A Cross-domain Data Sharing Scheme Based on Federated Blockchain

Honglin Mao, Jie Zhang, Yao Zhang(✉), and Xiaohong Li(✉)

College of Intelligence and Computing, Tianjin University, 300350, Tianjin, China

Abstract. Cross-domain data sharing involves exchanging or providing access to data between different systems or individuals, which is critical in domains such as healthcare, transportation, and data marketplaces. While blockchain technology addresses the single-point-of-failure issue inherent in cloud servers for cross-domain data sharing, existing blockchain-based methods face challenges such as complex permission management, high storage costs, and trust issues. To address these challenges, we propose a federated blockchain based cross-domain data sharing scheme that simplifies permission management by reducing the number of entities and specifying access rules, and decreases storage overhead through encrypted indexing. Our approach enhances trust by improving the ring signature algorithm to meet requirements for anonymity and accountability in sensitive data sharing. Specifically, the enhanced ring signature algorithm provides anonymous authentication and tamper-proof signatures while incorporating traceability features to balance privacy protection with security auditing. This scheme effectively resolves the challenges of privilege management, storage costs, trust crises, privacy protection, and security auditing in medical data sharing scenarios. Formal security proofs, performance analysis, and experimental results validate the scheme's security, efficiency, and feasibility. Experimental data shows a reduction in signing time by at least 30% and a 14% decrease in verification time.

Keywords: Ring signature · Cross-domain Data Sharing · Privacy protection · Signature traceability · Blockchain.

1 Introduction

Cross-domain data sharing refers to the exchange and integration of data across various domains, departments, systems, or organizations. It has become an essential mechanism for collaboration and information exchange, particularly in fields such as healthcare [36]. By enabling the integration of data from diverse domains, cross-domain data sharing not only ensures data integrity but also enhances data utilization efficiency, prevents information silos, and improves

Honglin Mao and Jie Zhang contributed equally to this work.

Corresponding authors: zzyy@tju.edu.cn; xiaohongli@tju.edu.cn.

system transparency and real-time capabilities [21]. Despite its benefits, security remains a critical concern in cross-domain data sharing, as it is necessary to safeguard data integrity, ensure compliance, and protect user privacy. Specifically, inter-domain authentication plays a vital role in securing data exchanges [10].

While several studies have proposed cloud-based cross-domain data sharing schemes that provide centralized platforms for data exchange, these solutions often face significant challenges, such as trust crises. Specifically, issues such as inefficiency and privacy leakage have been highlighted in practical applications [4], which negatively affect system performance and resource utilization [29]. With the rise of cryptographic techniques, cloud-based data sharing solutions leveraging Public Key Infrastructure (PKI) have been introduced. However, these schemes encounter difficulties, including the complexity of certificate management [22] and vulnerability to a single point of failure [17]. As a result, innovative technological solutions are essential to address these limitations, and recent schemes have focused on mitigating the single point of failure issue through novel computing paradigms [9].

Blockchain-based cross-domain data-sharing schemes address trust crises and mitigate single points of failure via decentralized networks, consensus mechanisms, and tamper-proof ledgers, thereby enabling trustless data verification without reliance on central authorities [28, 5]. Enhanced by smart contracts and heightened transparency, these frameworks ensure integrity, auditability, and secure data exchange. Nonetheless, three critical challenges persist: **Challenge 1:** the complexity of decentralized privilege management, which arises from conflicting requirements for multi-node authentication, granular access control, and auditable encryption, thereby limiting adaptability to diverse participant needs; **Challenge 2:** scalability issues inherent in blockchain's ledger replication protocol, where voluminous data records exacerbate communication latency, network congestion, and node storage costs, ultimately hindering efficiency; and **Challenge 3:** the privacy-audit trade-off, wherein transaction transparency—vital for auditing—exposes sensitive metadata and compromises participant anonymity despite cryptographic protections. In [14], blockchain's auditable and interoperable nature in healthcare data sharing is shown to potentially compromise user anonymity due to data traceability and insufficient privacy mechanisms, leading to data leaks, tampering, and unauthorized disclosures that adversely affect clinical data availability and patient care quality. Consequently, privacy and integrity in cross-domain data-sharing schemes [20] must be rigorously evaluated to ensure data security and validity [31].

To address these issues, we propose a federated blockchain-based distributed cross-domain data-sharing scheme that enhances interoperability and standardization across healthcare organizations, thereby facilitating efficient patient information exchange and reducing data loss from inconsistent formats. To resolve **Challenge 1**, we leverage Hyperledger Fabric's modular architecture for simplified access control and distributed authentication; for **Challenge 2**, we reduce storage, communication, and bandwidth overhead by storing only encrypted data indices; and to mitigate **Challenge 3**, we introduce an enhanced

traceable ring signature (TRS) algorithm that preserves signer anonymity while enabling traceability via an authority node (AN) [16]. Optimized with a stricter, more transparent traceability mechanism, our TRS algorithm guarantees correctness, anonymity, non-tamperability, and auditability, with experimental results demonstrating a 30% reduction in signing time and a 14% decrease in verification time, thereby underscoring the efficiency of our scheme for privacy protection in medical data sharing.

This paper makes four key contributions: (1) We propose a cross-domain data sharing scheme based on a federated blockchain, enhancing data interoperability and standardization across different domains. The scheme provides detailed descriptions of the system and security models while ensuring data tamper-resistance, integrity, and transparency through the use of federated blockchain, making data flow records traceable and auditable, and simplifying the rights management process. (2) We improve and design a new traceable ring signature algorithm. The scheme incorporates the ring signature algorithm into cross-domain data sharing to achieve key security attributes such as anonymity, unforgeability, and auditability. By establishing the authority node within the system and optimizing the signature algorithm, we achieve a balance between privacy protection and security auditing by implementing a tracing and accountability mechanism for the signer while maintaining conditional anonymity. (3) We conduct an extensive security proof, verifying the security properties of anonymity, unforgeability, and traceability for the signature algorithm. The scheme's security and effectiveness are demonstrated within the context of medical data sharing scenarios. (4) Comparative benchmark analysis showing our TRS outperforms existing ring signatures with 30% faster signing and 14% reduced verification latency while uniquely enabling authorized identity tracing—critical for regulatory compliance in multi-stakeholder systems.

2 Preliminaries

2.1 Ring Signature

Ring signatures (RS), pioneered by Rivest, Shamir, and Tauman [25], deliver cryptographic anonymity, unforgeability, and decentralized authentication [2] by eliminating centralized authorities through public key-based signing [7]. While foundational in systems like Monero [23] and enhanced via secure data-sharing variants [35, 6, 12, 1, 32, 18, 8, 33], conventional RS suffers from accountability deficits due to untraceable signers. Fujisaki et al.'s traceable RS (TRS) [11] overcomes this through conditional anonymity—preserving privacy while embedding authorized traceability to resolve transaction ambiguity [27]—enabling auditable applications in healthcare, smart grids, and IoT [24, 15, 26, 16] via privacy-audit equilibrium.

The basic structure of a ring signature scheme consists of three sub-algorithms: $Setup()$, $Sign()$, and $Verify()$. These sub-algorithms are detailed as follows:

- $Setup(1^\lambda) \rightarrow PP$: Initialization algorithm. Given the security parameter λ, this algorithm generates the public parameter PP along with the public-private key pair (pk, sk), where pk represents the user's public key and sk represents the user's private key.

- $Sign(m, sk, L) \rightarrow \sigma$: Signature algorithm. Given a message m, the signer uses their private key sk and the public key set L to generate the signature σ. Here, $L = \{PK_1, ..., PK_n\}$ denotes the set of public keys of the ring members.
- $Verify(\sigma, m, L) \rightarrow True/False$: Signature verification algorithm. Given a message m, the verifier checks the validity of the signature σ against the public key set L, and outputs a boolean result indicating whether the verification succeeded $True$ or failed $False$.

Traceable ring signatures extend the basic scheme by adding an additional sub-algorithm:

- $Trace(\sigma, m, IDs, L) \rightarrow s$: Signature tracing algorithm. Given a message m, the verifier checks the signature σ against the public key set L and the identity set IDs, and outputs the index of the signer s.

2.2 Bilinear Mapping

Definition 1. *(Bilinear Mapping): Let q be a large prime. Consider a q-order additive cyclic group G_1, along with a q-order multiplicative cyclic group G_T. A mapping $e : G_1 \times G_1 \rightarrow G_T$ is said to be bilinear if it satisfies the following properties:*

1) Bilinearity: $\forall P_1, P_2, Q_1 \in G_1$ and $\forall \phi, \varphi \in Z_q^*$, it holds that:

$$e(P_1 + P_2, Q_1) = e(P_1, Q_1)e(P_2, Q_1) \text{ and } e(\phi P_1, \varphi Q_1) = e(P_1, Q_1)^{\phi\varphi}$$

2) Non-degeneracy: $\exists P_1 \in G_1, \exists Q_1 \in G_1$ such that $e(P_1, Q_1) \neq 1_{G_T}$.

3) Computability: $\forall P_1 \in G_1, \forall Q_1 \in G_1$, there exists an efficient algorithm to compute $e(P_1, Q_1)$.

On elliptic curves, the bilinear mapping continues to meet these properties.

2.3 Computational Diffie-Hellman Problem

Definition 2. *(Computational Diffie-Hellman Problem, CDHP): Given a large prime q, suppose $(P, aP, bP) \in G_1$, where G_1 is a cyclic group of order q under addition, and $a, b \in Z_q^*$ are unknown, compute abP. The success probability of solving CDHP in is given by $Adv_{\mathcal{A}}^{CDHP}(\lambda) = Pr[\mathcal{M}(P, aP, abP) = abP]$, where $\mathcal{M}$ is a probabilistic polynomial-time algorithm. For any adversary $\mathcal{A}$, if $Adv_{\mathcal{A}}^{CDHP}(\lambda)$ is negligibly small in the probability polynomial time (PPT), then CDHP does not hold.*

3 Overview of Framework

In this section, we first give a framework description of a cross-domain data sharing scheme, including notation, entities, system architecture, and then give the security attribute conventions.

3.1 System Architecture

We present the system architecture of the cross-domain data sharing scheme as illustrated in Fig. 1. The scheme involves five different types of entities:

1. **AN (Authority Node)**: AN is a distributed blockchain node responsible for system initialization, user identity management, signature traceability, and accountability for failed verifications.

Table 1: Notations.

Symbol	Description
λ	Security Parameters
q	A Large Prime
G_1, G_T	Cyclic Groups of Order q
e	Bilinear Mapping $e : G_1 \times G_1 \to G_T$
P, Q	A Generator/Element in G_1
H_1, H_2	Secure Cryptographic Hash Functions
(y, S)	Private/Public Key Pairs of AN
z, P_{pub}	Master Key, Master Public Key
PP	System Parameters
$ABE.Enc, ABE.Dec$	CP-ABE Encryption/Decryption Indexing Operations
ID, k_{ID}	User Identification, Partial Private Key
v, Y_{ID}	User Secret Value, User Partial Public Key
(x, X)	User's Private/Public Key Pair
M	EHR Content
IDs	ID Collection
L	Public Key Collection
σ, M'	Signature, Short Encrypted Hash Index for EHR

2. **KGC (Key Generation Center)**: KGC is responsible for key distribution, generating a master key, public key, and providing partial private keys and identity IDs to users.
3. **DU (Data User)**: DU is an entity involved in generating, signing, uploading, and accessing Electronic Health Records (EHR), interacting with AN for authorization and permissions.
4. **EN (Entity Node)**: EN handles signature uploading and verification, storing data on the blockchain, generating EHR indexes, and comparing them to the blockchain for verification.
5. **Blockchain**: A platform that securely stores signatures, manages transactions, and records operations. Nodes like AN, KGC, and EN enable distributed key generation, data sharing, and auditing. All data access and modifications are logged, ensuring immutability and traceability of unauthorized actions, enhancing accountability.

This framework integrates Hyperledger Fabric's PBFT-based permissioned architecture with CP-ABE-enhanced encryption for Byzantine fault-tolerant EHR sharing. Fabric's modular design implements PBFT consensus to achieve 2/3 node integrity thresholds, coupled with multi-channel isolation for domain-specific transactional confidentiality. The v2.0+ private state databases enforce node authorization through smart contracts, while inspired by CP-ABE schemes [34, 13] to bind EHRs with dynamic attribute policies — decryption requires both PBFT-validated ledger permissions and cryptographically proven credentials (Table 1). This dual-layer approach (PBFT for Byzantine resilience, CP-ABE for fine-grained access) guarantees tamper-evident auditability and policy-driven data sovereignty in federated healthcare ecosystems.

As illustrated in Fig. 1, to provide a clearer understanding of our proposed scheme, we detail the interaction process of each entity. Prior to these inter-

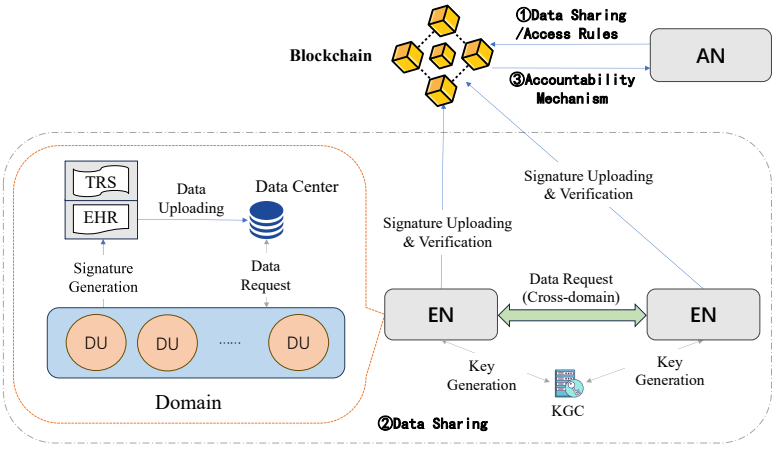

Fig. 1: Overview of Our Approach

actions, an initialization phase is required. This phase involves generating the system parameters PP, the master key and master public key, and the DU's identity ID along with its private/public key pair (x, X). The subsequent interaction processes are outlined as follows:

- **Publishing Access Rules**: To mitigate the complexity of permission management, all DUs negotiate and publish sharing rules and access policies—including DU identity, domain information, and data operations—on the blockchain platform, thereby enhancing transparency, preventing tampering, and streamlining permission allocation and management.
- **Data Sharing**: In sharing Electronic Health Records (EHRs), the solution uses traceable ring signature (TRS) technology to ensure data anonymity, while allowing traceability of the signer when needed, balancing privacy and security auditing. The EHR content is denoted as M, and DU generates a signature σ for M, which is verified by EN. To reduce blockchain storage and computation, EN stores M in a data center and uploads the encrypted index M' of the EHR and σ to the blockchain after validation. Access to the EHR is granted only after legitimate permission is verified. The data and signatures are standardized for inter-domain interoperability and compatibility.
- **Accountability Mechanism**: To ensure auditability, AN can identify the real signer from the signature σ. AN then generates an accountability report, which is sent to the relevant parties, holding the signer accountable. This mechanism not only ensures transparency of system operations but also enhances the security and trustworthiness of the system.

3.2 Security Attribute Convention

In this blockchain-based healthcare data sharing scenario, it is crucial to design traceable ring signature algorithms that enable accountability without requiring interaction between DUs during sensitive data generation. Additionally, AN should be able to identify the actual signer without interacting with the signer or the signature verifier. We analyze the security requirements of this

scenario by defining 3 core properties: anonymity, non-tamperability, and traceability. We assume that the scheme adheres to the security guarantees provided by CDHP.

To facilitate the proof of these security properties in subsequent sections, we introduce the following three types of oracles:

- **Oracle-Random(O_R)**: Outputs a random value.
- **Oracle-Corruption(O_C)**: Takes a public key pk_i as input and returns the corresponding private key sk_i.
- **Oracle-Signature(O_S)**: Takes data m as input and returns a signature σ.

The scheme must satisfy the following security properties:

Anonymity: The scheme satisfies anonymity if, for any adversary, given a set of identities $IDs = \{ID_1, ID_2, \ldots, ID_n\}$, with all DUs $ID_i \in IDs$, the probability of identifying the true signer from a signature σ is negligible, i.e., at most $1/n$.

Unforgeability: The scheme satisfies unforgeability if, for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the probability of successfully tampering with a valid signature is negligible.

Traceability: The scheme satisfies traceability if, for any $ID_i \in IDs$ within any ring, the probability of identifying the signer from the signature σ is negligible for any entity without AN privileges.

Auditability: After obtaining the traceability results, AN checks and evaluates the signature behavior to ensure compliance with security and privacy policies, detect any violations, and confirm adherence to system standards.

Accountability: After identifying the signer, AN reviews the signature behavior for compliance and enforces accountability according to regulations or system specifications, ensuring system security and transparency.

4 Methodology

In this section, we present the implementation of the proposed method based on the previously described framework. This includes defining access rules, data sharing processes with a detailed signature algorithm, and outlining the accountability mechanism. Before the implementation can proceed, the following initialization steps are required.

The cryptographic system initialization proceeds as follows: define security parameter λ and prime q, then instantiate hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \to G_1$. AN deploys blockchain-based consensus with traceability smart contracts, selecting private key $y \in \mathbb{Z}_q^*$ to compute public key $S = yP$. The KGC generates master key z and public key P_{pub}, while publishing system parameters $PP = \{\lambda, G_1, G_T, P, Q, e, H_1, H_2, S, P_{\text{pub}}\}$ on-chain. Private keys remain confidential through $ABE.Enc/Dec$ algorithms. For each user ID: (1) compute partial private key $k_{ID} = z + H_1(ID)$; (2) generate secret v to derive private key $x = vk_{ID}$; (3) calculate public key $X = vY_{ID}$ where $Y_{ID} = P_{\text{pub}} + H_1(ID)P$. Following initialization, all users possess certified key pairs (x, X), enabling execution of subsequent protocols (detailed in the next 3 subsections).

4.1 Publishing Access Rules

This subsection describes the process of defining and enforcing access rules for private data sharing, which consists of two main steps:

1. **Access Rules and Policy Definition**: All DUs define and publish access control rules and data-sharing policies on the blockchain. These policies specify the conditions under which specific private data can be accessed and by whom, thereby ensuring transparency and accountability.
2. **License Issuance**: AN verifies whether a user possesses the necessary permissions to perform a given operation on the data. If the user is authorized, EN issues a license, denoted as $lic = \{\text{seq}_l, X_s, op, t_r\}$, where seq_l is the license sequence number, X_s is the user identifier, op represents the type of operation, and t_r is the timestamp of issuance. This license serves as proof of the user's access rights and enables secure data interaction.

4.2 Data Sharing

When a patient visits a doctor, medical data is generated, triggering DU (acting as both doctor and data owner) to sign the EHR. AN first verifies the license lic to authenticate DU's signing authority. Upon successful validation, DU generates the EHR and initiates the signature phase as follows: let s denote DU's serial number and n represent the ring length, then proceed with the steps outlined below:

1. Compute $HQ_s = H_2(ID_s)$, and randomly select $\phi \in \mathbb{Z}_q^*$ and compute $\Phi = \phi P$.
2. For each $j = 1, 2, \ldots, n$ with $j \neq s$, randomly select $u_j \leftarrow \mathbb{Z}_q^*$ and compute $U_j = u_j P$. Then compute $h_j = H_1(L, M, U_j, \Phi)$.
3. Compute $V = v_s Q$, $\tau = H_1(L, M, V)$, $a_s = H_1(L, \Phi)$. Finally, compute $\Theta = (\tau + a_s)Q$.
4. Randomly select $\forall T_1, T_2, \ldots, T_n \in G_1$ and compute $Y_s = v_s(S + HQ_s)$, $T_R = v_s \sum_{i=1}^{n} T_i$, and $T_s = (a_s + c_s)T_R$, where $c_s = H_1(ID_s)$. Then randomly select $\forall \delta \leftarrow \mathbb{Z}_q^*$ and compute $U_s = \delta X_s - \sum_{j \neq s} (U_j + h_j X_j)$, where $h_j = H_1(L, M, U_j, \Phi)$, $j = 1, 2, \ldots, n$. Finally, compute $W = (h_s + \delta)\Theta$ and $Z = x_s W$, where $h_s = H_1(L, M, U_s, \Phi)$.
5. The final generated signature is $\sigma = (V, U_1, U_2, \ldots, U_n, Z, \Phi, T_1, T_2, \ldots, T_n, T_s, Y_s)$.

After the signature is generated, the following operations are performed:

1. DU sends the tuple (M, σ) to AN for storage, and subsequently transmits (M, σ) to its EN, which verifies the validity of σ as follows:
   (a) Compute $\tau = H_1(L, M, V)$, $a_s = H_1(L, \Phi)$, and $\Theta = (\tau + a_s)Q$.
   (b) For each $j = 1, 2, \ldots, n$, compute $h_j = H_1(L, M, U_j, \Phi)$, and then verify the equation $e(P, Z) = e(\sum_{j=1}^{n} (U_j + h_j X_j), \Theta)$.
   (c) If the equation is valid, then the EHR content M and the signature σ are deemed valid, and the process continues. Otherwise, the result is notified to DU, and a traceability request is sent to AN.

2. EN stores M within the data center, generates a short cryptographic index $M' = \text{ABE.Enc}(M)$ for M, where M' can be decrypted by $ABE.Dec$, and uploads (M', σ) to the blockchain.

When DU sends a request to EN along with the associated EHR number, the request can pertain to two types of data sharing processes: intra-domain and cross-domain. These correspond to intra-domain data sharing and cross-domain data sharing, respectively. The permission verification and data sharing process is described as follows:

1. EN verifies whether DU possesses the necessary access rights to the requested data, as stipulated by the license lic. If DU has the appropriate privileges, the process continues to the next step. Otherwise, the process terminates.
2. EN retrieves the relevant EHR data from its local server. Let the EHR data be denoted as M and the corresponding blockchain-stored data as M'. If $M' = ABE.Enc(M)$, EN extracts the tuple (M', σ) and proceeds to the next step. If this condition is not met, EN reports a verification failure to DU and forwards an accountability request to AN, as detailed in Section 4.3. Upon successful verification, EN shares the EHR data M with DU.

This process ensures the secure and authorized sharing of medical data, safeguarding both intra-domain and cross-domain data exchanges.

4.3 Accountability Mechanism

In the event of a signature verification failure or other medical-related incidents, DU forwards a tuple (M, σ) and an accountability request to AN. AN subsequently invokes a smart contract to identify the specific signer responsible for the incident using a traceability algorithm, thereby ensuring accountability. The process is outlined as follows:

1. AN receives the tuple (M, σ), extracts the signature σ, and computes $\Gamma = \sum_{i=1}^{n} T_i$, where Γ represents the aggregation of the values T_i.
2. For each $i \in \{1, 2, \ldots, n\}$, AN performs the following computations:
   - Calculate $a_s = H_1(L, \Phi)$, and then compute $c_i = H_1(ID_i)$.
   - Determine $d_i = a_s + c_i$, and compute $\Omega = d_i Y_s$, where Y_s is a public value associated with the signer.
   - AN performs signature validation through verification of the pairing equation $e(T_s, S + HQ_i) = e(\Gamma, \Omega)$, enabling signer traceability by identifying the accountable signer s from valid cryptographic commitments.
3. Based on the outcome of the signature trace, AN generates an accountability report and forwards it to the identified signer s.

5 Security Proof

In this section, we prove the security of the 3 security properties of anonymity, non-tamperability, and traceability, using the random Oracle and CDHP assumptions. The proof results show that our scheme is secure.

5.1 Correctness of Signature

Theorem 1. *Correctness of the signature is satisfied if AN is completely trustworthy and irreplaceable by any adversary $\mathcal{A}$, KGC is able to generate the master*

key and the master public key honestly, and DU as well as EN are able to execute the process correctly.

Proof. Correctness needs to be satisfied for signature verification, as shown in the following proof process:

$$\begin{aligned} e\left(\sum_{j=1}^{n}(U_j + h_j X_j), \Theta\right) &= e\Big(\delta X_s - U_s + U_s + h_s X_s, \Theta\Big) = e((\delta + h_s)X_s, \Theta) \\ &= e(v_s(\delta + h_s)Y_s, \Theta) = e\left(v_s(\delta + h_s)(z + c_s)P, \Theta\right) \\ &= e(x_s P, W) = e(P, x_s W) = e(P, Z) \end{aligned}$$

As above, the signature correctness can be verified.

5.2 Anonymity

Theorem 2. *In our scheme, for any PPT adversary $\mathcal{A}$, the information and identities of DUs cannot be exposed.*

Proof. Assuming that DUs, EN, AN, and KGC behave honestly, they can be classified as follows. The challenger $\mathcal{C}$ initializes the system, computes the system's public parameters, and returns them to the adversary $\mathcal{A}$.

In the initialization phase, the following two scenarios need to be discussed:

1. **Master Key Generation**: The master key is a random number, and the master public key is a random point on the elliptic curve group G_1. The probability of a third party guessing the master key and the master public key is negligible.

2. **User Information Generation**: Each user's ID is represented as a randomly generated bit string, and the IDs of different doctors are unique. The probability of guessing a private key is negligible since c_i is a random number generated by the hash function, and v_i is a random number chosen by the doctor. The public key, similarly, is a random point on G_1.

During the signing phase, assume the number of ring members is n. Let V be a random point on G_1, and $U_1, U_2, \ldots, U_n$ be n randomly selected points from G_1. Let δ and a_s be random numbers generated by the signer during the signing process. Since H_1 is a hash function and Φ is a different random point on G_1, h_j is a random number for $j = 1, 2, \ldots, n$, and consequently, U_s is a random point on G_1 that reveals no signer information. In summary, various random points and numbers are generated during the signature phase, and according to CDHP, these values cannot be computed or reversed. During the EHR acquisition phase, signature verification is required; however, the signature contains no identifying information about the signer, and access to blockchain-stored data is necessary. As both the verification and recording of transactions occur on the blockchain, with transaction data and addresses encrypted, the actual identity of the data user remains undisclosed.

In conclusion, assuming the number of ring members i.e., the ring size is n, the probability that adversary $\mathcal{A}$ correctly guesses the signer within the ring

is at most $1/n$. Given that the number of ring members is typically large, the probability of $\mathcal{A}$ successfully identifying the signer can be ignored. Thus, the identities of DUs, and other participants remain securely concealed.

5.3 Non-tamperability

Theorem 3. *For any PPT adversary $\mathcal{A}$, the probability of a signature being forged is negligible if CDHP holds.*

Proof. Let the adversary be denoted as $\mathcal{A}$ and the challenger as $\mathcal{C}$. The game is defined as solving CDHP. Given a random instance $(P, P_1 = aP, P_2 = bP) \in G_1^3$, where $a, b \in \mathbb{Z}_q^*$ are unknown to $\mathcal{C}$, $\mathcal{A}$ attempts to forge a valid signature. $\mathcal{C}$ will use $\mathcal{A}$'s forgery to compute abP. Assume there exists a user's public key that satisfies the relation $X_j = aP$, where j is the index of DU.

Initialization: The challenger $\mathcal{C}$ runs the algorithm to obtain the system parameters $PP = \{\lambda, G_1, G_T, P, Q, e, H_1, H_2, S\}$ and randomly chooses a value $z \in \mathbb{Z}_q^*$ as the master key, then computes the master public key $P_{\text{pub}} = zP$.

- O_C Query: If $i \neq j$, $\mathcal{C}$ answers the value $X_i \in G_1$ of $\mathcal{A}$'s public key. Otherwise, $\mathcal{C}$ returns $X_j = aP$ to the adversary $\mathcal{A}$.
- O_R Query: $\mathcal{C}$ generates a list L_h, which is initialized to be empty. Once $\mathcal{C}$ receives a query, it verifies the existence of the tuple (ID_i, c_i, v_i, X_i) in L_h, where ID_i is the identity, $c_i = H_1(ID_i)$, and v_i is the secret value. If present, $\mathcal{C}$ responds using such a record. Otherwise, $\mathcal{C}$ queries L_h to obtain a tuple (ID_i, c_i, v_i, X_i) and performs the following:
  - If $i = j$, $\mathcal{C}$ selects a value $v_i \in \mathbb{Z}_q^*$, computes the partial public key $Y_i = P_{\text{pub}} + H_1(ID_i)P$, and finally computes the public key $X_i = v_i Y_i$. $\mathcal{C}$ then adds the tuple (ID_i, c_i, v_i, X_i) to L_h.
  - Otherwise, $\mathcal{C}$ reselects a v_i value.
- O_C Query: If $i \neq j$, $\mathcal{C}$ returns the private key x_i to user i; otherwise, it terminates with $\perp$.
- Signature O_S Query:
  - Use O_C query: If $i \neq j$, $\mathcal{C}$ runs and knows the secret value v_i. Otherwise, $\mathcal{C}$ randomly chooses $\forall v \leftarrow \mathbb{Z}_q^*$ and computes $V = vQ$, where v represents the secret value.
  - Using O_R query: $\mathcal{C}$ computes $\tau = H_1(L, M, V)$ and $a_s = H_1(L, \Phi)$. Finally, $\mathcal{C}$ computes $\Theta = (\tau + a_s)Q$. Assume $\Theta = bP$. Then $\mathcal{C}$ randomly selects $\forall u_i \leftarrow \mathbb{Z}_q^*$ and computes $U_i = u_i P$. Subsequently, $\mathcal{C}$ computes $h_i = H_1(L, M, U_i, \Phi)$.
  - If $i = j$, $\mathcal{C}$ computes $HQ_i = H_2(ID_i)$.
  - For $i = j$, $\mathcal{C}$ randomly selects $\forall \delta \leftarrow \mathbb{Z}_q^*$ and computes $U_j = \delta X_j - \sum_{i \neq j} (U_i + h_i X_i)$, where $h_j = H_1(L, M, U_j, \Phi)$ and X_j is the public key of user j, $j = 1, 2, \ldots, n$. If the tuple $(L, M, U_j, \Phi, \hat{h}_j)$ exists in the list L_s (which is initialized to be empty) and j exists such that $h_j \neq \hat{h}_j$, return the signature O_S query, otherwise proceed to the next step.
  - $\mathcal{C}$ randomly selects $\forall T_1, T_2, \ldots, T_n \in G_1$, computes $Y_j = v_j(S + HQ_j)$, $T_R = v_j \sum_{i=1}^{n} T_i$, and finally computes $T_s = (a_s + c_j)T_R$, where $c_j =$

$H_1(ID_j)$. Then $\mathcal{C}$ computes $W = (h_j + \delta)\Theta$, $Z = x_j W$, where $h_j = H_1(L, M, U_j, \Phi)$.

If $\mathcal{A}$ can effectively tamper with a valid signature

$$\sigma^* = (V^*, U_1, U_2, \ldots, U_n, Z^*, \Phi^*, T_1, T_2, \ldots, T_n, T_s^*, Y_s^*)$$

on the tuple (L^*, M^*, Φ^*), $\mathcal{A}$ forges another valid signature as follow:

$$\hat{\sigma}^* = (\hat{V}^*, U_1, U_2, \ldots, U_n, \hat{Z}^*, \hat{\Phi}^*, T_1, T_2, \ldots, T_n, \hat{T}_s^*, \hat{Y}_s^*)$$

According to the signature bifurcation theorem, the probability of the following events is non-negligible: if $i = j$, then $h_j^* \neq \hat{h}_j^*$; if $i \neq j$, then $h_i^* = \hat{h}_i^*$. Thus, the following equation is introduced:

$$e(P, Z^*) = e\left(\sum_{i=1}^{n}(U_i + h_i^* X_i), \Theta\right) \stackrel{?}{=} e(P, \hat{Z}^*) = e\left(\sum_{i=1}^{n}(U_i + \hat{h}_i^* X_i), \Theta\right) \quad (1)$$

Subtracting from equation (2) gives:

$$e(P, Z^* - \hat{Z}^*) = e(a(h_j^* - \hat{h}_j^*)P, \Theta) = e(P, a(h_j^* - \hat{h}_j^*)(\tau + a_s)Q) \quad (2)$$

Thus, $a(h_j^* - \hat{h}_j^*)bP = Z^* - \hat{Z}^*$.

This leads to $abP = (Z^* - \hat{Z}^*)(h_j^* - \hat{h}_j^*)^{-1}$.

Final Conclusion: $\mathcal{A}$ solves CDHP with non-negligible probability. Therefore, under the CDHP assumption, our scheme satisfies unforgeability.

5.4 Auditability and Accountability

Theorem 4. *Our protocol enables tracing the real signer s by AN, which exclusively retains this capability. To verify a signature's authenticity, confirming the equation $e(T', Q + HQ_s) = e(\Gamma, \Omega)$ is essential.*

Proof. Once the actual signer s is identified, the following verification steps occur:

$$\begin{aligned} e(T', Q+HQ_s) &= e((a_s+c_s)T_R, Q+HQ_s) = e(d_s T_R, Q+HQ_s) \\ &= e\left(d_s v_s \sum_{i=1}^{n} T_i, Q+HQ_s\right) = e(d_s v_s \Gamma, Q+HQ_s) \\ &= e(d_s\Gamma, v_s(Q+HQ_s)) = e(d_s\Gamma, Y') = e(\Gamma, d_s Y') = e(\Gamma, \Omega) \end{aligned}$$

As above, the signature auditability and accountability can be verified.

5.5 Other Security Features

- **Data integrity**: Maintaining data integrity is crucial. During signature verification, the equation $e(P, Z) = e(\sum_{j=1}^{n}(U_j + h_j X_j), \Theta)$ is pivotal. Any change in M alters h_j, invalidating the equation and causing verification to fail. Uploading the signature and hash index of EHR reduces the risk of tampering.
- **Legitimacy of operation**: Ensuring user legitimacy is essential for secure data sharing. Signers must be authenticated, requiring appropriate licenses.

- **On-Blockchain traceability**: Traceability within the blockchain is crucial for accountability. Transactions must execute without interference, verified through unanimous agreement among network nodes. This decentralized validation mitigates the risk of a single point of failure.
- **Immutability**: The blockchain's immutability is integral to security. Consensus among nodes confirms block validity, making tampering difficult. Any modification disrupts the blockchain's integrity.
- **Verifiability**: Blockchain transparency allows comprehensive verification of transactions. Each transaction can be traced through the chain, facilitating signature validation using the ring signature mechanism.

6 Performance Analysis

6.1 Comparison of Ring Signature Schemes

Table 2: Comparison of different ring signature schemes, and "#" represents the baseline schemes

#	Architecture	Type of Ring Signature	ECC
[3]	Cloud Server	Certificateless, Untraceable	Pairing
[35]	Blockchain	Untraceable	Pairing
[30]	Blockchain	Untraceable	No Pairing
[15]	Blockchain	Certificateless, Traceable	Pairing
[19]	Web-based	Untraceable	Pairing
Ours	Federated Blockchain	Certificateless, Traceable	Pairing

We compared the ring signature scheme proposed in this paper with several existing schemes [3, 35, 30, 15, 19], highlighting the significant advantages of our approach. Our scheme leverages a federated blockchain architecture that offers superior security and control in restricted environments compared to cloud servers [3], web-based approaches [19], or other blockchains [30]; notably, its authentication-free, traceable ring signature type excels in data traceability. While the scheme in [15] also supports traceability, our approach—combined with the federated blockchain—significantly enhances data management and auditing efficiency, ensuring data integrity and transparency, and demonstrates superior performance (see Section 6.2 for details). In comparison to ring signature schemes lacking traceability [30, 19], our scheme provides robust data traceability and auditing capabilities while maintaining privacy protection, and thus stands out in terms of architectural design, traceability, and bilinear pairing support, making it particularly suitable for applications requiring high security and strict regulatory compliance.

6.2 Experimental Design and Results

This section evaluates the computational overhead and performance of various bilinear pairing ring signature schemes, including those in [15] and [19], which involve two roles. We compare these schemes against our proposed method and also analyze blockchain performance. The key research questions explored in this section include:

– **RQ1:** How does the computational overhead of our ring signature algorithm compare to other schemes?
– **RQ2:** How does the overall performance of our ring signature algorithm compare to other schemes?
– **RQ3:** What are the blockchain platform's key performance metrics, such as CPU usage and memory?

6.2.1 Experimental Environment

To evaluate the performance of the signature algorithm, we set up a simulation environment on a Dell host running Ubuntu 22.04.3, equipped with an Intel Core i7-10700 processor and 40GB of RAM. The environment was configured with Docker 24.0.5, Docker Compose 1.29.2, and Go 1.18.8 to deploy Hyperledger Fabric v2.2, utilizing the PBFT consensus protocol. Three peer nodes were deployed within the blockchain system. Additionally, we employed Hyperledger Caliper v0.4.2 to assess the blockchain's performance.

6.2.2 Comparison of Computational Overhead (RQ1)

Table 3 compares the functionality and computational overhead of various ring signature schemes. Our proposed scheme shares similar signing and verification phases with others. The computational overhead for each phase is detailed in Table 3, with n representing the ring size. Assuming n users generate n individual signatures, the following notations represent the different cryptographic operations:

– T_h: Execution time of the hash function mapping.
– T_{mp}: Execution time of scalar multiplication on G_1.
– T_{bp}: Execution time of bilinear pairing.
– T_{pr}: Execution time of data processing, such as encryption/decryption or other operations.

Table 3: Comparison of time overhead of different algorithms

#	Signing	Verification	Tracing
[15]	$(4n-1)T_{mp} + nT_h$	$nT_{mp}+2T_{bp} +3T_{pr}+nT_h$	$(3n-1)T_{mp}+(2+2n)T_{bp}$
[19]	$(2n+2)T_{mp}+2nT_h+2T_{pr}$	$2nT_{mp}+4T_{bp}+4T_{pr}+2nT_h$	—
Ours	$(n+8)T_{mp} + (n+3)\ T_h$	$(n+1)T_{mp}+2T_{bp}$ $+T_{pr}+(n+2)T_h$	$nT_{mp}+(n+1)T_h+2nT_{bp}$

Answer to RQ1: The computational overhead of our ring signature algorithm, as shown in Table 3, is generally more efficient than [19] in terms of signing and verification time, with fewer computational operations required for these phases. Compared to [15], our scheme offers similar performance in signing and verification but introduces a lower computational overhead in tracing operations.

6.2.3 Experimental Design and Results (RQ2)

In our study, we evaluated the performance of our cryptographic framework by measuring the execution times of signature algorithms and encryption oper-

https://github.com/hyperledger/fabric
https://github.com/hyperledger/caliper

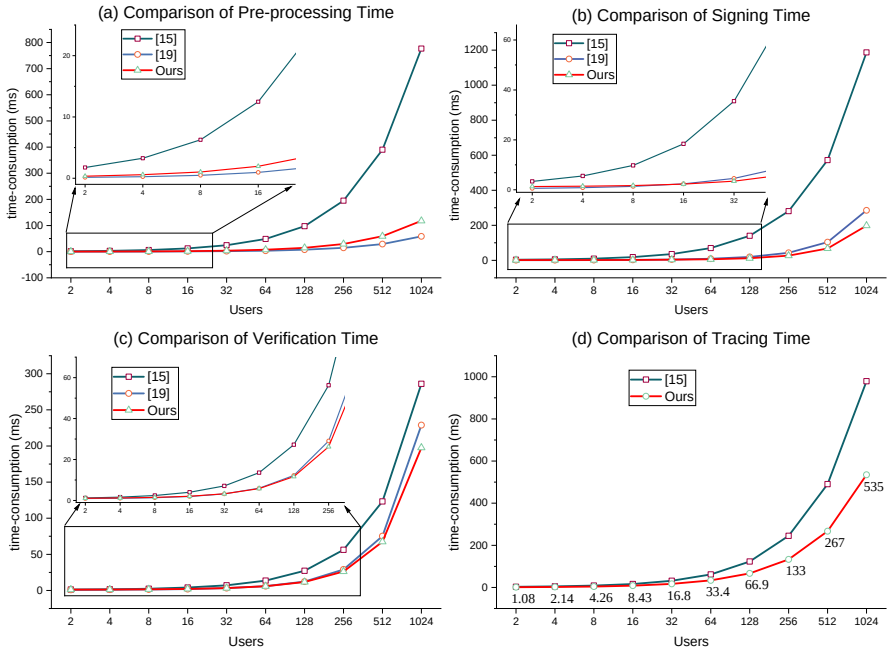

Fig. 2: Performance Comparison

ations using Go's testing framework on 1,000 anonymized 1KB medical data entries processed on the bn254 elliptic curve, which ensuring 128-bit security with a 254-bit base domain size. The results depicted in Figs. 2 show the time cost of our scheme for preprocessing in Fig. 2(a), signature generation in Fig. 2(b), verification in 2(c), and tracing in Fig. 2(d) for different numbers of users.

Answer to RQ2: The overall performance of our ring signature algorithm is significantly better than other schemes, achieving a 30% reduction in signature generation time compared to [19], an 83% decrease in time compared to [15], while also improving signature verification time by 14% and 31% compared to [15] and [19], respectively, all while introducing traceability and maintaining efficiency. Overall, our scheme effectively balances enhanced traceability with improved efficiency, also benefiting from the optimized performance of the hash function, bilinear mapping, and data processing operations.

6.2.4 Blockchain Throughput Performance (RQ3)

In the proposed blockchain system, we deployed Certificate Authority (CA) nodes within the Authority Node (AN), created a channel, and generated blocks. Entity Nodes (ENs) were deployed across two organizations, Org0 and Org1, representing different nodes, with DUs connecting to the peer0 and peer1 nodes within their respective organizations. Smart contracts were packaged and installed on the ENs, executed via a command-line script. We evaluated the per-

Data Availability Declaration: Due to privacy concerns and ethical considerations, the raw data cannot be shared publicly.

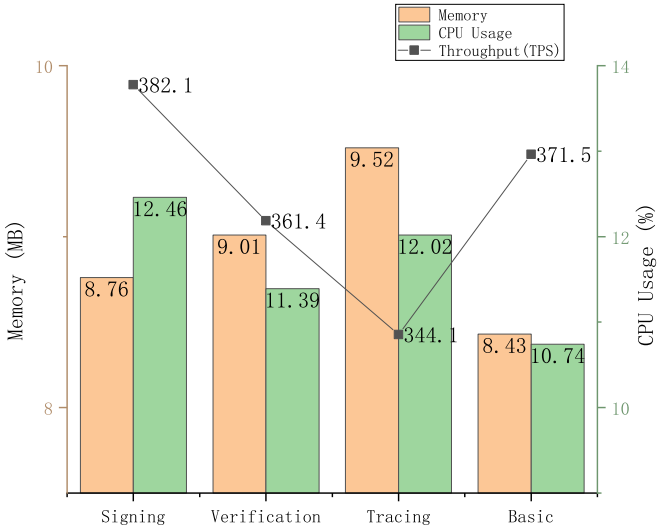

Fig. 3: Blockchain Performance

formance of the smart contracts invoked during the signing, verification, and tracing phases, comparing them to the benchmark BASIC. Test results, averaged over 100 iterations, show the throughput and resource utilization for various operations, as illustrated in Fig. 3. Signing achieves 382.1 transactions per second (TPS) with 8.76 MB memory usage and 12.46% CPU usage. Verification achieves 361.4 TPS with 9.01 MB memory usage and 11.39% CPU usage. Tracing achieves 344.1 TPS with 9.52 MB memory usage and 12.02% CPU usage. The benchmark BASIC achieves 371.5 TPS with 8.43 MB memory usage and 10.74% CPU usage.

Answer to RQ3: The key performance metrics of the blockchain platform include throughput values of 382.1 TPS for signing, 361.4 TPS for verification, and 344.1 TPS for tracing, as shown in Fig. 3. Additionally, the system achieved a transaction success rate of 100%. Importantly, the performance overhead grows little with the addition of ring signatures, demonstrating the efficiency of the proposed approach.

7 Conclusion and Outlook

This paper presents a cross-domain data sharing scheme using federated blockchain and an enhanced ring signature algorithm. Unlike traditional ring signatures, our scheme offers data integrity, conditional anonymity, traceability, auditability, and privacy protection. By incorporating authority nodes, we improve traceability and accountability while ensuring conditional anonymity. Our scheme effectively protects medical data privacy and is validated through formal analyses for suitability in medical data sharing.

8 Funding

This work is supported by the National Key R&D Program of China under Grant 2021YFF1201102.

References

1. Au, M.H., Liu, J.K., Susilo, W., Yuen, T.H.: Secure id-based linkable and revocable-iff-linked ring signature with constant-size construction. Theoretical Computer Science **469**, 1–14 (2013)
2. Bender, A., Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. In: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3. pp. 60–79. Springer (2006)
3. Bouakkaz, S., Semchedine, F.: A certificateless ring signature scheme with batch verification for applications in vanet. Journal of Information Security and Applications **55**, 102669 (2020)
4. Chandersekaran, C., Simpson, W., Trice, A.: Cross-domain solutions in an era of information sharing. In: The 1st international multi-conference on engineering and technological innovation: IMET2008, Orlando, FL. vol. 1, pp. 313–318 (2008)
5. Chen, J., Zhan, Z., He, K., Du, R., Wang, D., Liu, F.: Xauth: Efficient privacy-preserving cross-domain authentication. IEEE Transactions on Dependable and Secure Computing **19**(5), 3301–3311 (2021)
6. Chow, S.S., Lui, R.W., Hui, L.C., Yiu, S.M.: Identity based ring signature: Why, how and what next. In: Public Key Infrastructure: Second European PKI Workshop: Research and Applications, EuroPKI 2005, Canterbury, UK, June 30-July 1, 2005, Revised Selected Papers 2. pp. 144–161. Springer (2005)
7. Chow, S.S., Yap, W.S.: Certificateless ring signatures. Cryptology ePrint Archive (2007)
8. Deng, L., Li, S., Huang, H., Jiang, Y., Ning, B.: Certificateless ring signature scheme from elliptic curve group. Journal of Internet Technology **21**(3), 723–731 (2020)
9. Ezawa, Y., Kakei, S., Shiraishi, Y., Mohri, M., Morii, M.: Blockchain-based cross-domain authorization system for user-centric resource sharing. Blockchain: Research and Applications **4**(2), 100126 (2023)
10. Fan, K., Pan, Q., Wang, J., Liu, T., Li, H., Yang, Y.: Cross-domain based data sharing scheme in cooperative edge computing. In: 2018 IEEE International Conference on Edge Computing (EDGE). pp. 87–92. IEEE (2018)
11. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: International Workshop on Public Key Cryptography. pp. 181–200. Springer (2007)
12. Garjan, M.S., Kılıç, N.G.O., Cenk, M.: Supersingular isogeny-based ring signature. International Journal of Information Security Science **12**(1), 32–57 (2023)
13. Hong, H., Chen, D., Sun, Z.: A practical application of cp-abe for mobile phr system: a study on the user accountability. SpringerPlus **5**, 1–8 (2016)
14. Khatoon, A.: A blockchain-based smart contract system for healthcare management. Electronics **9**(1), 94 (2020)
15. Lai, C., Ma, Z., Guo, R., Zheng, D.: Secure medical data sharing scheme based on traceable ring signature and blockchain. Peer-to-Peer Networking and Applications **15**(3), 1562–1576 (2022)
16. Li, T., Wang, H., He, D., Yu, J.: Blockchain-based privacy-preserving and rewarding private data sharing for iot. IEEE Internet of Things Journal **9**(16), 15138–15149 (2022)
17. Li, Y., Liu, Q.: A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. Energy Reports **7**, 8176–8186 (2021)
18. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without

trapdoors. In: Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35. pp. 1–31. Springer (2016)
19. Lin, C.C., Chang, C.C., Zheng, Y.Z.: A ring signature based anonymity authentication scheme for group medical consultation. Symmetry **12**(12), 2009 (2020)
20. Liu, J., Wang, L., Yu, Y.: Improved security of a pairing-free certificateless aggregate signature in healthcare wireless medical sensor networks. IEEE Internet of Things Journal **7**(6), 5256–5266 (2020)
21. Mavrogiorgou, A., Koukos, V., Kouremenou, E., Kiourtis, A., Raikos, A., Manias, G., Kyriazis, D.: A cross-domain data marketplace for data sharing. In: Proceedings of the 2022 European Symposium on Software Engineering. pp. 72–79 (2022)
22. Millán, G.L., Pérez, M.G., Pérez, G.M., Skarmeta, A.F.G.: Pki-based trust management in inter-domain scenarios. Computers & Security **29**(2), 278–290 (2010)
23. Möser, M., Soska, K., Heilman, E., Lee, K., Heffan, H., Srivastava, S., Hogan, K., Hennessey, J., Miller, A., Narayanan, A., et al.: An empirical analysis of traceability in the monero blockchain. arXiv preprint arXiv:1704.04299 (2017)
24. Perera, M.N.S., Nakamura, T., Hashimoto, M., Yokoyama, H., Cheng, C.M., Sakurai, K.: A survey on group signatures and ring signatures: Traceability vs. anonymity. Cryptography **6**(1), 3 (2022)
25. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7. pp. 552–565. Springer (2001)
26. Russo, A., Anta, A.F., Vasco, M.I.G., Romano, S.P.: Chirotonia: A scalable and secure e-voting framework based on blockchains and linkable ring signatures. In: 2021 IEEE International Conference on Blockchain (Blockchain). pp. 417–424. IEEE (2021)
27. Scafuro, A., Zhang, B.: One-time traceable ring signatures. In: Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part II 26. pp. 481–500. Springer (2021)
28. Shen, M., Liu, H., Zhu, L., Xu, K., Yu, H., Du, X., Guizani, M.: Blockchain-assisted secure device authentication for cross-domain industrial iot. IEEE Journal on Selected Areas in Communications **38**(5), 942–954 (2020)
29. Singh, P., Masud, M., Hossain, M.S., Kaur, A.: Cross-domain secure data sharing using blockchain for industrial iot. Journal of Parallel and Distributed Computing **156**, 176–184 (2021)
30. Singh, S., Satish, D., Lakshmi, S.R.: Ring signature and improved multi-transaction mode consortium blockchain-based private information retrieval for privacy-preserving smart parking system. International Journal of Communication Systems **34**(14), e4911 (2021)
31. Sun, J., Fang, Y.: Cross-domain data sharing in distributed electronic health record systems. IEEE Transactions on Parallel and Distributed Systems **21**(6), 754–764 (2009)
32. Sun, S.F., Au, M.H., Liu, J.K., Yuen, T.H.: Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In: Computer Security–ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II 22. pp. 456–474. Springer (2017)

33. Tian, M., Zhang, Y., Zhu, Y., Wang, L., Xiang, Y.: Divrs: Data integrity verification based on ring signature in cloud storage. Computers & Security **124**, 103002 (2023)
34. Wang, S., Wang, H., Li, J., Wang, H., Chaudhry, J., Alazab, M., Song, H.: A fast cp-abe system for cyber-physical security and privacy in mobile healthcare network. IEEE Transactions on Industry Applications **56**(4), 4467–4477 (2020)
35. Wang, Z., Fan, J.: Flexible threshold ring signature in chronological order for privacy protection in edge computing. IEEE Transactions on Cloud Computing **10**(2), 1253–1261 (2020)
36. Xiao, Z., Li, Z., Liu, Y., Feng, L., Zhang, W., Lertwuthikarn, T., Goh, R.S.M.: Emrshare: A cross-organizational medical data sharing and management framework using permissioned blockchain. In: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). pp. 998–1003. IEEE (2018)