditional anonymity of user identities. Users can generate new key pairs and sign transactions with their existing cryptographic systems while their pseudonyms are updated in accordance with their public keys, achieving unlinkability through pseudonym updates. We efficiently manage data with Content Identifiers (CIDs) stored in IPFS, retaining only the CID and Zero-Knowledge Proofs on the blockchain, thus reducing storage burden. During cross-chain authentication, cryptographic configurations of parallel chains are stored on the relay chain, and transaction signatures are verified using the cryptographic systems of the source blockchain, ensuring full compatibility across chains.

Author Contributions: Conceptualization, Q.H. and M.T.; methodology, Q.H.; software, Q.H.; validation, Q.H. and M.T.; formal analysis, Q.H.; investigation, Q.H.; resources, M.T.; data curation, Q.H.; writing—original draft preparation, Q.H.; writing—review and editing, Q.H. and M.T.; visualization, Q.H.; supervision, M.T.; project administration, M.T.; funding acquisition, M.T.; experimental guidance, W.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Project 2022JJ50153 of the Hunan Provincial Natural Science Foundation of China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xue, L.; Huang, H.; Xiao, F.; Wang, W. A cross-domain authentication scheme based on cooperative blockchains functioning with revocation for medical consortiums. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 2409–2420. [CrossRef]
2. Ren, Y.S.; Ma, C.Q.; Chen, X.Q.; Lei, Y.T.; Wang, Y.R. Sustainable finance and blockchain: A systematic review and research agenda. *Res. Int. Bus. Financ.* **2023**, *64*, 101871. [CrossRef]
3. Ou, W.; Huang, S.; Zheng, J.; Zhang, Q.; Zeng, G.; Han, W. An overview on cross-chain: Mechanism, platforms, challenges and advances. *Comput. Networks* **2022**, *218*, 109378. [CrossRef]
4. Guo, Z.; Guo, S.; Zhang, S.; Song, L.; Wang, H. Analysis of cross-chain technology of blockchain. *Chin. J. Internet Things* **2020**, *4*, 35–48.
5. Wu, X. Cross-chain workflow model based on trusted relay. In *ACM Turing Award Celebration Conference-China, Proceedings of the ACM TURC 2021: ACM Turing Award Celebration Conference-China (ACM TURC 2021), Hefei, China, 30 July–1 August 2021*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 49–53.
6. Hildebrandt, T.; van Dongen, B.F.; Röglinger, M.; Mendling, J. Business process management. *Lect. Notes Comput. Sci.* **2019**, *11675*.
7. Shao, S.; Chen, F.; Xiao, X.; Gu, W.; Lu, Y.; Wang, S.; Tang, W.; Liu, S.; Wu, F.; He, J.; et al. IBE-BCIOT: An IBE based cross-chain communication mechanism of blockchain in IoT. *World Wide Web* **2021**, *24*, 1665–1690. [CrossRef]
8. Liang, X.; Zhao, Y.; Wu, J.; Yin, K. A privacy protection scheme for cross-chain transactions based on group signature and relay chain. *Int. J. Digit. Crime Forensics (IJDCF)* **2022**, *14*, 1–20. [CrossRef]
9. Voigt, P.; Von dem Bussche, A. The eu general data protection regulation (gdpr). *A Pract. Guid. 1st Ed. Cham Springer Int. Publ.* **2017**, *10*, 10–5555.
10. Gao, S.; Su, Q.; Zhang, R.; Zhu, J.; Sui, Z.; Wang, J. A Privacy-Preserving Identity Authentication Scheme Based on the Blockchain. *Secur. Commun. Networks* **2021**, *2021*, 9992353. [CrossRef]
11. Li, W.; Chen, L.; Lai, X.; Zhang, X.; Xin, J. Bpcex: Towards blockchain-based privacy-preserving currency exchange. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 3–6 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–9.

12. Zhang, H.; Chen, X.; Lan, X.; Jin, H.; Cao, Q. BTCAS: A blockchain-based thoroughly cross-domain authentication scheme. *J. Inf. Secur. Appl.* **2020**, *55*, 102538. [CrossRef]
13. Jiang, J.; Zhang, Y.; Li, J. A blockchain-based privacy-preserving scheme for cross-domain authentication. In Proceedings of the 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Wuhan, China, 9–11 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 992–999.
14. Tong, F.; Chen, X.; Wang, K.; Zhang, Y. CCAP: A complete cross-domain authentication based on blockchain for Internet of Things. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3789–3800. [CrossRef]
15. Chulerttiyawong, D.; Jamalipour, A. A blockchain assisted vehicular pseudonym issuance and management system for conditional privacy enhancement. *IEEE Access* **2021**, *9*, 127305–127319. [CrossRef]
16. Chen, B.; Wang, Z.; Xiang, T.; Yang, J.; He, D.; Choo, K.K.R. BCGS: Blockchain-assisted privacy-preserving cross-domain authentication for VANETs. *Veh. Commun.* **2023**, *41*, 100602. [CrossRef]
17. Guan, Z.; Zhou, X.; Liu, P.; Wu, L.; Yang, W. A blockchain-based dual-side privacy-preserving multiparty computation scheme for edge-enabled smart grid. *IEEE Internet Things J.* **2021**, *9*, 14287–14299. [CrossRef]
18. Sun, C.; Liu, J.; Jie, Y.; Ma, Y.; Ma, J. Ridra: A rigorous decentralized randomized authentication in VANETs. *IEEE Access* **2018**, *6*, 50358–50371. [CrossRef]
19. Zhao, C.; Guo, N.; Gao, T.; Deng, X.; Qi, J. PEPA: Paillier cryptosystem-based efficient privacy-preserving authentication scheme for VANETs. *J. Syst. Archit.* **2023**, *138*, 102855. [CrossRef]
20. Thyagarajan, S.A.; Malavolta, G.; Moreno-Sanchez, P. Universal atomic swaps: Secure exchange of coins across all blockchains. In Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 22–26 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1299–1316.
21. Li, Y.; Weng, J.; Li, M.; Wu, W.; Weng, J.; Liu, J.N.; Hu, S. Zerocross: A sidechain-based privacy-preserving cross-chain solution for monero. *J. Parallel Distrib. Comput.* **2022**, *169*, 301–316. [CrossRef]
22. Catalano, D.; Gennaro, R.; Howgrave-Graham, N.; Nguyen, P.Q. Paillier's cryptosystem revisited. In Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, PA, USA, 5–8 November 2001; pp. 206–214.
23. Trautwein, D.; Raman, A.; Tyson, G.; Castro, I.; Scott, W.; Schubotz, M.; Gipp, B.; Psaras, Y. Design and evaluation of IPFS: A storage layer for the decentralized web. In Proceedings of the ACM SIGCOMM 2022 Conference, Amsterdam, The Netherlands, 22–26 August 2022; pp. 739–752.
24. Singh, M.; Pati, D. Countermeasures to replay attacks: A review. *IETE Tech. Rev.* **2020**, *37*, 599–614. [CrossRef]
25. Brown, P.N.; Borowski, H.P.; Marden, J.R. Security against impersonation attacks in distributed systems. *IEEE Trans. Control Netw. Syst.* **2018**, *6*, 440–450. [CrossRef]
26. Nuiaa, R.R.; Manickam, S.; Alsaeedi, A.H. Distributed reflection denial of service attack: A critical review. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 5327. [CrossRef]
27. Conti, M.; Dragoni, N.; Lesyk, V. A survey of man in the middle attacks. *IEEE Commun. Surv. Tutorials* **2016**, *18*, 2027–2051. [CrossRef]
28. Zhang, Y.; Deng, R.H.; Bertino, E.; Zheng, D. Robust and universal seamless handover authentication in 5G HetNets. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 858–874. [CrossRef]
29. Koblitz, N.; Menezes, A. Pairing-based cryptography at high security levels. In *Proceedings of the IMA International Conference on Cryptography and Coding*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 13–36.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.