Cross-Chain Data Auditing for Medical IoT Data Sharing

Kuan Fan[1], Zhuoxuan Liu[1], Mingxi Liu[1], Yihong Wen[3], Ning Lu[1,2(✉)], and Wenbo Shi[1(✉)]

[1] School of Computer Science and Engineering, Northeastern University, Shenyang, China
{luning,shiwb}@neuq.edu.cn
[2] School of Computer Science and Technology, Xidian University, Xi'an, China
[3] The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China

Abstract. Secure medical IoT data sharing significantly improves medical collaboration and facilitates patients' medical treatment. Since block chain provides integrity and traceability management for medical data, many IoT medical data choose block chain as a storage medium. However, the isolation of block chain hinders data sharing between heterogeneous chains, so how to realize the secure sharing of medical IoT data in heterogeneous block chains and allow users to obtain correct and credible shared data is still a challenge. Existing data integrity verification techniques ensure the correctness of shared data by comparing off-chain data with metadata stored on-chain. However, these schemes ignore the consistency of shared data and the correctness of cross-chain data. This paper builds a cross-chain medical IoT data-sharing framework, introduces a relay chain, and verifies the consistency between data requests and actual storage through registration, auditing, and other methods. Based on this framework, this paper uses homomorphic signature technology and batch auditing to design a cross-chain audit protocol to verify the consistency of registered data attributes and the correctness of shared data. Security analysis and simulation experiments based on security reduction demonstrate the security and effectiveness of the proposed scheme.

Keywords: Cross-chain · Data sharing · Consistency verification · Data auditing

1 Introduction

Medical IoT data collection is becoming more extensive with the widespread use of medical sensor devices. For the data management of the Medical Internet of Things, most health systems establish data storage and query platforms based on block chain and share desensitization data with authorized users to carry out medical research or regional pathological statistics [1]. However, data users are unsatisfied with pulling data from a single block chain based shared platform because of the more comprehensive data usage needs. Considering the user's tolerance for data retrieval, if the data is retrieved

Q. Jiang et al. (Eds.): SPNCE 2022, LNICST 496, pp. 65–81, 2023.
https://doi.org/10.1007/978-3-031-30623-5_5

from different platforms, the serial time is high, and the data usage fee may be wasted due to data redundancy [2]. Cross-chain data sharing will be one of the feasible solutions.

At present, many scholars use attributes to represent medical data [3, 4]. Some medical organizations also publish various types of medical data, usually composed of attribute data [5]. Therefore, we use attributes to describe the medical Internet of Things data, as shown in Table 1. Each record contains an identification code, age, gender, blood pressure, heart rate, blood sugar, and other information. A relay chain is regarded as a data-sharing center in a cross- chain scenario to realize data sharing among heterogeneous chains [6]. After the resource chain registers the data size and attributes (age, sex, etc.) on the relay chain, the data demander sends a request to the relay chain, and the relay chain matches the registration data. The matched resource chain sends data to the data demander. The above process can realize cross-chain data sharing, but the following problems need to be solved to improve the quality of shared:

1. Data consistency: The size of the shared data provided by the resource chain is related to the revenue obtained, so the resource chain may exaggerate the data size when releasing the transmitted data. The resource chain may also provide data that is not related to the required data [7].
2. Data integrity: Due to the limited storage space of the resource chain, the resource chain may delete some data. When a user requests to delete data, the resource chain may return irrelevant data to the user as the requested attribute data [8].

Table 1. Multi-attribute medical IoT data

ID	Age	Sex	Blood pressure	Heart rate	Blood sugar
0001	35	0	120/80	70	5.0
0002	58	0	145/90	90	10.9
0003	67	1	120/80	70	5.5

Many scholars have studied data-sharing audits, which can be divided into two categories. The first type focuses on sharing data. These articles address privacy protection, integrity audit, dynamic modification of data, and other issues [9–12]. The second is about the data owner, including such issues as identity auditing, identity tracing, and key distribution [13]. The above research ensures the security and efficiency of data sharing in many aspects. However, they can not be applied to the cross-chain sharing scenario. It can not solve the problem mentioned in this paper.

This paper proposes a cross-chain audit scheme for medical multi-attribute data (CCDAS), which focuses on the data consistency and integrity problems encountered in cross-chain sharing. We introduce a relay chain to build the cross- chain data-sharing framework. We use the off-chain oracle to obtain the shared data information and audit data proof. The protocol includes registration, which ensures that the registered data scale is consistent with the actual data scale. After the audit, the data information is

written into the relay chain. In the data- sharing stage, the protocol uses different keys to encrypt the data's attributes and adopts the batch audit to ensure data integrity. Our main contributions are as follows:

1. This paper proposes a cross-chain data-sharing architecture based on the relay chain. The relay chain acts as a data-sharing center, handling data requests and sharing. The off-chain oracle is responsible for obtaining data on heterogeneous resource chains and auditing data proof.
2. This paper analyzes the data consistency and integrity in the cross-chain sharing scenario. Based on the Challenge-Response method and batch auditing, we designed the auditing protocol to ensure data consistency and integrity.
3. We provide the security analysis of the proposed protocols and demonstrate that the desired security requirements are satisfied. Performance evaluation shows that the scheme is practical.

2 Related Work

Existing cross-chain sharing architectures need to give more consideration to data security. Many scholars have studied this issue. Jiang et al. proposed a cross- chain solution to integrate multiple block chains for IoT data management. They built a consortium block chain to integrate multiple block chains for efficient and secure IoT data management [14]. Li et al. proposed a multi-domain authentication framework for cross-chain data sharing. They designed smart contracts to protect the confidentiality of the authentication data [15]. Qiao et al. proposed a dynamic autonomous cross-consortium chain mechanism in e-Healthcare. They developed a mechanism of cross-chain consensus to simplify heterogeneous node communication topology and improve node identity trustworthiness [2]. Zhao et al. proposed a secure and scalable access control model for cross-chain data sharing, which includes an access control strategy written by smart contracts and a storage system adopted by IPFS to reduce the cost of storage on the block chain [16]. Chang et al. proposed an epidemic data-sharing model based on a cross-chain mechanism. To improve the availability of sharing system, they constructed a model of epidemic data sharing between multiple consortium block chains based on a cross-chain mechanism to guarantee security [17]. Although the above solutions achieve security and efficient cross-chain data sharing, they need to conduct effective data integrity protection.

Existing data auditing schemes provide data integrity verification, but they are rarely able to verify data consistency. Zhou et al. propose an efficient certificates multi-copy integrity auditing scheme named MDSS, which entails massive overhead of certificate computation and management [18]. They improve the classic Merkle Hash Tree to achieve batch updates for multi-copy storage, which allows the communication overhead incurred for dynamics to be independent of the replica number. Xu et al. propose a block chain enabled deduplicate data auditing mechanism [19]. They first designed a client-side data deduplication scheme based on bilinear pairing technology to reduce the burden on users and service providers. They realized a reliable and efficient data auditing mechanism using block chain technology and a bilinear pairing cryptosystem. Hahn et al. propose a fast public auditing mechanism supporting dynamic verification [20]. They

study a new challenge-response protocol that significantly reduces the computational cost of TPA and increases the speed of verification of audit results. Zhang et al. propose a block chain based multi-cloud storage data auditing scheme with locating faults [21]. They present an arbitration mechanism to detect service disputes and effectively identify malicious service providers. Gao et al. propose an auditing scheme to verify the integrity of ciphertext keyword search data [22]. They study a relational authentication label (RAL) for verifying the relation in documents containing query keywords without exposing sensitive information in audit proofs. Although these protocols can realize efficient and secure data integrity auditing, they cannot verify the consistency between the requested data and the obtained data. These protocols are also not applicable to data auditing in cross-chain data sharing.

In summary, there needs to be a solution to achieve efficient auditing under cross-chain data sharing. Therefore, based on the homomorphic signature, we design an auditing protocol for cross-chain data-sharing consistency and integrity.

3 Problem Statement

In this section, we introduce system model, threat model, and Design goals.

Table 2 summarizes the notations in this paper.

Table 2. Notations described in the CCDAS.

Notation	Description
RC	Relay chain
DC	Demand chain
RSC	Resource chain
OCO	Off-chain oracle
OC	Oracle contract
HZp	Hash function in Zp
H_G	Hash function in G
M_G	Multiply function in G
P_G	Bilinear pairing operation
E_G	power operation in G

3.1 System Model

Multi-party healthcare data sharing provides health assessment services for the public. Currently, medical data from different institutions are often stored on various block chains. To improve the availability and correctness of heterogeneous chain shared data, a CCDAS system model is proposed in this paper.

In the CCDAS model, there are multiple entities: Demand Chain (DC), Resource Chain (RSC), Relay Chain (RC), Oracle contract(OC), and Off-chain Oracle(OCO).

- Demand Chain: requests the RSC for data that meets specific attributes and quantities.
- Resource Chain: owns a portion of medical data and shares this data with DC.
- Relay Chain: assists in cross-chain transactions.
- Oracle contract: includes user contracts and oracle contracts. User contracts are mainly used to receive DC requirements. The oracle contract can manage data according to the scheme's requirements and is primarily responsible for uploading the data sent by the oracle node to the chain and passing it to the user contract.
- Off-chain Oracle: runs the middle-ware of the oracle protocol, forming a distributed network outside the block chain nodes, which is responsible for collecting and verifying the data provided by the resource chain and passing it to the contract on the relay chain.

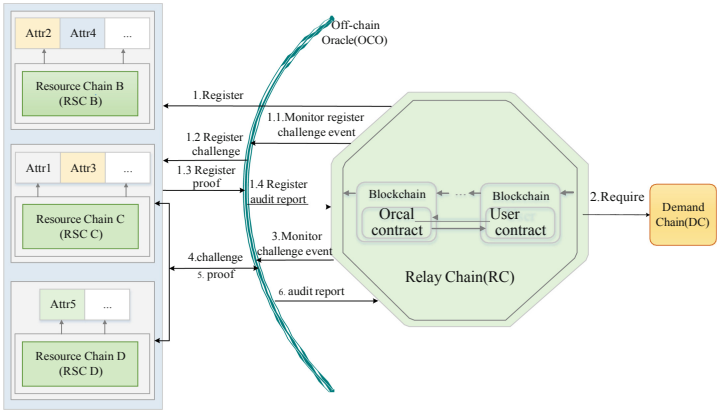

Fig. 1. System model

Combined with Fig. 1, we provide a detailed overview of the CCDAS protocol. The program is divided into two parts. The first section ensures that RSC registration data is consistent with actual data. The RSC sends registration information (data attributes, scale, and hash values) to the RC. Then, OCO monitors this event and sends a challenge message to the RSC. The RSC generates evidence and returns it to the OCO node. When the OCO node successfully verifies the registration evidence, the RC writes the registration information to the block chain. The second part ensures data integrity, preventing RSC from deleting registered source data and providing irrelevant data to DC. When the DC needs data, it sends the requirements (data attributes and size) to the RC. When the RC receives the request, it checks the registration information, finds the appropriate RSC, and generates a challenge. The requirements and challenges are then written into the user contract and the oracle contract is invoked to authorize OCO to audit the integrity of the data, including the registration attributes and data content. The OCO monitors the

OC in real time and sends requirements and challenges to the RSC under the contract. Upon receipt, the RSC generates data evidence and returns it to the OCO. OCO reviews the evidence and provides feedback to OC. If the audit is successful, RC calls OC to fetch the resource chain data and return it to the demand chain

3.2 Threat Model

We define the resource chain as a semi-honest entity. It will forge its data scale or attribute information during registration to improve the probability of sharing. At the same time, It also fakes multiple identities, using different identities to provide the same data. In addition, due to the consensus mechanism of other chains, it is necessary to ensure the consistency of cross-chain information. Therefore, the proposed CCDAS scheme should ensure data consistency and integrity and defend against the following attacks:

- Consistency attack: When dishonest RSCS register data with the relay chain, they will exaggerate the data size to increase the probability of data sharing for more benefit.
- Forgery attack: After a dishonest RSC registration is successful, part of the registration data is deleted to expand the storage space.
- Deduplication attack: Dishonest resource chains will forge multiple identities and send the same data to the DC in order to gain more profits.

3.3 Design Goals

The design goals of this scheme can be summarized as the following.

- **Registration data consistency**: The proposed scheme prevents malicious RSC from exaggerating the data size during the registration process to improve the probability of data being shared
- **Cross-chain data correctness and integrity**: This scheme prevents malicious RSC from deleting part of the registered source data.
- **Anti-data duplication**: The proposed scheme can prevent malicious RSC from forging multiple identities, providing the same data to the demand chain for more revenue.

3.4 Security Model

According to the threat model, RSC is a "semi-honest" entity. Therefore, according to the security requirements of CCDAS, the following security model is given to analyze the security of CCDAS.

Definition 3. During the audit process, it is not feasible to use forged proof to pass the integrity audit of the CCDAS protocol.

- **Initialization**: The proof unforgeable game between challenger C and adversary A is constructed. C constructs the algorithm B_A, and simulates the CCDAS environment for A. A generates proof by inquiring B_A, and B_A verifies the proof. B_A and A are the verifier and the prover respectively.

– **Query**: In this phase, A can make the hash queries and signature queries.

  (a) Hash queries: A queries B_A for the hash value of some data blocks (a_i, j, m_{ij}), and B_A calculates and sends it to A.
  (b) Signature queries: A queries the signature of the data block (a_i, j, m_{ij}), C runs the SigGen algorithm to calculate the signature of (a_i, j, m_{ij}), and sends to A.

– **Challenge**: B_{A} checks A with a random challenge $Chal_t$ consisting of some blocks that have not been queried. Aording to $Chal_t$, A generates the corresponding signatures σ_{t} and $proof_t$, and then returns them to C.
– **Forged output**: If A can forge proof based on the challenge and pass the verification of C, then A wins the game.

Definition 4. The CCDAS protocol is resistant to consistency attacks if a dishonest RSC can not pass the audit on registration.

Definition 5. The CCDAS protocol is resistant to data duplication attacks, if a dishonest RSC cannot pass the verification using the same data.

4 The Proposed CCDAS Scheme

4.1 Main Idea

The CCDAS protocol is divided into two parts. The first part mainly solves the consistency of RSC data and its registration information, which is realized by bilinear pairing and hash function [23]. After the successful audit, write the data attribute, data scale, and hash value to the RC to complete the registration. Data attributes and data scales are used for data-sharing queries. The hash value is used to assist the second part of the protocol in implementing attribute consistency audit and data content integrity audit. The second part ensures attribute consistency, data integrity, and correctness of cross-chain data through bilinear pairing and BLS signatures. But different from traditional auditing, this scheme assigns a public-private key pair to each attribute. It adopted batch auditing to complete the consistency of data attributes, registered attributes, and data content integrity. In addition, the CCDAS protocol can resist traditional audit attacks and resist the attack behavior of resource chain forging identities to provide the same data multiple times.

4.2 Construction of CCDAS

Setup: Given a security parameter λ, and a prime k, it outputs the system public parameters $params = (p, g, G, G1, H, h, e)$, where p is the large prime order of multiplicative cyclic groups G and G_1, g is the random generator of G and G_1, $e : G \times G \leftarrow G_1$ is a bilinear pairing, $H^* : 0, 1 \leftarrow G$ and $h : 0, 1^* \leftarrow Z_p^*$ are collision-resistant hash function.

KeyGen: As shown in Fig. 2, the RSC divides data into blocks according to attributes $F = \{a_1 : A_1, a_2 : A_2, \ldots, a_n : A_n\}$, where a_i is the attribute name, $A_i = \{m_{i1}, m_{i2}, \ldots, m_{ik}\}$ is the data set of a_i and n is the number of attribute. The RC randomly generates n signing key pair $(ssk_i, spk_i)_{1 \le i \le n}$, and chooses random values $a_i \leftarrow Z_p$, calculate $v_i \leftarrow g_i^{\alpha}$. The private key for attribute a_i is (a_i, ssk_i), and the public key v_i for a_i is $\left(g_i^{\alpha}, spk_i\right)$.

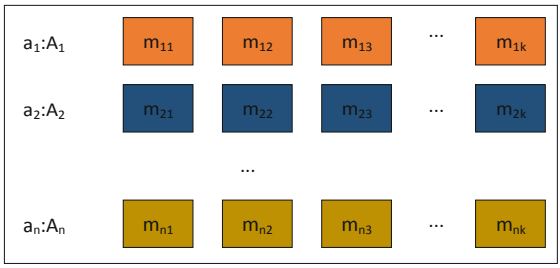

Fig. 2. Attribute data

SigGen: The RSC choose n random values $\{u_i \leftarrow G\}_{1 \le i \le n}$, and calculate the signature $\sigma_{ij} = \left(H\left(m_{ij}\right) \cdot u^{m_{ij}}\right)^{\alpha_i}$ of each data block in the A_i. The data attribute signature set is $\phi = \{\sigma_i\}_{1 \le i \le n}$, where $\sigma_i = \sum_{j=1}^{k} \sigma_{ij}$, the RSC choose random value $r \leftarrow Z_p$, calculate the lable $\tau = \left(g^r, g^{r \cdot h(F)}\right)$ of F.

RegisterChal: The RSC sends registration information $\left\{attr_i, H\left(\sum_{j=1}^{k} m_{ij}\right), size\right\}_{1 \le i \le n}$ to the RC, where $attr_i$ represents an attribute of RSC, $H\left(\sum_{j=1}^{k} m_{ij}\right)$ is the hash value of the accumulated data from $attr_i$, and k is the number of $attr_i$.After the RC receives the information, it generates $n*k$ random number $\left\{r_{ij}\right\}$ to form registration challenge *ReChal*, where n is the number of attribute from RSC. RC writes *ReChal* into the oracle contract to register "Register Challenge event". The off-chain oracle node listens to the "Register Challenge event" and sends challenge information *ReChal* to the RSC.

RegisterProof: The RSC is calculated according to the following formula:

$$H_{attr_i} = \sum_{j=1}^{k} H\left(m_{ij}\right)^{r_{ij}} \tag{1}$$

RegisterProof is $ReProof = \left\{H_{attr_i}\right\}_{1 \le i \le n}$. The RSC returns *ReProof* to the off-chain oracle node.

RegisterAudit: The off-chain oracle node calculates the verification equation according to the following formula:

$$e\left(\prod_{i=1}^{n}\left(\prod_{j=1}^{k} H\left(m_{ij}\right)^{r_{ij}}\right), g\right) \stackrel{?}{=} e\left(\prod_{i=1}^{n}\left(\prod_{j=1}^{k} H\left(m_{ij}\right),\right), \prod_{i=1}^{n}\left(\prod_{j=1}^{k} g^{r_{ij}}\right)\right) \tag{2}$$

If the equation is equal, it means that the RSC does have the number of data shown in the registration information.

ChalGen: The RC receives the data request information $Reinfor = \{attr_i, k\}_{1 \le i \le y}$ sent by the DC, where $attr_i$ represents attribute name, k is the number of the data for $attr_i$, and y is the number of required attribute. RC randomly extracts t pieces of data from each attribute data, and generates $y * t$ random value w_{ij} to generate challenge $Chal = \left(t, w_{ij}\right)_{1 \le i \le y, 1 \le j \le t}$. The RC writes the challenge into the user contract and registers it as"Challenge Event" on the oracle contract. The off-chain oracle node listens to the challenge event and sends data requirements *Reinfor* and challenge information *Chal* to the RSC according to the event requirements.

DataProofGen: The RSC receives challenge and requirement, calculates proof according to the following formula:

$$\mu_i = \sum\nolimits_{j=1}^{t} w_{ij} \cdot m_{ij} \tag{3}$$

$$\xi_i = \prod\nolimits_{j=1}^{t} \sigma_{ij}^{w_{ij}} \tag{4}$$

The attribute proof is $DataProof = (\mu_i, \xi_i)$.

DataVerify: The off-chain oracle node calculates the verification equation according to the following formula:

$$Pr_1 = e\left(\prod\nolimits_{i=1}^{y} \xi_i, g\right) \tag{5}$$

$$Pr_2 = e\left(\prod\nolimits_{i=1}^{y} \left(\prod\nolimits_{j=1}^{k} H\left(m_{ij}\right)^{w_{ij}} \cdot u_i^{\mu_i}\right), \prod\nolimits_{i=1}^{y} v_i\right) \tag{6}$$

If $Pr1 == Pr2$, it means that the attributes owned by RSC are the same as the registered attributes, and it also means that the content of the registration data has not changed; otherwise, the transaction is terminated.

In order to prevent data conflict, the off-chain oracle node randomly selects the data labels of two RSCs $\tau_1 = (g^{r}{}_1, g^{r}{}_2{}^{.h(F}{}_1{}^{)})$ and $\tau_2 = (g^{r}{}_2 , g^{r}{}_1{}^{.h(F}{}_2))$,verify according to the following equation

$$e((g^{r1}, g^{r2.h(F_1)})) \stackrel{?}{=} e(g^{r2}, g^{r1.h(F_2)}) \tag{7}$$

If the verification passes, it means that F_1 and F_2 are different from datasets, otherwise, terminate the transaction. Off-chain oracle nodes repeat verification until all RSC verification are completed. The off-chain oracle will send two audit success messages to RC.

5 Security Analysis

Theorem 1: Under the computational CDH problem, CCDAS is resistant to forgery attacks.

Proof: Suppose A can successfully forge some data blocks proofs ξ with non- negligible probability $\in$, and the forgery-proof game is won by verification. In that case, BA can solve the CDH hard problem with negligible probability [24].

Initialization: Given $\alpha \in Z_p$, $\beta \in Z_p$, g is generator of group G, α_i is the private key of attribute a_i in the RSC, g^{α_i} is the public key of α_i. Let $u = g^{\theta}$, where $\theta \in Z_p$ be the random value chosen by B_A. The input value of B_A are g^{α_i}, and g^{β}, and then B_A can solve the *CDH* problem with a non-negligible probability and output $g^{\alpha_i \beta}$.

Hash-Oracle: A queries B_A for the hash value $H(m_{ij})$ of data block m_{ij}:

If m_{ij} is in the hash list $H = \{m_{ij}, H(m_{ij})\}$, B_A obtains data $\{k_0, i, j, m_{ij}, h_{m_{ij}}\}$ from the list, and then reply A $H(m_{ij}) = h_{m_{ij}}$.

If m_{ij} is not in the hash list, B_A randomly selects a number from $k_0 = \{0, 1\}$, where $P_r[k_0 = 0] = \Theta$, random value $r_{ij} \leftarrow Z_p$. When $k_0 = 0$, B_A calculates $h_{m_{ij}} = g^{r_i j}$ When $k_0 = 1$, B_A calculates $h_{m_{ij}} = (g^{\beta})^{r_{ij}}$. B_A store $\{k_0, i, j, m_{ij}, h_{m_{ij}}\}$ into the hash list, and reply A $H(m_{ij}) = h_{m_{ij}}$.

Signature-Oracle: In order to ensure that the interaction between B_A and A is the same as the actual attack, the B_A maintains the signature list $sig = \{i, j, m_{ij}, \sigma_{ij}\}$, and responds to the request of the data block m_{ij} signature according to the signature list.

1. If the signature of m_{ij} is in the signature list, B_A obtains signature σ_{ij} from the list, and then reply A the signature.
2. If the signature of σ_{ij} is not in the signature list, B_A finds the hash list corresponding to $H(m_{ij})$. When the target entry does not exist in the hash table, the B_A queries the oracle again.

   (a) If the corresponding record is in the hash list and $k_0 = 0$, then B_A selects $H(m_{ij}) = g^{r_{ij}}$ according to the Hash-Oracle, and generates the signature as follows:

$$\begin{aligned} \sigma_{ij} &= (H(m_{ij}) \cdot u^{m_{ij}})^{\alpha_i} \\ &= (g^{\alpha_i})^{r_{ij}} \cdot g^{\alpha_i})^{\theta(m_{ij})} \end{aligned} \tag{8}$$

   B_A adds the data $\{i, j, m_{ij}, \theta_{ij}\}$ to the signature list, and sends σ_{ij} to A.

   (b) When $k_0 = 1$, the B_A refuses to respond to corresponding signature to A.

Challenge: Suppose the B_A generation challenge $chal = \{(i, j, w_{ij}), 1 \le i \le n, 1 \le j \le k\}$. There is a tuple in the *chal* that is not in the signature list.

Forged Output: A generates a legal proof $\{\mu', \xi'\}$ depend on the *chal*. According to the formula (5) and formula (6), it can be known that:

$$e\left(\prod_{i=1}^{n} \xi', g\right) = e\left(\prod_{i=1}^{n} \left(\prod_{j=1}^{k} H(m_{ij})^{w_{ij}} \cdot u_i^{\mu'}\right), \prod_{i=1}^{n} g^{\alpha_i}\right) \tag{9}$$

At the same time, A cannot request the Signature Oracle for the data $\left(i^*, j^*, w_{ij}^*\right)$ from the *chal*. This means that the hash value $h_{m_{ij}}$ of $\left(i^*, j^*, w_{ij}^*\right)$ can be found in the Hash list, and the signature list has no record for the data $\left(i^*, j^*, w_{ij}^*\right)$. B_A queries the signature list to obtain signatures for other challenge values. If the target records does not exist, then the BA queries the Hash-Oracle or Signature- Oracle. If $\left(i^*, j^*, w_{ij}^*\right)$, $k_0 = 0$, B_A rejects the hash value $H(m_{ij})$, otherwise, B_A can solve the CDH problem.

In the Hash list, $k = 0$ for the challenge value $\left(i^*, j^*, w_{ij}^*\right)$, and $k = 1$ for other challenge values, so the right side of formula (9) can be expressed as:

$$\begin{aligned} &e\left(\left(\prod_{i,j\in chal, i,j\neq i^*,j^*} \left(g^{r_{ij}}\right)^{w_{ij}}\right) \cdot \prod_{i,j\in chal} u_i^{\mu'} \cdot \left(g^{\beta r_{i^*j^*}}\right)^{g^{w_{i^*j^*}}}, \prod_{i,j\in chal} g^{\alpha^i}\right) \\ &= e\left(\left(g^{\beta\alpha^* r_{i^*j^*}}\right)^{w_{i^*j^*}}, g\right) \cdot e\left(g^{\sum_{i,j\in chal, i,j\neq i^*,j^*} r_{ij}\cdot w_{ij}\cdot\alpha_i} \cdot g^{\sum_{i,j\in chal, i,j\neq i^*,j^*} \theta_i\cdot\alpha_i\cdot\mu'}, g\right) \end{aligned} \tag{10}$$

The solution to the computational CDH hard problem is:

$$g^{\alpha_i^*\beta} = \left(\xi'^{(g^{\sum_{i,j\in chal, i,j\neq i^*,j^*} r_{ij}\cdot w_{ij}\cdot\alpha_i} \cdot (g^{\sum_{i,j\in chal, i,j\neq i^*,j^*} \theta_i\cdot\alpha_i\cdot\mu'} \cdot g)^{-1})^{-1}}\right)^{\frac{1}{w_{i^*j^*}\cdot r_{i^*j^*}}} \tag{11}$$

Probabilistic Analysis of Reduction: Analyze the probability that B_A uses A forged proof to solve a computational CDH problem, for three things:

- E_1: B_A does not reject all of A's inquiries to the signature oracle.
- E_2: A generates a legal proof according to the *chal*.
- E_3: After the E_2 event, $k_0 = 1$ for i^* in the Hash list.

If A can succeed in all of the above events, then the probability that B_A successfully solves the computational CDH problem is:

$$\begin{aligned} &Pr\left[E_1 \bigcap E_2 \bigcap E_3\right] \\ &= Pr\left[E_1\right] \cdot Pr[E_2|E_1] \cdot Pr\left[E_3|E_2 \bigcap E_1\right] \\ &= \Theta^{n_s} \epsilon_1 (1 - \Theta) \end{aligned} \tag{12}$$

$\Theta = n_s/(n_s + 1)\sqrt{a^2 + b^2}$, where n_s is the number of signatures, then the probability $Pr\left[E_1 \bigcap E_2 \bigcap E_3\right]$ is at least $\epsilon_1/\hat{e}(n_s + 1)$ where $\hat{e}$ is a natural logarithm. Since ϵ_1 is nonnegligible, B_A can solve the CDH problem with a non-negligible probability, but this contradicts the CDH difficulty problem, so CCSA can resist the forged proof attack initiated by A.

Theorem 2. In the CCDAS scheme, the data registered by the RSC to the RC must be real data.

Proof: In the registration phase, RSC sends data $(attr_i, H\left(\sum_{j=1}^{k} m_{ij}\right))_{1\leq i\leq n}$ to the RC. RC generates $n*k$ random numbers r_{ij} based on the received data and returns it to the RSC as a challenge. After the RSC receives the challenge, it calculates the proof $H_{attr_i} = \prod_{j=1}^{k} H(m_{ij})^{r_{ij}}$ and sends it to the RC. The RC inputs the generated challenge and the proof into the audit formula as parameters and uses the bilinear pairing property to verify the data scale. In the verification phase, the random value r_{ij} and the hash value $\prod_{j=1}^{k} H(m_{ij})$ generated by RC are input into the audit equation as parameters, so RSC cannot exaggerate the data size during the verification process. Even if a malicious RSC passes audit verification by falsifying data, RC retains the data hash value, which requires RSC to share the registered data.

Theorem 3. In the CCDAS protocol, RSC can not pass the verification with the same data.

Proof: The RSC generates tags $\tau = \left(g^r, g^{r.h(F)}\right)$ for file F in the SigGen algorithm. In the DataVerify algorithm, OCO randomly selects the tags $\tau_1 = \left(g_1^r, g^{r_2.h(F_1)}\right)$ and $\tau_2 = \left(g_2^r, g^{r_1.h(F_2)}\right)$ of the files F_1 and F_2, and uses the bilinear pairing technique and hash function to verify whether the two tags are equal. Since $h(F_1)$ and $h(F_2)$ are equal, the data tags are equal when the malicious RSC provides the same data. Therefore, it can prevent the malicious RSC from providing the same data and resist data deduplication attacks.

6 Performance Analysis

In this section, we first analyze the performance of CCDAS from the theoretical level, and then conduct simulation experiments.

6.1 Theoretical Analysis

We compare our CCDAS with the scheme of MHT, which has the same functions [25]. We divide the main process of CCDAS into three stages: initialization, register auditing, and data auditing. Table 3 compares the calculation costs of the MHT and the CCDAS. For initialization, the difference between the MHT and the CCDAS is about coefficient n. Although fewer multipliers for nk in the CCDAS, $E_G + M_G$ is significantly more than H_{Z_p}. For register auditing, nk in the CCDAS is less than the MHT, though $k(H_G + E_G)$ is more than the costs in the MHT. In data auditing, there are significantly fewer multipliers in yl, though introducing $l(E_G + M_G)$ and $y(4M_G + E_G)$.

Table 3. Comparison of calculation costs

Scheme	Initialization	Register auditing	Data auditing
MHT	$2nk(H_{Zp}) + E_G + nk(HG + MG + 2EG)$	$4PG + 2HG + nk(HG + 3EG + 2MG)$	$4PG + 2HG + yl(HG + 3EG + 2MG)$
CCDAS	$n(E_G + M_G) + 2E_G + h_{Zp} + nk(HG + MG + 2EG)$	$k(HG + EG) + HG + 2PG + nk(2HG + 2EG)$	$l(EG + MG) + 2PG + 4EG + yl(HG + EG + MG)$

Table 4 compares communication costs about the MHT and the CCDAS. In initialization, $n|G|$ is significantly less than $nk|G|$, so communication costs are obvious superiority. In register auditing, the main advantage of the CCDAS is less multiplier in coefficient nk. But the CCDAS has extra $n(2|Z_p| + |G|) + nk|Z_p| + |G|$. So the actual size will depend on the instantiation of the coefficients. In data auditing, it is obvious that $k(|G| + |Z_P|) + 2ty|Z_p|$ is less than $ty(2|Z_p| + |G|)$.

Table 4. Comparison of communication costs

Scheme	Initialization	Register auditing	Data auditing
MHT	$2\|Zp\| + 2\|G\| + nk\|G\|$	$nk(2\|Zp\| + \|G\|) + \|Zp\| + 2\|G\|$	$ky(2\|Zp\| + \|G\|) + \|Zp\| + 2\|G\|$
CCDAS	$3\|G\| + \|Zp\| + n\|G\|$	$n(2\|Zp\| + \|G\|) + nk\|Zp\| + \|G\|$	$k(\|G\| + \|Zp\|) + 2ky\|Zp\| + \|Zp\| + \|G\|$

6.2 Experimental Analysis

This article selects the Medical Data dataset, which contains 14 attributes [26]. We set up a simulation environment and constructed experiments to compare the practical time cost of CCDAS, MHT, and Dredas [27]. The experiments were run on a computer with a 3.50 GHz Intel i7-4710HQ CPU and 16 GB RAM. The experiment uses the pair-based cryptographic library JPBC (java encapsulation of the PBC library) [28] to implement the encryption algorithm and sets the security parameter to 256bit. For the initialization and register auditing stages, We set $n = 10$, and k goes from 100 to 1000. For the data auditing stage, we set $l = 5$, and y goes from 100 to 1000. The results of each experiment are calculated 20 times and averaged.

Figure 3 shows the computational overhead of MHT, CCDAS, and Dredas for the initialization phase. Since CCDAS and Dredas require registration time, we can see from the figure that MHT has advantages over CCDAS and Dredas. And as the number of data blocks increases, the advantage becomes larger.

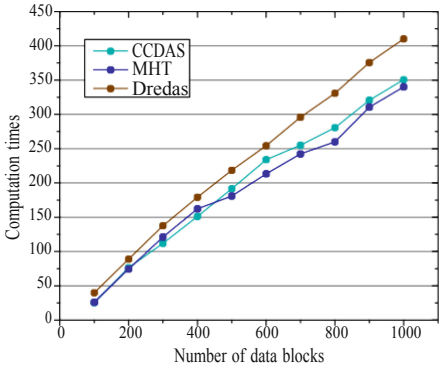

Fig. 3. Initialization time comparison

Figure 4 shows the calculation overhead of MHT, CCDAS, and Dredas for the register audit phase. We can see from the figure CCDAS has apparent advantages over MHT and Dredas. With the increase in the number of data blocks, the slope of the curve of MHT and Dredas gradually increases while the angle of CCDAS remains flat.

Figure 5 shows the computational overhead of the MHT, CCDAS, and Dredas about data auditing phases. The figure shows that CCDAS has an obvious advantage over MHT and Dredas. And with the increase in the number of data blocks, the slope of CCDAS is no longer stable, while MHT and Dredas keep a steady growth. In addition, CCDAS has a significantly lower slope than MHT and Dredas. In this stage, CCDAS combines each attribute in batches to complete the data audit, while MHT and Dredas protocols are single audit, so CCDAS has a significant advantage in computing overhead.

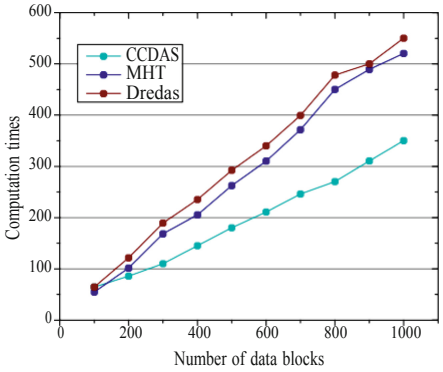

Fig. 4. Register auditing time comparison

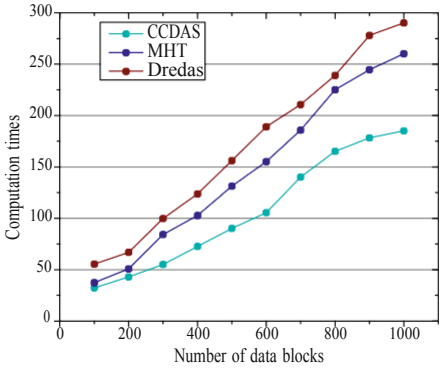

Fig. 5. Data auditing time comparison

7 Conclusion

With the increasingly extensive application of block chain data management technology and framework, cross-chain data sharing has gradually become a more common data circulation business. In this paper, we propose a cross-chain shared data audit scheme to ensure the consistency and integrity of shared data. Firstly, we propose a cross-chain data-sharing architecture based on the relay chain. Secondly, we analyze the data consistency and data. Based on the Challenge-Response method and the idea of batch auditing, we designed the audit protocol. Thirdly, the security analysis demonstrates that the desired security requirements are satisfied. Performance evaluation and comparison show that the scheme is practical. For future work, we will focus on defining and improving the usability of the proposed solution.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Nos. 62072092, 62072093, U1708262, and 62102075); the China Postdoctoral Science Foundation(No. 2019M653568); the Key Research and Development Project of Hebei Province (No. 20310702D); the Natural Science Foundation of Hebei Province (No. F2020501013); the Fundamental Research Funds for the Central Universities (No. N2023020).

References

1. Shen, M., Deng, Y., Zhu, L., et al.: Privacy-preserving image retrieval for medical IoT systems: a blockchain-based approach. IEEE Network **33**(5), 27–33 (2019)
2. Qiao, R., Luo, X.Y., Zhu, S.F., et al.: Dynamic autonomous cross consortium chain mechanism in e-healthcare. IEEE J. Biomed. Health Inform. **24**(8), 2157–2168 (2020)
3. DeBaun, M.R., et al.: American society of hematology 2020 guidelines for sickle cell disease: prevention, diagnosis, and treatment of cerebrovascular disease in children and adults. Blood Adv. **4**(8), 1554–1588 (2020)
4. Pournaghi, S.M., Bayat, M., Farjami, Y.: MedSBA: a novel and secure scheme to share medical data based on block chain technology and attribute-based encryption. J. Ambient Intell. Humaniz. Comput. **11**(11), 4613–4641 (2020)

5. Lo, B., Field, M.J.: Institute of Medicine. Conflict of interest in medical research, education, and practice (2009)
6. Wang, H., He, D., Wang, X., et al.: An electricity cross-chain platform based on sidechain relay. J. Phys. Conf. Ser. **1631**(1) 012189. IOP Publishing (2020)
7. Zhu, Z., Qi, G., Zheng, M., et al.: Block chain based consensus checking in decentralized cloud storage. Simul. Model. Pract. Theory **102**, 101987 (2020)
8. Wang, H., He, D., Yu, J., et al.: RDIC: a blockchain-based remote data integrity checking scheme for IoT in 5G networks. J. Parallel Distrib. Comput. **152**, 1–10 (2021)
9. Huang, L., Zhang, G., Yu, S., et al.: SeShare: secure cloud data sharing based on block chain and public auditing. Concurr. Comput. Pract. Exp. **31**(22), e4359 (2019)
10. Hardjono, T., Howard, G., Scace, E., et al.: Towards an Open and Scalable Music Metadata Layer (2019). arXiv preprint arXiv:1911.08278
11. Fan, K., Liu, M., Dong, G., et al.: Enhancing cloud storage security against a new replay attack with an efficient public auditing scheme. J. Supercomput. **76**(7), 4857–4883 (2020)
12. Duan, H., Du, Y., Zheng, L., et al.: Towards practical auditing of dynamic data in decentralized storage. IEEE Trans. Dependable Secure Comput. (2022)
13. Xue, J., Xu, C., Zhao, J., et al.: Identity-based public auditing for cloud storage systems against malicious auditors via block chain. Sci. China Inf. Sci. **62**(3), 1–16 (2019)
14. Jiang, Y., Wang, C., Wang, Y., et al.: A cross-chain solution to integrating multiple block chains for IoT data management. Sensors **19**(9), 2042 (2019)
15. Li, D., Yu, J., Gao, X., et al.: Research on multidomain authentication of IoT based on cross-chain technology. Secur. Commun. Netw. **2020** (2020)
16. Zhao, F., Yu, J., Yan, B.: Towards cross-chain access control model for medical data sharing. Procedia Comput. Sci. **202**, 330–335 (2022)
17. Chang, L., Yu, M., Xu, Z., Yuan, L., Bo, L.: Research on epidemic data sharing model based on cross-chain mechanism. In: Liang, Q., Wang, W., Liu, X., Na, Z., Zhang, B. (eds.) Communications, Signal Processing, and Systems. CSPS 2021. LNEE, vol. 878, pp. 424–430. Springer, Singapore (2022). https://doi.org/10.1007/978-981-19-0390-8_52
18. Zhou, L., Fu, A., Yang, G., et al.: Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics. IEEE Trans. Dependable Secure Comput. (2020)
19. Xu, Y., Zhang, C., Wang, G., et al.: A block chain-enabled deduplicatable data auditing mechanism for network storage services. IEEE Trans. Emerg. Top. Comput. **9**(3), 1421–1432 (2020)
20. Hahn, C., Kwon, H., Kim, D., et al.: Enabling fast public auditing and data dynamics in cloud services. IEEE Trans. Serv. Comput. (2020)
21. Zhang, C., Xu, Y., Hu, Y., et al.: A blockchain-based multi-cloud storage data auditing scheme to locate faults. IEEE Trans. Cloud Comput. (2021)
22. Gao, X., Yu, J., Chang, Y., et al.: Checking only when it is necessary: Enabling integrity auditing based on the keyword with sensitive information privacy for encrypted cloud data. IEEE Trans. Dependable Secure Comput. (2021)
23. Ali, I., Chen, Y., Ullah, N., et al.: Bilinear pairing-based hybrid signcryption for secure heterogeneous vehicular communications. IEEE Trans. Veh. Technol. **70**(6), 5974–5989 (2021)
24. Dupressoir, F., Zain, S.: Machine-checking unforgeability proofs for signature schemes with tight reductions to the computational diffie-hellman problem. In: 2021 IEEE 34th Computer Security Foundations Symposium (CSF), pp. 1–15. IEEE (2021)
25. Wang, C., Chow, S.S.M., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for secure cloud storage. IEEE Trans. Comput. **62**(2), 362–375 (2013)
26. Kaggle. https://www.kaggle.com/datasets

27. Fan, K., Bao, Z., Liu, M., et al.: Dredas: decentralized, reliable and efficient remote outsourced data auditing scheme with block chain smart contract for industrial IoT. Futur. Gener. Comput. Syst. **110**, 665–674 (2020)
28. JPBC Library. http://gas.dia.unisa.it/projects/jpbc/