Jiang *et al. J Surveill Secur Saf* 2020;1:61-78
DOI: 10.20517/jsss.2020.09

Journal of Surveillance, Security and Safety

Original Article

Open Access

Check for updates

Stereo storage structure assisted one-way anonymous auditing protocol in e-health system

Ling-Hong Jiang[1], Chen Wang[1], Jian Shen[1,2,3]

[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, Jiangsu, China.
[2]Cyberspace Security Research Center Peng Cheng Laboratory, Shenzhen 518000, Guangdong, China.
[3]Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450000, Henan, China.

Correspondence to: Prof. Jian Shen, Jiangsu Engineering Center of Network Monitoring, School of Computer and Software, Nanjing University of Information Science and Technology, 219 ningliu Road, Nanjing 210044, Jiangsu, China.
E-mail: s_shenjian@126.com

How to cite this article: Jiang LH, Wang C, Shen J. Stereo storage structure assisted one-way anonymous auditing protocol in e-health system. *J Surveill Secur Saf* 2020;1:61-78. http://dx.doi.org/10.20517/jsss.2020.09

Received: 9 Apr 2020 **First Decision**: 29 Jun 2020 **Revised**: 30 Jul 2020 **Accepted**: 13 Aug 2020 **Available online**: 24 Sep 2020

Academic Editor: Stefanos Gritzalis **Copy Editor**: Cai-Hong Wang **Production Editor**: Jing Yu

Abstract

Aim: With the popularity of cloud storage, data integrity has become a hot research spot. As clients' data is outsourced to the cloud, how to prevent clients' privacy from being leaked has become an urgent problem to be solved. In addition, the design of the storage structure in the cloud is also a challenge. To solve the above problem, we focus on enabling data integrity verification in the medical environment with clients' privacy protection and a novel storage structure assisted.

Methods: By leveraging the one-way anonymous key agreement and the novel stereo storage structure, a novel stereo storage structure assisted one-way anonymous auditing protocol in e-health system is proposed. First, the one-way anonymous auditing protocol can realize the adaptive anonymity of clients in the e-health system. Second, the novel stereo storage structure can implement the storage and fast search of medical data.

Results: The theoretical analyses indicate that the proposed scheme is secure under the Computational Diffie-Hellman problem and Discrete algorithm problem and it has a decent performance in computational overhead. Besides, the simulation results demonstrate that the computational cost of the user is constant.

Conclusion: To protect the user's private information in e-health system, we propose a stereo storage structure assisted one-way anonymous auditing protocol in this paper. In the proposed scheme, fast searching of data,

the one-way anonymity and the data auditing with mutual supervision are supported, which is necessary for the patients and the medical personnel in a real e-health scenario.

Keywords: Data integrity, stereo storage structure, one-way anonymous

1 INTRODUCTION

The development of society is inseparable from the advancement of science and technology. As the huge development of the internet industry and the rise of data applications such as artificial intelligence and big data spread throughout people's lives, people begin to generally realize the importance of data. The world is gradually stepping into the era of data dominance, and all walks of life are generating data all the time. Due to the sheer amount of data that is ever increasing, traditional storage methods cannot meet the needs of the people anymore which leads to the emergence of cloud storage.

Cloud platform provides individuals and organizations with powerful storage services which brings great benefits as follows: (1) users only need to pay for the actual storage without worrying about insufficient storage resources; (2) with data hosted to cloud platform, local data will no longer be stored, which can reduce the purchase cost and energy consumption cost of storage equipment; (3) the maintenance of data storage can be left to cloud service providers (CSP) to save the cost of maintaining large amounts of data for users; (4) cloud data can form linkage with local data to make redundant backup to each other; and (5) users can easily access the data in the cloud through web interface or application.

While cloud storage has many advantages, there are also some security threats[1,2]. On the one hand, the cloud infrastructure may suffer some inevitable hardware or software failures or external attacks that lead to data corruption could occur, but cloud server providers could deliberately hide the fact of data corruption for the sake of their business reputation. On the other hand, the outsourced data stored in the cloud might suffer from illegal behaviors from CSP for commercial benefits. What is more, the outsourcing of data results in users no longer physically owning the data, so users cannot even verify whether their data is complete, available, or secure.

Therefore, how to guarantee the data integrity and the privacy of users on the cloud server has become a key issue for cloud storage services. More data security issues are increasingly prominent such as whether the user data is securely stored or whether the user privacy is leaked and so on[3-5]. When it comes to the electronic heath system, a physician records the information of patients' medical treatment electronically through electronic health records which involves the collection, quality control, transmission, storage, statistics, and utilization of patient information. Obviously, it is difficult for traditional storage methods to screen and retrieve typical health records for medical statistics and scientific research. An electronic health system can not only retrieve all kinds of medical records quickly, but also make the previously laborious process of obtaining medical statistics become very simple and fast, providing first-hand data for scientific research and teaching. Nevertheless, such information often contains confidential and sensitive information, and the disclosure or falsification of such information may damage the reputation and property of patients.

In order to address these issues, considerable efforts have been made. Among existing proposals, great amounts of cloud data integrity auditing schemes based on privacy protection have been proposed[6-9]. To verify whether the outsourced data remains intact, file owners or auditors can challenge the cloud server with low communication overheads and computation costs.

Motivation of this paper: Medical data include patients' information such as admissions, discharges, transfers, e-health system patient records, diagnoses, treatments, medical images, economic/financial data, and so on. The quality, confidentiality, and integrity of medical data will affect the real-time, short-term, and long-term performance of the application. First, it will directly affect the daily management and treatment of patients. Second, the application of software and systems for obtaining information and decision support may be affected. Third, there are unknown impacts data storage failure may cause on medical research which can lead to irreparable consequences. At present, researchers have designed many protection schemes for data in the cloud. However, there is no complete data protection scheme specially designed for medical data.

1.1 Our contributions

To solve the above security protection problem of cloud medical data, this paper designs a one-way anonymous auditing protocol in the e-health system. The contributions of this paper can be summarized as follows.

1.1 1 A novel stereo storage structure is proposed to assist the auditing protocol in the e-health system

As stated above, medical data consists of a variety of data types. Therefore, we propose a novel data storage structure to store medical data, which can achieve fast search of data. In addition, the design of this structure saves the storage overhead of index tables.

1.1.2 A one-way anonymous e-health system model is presented

In view of the current status of the medical environment, for better protection of the privacy of patients, we propose an e-health system model that supports one-way anonymity, which means patients in this system model can keep their identities anonymous. Simultaneously, medical personnel identity information is disclosed in the system, so that patients can find the responsible person when a medical accident occurs.

1.1.3 An auditing protocol aiming to support both physician and patient validation is provided

This scheme innovatively enables patients and attending physicians to independently verify the integrity of their commonly relevant medical data. In other words, both patients and their attending physicians can verify whether medical data file in the cloud is correct and complete. In addition, it can promote information exchange and mutual supervision between physicians and patients.

1.2 Related works

In the past few years, data integrity in the cloud has received much attention as a core security issue. Hereafter, abundant security models and data protection schemes have been proposed by researchers around the world to solve the integrity audit problem of outsourced data[10,11]. In 2003, Deswarte *et al.*[12] first put forward the theoretical model of remote verification of data integrity of untrusted servers based on the Diffie-Hellman key agreement protocol. The proposed model consists of only two entities, the user and the cloud server provider. The user can directly initiate data integrity verification to the cloud service provider, laying a foundation for the subsequent cloud data auditing protocol. At that time, cloud storage was not yet widespread, and only a few users outsourced a small amount of data on remote servers, so that the protocol did not take into account a situation where a large community of users are storing a great deal of data on cloud servers which we see today. Once the data stored by the user on the remote server is too large, the computing overhead on the user side cannot be borne by ordinary computers, and the protocol cannot work normally. Thus, to solve that problem, a third-party auditor entity is introduced to validate the integrity of the outsourced data in the cloud.

With a growing number of users using the storage service on the cloud, cloud data auditing protocols are rapidly being developed, and many scholars are proposing plentiful valuable solutions. In 2007, Ateniese *et al.*[13]

firstly put forward a notion of Provable Data Possession to confirm the outsourced data possession on the untrusted cloud, which is based on RSA homomorphic linear verification and supports third-party public auditing. However, the dynamic update of data is not supported in this scheme, and this scheme cannot protect users' privacy. In the same year, Juels *et al.*[14] proposed a model named Proof of Retrievability, as well as presented a practical scheme which supports the integrity verification of data and the recovery of damaged data. Nevertheless, this scheme has a limited number of times to verify data integrity and does not support dynamic auditing or batch auditing. Since then, to solve the aforementioned problems, many scholars have devoted themselves to making improvements based on these two schemes, and they have made great progress in supporting more performance such as batch auditing, operating efficiency, and dynamic data update. Nevertheless, few people paid attention to the problem that these schemes leak users' private data to third-party auditors in the process of auditing. In 2010, Wang *et al.*[15] first proposed an auditing scheme that can be publicly verified to support user privacy protection. This scheme is based on public key homomorphic label technology so that the auditor can perform auditing without obtaining all the data of the user which greatly increases the operating efficiency of the system. The scheme also uses a random masking technique which makes it impossible for third party auditors to obtain users' private information through the verification returned by cloud service providers. In addition, the auditing protocol supports dynamic update of data, batch auditing, and multiple auditing tasks that can be performed simultaneously. It was later confirmed that there were still security risks. Therefore, in 2011, Wang *et al.*[16] improved the system for the security but caused a huge computing burden on the cloud server, greatly reducing the efficiency of system operation. In terms of this problem, in 2015, Worku *et al.*[17] increased the efficiency of system operation while ensuring data security, but unfortunately, it did not support dynamic data operations.

Besides storage data, users would like to perform updates to outsourced data directly in the cloud. Based on this, Wang *et al.*[18] proposed a relatively complete protocol which can support data update, user privacy protection, and batch auditing, but it will lead to the problem of high computing cost on the client side. Then, Garg *et al.*[19] designed a protocol that can minimize the computational complexity for the client during the system setup phase, which is publicly verifiable and supports dynamic operations on data.

After that, many multi-user modification and user revocation schemes have been proposed[20-23]. However, the above scheme cannot solve the problem of data redundancy well. To solve that problem, Wu *et al.*[24], Daniel and Vasanthi[25] removed redundant data from the cloud server which saved the storage cost of cloud service providers and greatly improved the efficiency of data validation. However, none of the above schemes have been designed specifically for images stored on the cloud, thus Tang *et al.*[26] proposed an efficient real-time integrity auditing protocol specially designed for cloud images, which also supported fair arbitration. In 2019, based on a new primitive fuzzy identity, Zhao *et al.*[27] presented a dynamic auditing protocol for the integrity verification of big data. This scheme applies fuzzy identity to the integrity verification of big data for the first time.

However, the above existing solutions cannot be well applied to the e-health systems due to the special relationship between medical staff and patients, and the particularity of medical data. Therefore, we explore a novel storage structure for storing medical data for the e-health system and design a one-way anonymous auditing protocol in this paper.

1.3 Organization

The rest of this paper consists of the following parts: We first introduce the preliminaries in Section 2, mainly including some definitions and basic properties about bilinear pairing and one-way anonymous key agreement required for this paper. Then, we describe the system architecture that contains the proposed system model, system components, and stereo storage structure in Section 3. In Section 4, we formalize

the security model of the proposed one-way anonymous auditing protocol. In Section 5, a detailed description of the proposed scheme is demonstrated. After that, a security analysis is presented in Section 6. In addition, performance analysis of our stereo storage structure assisted one-way anonymous auditing protocol in e-health system is given in Section 7. Finally, Section 8 concludes the findings of the paper.

2 PRELIMINARIES

Necessary preliminaries mainly including some definitions and basic properties about bilinear pairing and one-way anonymous key agreement required for this paper are introduced in this section.

2.1 Bilinear pairing

Let G_1 and G_2 be two groups of the same prime order q. Let G_1 be an additive group, and let G_2 be a multiplicative group. A mapping e on (G_1, G_2): $G_1^2 \rightarrow G_2$ satisfying the following properties is named a cryptographic bilinear map[28].

2.1.1 Bilinearity

$e(aP,bQ) = e(P,Q)^{ab}$ for all $P,Q \in \mathsf{G}_1$ and $a,b \in Z_q^*$. This can be expressed in the following manner. For $P,Q,R \in \mathsf{G}_1$, $e(P+Q,R) = e(P,R)e(Q,R)$ and $e(P-Q,R) = e(P,R)e(Q,R)^{-1}$.

2.1.2 Non-degeneracy

If P is a generator of G_1, then $e(P, P)$ is a generator of G_2. That is to say, $e(P, P) \neq 1$.

2.1.3 Computability

e is efficiently computable.

2.2 One-way anonymous key agreement

One-way anonymous key agreement was proposed by Kate *et al.*[29]. Suppose Alice ID_A and Bob ID_B are clients of the same key generation center, whose master secret is s and $d_i = s \cdot H(ID_i)$ for clients with their identity ID. Then, clients can compute a shared key by using their own privacy key and the identity ID of the other participant. What is more, suppose Alice wants to remain anonymous with Bob. Hereafter, the key agreement protocol process can be roughly divided into the following two parts: (1) first, Alice computes $Q_A = H(ID_A)$ and $Q_B = H(ID_B)$. Finally, randomly chooses an integer $r_A \in_R Z_q^*$, computes $P_A = r_A \cdot Q_A$ as Alice's pseudonym and sends it to Bob; (2) after received Alice's pseudonym, Bob computes $K_{AB} = e(P_A, d_B)$. Then, Alice and Bob have the same shared key $K_{AB} = e(d_A, Q_B) = e(Q_A, Q_B)^{r_A \cdot s} = e(P_A, d_B)$.

3 SYSTEM MODEL AND DATA STRUCTURE

3.1 System model

Stereo storage structure assisted one-way anonymous auditing scheme in e-health system involves four entities: key generation center, users, the third-party auditor, and cloud server. Figure 1 illustrates the relationship between those four entities.

3.1.1 User

In our model, patients and physicians are considered as the two main electronic health system (EHS)-related personnel types. For instance, when a patient seeks a diagnosis through interview by a physician in EHS, the patient needs to inform the physician of his or her own information at first. To realize the privacy protection of the patient's identity, our scheme will set up a false name for the patient based on the patient's identity ID to interact with the physician. A physician needs to generate patients' electronic health records (EHRs), which contains basic information about the physician and the patient as well as the patient's medical data, and upload it to the cloud. Although physicians and patients are two different entities, their

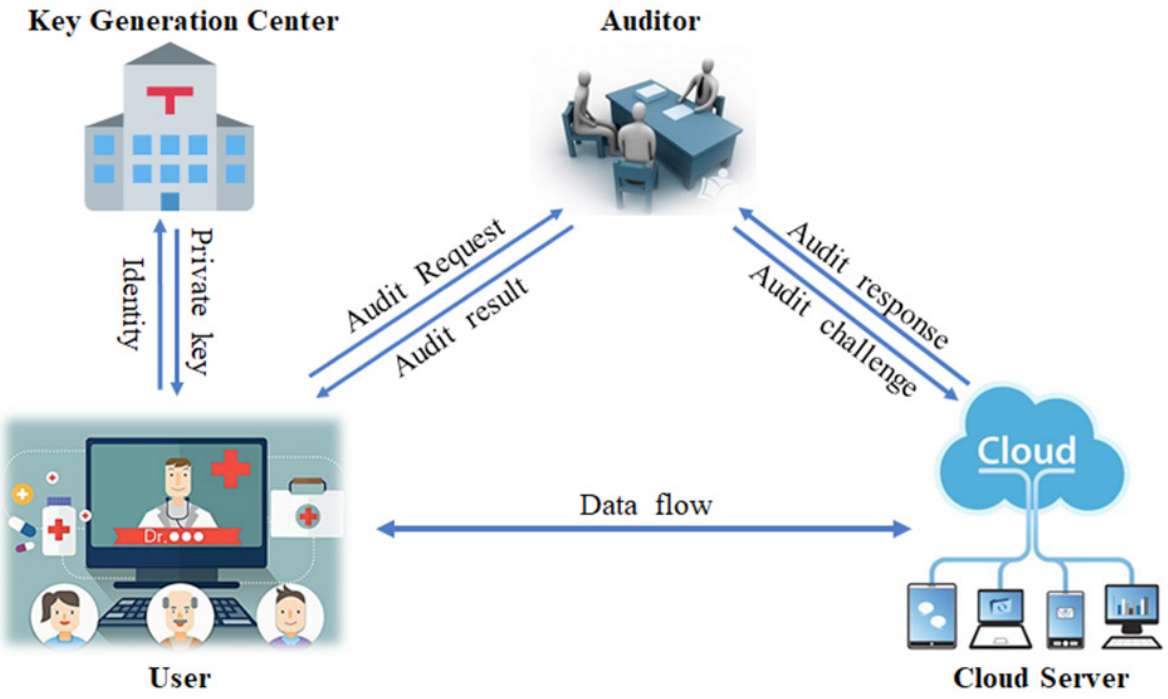

Figure 1. The proposed system model

functional needs for data in the EHS are similar. Therefore, we consider the physicians and the patient as one object in this system. As user, both physicians and patients can access the relevant EHRs and validate the integrity of their data by authorizing the TPA.

3.1.2 Key generation center

The key generation center is a trusted party in e-health system responsible for setting system parameters and generating the corresponding privacy key based on the client's identity and distributing it to the user.

3.1.3 Cloud server

It is supposed that the cloud server is a terminal that provides unlimited computing and storage capacity. Users can upload data through the cloud storage service and share it with other users. During the data integrity auditing process, Cloud server (CS) can respond to the challenges that users delegate to third-party auditor (TPA).

3.1.4 The TPA

TPA is a public verifier, which is assumed to be a terminal with unlimited computing and storage capability. TPA provides data auditing services and is entrusted by users to verify the integrity of cloud data.

3.2 System components

Stereo storage structure assisted one-way anonymous auditing protocol in e-health system consists of the following four algorithms: *Setup, KeyGen, Extract,* and *Audit*. Specifically, these algorithms are described as follows:

$Setup(1^{\kappa}) \rightarrow (para, msk)$: On input 1^{κ} where κ is a security parameter, the system setup algorithm, which is a probabilistic algorithm run by the Key generation center (KGC), generates the public parameter *PP* for the system and a master secret key *msk* for the KGC itself.

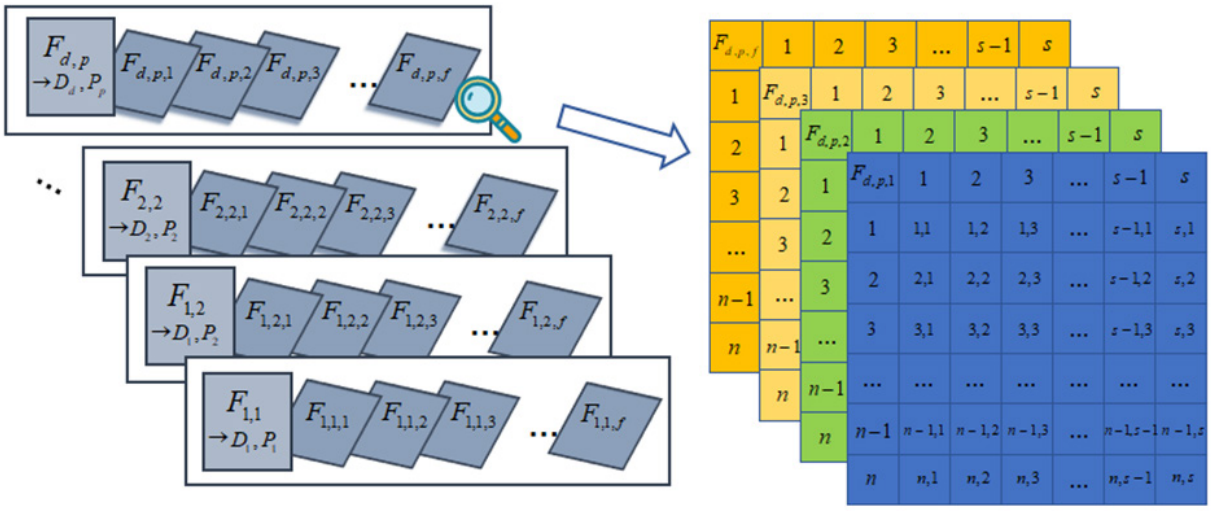

Figure 2. The presented data structure model

$KeyGen(PP,\ msk\ ,\ ID_A,\ ID_B) \rightarrow (d_A, d_B, K_{AB}, KAB)$: This algorithm is a probabilistic algorithm implemented by KGC. The public parameter *PP*, the master key *msk*, and patient's identity ID_A along with physician's identity ID_B are the inputs, and *KeyGen* generates a private key d_A for patient A and a private key d_B for physician B. This algorithm outputs a session key K_{AB} and secret key KAB for auditing.

$Extract(PP, F, KAB) \rightarrow (F^*, \tau, \{\sigma_i\}_{i\in[1,n]})$: This algorithm is a probabilistic algorithm run by a user. The user is given system parameters *PP*, key KAB, file *F* and its file name. It outputs a verifiable file tag τ, a set of block authenticators $\{\sigma_i\}_{i\in[1,n]}$ of the processed file blocks $\{\chi_i\}_{i\in[1,n]}$.

$Audit(PP, \tau) \rightarrow \{0,1\}$: This algorithm is a probabilistic algorithm jointly run by the auditor and cloud server. It outputs 1 to indicate all of the data block can be verified to be original and integrated by τ.

3.3 Stereo storage structure

The novel stereo storage structure proposed in this paper is aimed to realize fast retrieval and query of data and assist the auditing protocol in the e-health system. As is shown in Figure 2, a three-dimensional storage structure is designed to store mass amounts of medical data from the users. Specifically, each plane of the three-dimensional structure on the left part of the figure contains a header file and a series of *f* diagnosis and treatment files of a certain physician corresponding to a certain patient. The header file contains the identity information of the physician and the patient, which is convenient for quick search of the file. Here, $1 \le f \le \mathcal{N}$, and $\mathcal{N}$ is the upper limit of file number of each plane in the stereo storage structure. And those medical files contained in one plane can be generated, shared with, and verified for integrity by both of the specific physician and the patient. In other words, all diagnosis and treatment files of a physician D_d for one of his/her patients P_p are stored in the same plane. For example, $F_{1,1,f}$ represents the *f*-th files of the physician D_1 and the patient P_1, and $F_{1,2,f}$ represents the *f*-th files of the physician D_1 and the patient P_2. In the same way, the patient P_2 can also consult with the physician D_2, during which a series of files will be generated. In this e-health system, we suppose the user set contains a set of physician *D* and a set of patient *P*, and the index of the physician and patient is *d* and *p*, respectively. Here, the *f*-th files of the physician D_d and the patient P_p is denoted as $F_{d,p,f}$, and the header file of this series of files in the same plane is represented as $F_{d,p}$. In addition, the *f* files corresponding to one of the planes are shown on the right in the figure, which together form a smaller three-dimensional storage structure. Each plane in the right picture represents a file. In order to better process the file data, we uniformly divide each file into *n* blocks and each block

comprises *s* sectors. Each file and each plane of the stereo storage structure stores data as follows.

$$F_{d,p,f} = \{\chi_{x,y}\}_{1\le x\le n, 1\le y\le s}$$

$$F_{d,p} = \{F_{d,p,f}\}_{d\in D, p\in P, 1\le f\le N}$$

Furthermore, there are a warrant list of corresponding files in the header file of each plane in the structure for the auditing of the log information, which include the file origin, file type, and consistency of outsourced files. Based on this stereo storage structure, we can quickly search any user's file and the corresponding data block fragments to assist one-way anonymous auditing protocol. Additionally, dynamic data updates are an important part of the auditing schemes. However, due to the particularity of medical data, changes in the data may cause irreversible effects on the medical data. Therefore, dynamic data updates in this paper need both patients' and their attending physicians' authorization; however, those updates will not change the division of the original file.

4 SECURITY MODEL

The following security model of the stereo storage structure assisted one-way anonymous auditing scheme is proposed by designing a series of games between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. Taking into account in our security model the fact that the cloud server may modify or remove the data in the cloud due to software and hardware failure or man-made destruction, we view the untrusted cloud server as an adversary $\mathcal{A}$ and the user as a challenger $\mathcal{C}$. The formalized security model of the game is as follows:

(1) Setup. Once security parameter κ is inputted in the system, the challenger $\mathcal{C}$ runs the system Setup algorithm, and generates the system public parameter *PP* and a master secret key *msk*. Then, the challenger $\mathcal{C}$ sends the system public parameters *PP* to $\mathcal{A}$.

(2) Query. In this process, $\mathcal{A}$ can spontaneously issue the following two queries to $\mathcal{C}$:
KeyGen Queries: At first, $\mathcal{A}$ queries the secret key for the patient P_A and physician Q_B. Then, $\mathcal{C}$ runs the KeyGen algorithm in the system to generate a secret key *KAB* and sends the secret key to $\mathcal{A}$.
Extract Queries: Then, in these queries, $\mathcal{A}$ adaptively make queries of the signatures for the file M. After $\mathcal{C}$ runs the KeyGen algorithm and gets the secret key, $\mathcal{C}$ runs the Extract algorithm to generate the signatures of the file M. Next, $\mathcal{C}$ sends the signatures of the file M to $\mathcal{A}$.

(3) Challenge. In this phase, $\mathcal{A}$ plays the role of a prover to yield a valid proof and $\mathcal{C}$ acts as a verifier to check out the correctness of the proof. The challenger $\mathcal{C}$ samples a series of random numbers and sends the challenge $chal = \{i, s_i\}_{i\in I}$ to $\mathcal{A}$.

(4) Output. Once receiving the challenge from the challenger $\mathcal{C}$, the adversary $\mathcal{A}$ generates corresponding proof *P* and feedback to $\mathcal{C}$. If this proof *P* can be verified by $\mathcal{C}$ with a non-negligible probability, that is to say, this game ends and $\mathcal{A}$ ultimately successful in the game above.

5 OUR PROPOSED SCHEME

Our proposed scheme is demonstrated in four phases in this section. Firstly, in the system setup phase, the KGC sets the system public parameters and a master secret key. Secondly, the KGC generates privacy keys for users and secret keys for auditing in the registration phase. Next, in the storage phase, users upload and update files to the cloud along with file warrants, authenticators, and tags. Finally, in the integrity verification phase, TPA is entrusted by the data owner to verify corresponding data integrity. Note that for simplicity, some primary notations used throughout the paper are summarized in Table 1. Moreover, the scheme is described in detail as follows:

Table 1. Main notations in the proposed scheme

Notation	Description
H_1, H_2, H_3, H_4	Four hash functions
msk	The master secret key
d_i	The secret key of user i
P_i	The pseudonym of user i
K_{AB}	The session key of user A and B
KAB	The auditing secret key of user A and B
$\tau, \{\sigma_i\}_{i\in[1,n]}$	The file tag and set of block authenticators
Λ, V_N, T_N	The warrant, version number, and time stamp of outsourced files
χ_{ij}	The i-th block j-th sector data of file

5.1 System setup: Setup

Once taking a security parameter κ as input, the KGC randomly selects two multiplicative cyclic groups G and G_T with prime order q, where g is a generator of G. e: $G \times G \longrightarrow G_T$ denotes a bilinear map. After that, the KGC picks an integer $a \in_R Z_q^*$ at random and computes $g_1 = g^a$ where $g \in G$.

Next, $v_0, v_1, \cdots, v_\ell, u_1, \cdots, u_s \in_R G$ are uniformly chosen at random. Four collision-resistant hash functions are chosen as follows: $H_1, H_2, H_4 : \{0,1\}^* \to G$ and $H_3 : \{0,1\}^* \to \{0,1\}^\ell$. So, the system public parameter is $PP = (g, g_1, g_2, v_0, v_1, \cdots, v_\ell, u_1, \cdots, u_s \in_R G, H_1, H_2, H_3, H_4)$. Finally, the master secret key msk is set as $msk = g_2^a$ with $g_2 \in G$ and keeps the msk in secret by the KGC.

5.2 Registration: *KeyGen*

The KGC runs the *KeyGen* algorithm to yield a shared secret key for users with the msk and public parameter PP. The registration procedure consists of two phases: *PrivacyKeyGen* and *SecretKeyGen*.

(1) *PrivacyKeyGen*: First, the KGC generates and distributes the corresponding private key for every user who may be a patient or a consultant in e-healthy system. In detail, the KGC computes Q_i based on user's identity as $Q_i = H_1(ID_i)$. Then, KGC calculates user privacy key as:

$$d_i = g_2^a \cdot H_1(ID_i) \quad (1)$$

For example, KGC independently yields a private key d_A for patient A, and a private key d_B for the attending physician B. Then, the KGC sends d_i to ID_i. After receiving the d_i, user validates ID_i by calculating:

$$e(d_i, g) \stackrel{?}{=} e(g_2, g_1) \cdot e(H_1(ID_i), g) \quad (2)$$

If the above equation is true, the user ID_i adopts the private key d_i; otherwise, the KGC fails to generate a valid privacy key.

(2) *SecretKeyGen*: To protect the identity of patient A, patient A randomly chooses a number $r_A \in_R Z_q^*$, creates a pseudonym $P_A = r_A \cdot Q_A$, and sends it instead of his or her actual identity to B. Then, A and B can calculate a session key K_{AB}, and this algorithm produces a secret key KAB for auditing. The specific algorithm is as follows:

$$\begin{aligned} K_{AB} &= e(d_A, Q_B) = e(P_A, d_B) \\ KAB &= g_2^a \cdot H_2(K_{AB}) \end{aligned} \quad (3)$$

5.3 Storage: *Extract*

The storage procedure contains the following three phases: *WarrantGen*, *AuthenticatorGen*, and *TagGen*.

(1) *WarrantGen*: When user uploads or updates a new medical data, the corresponding file information will be updated. For confirming some additional information about the source, type, and consistency of the files

outsourced to the cloud, the user generates a warrant Λ which includes the pseudonym of A, the identity hash value Q_i of attending physician B, and medical file information such as file type *filetype*, version number V_N, time stamp T_N, etc. For example, $\Lambda = P_A \,||Q_B\,||V_N\,||T_N\,||filetype$. Here, the N denotes the index of different medical files. Then, the following is calculated:

$$\vec{\Lambda} = (\zeta_1, \cdots, \zeta_\ell) \leftarrow H_3(\Lambda) \tag{4}$$

The patient A picks a random number $t_\Lambda \in_R Z_q^*$, and generates an authorization:

$$\delta_\Lambda = (KAB \cdot (v_0 \cdot \prod_{j=1}^{\ell} v_j^{\zeta_j})^{t_\Lambda}, g^{t_\Lambda}) \tag{5}$$

Finally, the patient A sends the warrant pair $(\Lambda, \delta_\Lambda) = (\Lambda, (\alpha, \beta))$ to attending physician B. Then, the attending physician B validates the warrant pair by calculating:

$$e(\alpha, g) \stackrel{?}{=} e(g_2, g_1) \cdot e(H_2(K_{AB}), g) \cdot e(v_0 \prod_{j=1}^{\ell} v_j^{\zeta_j}, \beta) \tag{6}$$

If the above equation is true, the attending physician B accepts the authorization δ_Λ; otherwise, the patient A fails to generate a valid warrant.

(2) *AuthenticatorGen*: Given a medical file F to be outsourced, the user first splits F into n blocks, and each contains s sectors: $F \rightarrow \{\chi_{i,j}\}_{n \times s}$, where $\chi_{i,j} \in_R Z_q^*$. For each file F, choose a random number $t_\vartheta \in_R Z_q^*$, and for the i-th block, yield a block authenticator as follows:

$$\sigma_i = KAB \cdot (H_4(\Lambda \,||FID\,||i) \cdot \prod_{j=1}^{s} u_j^{\chi_{i,j}})^{t_\vartheta} \tag{7}$$

(3) *TagGen*: A random name *FID* is chosen for a file from Z_q^*, and s random elements $u_1, \cdots, u_s \in G$. Set $\tau_0 = \Lambda \,||FID\,||n\,||u_1\,||\,...\,||u_s\,||g^{t_\Lambda}\,||g^{t_\vartheta}$. Then, the user generates file tag τ based on τ_0 and K_{AB} to guarantee the integrity of each distinct file information.

$$\tau = \tau_0 \,||S.Sign(\tau_0)_{K_{AB}} \tag{8}$$

Hereafter, the user sends the file tag τ to the TPA. Besides, $KP = e(H_2(K_{AB}), g)$ can be pre-computed and sent to TPA. In addition, the processed file F^* that comprises F, FID, Λ, δ_Λ, and σ_i is uploaded to the CS and can be stored in the proposed stereo storage structure and removed from the user's local side.

5.4 Integrity verification: *Audit*

The auditing procedure contains following three phases: *Challenge*, *Response,* and *Verification*. And the process of integrity verification is shown in Figure 3.

(1) *Challenge*: First, the TPA confirms whether the file tag τ of outsourced file can pass the verification by retrieving τ from the CS and performing $S.Vrf(\tau_0, K_{AB})$. If the file tag τ of outsourced file cannot pass the verification, then the auditing task will not be executed, and the protocol aborts; otherwise, the TPA will analyze τ_0 to acquire the total number n of outsourced file blocks. The TPA picks a random nonempty subset $I \subseteq [1, n]$ and a number of values $s_i \in_R Z_q^*$ at random, for each $i \in I$. Then, the TPA distributes the challenge set $C = \{(i, s_i)_{i \in I}\}$ and corresponding file identifier FID to the CS. After that, the TPA can compute $WP = e(H_4(\Lambda \,||\, FID\,||i), g^{t_\vartheta})^{\sum_{i \in I} s_i}$ in advance for the final verification.

(2) *Response*: CS locates to the corresponding file F^* in the stereo storage structure upon receiving a challenge C and its file identifier FID from the TPA. Then, the CS computes $\chi_j = \sum_{i \in I} s_i \cdot \chi_{i,j} \bmod q, j \in [1, n]$ and $\sigma = \prod_{i \in I} \sigma_i^{s_i}$. After that, the CS sends to the TPA a proof P that consists of $\chi_1, \cdots, \chi_s, \sigma$ and corresponding authorization δ_Λ.

(3) *Verification*: Once receiving the proof P, with public system parameter PP and file tag τ, the TPA first verifies the validity of δ_Λ by demonstrating the equation (6), and then, verifies aggregate block authenticator σ as follows:

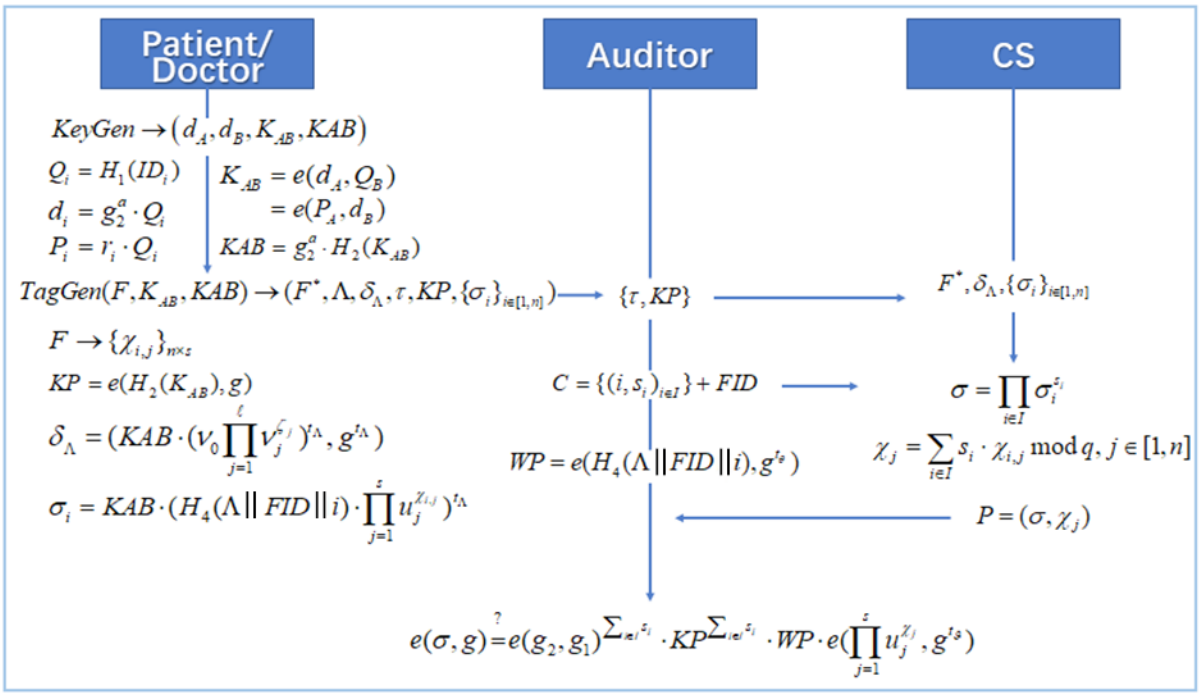

Figure 3. The process of integrity verification

$$e(\sigma, g) \stackrel{?}{=} e(g_2, g_1)^{\sum_{i\in I} s_i} \cdot KP^{\sum_{i\in I} s_i} \cdot WP \cdot e(\prod_{j=1}^{s} u_j^{\chi_j}, g^{t_\theta}) \quad (9)$$

If the equation (9) is true, the challenged outsourced file in the cloud is verified as intact; otherwise, the challenged file is corrupted. In the above auditing process, TPA can also audit the details of the challenged file warrant. That is, the proof *P*, which will be fed back by CS, should contain more file details.

6 SECURITY ANALYSIS

We analyzed the soundness of our scheme at first. That is, if all the entities are honest in this identity-based one-way anonymous e-health system, then the processed files and log warrants about medical data can be audited correctly. Then, we propose a simple security analysis for this scheme.

Theorem 1: In an appropriate registration process, the KGC is supposed to generate a correct privacy key for the user. In addition, the patient always produces a valid log warrant for his or her attending physician to render certain the authenticity of medical data. If the outsourced file in the cloud is not corrupted or tampered with, then the proof yielded by CS will be confirmed as valid.

Proof: As shown in Equation (2), we can confirm the correctness directly. Since patient A and the attending physician B have the shared auditing key, it follows that:

$$\begin{aligned} e(\alpha, g) &= e(KAB \cdot (v_0 \prod_{j=1}^{\ell} v_j^{\zeta_j})^{t_\Lambda}, g) \\ &= e(KAB, g) \cdot e((v_0 \prod_{j=1}^{\ell} v_j^{\zeta_j})^{t_\Lambda}, g) \\ &= e(g_2^a \cdot H_2(K_{AB}), g) \cdot e(v_0 \prod_{j=1}^{\ell} v_j^{\zeta_j}, g^{t_\Lambda}) \\ &= e(g_2, g_1) \cdot e(H_2(K_{AB}), g) \cdot e(v_0 \prod_{j=1}^{\ell} v_j^{\zeta_j}, g^{t_\Lambda}) \\ &= e(g_2, g_1) \cdot KP \cdot e(v_0 \prod_{j=1}^{\ell} v_j^{\zeta_j}, g^{t_\Lambda}) \end{aligned}$$

Therefore, Equation (6) holds.

Note that, $\chi_j = \sum_{i \in I} s_i \cdot \chi_{i,j} \bmod q$ for all $j \in [1, s]$ and

$$\begin{aligned}\sigma &= \prod_{i \in I} \sigma_i^{s_i} \\ &= \prod_{i \in I} KAB^{s_i} \cdot \prod_{i \in I} ((H_4(\Lambda \,||\, FID \,||i) \cdot \prod_{j=1}^{s} u_j^{\chi_{i,j}})^{t_\vartheta})^{s_i} \\ &= (g_2^a)^{\sum_{i \in I} s_i} \cdot H_2(K_{AB})^{\sum_{i \in I} s_i} \cdot (\prod_{i \in I} H_4(\Lambda \,||\, FID \,||i)^{s_i} \cdot \prod_{j=1}^{s} u_j^{\sum_{i \in I} s_i \chi_{i,j}})^{t_\vartheta}\end{aligned}$$

It follows that:

$$\begin{aligned}e(\sigma, g) &= e(g_2^a, g)^{\sum_{i \in I} s_i} \cdot e(H_2(K_{AB}), g)^{\sum_{i \in I} s_i} \cdot e((\prod_{i \in I} H_4(\Lambda \,||\, FID \,||i)^{s_i} \cdot \prod_{j=1}^{s} u_j^{\sum_{i \in I} s_i \chi_{i,j}}), g)^{t_\vartheta} \\ &= e(g_2, g_1)^{\sum_{i \in I} s_i} \cdot KP^{\sum_{i \in I} s_i} \cdot e(H_4(\Lambda \,||\, FID \,||i)^{s_i}, g^{t_\vartheta})^{\sum_{i \in I} s_i} \cdot e(\prod_{j=1}^{s} u_j^{\chi_j}, g^{t_\vartheta}) \\ &= e(g_2, g_1)^{\sum_{i \in I} s_i} \cdot KP^{\sum_{i \in I} s_i} \cdot WP \cdot e(\prod_{j=1}^{s} u_j^{\chi_j}, g^{t_\vartheta})\end{aligned}$$

Theorem 2: Here, we suppose that the signature algorithm is efficient and secure, and can generate file tags validly and correctly. And it is supposed that the Computational Diffie-Hellman (CDH) assumption holds in bilinear groups. The identity-based one-way anonymous scheme is secure against adaptive simulation. In detail, neither an untrusted cloud server nor the adversary $\mathcal{A}$ can forge a valid proof to get through the verification of the auditor successfully if the data in the cloud is tampered with or corrupted.

Proof: We utilize the theory of knowledge proof and a series of security games to prove this theorem which can acquire the challenged data blocks in the aforementioned game. When the adversary $\mathcal{A}$ interacts with the challenger $\mathcal{C}$ and generates a valid proof *P*, adversary $\mathcal{A}$ can successfully pass the verification for the challenged data blocks in the aforementioned game; there is a constructed knowledge extractor that can capture the challenged data blocks. It is assumed that the adversary $\mathcal{A}$ can get through the TPA's verification successfully without keeping the outsourced file integrity. Then, we can capture the whole challenged data blocks through the interaction between the constructed knowledge extractor and the proposed scheme.

Game 0: The challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ behave in Game 0 in a manner similar to that described in Section 4. First, the challenger $\mathcal{C}$ executes the preprocessing Setup algorithm to obtain the public parameter *PP* and a master secret key *msk*, and then sends *PP* to the adversary $\mathcal{A}$. Next, $\mathcal{C}$ performs the *KeyGen* algorithm to obtain the secret key of user. Then, $\mathcal{A}$ picks a list of data blocks and queries the signatures of them. According to the queries, $\mathcal{C}$ executes the Extract algorithm to generate corresponding signatures for the data blocks and transmit these requested signatures to the $\mathcal{A}$. After that, $\mathcal{C}$ sends a challenge to $\mathcal{A}$, and $\mathcal{A}$ generates corresponding proof to $\mathcal{C}$. Finally, $\mathcal{A}$ succeeds and the game aborts if the proof can get through the verification of $\mathcal{C}$ successfully with non-negligible probability.

Game 1: This game is identical to Game 0 with one difference. The challenger $\mathcal{C}$ keeps a list of query records about the requested signature of $\mathcal{A}$. If the adversary $\mathcal{A}$ is able to yield a aggregate signature, which is valid under the verification of the challenger $\mathcal{C}$ and is not generate by $\mathcal{C}$, the game aborts and the adversary $\mathcal{A}$ succeeds.

Analysis: It is supposed that $\mathcal{A}$ wins in the Game 1 with non-negligible probability. With this in mind, we can construct a simulator in our scheme to solve the CDH problem in bilinear groups. Given a group *G* with prime order q, $g, g^\alpha, h \in G$ as input, the simulator is to generate h^α by interacting with $\mathcal{A}$. The simulator acts like the challenger and runs as follows:

(1) The simulator randomly chooses an element $x \in_R Z_q^*$, and yields the public parameters as $g_1 = g^x$, $g_2 = h$ and the master secret key $msk = g_2^a$. Next, it randomly picks integers $\omega_j, \varpi_j \in_R Z_q^*$, and sets $u_j = g_2^{\omega_j} g^{\varpi_j}$. There is a random oracle H_4. The simulator stores a list of queries in the game and responses to the challenger $\mathcal{C}$ in a consistent manner by controlling the random oracle.

(2) When processing a file *F*, the simulator first yields a secret key for user as *KAB* by executing KeyGen algorithm. Hereafter, the simulator picks a random unique identifier for file *F* and a random element $\tilde{x} \in Z_q^*$, and yields $g^{t_\vartheta} = (g^\alpha)^{\tilde{x}}$. For every data block i, the simulator picks random values $\chi_i \in_R Z_q^*$ and sets:

$$H_4(\Lambda \,||\, FID \,||i) = g^{\gamma_i} / (g_2^{\sum_{j=1}^{s} \omega_j \chi_{i,j}} g^{\sum_{j=1}^{s} \varpi_j \chi_{i,j}}) \quad (10)$$

Based on equation (10), we have:

$$(H_4(\Lambda \,||\, FID \,||i) \cdot \prod_{j=1}^{s} u_j^{\chi_{ij}})^{t_\vartheta} = (g^{\gamma_i})^{t_\vartheta} \quad (11)$$

In addition, the simulator computes the block authentication for file block x_i as $\sigma_i = KAB \cdot (H_4(\Lambda \,||\, FID \,||i) \cdot \prod_{j=1}^{s} u_j^{\chi_{i,j}})^{t_\vartheta} = KAB \cdot (g^{\gamma_i})^{t_\vartheta}$. From the perspective of $\mathcal{A}$, σ_i is computationally indistinguishable from the real value.

(3) With the constant interaction, the simulator sends the processed files F^* to the adversary $\mathcal{A}$, which contains $\{\{\sigma_i\}_{i\in[1,n]}, \delta_\Lambda, FID\}$. Then, $\mathcal{A}$ outputs a forgery $\tilde{\sigma}$ with a non-negligible probability. Finally, if the adversary $\mathcal{A}$ is succeed to pass the validation, but the aggregate authentication $\tilde{\sigma}$ is unequal to the excepted aggregate authentication σ calculated by the simulator, then the game aborts.

According to the correctness of the proposed protocol, it is obvious that a correct proof $\chi_1, \cdots, \chi_s, \sigma$ can get through the verification successfully of the equation as follow:

$$e(\sigma, g) = e(g_2, g_1)^{\sum_{i\in I} s_i} \cdot KP^{\sum_{i\in I} s_i} \cdot WP \cdot e(\prod_{j=1}^{s} u_j^{\chi_j}, g^{t_\vartheta}) \quad (12)$$

Suppose the adversary $\mathcal{A}$ forges a proof $\tilde{\chi}_1, \cdots, \tilde{\chi}_s, \tilde{\sigma}$ which is different from the correct proof. Next, compute the following equation:

$$e(\tilde{\sigma}, g) = e(g_2, g_1)^{\sum_{i\in I} s_i} \cdot KP^{\sum_{i\in I} s_i} \cdot WP \cdot e(\prod_{j=1}^{s} u_j^{\tilde{\chi}_j}, g^{t_\vartheta}) \quad (13)$$

It is obvious that $\tilde{\chi}_j \neq \chi_j$, otherwise $\tilde{\sigma} = \sigma$. Then, define a set $\{\Delta\chi_j = \tilde{\chi}_j - \chi_j\}_{j\in[1,s]}$, which means at least one element of $\Delta\chi_j$ is non-zero. After that, divide equation (13) by equation (12) and get the following equation:

$$e(\tilde{\sigma}/\sigma, g) = e(\prod_{j=1}^{s} u_j^{\Delta\chi_j}, g^{t_\vartheta}) = e(\prod_{j=1}^{s} (g_2^{\omega_j} g^{\varpi_j})^{\Delta\chi_j}, (g^\alpha)^{\tilde{x}}) \quad (14)$$

It further implies:

$$e(\tilde{\sigma} \cdot \sigma^{-1} \cdot (g^\alpha)^{-\tilde{x} \cdot \sum_{j=1}^{s} \varpi_j \cdot \Delta\chi_j}, g) = e(h, g^\alpha)^{\tilde{x} \cdot \sum_{j=1}^{s} \omega_j \cdot \Delta\chi_j} \quad (15)$$

Finally, we can get the value of h^α as follow:

$$h^\alpha = (\tilde{\sigma} \cdot \sigma^{-1} \cdot (g^\alpha)^{-\tilde{x} \cdot \sum_{j=1}^{s} \varpi_j \cdot \Delta\chi_j})^{1/(\tilde{x} \cdot \sum_{j=1}^{s} \omega_j \cdot \Delta\chi_j)} \quad (16)$$

As long as the $\tilde{x} \cdot \sum_{j=1}^{s} \omega_j \cdot \Delta\chi_j \neq 0 \bmod q$, the above equations are valid and can be structured to solve the CDH problem. The probability of solving the CDH problem is equal to the probability of $1 - \Pr[\tilde{x} \cdot \sum_{j=1}^{s} \omega_j \cdot \Delta\chi_j = 0 \bmod q] = 1 - 1/q$, which is contradictory with the assumptions of the CDH problem. It means that if the adversary $\mathcal{A}$ has a different probability of success in Game 0 versus Game 1, which is non-negligible, then the simulator can be constructed to solve the CDH problem.

Game 2: Game 2 is similar with Game 1, except the following difference. The challenger $\mathcal{C}$ keeps interaction with the adversary $\mathcal{A}$ and holds all the processed outsourced files that have been sent to $\mathcal{A}$. In the process of the proposed auditing protocol, if the aggregate authenticator $\tilde{\sigma}$ yielded by $\mathcal{A}$ is not equality to the aggregate authenticator σ of the challenged file blocks, then the game aborts and the adversary $\mathcal{A}$ succeeds.

Analysis: Suppose the adversary $\mathcal{A}$ wins in this Game with a non-negligible probability. Hereafter, a simulator is constructed to work out the Discrete algorithm (DL) problem if the adversary $\mathcal{A}$ can succeed in this game. Given a group G with prime order q, $g, h \in G$ as input, the target of the simulator is to yield α by interacting with $\mathcal{A}$, which satisfies $h = g^{\alpha}$. The simulator behaves like $\mathcal{C}$ in Game 2, but with the following differences:

(1) Before processing a file F, the simulator first performs the *KeyGen* algorithm and yields a secret key for user as *KAB*. Then, following the process of the presented scheme in this paper, the simulator uses $u_j = g_2^{\omega_j} g^{\varpi_j}$ for each $1 \le j \le s$, where $\omega_j, \varpi_j \in_R Z_q^*$.

(2) The simulator keeps interacting with $\mathcal{A}$ to execute the auditing protocol proposed in this paper. As described in Game 1, if the aggregate file sectors $\tilde{\chi}_j$ generated by the adversary $\mathcal{A}$ is not equal to the aggregate file sectors χ_j of the challenged sectors, then the game aborts and the adversary $\mathcal{A}$ succeeds. It is easy to know that $\tilde{\sigma} = \sigma$ for the reason that Game 1 is not aborted. Next, with this in mind, compared with equation (12) and equation (13), we can get the following equation:

$$e(\prod_{j=1}^{s} u_j^{\tilde{\chi}_j}, g^{t_g}) = e(\prod_{j=1}^{s} u_j^{\chi_j}, g^{t_g}) \tag{17}$$

It further indicates that:

$$\prod_{j=1}^{s} u_j^{\tilde{\chi}_j} = \prod_{j=1}^{s} u_j^{\chi_j} \tag{18}$$

In addition, set $\{\Delta\chi_j = \tilde{\chi}_j - \chi_j\}_{j\in[1,s]}$, which means at least one element of $\Delta\chi_j$ is non-zero. After that, compute:

$$\prod_{j=1}^{s} u_j^{\Delta\chi_j} = h^{\sum_{j=1}^{s}\omega_j\Delta\chi_j} g^{\sum_{j=1}^{s}\varpi_j\Delta\chi_j} = 1 \tag{19}$$

Finally, the value of a is as follow:

$$\alpha = -\frac{\sum_{j=1}^{s}\varpi_j\Delta\chi_j}{\sum_{j=1}^{s}\omega_j\Delta\chi_j} \bmod q \tag{20}$$

As long as $\sum_{j=1}^{s}\omega_j\Delta\chi_j \neq 0 \bmod q$, the above equations are valid and can be structured to work out the DL problem. The probability of solving the DL problem is the same as the probability of $1 - \Pr[\sum_{j=1}^{s}\omega_j\Delta\chi_j \neq 0 \bmod q] = 1 - 1/q$, which is contradictory with the assumption of the DL problem. It means that if the adversary $\mathcal{A}$ has a different probability of success in Game 1 and Game 2, which is non-negligible, then the simulator can be constructed to solve the DL problem. To summarize, the proposed one-way anonymous auditing protocol is secure and can be proven by uniting Game 0, Game 1, and Game 2.

7 PERFORMANCE ANALYSIS

In this section, we first compare our scheme with the related schemes in terms of various characteristics. In Table 2, we can clearly conclude that our solution can better satisfy all the major characteristics.

Then, we give the numerical analysis of the computation overhead of the proposed stereo storage structure assisted one-way anonymous auditing protocol and then evaluate the performance of our scheme. In Table 3, we analyze and present the computation overhead of each algorithm respectively in the proposed scheme. Primarily, the following notations are defined to represent the various operations in the specific algorithms of each phase. The symbols $\mathbb{M}$, $\mathbb{E}$, and $\mathbb{H}$ denote a multiplication operation, a exponentiation operation and a hashing operation in G, respectively. In this paper, H_1, H_2, and H_3 are not distinguished and all can be expressed as $\mathbb{H}$. Similarly, the symbols $\mathbb{M}_T$ and $\mathbb{E}_T$ are respectively expressed as a multiplication operation and a exponentiation operation in G_T. $\mathbb{A}_q$ and $\mathbb{M}_q$ are indicated as one addition operation and one multiplication operation in Z_q, respectively. And $\mathbb{P}$ represents a bilinear pairing evaluation operation $e: G \times G \to G_T$. Considering that both g_1 and g_2 are public system parameters in our protocol, then $e(g_2, g_1)$ can be calculated in advance and viewed as a public value.

Table 2. Characteristics comparison with related schemes

Schemes	Public verifiability	Certificate management simplification	Privacy protection	Dynamic operations
Worku *et al*.[17]	√	×	√	√
Garg *et al*.[19]	√	×	×	√
Daniel and Vasanthi[25]	√	×	√	×
Zhao *et al*.[27]	×	√	×	×
Jiang *et al*. (this study)	√	√	√	√

Table 3. Computational overhead of the proposed scheme

Phases	KGC	User (physician)	User (patient)	TPA	CS
Setup	$2\mathbb{E}$	/	/	/	/
KeyGen(a)	$\mathbb{M}+\mathbb{H}$	$2\mathbb{P}+\mathbb{H}+\mathbb{M}_T$	$2\mathbb{P}+\mathbb{H}+\mathbb{M}_T$	/	/
KeyGen(b)	/	$\mathbb{P}+\mathbb{H}+\mathbb{M}$	$\mathbb{P}+2\mathbb{H}+\mathbb{M}_q+\mathbb{M}$	/	/
Extract(a)	/	$3\mathbb{P}+2\mathbb{H}+2\mathbb{M}_T+\ell\mathbb{M}$	$2\mathbb{E}+\mathbb{H}+(\ell+1)\mathbb{M}$	/	/
Extract(b)	/	$\mathbb{E}+\mathbb{H}+(s+1)\mathbb{M}$	/	/	/
Audit(b)	/	/	/	/	$n\lvert I\rvert\mathbb{M}_q+n(\lvert I\rvert-1)\mathbb{A}_q+(\lvert I\rvert-1)\mathbb{M}+\lvert I\rvert\mathbb{E}$
Audit(c)	/	/	/	$(s+1)\mathbb{E}+\mathbb{H}+(\lvert I\rvert-1)\mathbb{A}_q+3\mathbb{P}+3\mathbb{E}_T+(s+1)\mathbb{M}+\mathbb{M}_T$	/

KGC: Key generation center; TPA: third-party auditor; CS: cloud server

Therefore, the computation overhead of $e(g_2,g_1)$ is not contained in Table 3. Furthermore, the symbols $S.Sign$ and $S.Vrf$ are used to denote the signature and verification file tag processes. Hereafter, as shown in Table 3, *Setup* is a system preprocessing phase, which is performed by KGC and needs $2\mathbb{E}$. In the algorithm of *KeyGen(a)*, KGC needs $\mathbb{M}+\mathbb{H}$ operations to generate a privacy key for user, and both the physician and the patient need $2\mathbb{P}+\mathbb{H}+\mathbb{M}_T$ operations to verify the validity of the private key distributed by KGC. In the algorithm of *KeyGen(b)*, the patient performs one $\mathbb{H}$ operation and one $\mathbb{M}_q$ operation more than the physician to generate a pseudonym. To process a medical file, patient firstly yields a warrant for the physician, which needs $2\mathbb{E}+\mathbb{H}+(\ell+1)\mathbb{M}$ operations. Then, the physician verifies the validity of the warrant, which needs $3\mathbb{P}+2\mathbb{H}+2\mathbb{M}_T+\ell\mathbb{M}$ operations. ℓ denotes the string length of warrant. The amount of file data blocks and sectors are expressed as n and s. After that, physician performs another $\mathbb{E}+\mathbb{H}+(s+1)\mathbb{M}$ operation to generate a block authenticator. After receiving a challenge from TPA, CS executes $n\lvert I\rvert\mathbb{M}_q+n(\lvert I\rvert-1)\mathbb{A}_q+(\lvert I\rvert-1)\mathbb{M}+\lvert I\rvert\mathbb{E}$ operations to yield a proof P, where the $\lvert I\rvert$ is indicated as a set of non-empty challenge file randomly selected by TPA for auditing. Finally, TPA performs $(s+1)\mathbb{E}+\mathbb{H}+(\lvert I\rvert-1)\mathbb{A}_q+3\mathbb{P}+3\mathbb{E}_T+(s+1)\mathbb{M}+\mathbb{M}_T$ operations to verify data integrity in the cloud.

Figure 4 shows the computational cost of each entity in the proposed scheme for auditing an outsourced medical file with various numbers of data blocks. In this scheme, the time costs of TPA to prepare a challenge $\lvert I\rvert$ is not taken into account, for TPA can sample a series of random elements by running offline. In the experiments, we set $\ell=160$ in this scheme and each file block consists of 160 sectors, which means that it has around 4 KB of size. Moreover, we compare the efficiency of processing a 1 MB file by set challenge data block as 20, 40, ... , 100, 200, respectively.

The simulation results of Figure 4 demonstrate that the computational cost of the user is independent of the number of data blocks in the file in carrying out the extraction algorithm. Specifically, this experiment of our scheme only considers the case that patients generate warrants for files, which can be verified by physicians and generate file tags for those files, so the calculation cost of physicians is slightly higher than that of patients, which is in line with the theoretical computational overhead analysis of the proposed scheme shown in Table 3. In addition, if it is necessary, the division of work between the physician and the patient is interchangeable during the file processing phase. After that, in the audit phase, TPA has transferred part of the calculate task to CS. Therefore, we can conclude that, as shown in Figure 4, with the increase of data blocks, the calculation cost of CS increases gradually.

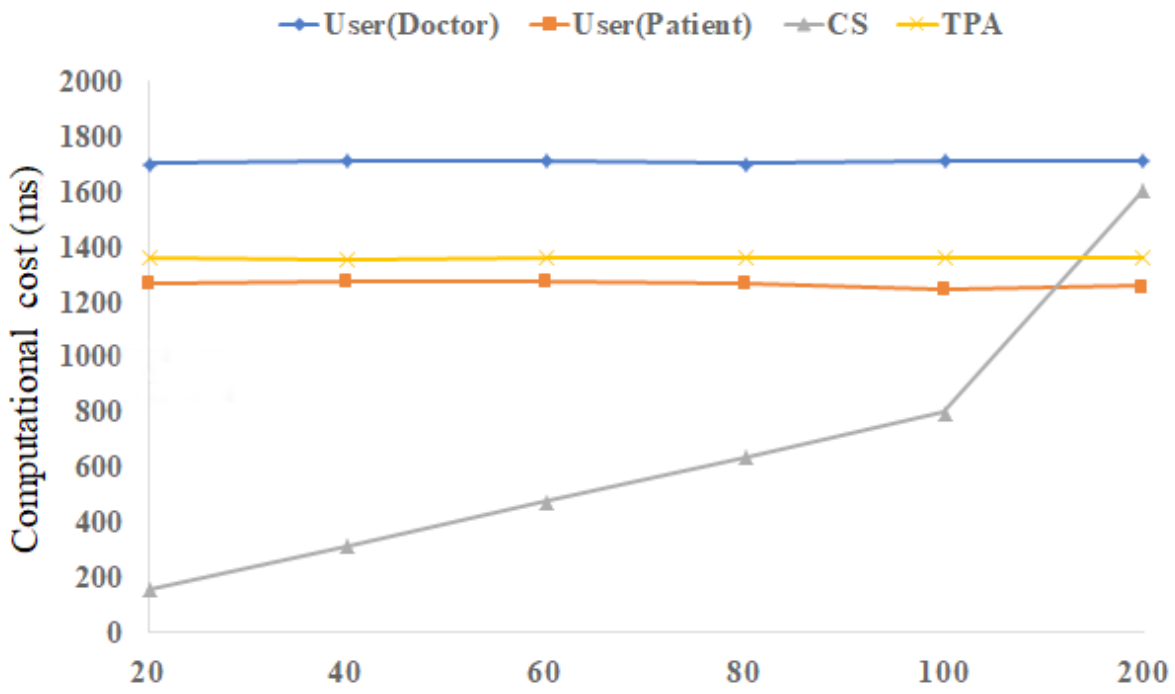

Figure 4. The computational cost of each entity in the proposed scheme

8 CONCLUSION

In this paper, we proposed a stereo storage structure assisted one-way anonymous auditing protocol aiming the e-health system for the particularity of medical data. In our scheme, medical data can be reviewed, used and verified for integrity by relevant medical personnel and relevant patients. Besides, both the file origin and the file integrity of medical data in EHS can be verified. In addition, the proposed stereo storage structure can effectively assist the storage and quick search of various types of medical data. Both the security analyses and experimental results demonstrate that the proposed scheme in this paper is efficient and secure in the cloud.

DECLARATIONS

Authors' contributions

Made substantial contributions to conception and design of the study and write the manuscript: Jiang LH
Provided administrative, technical, and material support: Wang C, Shen J

Availability of data and materials

The related data used to support the findings of this study are included within the article.

Financial support and sponsorship

This work is supported by the National Natural Science Foundation of China (No. U1836115, No. 61672295, No. 61922045, No. 61672290), the Natural Science Foundation of Jiangsu Province (No. BK20181408), Henan Key Laboratory of Network Cryptography Technology (No. LNCT2019-A01), the Peng Cheng Laboratory Project of Guangdong Province (No. PCL2018KP004), the 2020 Research Innovation Program for Postgraduates of Jiangsu Province (No. KYCX20-0936), the CICAEET fund, and the PAPD fund.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Qian L, Luo ZG, Du YJ, Guo LT. Cloud computing: an overview. Proceedings of the IEEE International Conference on Cloud Computing; 2009 Sep 21-25; Bangalore, India. Springer; 2009. pp. 626-31.
2. Ion I, Sachdeva N, Kumaraguru P, Čapkun S. Home is safer than the cloud!: privacy concerns for consumer cloud storage. Proceedings of the Seventh Symposium on Usable Privacy and Security; 2011 Jul 14-16; Pittsburgh, PA, USA. ACM; 2011. pp. 1-20.
3. Yu Y, Au MH, Ateniese G, Huang XY, Susilo W, et al. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. IEEE T Inf Foren Sec 2016;12:767-78.
4. Kang B, Wang J, Shao D. Attack on privacy-preserving public auditing schemes for cloud storage. Math Probl Eng 2017;2017:8062182.
5. Li Y, Yu Y, Yang B, Min G, Wu H. Privacy preserving cloud data auditing with efficient key update. Future Gener Comp Sy 2018;78:789-98.
6. Mehmood A, Natgunanathan I, Xiang Y, Hua G, Guo S. Protection of big data privacy. IEEE access 2016;4:1821-34.
7. More S, Chaudhari S. Third party public auditing scheme for cloud storage. Procedia Computer Science 2016;79:69-76.
8. Shen W, Yu J, Xia H, Zhang H, Lu X, et al. Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium. J Netw Comp Appl 2017;82:56-64.
9. Wang H, He D, Yu J, Wang ZW. Incentive and unconditionally anonymous identity-based public provable data possession. IEEE T Serv Comput 2019;12:824-35.
10. Balasubramanian V, Mala T. Cloud data integrity checking using bilinear pairing and network coding. Cluster Comput 2019;22:6927-35.
11. Yang G, Yu J, Shen W, Su Q, Fu Z, et al. Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability. J Syst Software 2016;113:130-9.
12. Deswarte Y, Quisquater JJ, Sadane A. Remote integrity checking. In: Working Conference on Integrity and Internal Control in Information Systems. Springer; 2003. pp. 1-11.
13. Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, et al. Provable data possession at untrusted stores. Proceedings of the 14th ACM conference on Computer and communications security; 2007 Oct 29- Nov 2; Alexandria, Virginia, USA. ACM; 2007. pp. 598-609.
14. Juels A, Kaliski BS. PORs: proofs of retrievability for large files. Proceedings of the 14th ACM conference on Computer and communications security; 2007 Oct 29- Nov 2; Alexandria, Virginia, USA. ACM; 2007. pp. 84-97.
15. Wang C, Wang Q, Ren K, Lou WJ. Privacy-preserving public auditing for data storage security in cloud computing. Proceedings of the 29th IEEE Conference on Computer Communications; 2010 Mar 15-19; San Diego, CA, USA. IEEE ComSoc; 2010. pp. 1-9.
16. Wang Q, Wang C, Li J, Ren K, Lou W. Enabling public verifiability and data dynamics for storage security in cloud computing. Proceedings of the 14th European Symposium on Research in Computer Security; 2009 Sep 21-23; Saint-Malo, France. Springer; 2009. pp. 355-70.
17. Worku SG, Xu C, Zhao J, He X. Secure and efficient privacy-preserving public auditing scheme for cloud storage. Comput Electr Eng 2014;40:1703-13.
18. Wang C, Chow SSM, Wang Q, Ren K, Lou W. Privacy-preserving public auditing for secure cloud storage. IEEE T Comput 2013;62:362-75.
19. Garg N, Bawa S, Kumar N. An efficient data integrity auditing protocol for cloud computing. Future Gener Comput Syst 2020;109:306-16.
20. Yuan J, Yu S. Public integrity auditing for dynamic data sharing with multiuser modification. IEEE Trans Inform Forensic Secur 2015;10:1717-26.
21. Suguna M, Mercy Shalinie S, Sivaranjani R. Integrity verification for shared data in group with user revocation. In: Zungeru AM, Subashini S, Vetrivelan P, editors. Wireless Communication Networks and Internet of Things. Singapore: Springer; 2019. pp. 41-9.
22. Wang X, Weng J, Ma J, Yang X. Cryptanalysis of a public authentication protocol for outsourced databases with multi-user modification. Inform Sciences 2019;488:13-8.
23. Zhang Y, Yu J, Hao R, Wang C, Kui R. Enabling efficient user revocation in identity-based cloud storage auditing for shared big data. IEEE T Depend Secure 2020;17:608-19.
24. Wu Y, Jiang ZL, Wang X, Yiu SM, Zhang P. Dynamic data operations with deduplication in privacy-preserving public auditing for secure cloud storage. Proceedings of the IEEE International Conference on Computational Science and Engineering and IEEE International Conference on Embedded and Ubiquitous Computing; 2017 Jul 21-24; Guangzhou, Guangdong, China. IEEE; 2017. pp. 562-7.
25. Daniel E, Vasanthi NA. LDAP: a lightweight deduplication and auditing protocol for secure data storage in cloud environment. Cluster Comput 2019;22:1247-58.
26. Tang X, Huang Y, Chang C, Zhou L. Efficient real-time integrity auditing With privacy-preserving arbitration for images in cloud storage

system. IEEE Access 2019;7:33009-23.
27. Zhao C, Xu L, Li J, Wang F, Fang H. Fuzzy identity-based dynamic auditing of big data on cloud storage. IEEE Access 2019;7:160459-71.
28. Barua R, Dutta R, Sarkar P. Extending Joux's protocol to multi party key agreement. In: Johansson T, Maitra S, editors. Progress in Cryptology - INDOCRYPT 2003. Berlin: Springer Berlin Heidelberg; 2003. pp. 205-17.
29. Kate A, Zaverucha G, Goldberg I. Pairing-based onion routing. In: Borisov N, Golle P, editors. Privacy Enhancing Technologies. Berlin: Springer Berlin Heidelberg; 2007. pp. 95-112.