Hash Forest Structure Assisted Bi-auditing Protocol with Multiuser Modification in E-health Systems

Linghong Jiang[1], Jian Shen[1,2], Sai Ji[1,3], Yihui Dong[1], Tiantian Miao[1]

[1] School of Computer and Software, Nanjing University of Information Science and Technology, China
[2] Cyberspace Security Research Center, Peng Cheng Laboratory, China
[3] School of Information Engineering, Suqian College, China
jianglinghong@126.com, s_shenjian@126.com, jisai@nuist.edu.cn, dongyh99@126.com, mtt_0106@126.com

Abstract

Due to the increasing volume of medical data, storing medical data in the cloud has become a trend of the times. With the popularity of cloud computing, how to ensure the integrity of medical data in the cloud is an urgent problem to be solved. Also, how to realize the following functions such as supporting for dynamic operation, multiuser modification, and user revocation are also further challenges for data integrity verification after fully consider the particularity of the medical scenario. In this paper, a new primitive of bi-auditing is put forward given the particularity of the medical scene. According to the different configurations and requirements of medical personnel and patients, two auditing schemes are designed to support different users along with various functions. On the one hand, a novel hash forest structure is designed to provide medical personnel with dynamic operations on data. Besides, the proposed structure supports medical personnel to perform multiuser modification operations on relevant data and supports the revocation of illegal users. On the other hand, considering the weak security awareness and low device configuration on the patient side, the key update is provided for the patient to deal with the key exposure problem. Both theoretical and experimental analyses demonstrate that the proposed scheme is secure and has a decent performance in computational overhead.

Keywords: Cloud computing, Data outsourcing, Cloud storage auditing, Data integrity verification, Dynamic data

1 Introduction

Storing medical data in the cloud has become the development trend of the times, because doing so will not only help to solve the problem of storing the huge volume of medical data, but also help to realize the upward and downward linkage of medical data, and realize regional collaborative medical treatment. However, the data stored in the cloud may be destroyed or lost due to inevitable hardware failures, software errors and human errors [1-2]. In recent years, outsourced data integrity verification has attracted extensive attention and research, and many remote data integrity verification schemes have been proposed. In addition, schemes that focus on more functions of cloud storage auditing have been proposed successively, such as high efficiency, data privacy protection, identity privacy protection, dynamic data operation, data sharing, etc. But at present, there are still no auditing schemes that can fully fit the particularity of the medical scene. Therefore, it is still of great significance to verify the integrity of medical data in the cloud for special medical scenarios.

In addition to the above failures and errors, data in the cloud may be subject to various external security attacks. Medical data are often well classified, and patients' sensitive data can be accessed from the cloud records of hospital physicians, emergency rooms, outpatient services, and health care organizations. Hackers can easily find data of interest, such as name, home address, email address, birthday, and even insurance policy number, diagnosis results, etc. In this way, hackers can use the data to forge false identities and fake insurance policies to seek medical treatment, buy medicine and so on. Several recent incursions in the medical industry have reportedly been caused by insiders. In addition to the benefits driven by health information, personal curiosity may also lead to improper access or data theft. Therefore, these organizations need to strictly restrict access rights to ensure that the data can only be accessed by relevant personnel. Therefore, it is necessary to support the revocation of illegal users in the e-health system.

Motivation of this paper: At present, only a few data integrity auditing schemes have been specifically studied for special scenarios in the medical environment. With the improvement of living standards, patients are no longer satisfied with knowing the results of their own medical data only in the paper medical records. In addition, paper-based

*Corresponding Author: Jian Shen; E-mail: s_shenjian@126.com

DOI: 10.53106/160792642021072204019

medical records are not easy to be kept, which makes patients more urgent to control electronic medical data. What's more, support the dynamic operation of multiple medical personnel on the same patient's medical data is the current development trend of medical data. For example, the medical data in the process of emergency/hospitalization is often generated and modified by multiple medical personnel. Then, a new security challenge has been brought, and the issue of user security revocation in the medical system is needed to be concerned. Given the above problems, this paper mainly focuses on the different configurations and requirements between the medical personnel and the patient in the medical system. The scheme will be designed more in line with the actual needs of these two different types of users for medical data.

1.1 Our Contributions

(1) A concept of bi-auditing is presented. In this paper, bi-auditing means auditing on two levels. The first one refers to the integrity of the same data can be verified in different ways. The second is that the integrity of the data can be verified for diverse types of users. In such a scheme, data integrity verification can be implemented adaptively according to user-differentiated requirements and their resource configuration.

(2) A novel hash forest structure is provided in the protocol. This hash forest structure can effectively support fully dynamic operations, constant auditing metadata and lightweight batch auditing. Specifically, these properties mean that group members can perform modifications, and the size of the validation material for the data integrity check is independent of the number of users and the size of the data, and the overhead of batch auditing is much lower than that of the general scheme.

(3) A bi-auditing protocol for different types of users is designed for the e-health system. On the one hand, for the patient users, due to the weak awareness of security and their low resource allocation, user's secret keys are more likely to be leaked. Therefore, a strong key-exposure resilient auditing is taken into account in this protocol for patients. On the other hand, in many scenarios, a patient may be diagnosed and treated by multiple medical personnel, such as emergency surgery. Then, the user's medical data may be jointly formulated by multiple medical personnel. Therefore, considering the particularity of the medical scene, it is necessary to design an auditing protocol with multiuser modification, secure user revocation and public auditing for medical personnel.

1.2 Related Works

As a service hotspot in the field of cloud computing, in recent years, scholars and experts at home and abroad have conducted extensive research on cloud storage. In 2003, Deswarte et al. [3] first proposed the concept of remote data integrity auditing based on public key cryptography, which was used to verify data integrity stored on untrusted servers. However, the designed scheme did not consider the need to store large amounts of data in the cloud, and the computational cost of the solution was relatively high. Even so, the proposal of scheme [3] provided ideas for subsequent scholars on cloud data auditing research. In 2007, Ateniese et al. [4] introduced the concept of public auditing and proposed a model of provable data possession (PDP). In the PDP model, when users want to obtain data integrity information, they only need to send certain data block subsets as a challenge. In the same year, A. Juels et al. [5] proposed the model of proofs of retrievability (POR). The POR model can recover the data with a certain probability after the cloud data is damaged, but is prone to cause a lot of computational cost and communication cost. For the shortcomings of the POR model, most of the subsequent researches [6-7] are still based on the PDP model, and various functional extensions have been made. In the following, we classify auditing protocols from different perspectives.

According to the different roles of auditors, data integrity auditing can be divided into private auditing protocols and public auditing protocols. The private auditing protocol means that the data stored in the cloud can only be verified by the data owner. However, this process causes a heavy burden of computing resources and a lot of overhead for the data owner. Typical schemes are proposed in [8-9]. To solve the above problems, a third-party entity is introduced into the model of public auditing and is delegated the data auditing task by the owner. Typical schemes are proposed in [10-11]. But it comes with a new challenge, how to protect users' data privacy in the cloud. Therefore, Wang et al. [12] introduced the data privacy protection technology to the field of auditing. To improve the efficiency of public auditing, He et al. [13] first proposed a batch auditing scheme, which supports TPA to perform multiple auditing tasks. Subsequently, in terms of public auditing, data privacy protection and batch auditing schemes [14-15] were successively proposed and improved.

According to whether the data in the cloud supports dynamic operations, auditing protocols can be further divided into integrity auditing for static data and integrity auditing for dynamic data. The PDP model proposed in scheme [4] can only support static data auditing. That is to say, users can no longer perform dynamic operations on data after uploading their data to the cloud [16]. However, with the improvement of users' needs and the consideration of various aspects of performance, it is inevitable to support users to operate dynamically on the data in the cloud. The first auditing scheme supporting fully dynamic operation was proposed by Erway et al. [17], which realized the combination of dynamic data structure and verification auditing. Since then, a series of data structures [18-19] have been introduced into the auditing scheme. After that, supporting multi-user collaborative office has become the trend of cloud data development. In 2017, Wang et al. [20] proposed a data integrity auditing scheme that supports the revocation of group users. However, this scheme cannot resist the security problems caused by the collusion between malicious revoked users and the cloud. Henceforth, how to realize the security revocation of invalid and malicious group members and how to deal with the files operated by the revoked members have become a research hotspot [21-22].

However, the schemes in the above classification still have the problems of insufficient security and low efficiency. In addition, there is no auditing protocol that can meet the above characteristics at the same time. To sum up, it is necessary to design an auditing scheme based on the above aspects for the medical environment.

1.3 Organization

The remaining chapters of this paper are organized as follows: We first describe the system architecture that contains the system model and the security model of the proposed scheme in Section 2. Then, we demonstrate an overview of the proposed scheme, present the bi-auditing hash tree and hash forest structure, and represent a detailed description of the proposed scheme in Section 3. Besides, a security analysis is performed in Section 4. After that, Section 5 presents the performance analysis of our scheme. Finally, we conclude the findings of the paper in Section 6.

2 Models

2.1 System Model

The new paradigm of bi-auditing proposed in this paper includes two implications. One is to double-verify the medical data so that the accuracy of verification results can be enhanced. The other is to verify data integrity for different types of users, so as to improve the actual availability of the scheme. For the medical environment, the scheme can realize data integrity verification adaptively according to the user-differentiated needs and the resource allocation of patients and medical personnel. We focus on the deployment of the system model, which contains of following five entities as shown in Figure 1: a group of user medical personnel, patients, system service, a third part auditor (TPA), and cloud services.

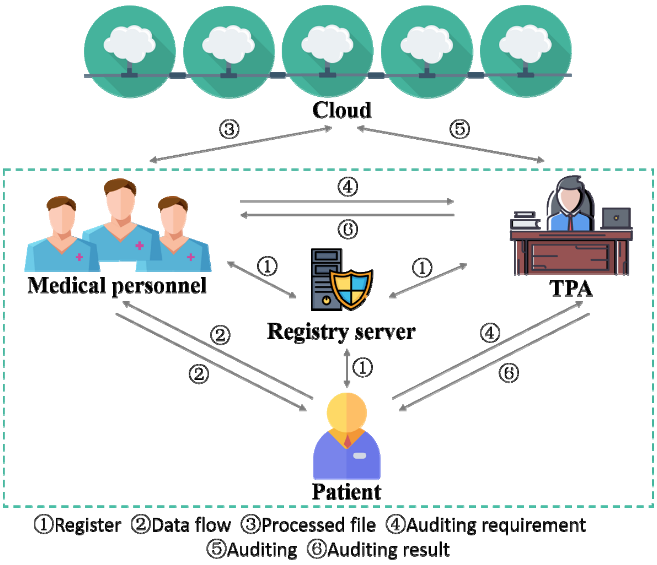

Cloud
③
⑤
④
⑥
Medical personnel
①
①
TPA
Registry server
②
②
④
⑥
①
Patient
①Register ②Data flow ③Processed file ④Auditing requirement
⑤Auditing ⑥Auditing result

Figure 1. The proposed user-differentiated system model

Medical personnel: including the patient's attending doctor and qualified medical personnel. Based on the patient's condition assessment, medical personnel will develop appropriate diagnosis and treatment records, including examination, treatment, and care plans. After the treatment, the complete medical file is formed and finally stored in the cloud. Medical personnel have the authority to review and modify the medical data of the patient in charge before the patient file is archived. In addition, this scheme supports secure user revocation of medical personnel.

Patients: the main user object in the medical system. Generally, they only have authority to access their own relevant medical data. Taking into account the particularity of the medical system, this protocol does not support the revocation of patient status.

System server: the e-health system server distributes initial keys for medical personnel, patients, and the TPA.

TPA: with nearly unlimited computing and storage capacity. After receiving the user's auditing requirements, TPA verifies the integrity of the data in the cloud and feeds back the auditing results to the user.

Cloud services: can provide almost unlimited computing resources and storage capacity, with ultra-fast computing speed.

2.2 Security Model

In real-world scenarios, cloud service providers are not always honest and trustworthy for some business interests. According to [24], the possible attack patterns for a malicious CS are described below.

Forge attack: the malicious CS can construct a valid sector authenticator without knowing the privacy key.

Replay attack: the malicious CS can generate a proof from the previous proof, without retrieving the actual challenged.

Replace attack: the malicious CS can choose an uncorrupted and valid pair of data sector and sector authenticator to replace the challenged pair of data sector and sector authenticator.

3 Our Construction

3.1 Overview

In the medical environment, according to the user-differentiated resource allocation and requirements, the scheme can implement the distinct data integrity verification protocol adaptively. This paper proposes a multi-user modification process in e-health system, to elaborate the process of medical personnel modifying patient medical data. The proposed scheme makes the authenticator accurate to the sector, which is suitable for the medical scene, and can realize the accountability of the relevant medical personnel. In this paper, the attending physician of the patient is regarded as the creator of the medical data, the specific process is shown in Figure 2.

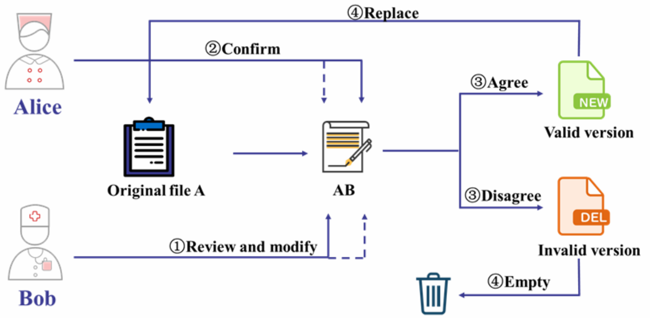


Figure 2. The proposed multi-user modification process in e-health system

First of all, we treat Alice as the attending physician for a patient and is responsible for generating and updating the patient's medical data original file A. Bob can be any medical personnel involved in treating the patient and has the authority to review and modify the patient's medical data. Specifically, if Bob finds that some data of original file A may have errors during the review process, he can modify it and generate file AB. Then, Alice confirms the modified file AB. If Alice agrees that Bob's modification can make the data more complete, the file AB becomes the valid version and replaces the original file A. Otherwise, the file AB becomes an invalid version and is eventually cleared.

3.2 The Proposed Bi-auditing Hash Tree and Hash Forest Structure

In this paper, we propose a novel bi-auditing hash tree and hash forest structure assisted data integrity verification with multiuser modification, as shown in Figure 3 and Figure 4.

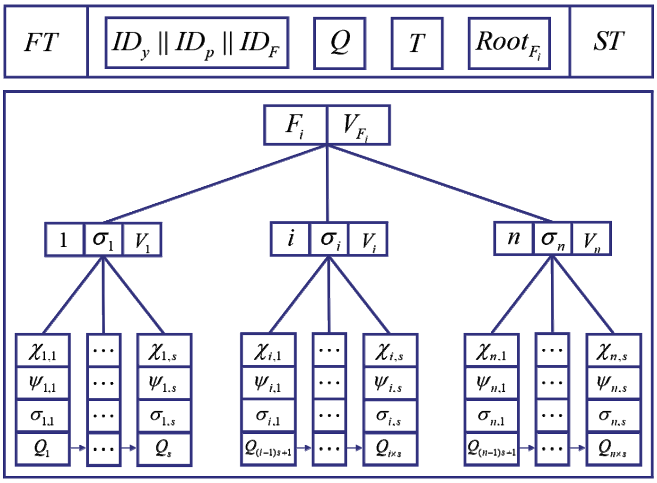


Figure 3. The proposed bi-auditing hash tree structure

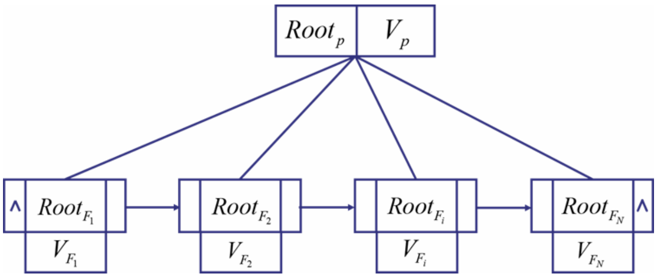


Figure 4. The proposed hash forest structure

In the e-health system, to complete the data integrity verification of massive users more efficiently, the design scheme needs to achieve the rapid search of data while meeting the storage requirements of massive data. When the traditional binary tree structure is used to store a large amount of data, the depth of the binary tree structure will be enlarged. Then, the disk input/output will be read and written too frequently, which will reduce the query efficiency. To solve the above problem, the basic idea of reducing tree depth is to use a multi-branches tree structure. Therefore, combined with the file preprocessing process of the traditional auditing scheme, an $n*s$ tree structure that supports the fast search of file blocks is proposed.

Specifically, the bi-auditing hash tree structure proposed in this paper is based on the medical file of the patient. A user file is divided into n data blocks χ_i, where $1 \le i \le n$. However, the data block χ_i is too large to be processed in Z_p. Therefore, each data block χ_i is divided into s blocks $\chi_{i,j}$, where $\chi_{i,j} \in Z_q$ and $1 \le j \le s$. The processed files are stored in a tree structure, as shown in Figure 3. $\chi_{i,j}$ represents the value of each data block. $\psi_{i,j}$ denotes the log tag corresponding to each data block. $\sigma_{i,j}$ is the authenticator generated by the user, which embeds his/her own key. Q_i signifies the pointer of leaf node to the adjacent node, which is convenient for quick index after data sector is inserted or deleted. For the i-th block χ_i, the corresponding node stores a triple (i, σ_i, V_i), where i is the block index number, σ_i is the authenticator aggregation of s relevant sectors belonging to the i-th data block. And V_i is the hash value of σ_i. For the root node, which stores a tuple (F_i, V_{F_i}), F_i represents the i-th file of patient ID_p, V_{F_i} denotes the hash value of the file F_i. A bi-auditing hash tree represents a medical file of a patient, and each tree contains a tree root. The tree root is used to store the relevant file information and log information, which is conducive to the rapid indexing of patients' files.

$$T_{ID}{=}ID_k \,||\, ID_p \,||\, F_i$$

$Root_{F_i}$ is the value after embedding the patient's key into the hash value of the root node. FT and ST represent pointers that can be linked to the preceding and subsequent adjacent trees, respectively. By linking all the medical files of a patient as shown in Figure 4, the proposed hash forest structure can help the patient to verify the integrity of medical data in a lightweight and fast way. $Root_p$ is the hash value of the constructed hash forest for patient p.

3.3 Description of Our Scheme

According to the particularity of medical environment, the scheme proposed in this paper will be respectively elaborated for medical personnel and patient users. Note that for ease of reading, Table 1 summarizes some of the main notations used in the paper as follows:

Table 1. Notation

a	The system master key
n, s	The number of blocks and sectors for file F
F_i	The i-th file of patient p
$\chi_{i,j}$	The data of the j sector of the i-th block in the file
Λ	The aggregate authentication value
τ, T_{ID}	The file tag and tree ID of file F
ID_i, ID_p	The identity of medical personnel i and patient p
H_1, H_2, H_3	Three hash functions
d_i, SK_p, SK_{TPA}	The secret key of medical personnel i, patient p and TPA
$\psi_{i,j}, \sigma_{i,j}$	The log information and block authenticator of data $\chi_{i,j}$
$\mathbb{U}_0, \mathbb{U}_k, \mathbb{U}_R$	The index set of data modified by attending doctor, medical personnel k and revoked users

3.3.1 Medical Personnel

This part of the protocol is mainly designed for medical personnel to achieve more detailed data sector verification of medical files.

(1) **SysSetup Phase.** In this phase, the e-health system server in charge of generates the system master secret key, public parameters and distributes the initial keys for medical personnel.

- Taking as input a secure parameter κ, the e-health system server randomly chooses two cyclic groups G_1 and G_2 with prime order q, and selects an element $g \in G_1$. Randomly pick two cryptographic hash functions, $H_1, H_3 : \{0,1\}^* \to G_1$ is used to map any length string to G_1 and $H_2 : \{0,1\}^* \to Z_q^*$ is used to map any length string to a finite field.
- The server of e-health system randomly selects a random integer $a \in Z_q^*$, and computes $g_1 = g^a$. As the system master key, a is used to generate the initial key for medical personnel and is kept secret.
- The e-health system server generates the corresponding key $d_i = H_1(ID_i)^a$ for medical personnel i according to the medical personnel ID. In order to distinguish, the medical personnel involved in the diagnosis and treatment of the same patient can be treated as a group of K users, which includes an attending doctor and $K-1$ medical qualified personnel. And the corresponding secret keys are $d_0 = H_1(ID_0)^a$ and $d_k = H_1(ID_k)^a, 1 \le k \le K-1.$. After receiving the key, the medical personnel i can verify it through the following equation.

$$e(d_i, g) \overset{?}{=} e(H_1(ID_i), g_1) \qquad \textbf{(1)}$$

If the equation is true, the key is correct; otherwise, the e-health system server is requested to resend the key.
- The e-health system server also calculates $\theta_k = g^{d_0/d_k}$ and publishes the public parameters as $PK = (H_1, H_2, H_3, g, g_1, \theta_k)$.

(2) **Preprocess Phase.** At this stage, the attending doctor first generates initial medical records for the

patient, preprocesses the files and uploads them to the e-health system.

- Multiple files would be produced during the diagnosis and treatment of a patient. To facilitate the file processing, suppose that the attending doctor divides each medical file $F \in (0,1)^*$ into n blocks, and each block χ_i comprises s sectors, that is, $F = (\chi_{i,j})_{n\times s}$. Note that, patient p owns the set of files $F_p = \{F_i\}_{1\le i\le N}$, N is the upper limit of the number of files for a patient.
- Randomly select an integer $\xi \in Z_q^*$, generate set $\{g^{\xi^j}\}, 1 \le j \le s$. The attending doctor calculates the authenticator

$$\sigma_{i,j} = (\psi_{i,j} \cdot g^{\chi_{i,j}\cdot\xi^j})^{d_0} \tag{2}$$

for each sector $\chi_{i,j}$, where $\psi_{i,j} = H_2(i\,||\,j\,||\,k\,||\,t_i\,||\,v_n)$, i is the index of data block χ_i, j is the index of sector $\chi_{i,j}$ in data block χ_i, k is the index of user in the group, t_i is the time stamp and v_n is the version number.

- The attending doctor sends these files and tuples $(\chi_{i,j}, \psi_{i,j}, \sigma_{i,j}, \mathbb{U}_0)$ to the system server, $\mathbb{U}_0$ is the index set of the data generated by attending doctor, and these files can be viewed and modified by the rest of the doctors in the group.

(3) **DataModify Phase.** Medical qualified personnel are allowed to modify files in the e-health system.

- Specifically, as the medical personnel modifies the sector data, the corresponding sector log changes to $\psi'_{i,j} = H_2(i\,||\,j\,||\,k'\,||\,t'_i\,||\,v'_n)$, and the authenticator changes to

$$\sigma'_{i,j} = (\psi'_{i,j} \cdot g^{\chi'_{i,j}\cdot\xi^j})^{d_k} \tag{3}$$

accordingly.

- After modification, the medical personnel send the file and the tuple $(\chi'_{i,j}, \psi'_{i,j}, \sigma'_{i,j}, \mathbb{U}_k)$ to the system server, $\mathbb{U}_k$ is the index set of the data modified by medical personnel. At this time, there are two versions before and after modification in the e-health system, and the valid version is left after confirmation by the attending doctor.

(4) **Upload Phase.** According to the files to be uploaded, the e-health system constructs a bi-auditing hash tree described in section 3.2 for each medical record. After that, the medial record and root value will be sent to the patient. If the patient agrees with the current diagnosis and treatment data, the root value will be encrypted and returned to the e-health system.

- According to the files to be uploaded, the e-health system constructs a bi-auditing hash tree described in section 3.3 for each medical record. If there is no medical personnel to modify the original data, then calculate the aggregated block authenticators as follow:

$$\sigma_i = \prod_{(i,j)\in\mathbb{U}_0} \sigma_{i,j} \tag{4}$$

On the contrary, if the original data is modified by the set of medical personnel $\mathbb{U}_k$, then the aggregated block authenticators computed as follow:

$$\sigma_i = \prod_{(i,j)\in\mathbb{U}_0} \sigma_{i,j} \cdot \prod_{(i,j)\in\mathbb{U}_k} \sigma_{i,j} \tag{5}$$

where $\overrightarrow{\chi_i} = (0,\cdots,\chi_{i,j},\cdots)$, $(i,j)\in\mathbb{U}_0$ and $\vec{\chi}'_i = (0,\cdots,\chi'_{i,j},\cdots)$, $(i,j)\in\mathbb{U}_k$.

- Computes the auxiliary value of the bi-auditing hash tree $\{V_i\}_{1\le i\le n}$, in which $V_i = H_3(\sigma_i)$. Then the root value $V_{F_i} = H_3(V_1\,||\,V_2\,||\cdots||\,V_n)$ of the bi-auditing hash tree corresponding to the file F_i can be obtain.
- The e-health system generates a file tag $\tau = T_{ID}\,||\,V_{F_i}\,||\,\{V_i\}_{1\le i\le n}\,||\,SIG(T_{ID}\,||\,V_{F_i}\,||\,\{V_i\}_{1\le i\le n})$. F_i is the index number of the file, which can also be the file name. The e-health system uploads file tag τ, the set of authenticators $\Phi = \{g^{\xi^j}\}$ along with file F_i to the cloud, and sends the tuple (F_i, V_{F_i}) to the corresponding patient.

(5) **UserRevo Phase**. Due to the revoked user may upload or even modify many files, the process of downloading and re-signing relevant data in the cloud will cause a large computational overhead. Therefore, this protocol adopts Shamir Secret Sharing technology and uses multiple cloud nodes to re-sign the relevant data sectors.

- The system server generates $\eta = (d_0 + \varepsilon)/d_k$, where $\varepsilon \in Z_q^*$. $\lambda = d_0/(d_0+\varepsilon)$ is also produced and sent to valid group users and the TPA as part of the PK. The e-health system server runs the Shamir secret sharing scheme and generates N points $(j, f(j))$ of a $M-1$ degree polynomial $f(x) = \eta + a_1x + a_2x^2 + \ldots + a_{(M-1)}x^{M-1}$. N points will be sent to N nodes of a cloud server.
- Then any M cloud nodes can reconstruct η and calculate

$$\sigma''_{i,j} = (\sigma'_{i,j})^{\eta} = (\psi'_{i,j} \cdot g^{\chi'_{i,j}\cdot\xi^j})^{d_0+\varepsilon} \tag{6}$$

where $(i,j)\in\mathbb{U}_R$, and the $\mathbb{U}_R$ is the number of revoked medical personnel.

(6) **Challenge Phase.** In order to ensure the integrity of medical data in the cloud, medical personnel can entrust TPA to verify the integrity of outsourcing data. In the medical environment, the devices of medical personnel perform better than the patient's, which also has a more powerful capacity of calculation. Thus, a more detailed data verification protocol is designed for medical personnel to ensure the integrity of cloud data in this paper.

- After receiving the auditing request from the attending doctor, TPA first verifies the validity of the file tag τ. If the file tag is valid, TPA can resolve n and s from the file tag τ. However, when τ is verified to be invalid, the protocol aborts.
- Taking a single file as an example, TPA randomly chooses c data sectors in n data blocks as a challenge data sector set $C=\{(i,j)\,|\,1\le i\le n, 1\le j\le s\}$, and picks two random integers $r, \nu \in Z_q^*$.
- Suppose that the selected c data sectors are modified by a set of users, denoted by U, which $0\le |U| \le K-1$, and generate set $\eta=\{\theta_k{}^r\}_{k\in U}$. If the set C contains sectors last modified by any revoked user, add λ^r to set η. Send the challenge $Chal=\{C,\nu,\eta,g^r\}$ to the cloud server.

(7) **Prove Phase.** After receiving the challenge from TPA, the cloud server generates a corresponding proof and returns it to TPA.

- According to the challenge, the cloud server locates the data sectors according to the challenge and computes $\varphi=\prod_{(i,j)\in C}\psi_{i,j}{}^{r_i}$, in which $r_i=\nu^i \bmod q$, $i\in C$.
- Calculate $\phi=f_{\vec{A}}(\nu) \bmod q$, where $\vec{A}=(0,\sum_{i\in D} r_i\chi_{i,1}, \sum_{i\in D} r_i\chi_{i,2},\cdots,\sum_{i\in D} r_i\chi_{i,s})$. Divide the polynomial $f_{\vec{A}}(x)-f_{\vec{A}}(\nu)$ with $(x-\nu)$ using polynomial long division, and denote the coefficients vector of the resulting quotient polynomial by $\vec{\omega}=(\omega_1,\omega_2,\cdots,\omega_s)$, i.e., $f_{\vec{\omega}}(x)\equiv\dfrac{f_{\vec{A}}(x)-f_{\vec{A}}(\nu)}{x-\nu}$. With $\vec{\omega}$, computes

$$\Gamma=\prod_{j=1}^{s}(g^{\xi^j})^{\omega_j}=g^{f_{\vec{\omega}}(\xi)} \qquad \textbf{(7)}$$

- For sectors never modified by any medical personnel or only modified by attending doctor, compute $\Lambda_{i,j}=e(\sigma_{i,j},g^r)$. For sectors in C that were last modified by medical personnel u_k, $k\in U$, compute $\Lambda_{i,j}=e(\sigma_{i,j},\theta_k{}^r)$. For sectors in C that were last modified by revoked medical personnel, compute $\Lambda_{i,j}=e(\sigma''_{i,j},\lambda^r)$. Aggregate Λ_i as

$$\Lambda=\prod_{(i,j)\in C}\Lambda_{i,j}{}^{r_i} \qquad \textbf{(8)}$$

and send the proof $\{\varphi,\phi,\Gamma,\Lambda\}$ to the TPA.

(8) Verify Phase.

- Compute $\varphi=\prod_{(i,j)\in C}\psi_{i,j}{}^{r_i}$. Verify the integrity of file as

$$\Lambda\cdot e(\theta_0{}^{-\phi},g^r)\overset{?}{=}e(\varphi,\theta_0{}^r)\cdot e(\Gamma^r,\theta\cdot\theta_0{}^{-\nu}) \qquad \textbf{(9)}$$

- If the above equation holds, the data stored in the cloud is intact, otherwise, it is not.

3.3.2 Patients

This part is the data integrity verification protocol designed for patients, which aims to give patients more authority over their medical data.

(1) SysSetup Phase. This phase is identical to the SysSetup Phase of medical personnel in section 3.3.1, with the following difference.

- The e-health system server randomly selects a $a\in Z_q^*$ as system master secret key for generating privacy keys for patients, and computes $g_1=g^a$.
- Calculates and sends the privacy key $sk_p=H_1(ID_p)^a$ to patient p. After receiving the key, the patient p can verify it through the following equation.

$$e(sk_p,g)\overset{?}{=}e(H_1(ID_p),g_1) \qquad \textbf{(10)}$$

If the equation is true, the key is correct; otherwise, the e-health system server is requested to resend the key. The public key of patient p is $pk_p=g^{sk_p}$.

- The e-health system server randomly picks an integrity $sk_{TPA}\in Z_q^*$ and send it to TPA as TPA's secret key. The public key of TPA is $pk_{TPA}=g^{sk_{TPA}}$.
- Select an element $u\in G_1$ and set system public parameters as $PK=(g,g_1,u,H_1,H_2,pk_p,pk_{TPA})$.

(2) **KeyUpdate Phase.** Due to the weak security awareness and low security configuration of the patient client, the e-health system server provides patients with key updates to address the key-exposure issues.

- TPA calculates and sends update messages $M_t=H_1(t)^{sk_{TPA}}$ to the patient users at regular intervals.
- Patients can verify whether the update message M_t is valid according to the following equation.

$$e(M_t,g)\overset{?}{=}e(H_1(t),pk_{TPA}) \qquad \textbf{(11)}$$

- If the update information M_t is valid, the patient updates the sign secret key in time period t as

$$SK_t=H_1(t)^{sk_p}\cdot M_t \qquad \textbf{(12)}$$

(3) **AuthGen Phase.** Generally, patients do not have relevant professional knowledge and have fewer

computing resources, so we design an auditing protocol for patients with relatively simple functions without high overhead.

- After the patient's medical data is finally modified and confirmed in the e-health system during the Uploading Phase in section 3.3.1, the system server sends each file and the root value of the constructed tree to the patient. After the file is confirmed, the patient randomly picks $\alpha \in Z_q^*$, and calculate $\gamma = g^{\alpha}$. Then, the label

$$Root_{F_i} = SK_t \cdot (H_2(t \| T_{ID}) \cdot u^{V_{F_i}})^{\alpha} \qquad \textbf{(13)}$$

  is calculated for the corresponding root value and returned the tuple $(t, \gamma, Root_{F_i} \cdots Root_{F_N})$ to the e-health system server.
- The e-health system server receives the labels from patient and gets a forest value

$$V_p = H_3(V_{F_1} \| \cdots \| V_{F_N}) \qquad \textbf{(14)}$$

  which is an aggregation of labels about all files of the patient, and finally will be sent to the cloud server.

(4) **Challenge Phase.** TPA first verifies the validity of the file tag, and selects $s_{F_i} \in Z_q^*$, $\mathbb{F}_i \subseteq [F_1, F_N]$, and send the challenge $Chal = \{F_i, s_{F_i}\}_{F_i \in \mathbb{F}_i}$ to the cloud.

(5) **Prove Phase.** After receiving the challenge from TPA, the cloud server generates a corresponding proof and returns it to TPA.

- When the cloud server receives the challenge, the aggregate label is calculated as $Root_p = \prod_{F_i \in \mathbb{F}_i} Root_{F_i}^{s_{F_i}}$.
- Computes a linear combination of file $\pi = \sum_{F_i \in \mathbb{F}_i} s_{F_i} \cdot V_{F_i}$.
- Send an auditing proof $P = \{\pi, Root_p, t\}$ to the TPA.

(6) Verify Phase.

- After receives the cloud server's proof, TPA verifies the following equations.

$$e(g, Root_p) \overset{?}{=} e(pk_p \cdot pk_{TPA}, H_1(t)^{\sum_{F_i \in \mathbb{F}_i} s_{Fi}}) \qquad \textbf{(15)}$$
$$\cdot e(\gamma, (\prod_{F_i \in \mathbb{F}_i} H_2(t \| T_{ID})^{s_{Fi}}) \cdot u^{\pi})$$

- If the above equation holds, the data stored in the cloud is intact, otherwise, it is not.

4 Security Analysis

In this section, we will analyze and demonstrate the correctness and security of the proposed scheme. That is, if all entities in the e-health system are functioning properly, then the user registration information, file log and the processed data generated by the proposed scheme can be properly audited. We will clarify the security of the relevant protocols from two aspects of medical personnel and patients according to the characteristics of the scheme proposed in this paper.

Theorem 1: In the successful system setup and phase, medical personnel and patients always accept the private key generated by the system server. The file processing related operations are performed correctly if the corresponding medical personnel and patients are honest. If the auditing file is correctly stored in the cloud, the proof generated by the cloud server will be proved to be valid. In other words, the following equation holds.

Proof:

- **Medical personnel:** From the description of *SysSetup Phase* in section 3.3.1, the correctness of Equation (1) is intuitive. Owing to

$$\begin{cases} \Lambda_{i,j} = e(\sigma_{i,j}, g^r) = e(\psi_{i,j} \cdot g^{\chi_{i,j} \cdot \xi^j}, g)^{d_0 \cdot r}, (i,j) \in \mathbb{U}_0 \\ \Lambda_{i,j}{}' = e(\sigma_{i,j}{}', \theta_k{}^r) = e(\psi_{i,j}{}' \cdot g^{\chi_{i,j}{}' \cdot \xi^j}, g)^{d_0 \cdot r}, (i,j) \in \mathbb{U}_k \\ \Lambda_{i,j}{}'' = e(\sigma_{i,j}{}'', \lambda^r) = e(\psi_{i,j}{}' \cdot g^{\chi_{i,j}{}' \cdot \xi^j}, g)^{d_0 \cdot r}, (i,j) \in \mathbb{U}_R \end{cases} \qquad \textbf{(16)}$$

and

$$\prod_{(i,j) \in C} g^{r_i \cdot \chi_{i,j} \cdot \xi^j} = g^{f_{\underset{A}{\rightarrow}}(\xi)} \qquad \textbf{(17)}$$

it follows that

$$\begin{aligned} \Lambda &= \prod_{(i,j) \in C} \Lambda_{i,j}{}^{r_i} \\ &= \prod_{(i,j) \in C} e(\psi_{i,j} \cdot g^{\chi_{i,j} \cdot \xi^j}, g)^{d_0 \cdot r \cdot r_i} \qquad \textbf{(18)} \\ &= e(\prod_{(i,j) \in C} \psi_{i,j}{}^{r_i}, g)^{d_0 \cdot r} \cdot e(g^{f_{\underset{A}{\rightarrow}}(\xi)}, g)^{d_0 \cdot r} \end{aligned}$$

Then,

$$\begin{aligned} \Lambda \cdot e(\theta_0{}^{-\phi}, g^r) &= e(\prod_{(i,j) \in C} \psi_{i,j}{}^{r_i}, g)^{d_0 \cdot r} \\ &\quad \cdot e(g^{f_{\underset{A}{\rightarrow}}(\xi)}, g)^{d_0 \cdot r} \cdot e(g^{-\phi}, g)^{d_0 \cdot r} \\ &= e(\varphi, \theta_0{}^r) \cdot e(g^{f_{\underset{\omega}{\rightarrow}}(\xi)}, g^{\xi - \nu})^{d_0 \cdot r} \qquad \textbf{(19)} \\ &= e(\varphi, \theta_0{}^r) \cdot e(\Gamma^r, \theta \cdot \theta_0{}^{-\nu}) \\ &= e(\varphi, \theta_0{}^r) \cdot e(\Gamma^r, \theta \cdot \theta_0{}^{-\nu}) \end{aligned}$$

- **Patient:** From the description of *SysSetup Phase* and *AuthGen Phase* in section 3.3.2, the correctness of Equation (10) and Equation (11) is intuitive. With the challenge of $Chal = \{F_i, s_{F_i}\}_{F_i \in \mathbb{F}_i}$ and proof $\pi = \sum_{F_i \in \mathbb{F}_i} s_{F_i} \cdot V_{F_i}$, we can verify the equation as follows:

$$
\begin{aligned}
& e(g, Root_p) \\
& = e(g, \prod_{F_i \in F_i} Root_{F_i}{}^{s_{Fi}}) \\
& = e(g, \prod_{F_i \in F_i} SK_t{}^{\alpha \cdot s_{Fi}}) \cdot \\
& e(g, \prod_{F_i \in F_i} (H_2(t \| T_{ID}) \cdot u^{V_{F_i}})^{\alpha \cdot s_{Fi}}) \\
& = e(g^{SK_p + SK_{TPA}}, \prod_{F_i \in F_i} (H_1(t))^{s_{Fi}}) \cdot \\
& e(\gamma, (\prod_{F_i \in F_i} H_2(t \| T_{ID})^{s_{Fi}}) \cdot u^{\sum_{F_i \in F_i} V_{F_i} \cdot s_{Fi}}) \\
& = e(PK_p \cdot PK_{TPA}, H_1(t)^{\sum_{F_i \in F} s_{Fi}}) \cdot \\
& e(\gamma, (\prod_{F_i \in F_i} H_2(t \| T_{ID})^{s_{Fi}}) \cdot u^{\pi})
\end{aligned} \qquad \textbf{(20)}
$$

Hence, the above equation holds.

5 Performance Analysis

In this section, we evaluate the performance of the proposed scheme. First of all, we compare the performance of our scheme with several classical schemes. Then, we provide the experimental results of this scheme.

5.1 Efficiency Evaluation

In the e-health system, to complete the data integrity verification of massive users more efficiently, the design scheme needs to achieve the rapid search of massive data.

When the traditional binary tree structure is used to store a large amount of data, the depth of the binary tree is enlarged. As a result, the read/write frequency of disk I/O will be high, which will lead to inefficient queries. Therefore, the hash forest structure proposed in this paper can not only solve the above problem, but also support user-differentiated auditing protocol. Besides, the proposed scheme can support multiple functions. The functionality comparison with existing related schemes is shown in Table 2. Compared with these typical schemes, our scheme can satisfy all the following properties: certificate management simplification, dynamic operation, multiuser modification, user revocation, and key-exposure resilience, while others cannot.

Table 2. Functionality comparison with existing related schemes

Schemes	Certificate management simplification	Dynamic operation	Multiuser modification	User revocation	Key-exposure resilient
Wang et al. [16]	Yes	Yes	No	No	No
Wang et al. [23]	Yes	No	No	No	No
Yu et al. [24]	Yes	Yes	No	No	Yes
Ours	Yes	Yes	Yes	Yes	Yes

5.2 Computational Cost

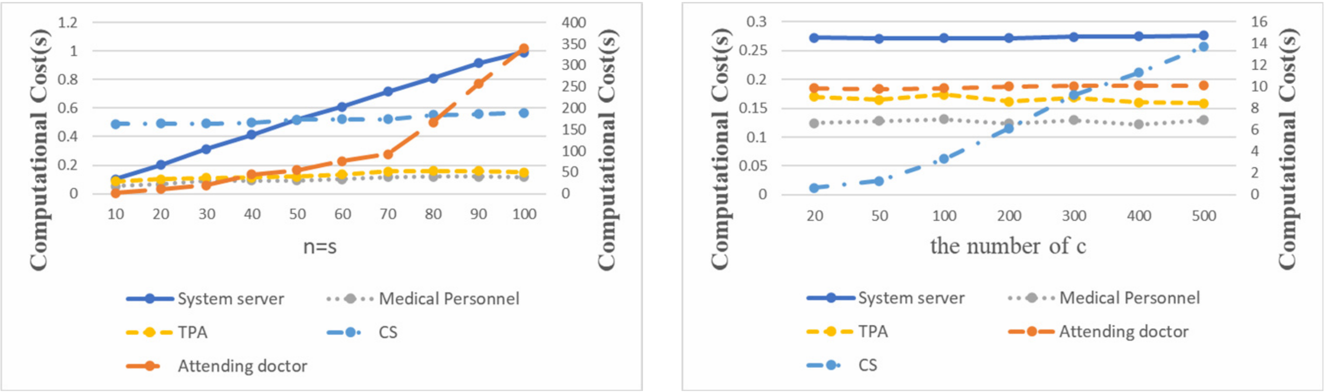
In this paper, the computation cost of hash operation is far less than that of exponentiation operation and multiplication operation. To simplify, we ignore hash operations in our evaluation in this chapter. And we use T_{exp} , T_{mul} , T_P to represent the computing cost of one exponentiation operation, one multiplication operation and one pairing operation, respectively.

It can be seen from the description of the scheme in section 3.3 that our scheme mainly includes two parts of protocol. In section 3.3.1, the system server first performs $2KT_{exp}$ operations to generate system master key, system public parameters and medical personnel's secret keys, where K is the number of medical personnel. To verify the validity of secret key, the computational cost of each medical personnel is $2T_P$ operations. In data Preprocess phase, the computational cost of the attending doctor is $2snT_{exp}$ operations and snT_{mul} operations, where n is the number of file blocks and s is the number of data sectors. In DataModify phase, we set $|\mathbb{U}_k|$ as the amount of data modified by the medical personnel, then the computational cost needs $2|\mathbb{U}_k|T_{exp}$ operations and $|\mathbb{U}_k|T_{mul}$ operations. In Upload phase, the system server conducts $(s-1)T_{mul}$ operations. In UserRevo phase, the system server performs $|\mathbb{U}_R|T_{exp}$ operations, where $|\mathbb{U}_R|$ is the number of revoked medical personnel. In Challenge phase, to generate the challenge $Chal$, the TPA conducts $|U|T_{exp}$ operations, where $|U|$ is the number of medical personnel who have modified data in the challenged file. In Prove phase, the cloud server executes cT_P operations, cT_{exp} operations and $(c-1)T_{mul}$ operations to yield a proof. Finally, in Verify phase, TPA performs $3T_P$ operations, $5T_{exp}$ operations and $3T_{mul}$ operations to verify the proof.

Similarly, in section 3.3.2, the system server performs $(p+3)T_{exp}$ operations to yield master key and public parameters of the e-health system and privacy keys of patients. And each patient executes $2T_P$

operations to verify the validity of the privacy key. In KeyUpdate phase, the computational cost of the TPA is $1T_{mul}$ operation. And each patient performs $2T_P$ operations to verify the validity of the update message, which can be done off-line. If the update message is valid, the patient conducts $1T_{exp}+1T_{mul}$ operations to update the sign secret key. In AuthGen phase, the patient performs $NT_{exp}+2NT_{mul}$ operations. In Prove phase, the TPA executes $|\mathbb{F}_i|T_{exp}+(2|\mathbb{F}_i|-1)T_{mul}$ operations. In Verify phase, the patient conducts $(2|\mathbb{F}_i|+2)T_{exp}+3T_{mul}+3T_p$ operations to verify the proof.

Figure 5 shows the computational cost of each entity in the proposed scheme, which is described in section 3.3.1. As shown in Figure 5, the computational cost of system server and attending doctor increases with the increase of n and s. Besides, the computational cost of cloud server is linearly related to the number of c.

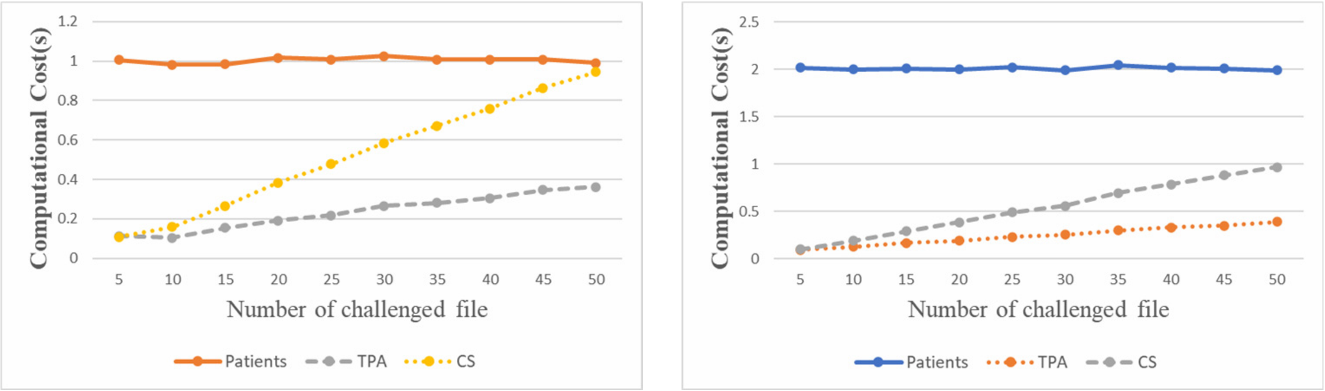


(a)

(b)

Figure 5. Computational cost of each entity in section 3.3.1

Figure 6 shows the computational cost of each entity in the proposed scheme, which is described in section 3.3.2. In this experiment, we set the number of patients as 500000, and the computational cost is about 3350 seconds. Because the computational cost of system server is only linearly related to the number of patients, we will not discuss it here. As shown in Figure 6, we set the maximum number of files owned by patients to 50 in Figure 6(a). And in Figure 6(b), we set the maximum number of files N owned by users to 100. As we can see, the computational cost of patients is were nearly one second and two seconds, respectively, which is the time required by the whole process. Besides, the computational cost of the TPA and CS is linearly related to the number of $|\mathbb{F}_i|$.



(a)

(b)

Figure 6. Computational cost of each entity in section 3.3.2

6 Conclusion

In this paper, we propose a new hashing forest structure and construct an adaptive dynamic auditing scheme that can be implemented according to the different configurations and needs of medical personnel and patients in the medical scenario. In our scheme, the proposed structure enables medical personnel to perform multi-user modification operations on relevant data. Support for the revocation of illegal users with ensuring the full availability of relevant medical data. Also, the scheme provides patients with key updates to solve the key exposure problem. Specifically, the performance analysis shows that our scheme is secure and efficient.

Further work will optimize the construction of the proposed scheme to improve the patient's batching auditing protocol. Besides, the work will be evaluated in a real-world environment.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grants No. U1836115, No. 61672295, No. 61922045, 61877034, the Natural Science Foundation of Jiangsu Province under Grant No. BK20181408, the Peng Cheng Laboratory Project of Guangdong Province PCL2018 KP004, the CICAEET fund, and the PAPD fund.

References

[1] K. Ren, C. Wang, Q. Wang, Security challenges for the public cloud, *IEEE Internet Computing*, Vol. 16, No. 1, pp. 69-73, January-February, 2012.

[2] M. Sookhak, A. Gani, H. Talebian, A. Akhunzada, S. U. Khan, R. Buyya, A. Y. Zomaya, Remote data auditing in cloud computing environments: A survey, taxonomy, and open issues, *Acm Computing Surveys*, Vol. 47, No. 4, pp. 1-34, July, 2015.

[3] Y. Deswarte, J.-J. Quisquater, A. Saïdane, Remote integrity checking, In: S. Jajodia, L. Strous (Eds.), Integrity and Internal Control in Information Systems VI. IICIS 2003. Working Conference on Integrity and Internal Control in Information Systems, Springer, Boston, Massachusetts, 2004, pp. 1-11.

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song, Provable data possession at untrusted stores, *Proceedings of the 14th Acm Conference on Computer and Communications Security*, Alexandria, Virginia, USA, 2007, pp. 598-609.

[5] A. Juels, B. S. Kaliski, Pors: Proofs of retrievability for large files, *Proceedings of the 14th ACM Conference on Computer and Communications Security*, Alexandria, Virginia, USA, 2007, pp. 584-597.

[6] H. Wang, Y. Zhang, On the knowledge soundness of a cooperative provable data possession scheme in multicloud storage, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 1, pp. 264-267, January, 2014.

[7] H. Wang, Identity-based distributed provable data possession in multicloud storage, *IEEE Transactions on Services Computing*, Vol. 8, No. 2, pp. 328-340, March-April, 2015.

[8] R. Curtmola, O. Khan, R. Burns, Robust remote data checking, *Proceedings of the 4th ACM International Workshop on Storage Security and Survivability*, Alexandria, Virginia, USA, 2008, pp. 63-68.

[9] D. Cash, A. Küpçü, D. Wichs, Dynamic proofs of retrievability via oblivious ram, *Journal of Cryptology*, Vol. 30, No. 1, pp. 22-57, January, 2017.

[10] J. Shen, J. Shen, X. Chen, X. Huang, W. Susilo, An efficient public auditing protocol with novel dynamic structure for cloud data, *IEEE Transactions on Information Forensics and Security*, Vol. 12, No. 10, pp. 2402-2415, October, 2017.

[11] L. Zhou, A. Fu, G. Yang, H. Wang, Y. Zhang, Efficient certificateless multi-copy integrity auditing scheme supporting data dynamics, *IEEE Transactions on Dependable and Secure Computing*, pp. 1-1, August, 2020.

[12] C. Wang, Q. Wang, K. Ren, W. Lou, Privacy-preserving public auditing for data storage security in cloud computing, *2010 Proceedings IEEE INFOCOM*, San Diego, California, USA, 2010, pp. 1-9.

[13] K. He, C. Huang, J. Wang, H. Zhou, X. Chen, Y. Lu, L. Zhang, B. Wang, An efficient public batch auditing protocol for data security in multi-cloud storage, *2013 8th ChinaGrid Annual Conference*, Los Alamitos, California, USA, 2013, pp. 51-56.

[14] Y. Wu, Z. L. Jiang, X. Wang, S. M. Yiu, P. Zhang, Dynamic data operations with deduplication in privacy-preserving public auditing for secure cloud storage, *IEEE International Conference on Computational Science and Engineering and IEEE International Conference on Embedded and Ubiquitous Computing*, Guangzhou, China, 2017, pp. 562-567.

[15] J. Han, Y. Li, J. Liu, M. Zhao, An efficient lucas sequence-based batch auditing scheme for the internet of medical things, *IEEE Access*, Vol. 7, pp. 10077-10092, 2019.

[16] H. Wang, D. He, J. Yu, Z. Wang, Incentive and unconditionally anonymous identity-based public provable data possession, *IEEE Transactions on Services Computing*, Vol. 12, No. 5, pp. 824-835, September-October, 2019.

[17] C. C. Erway, A. Kupcu, C. Papamanthou, R. Tamassia, Dynamic provable data possession, *ACM Transactions on Information and System Security*, Vol. 17, No. 4, Article No. 15, April, 2015.

[18] H. Tian, Y. Chen, C. C. Chang, H. Jiang, Y. Huang, Y. Chen, J. Liu, Dynamic-hash-table based public auditing for secure cloud storage, *IEEE Transactions on Services Computing*, Vol. 10, No. 5, pp. 701-714, September-October, 2017.

[19] L. Rao, H. Zhang, T. Tu, Dynamic outsourced auditing services for cloud storage based on batch-leaves-authenticated merkle hash tree, *IEEE Transactions on Services Computing*, Vol. 13, No. 3, pp. 451-463, May-June, 2020.

[20] B. Wang, B. Li, H. Li, Panda: Public auditing for shared data with efficient user revocation in the cloud, *IEEE Transactions on Services Computing*, Vol. 8, No. 1, pp. 92-106, January-February, 2015.

[21] J. Yuan, S. Yu, Public integrity auditing for dynamic data sharing with multiuser modification, *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 8, pp. 1717-1726, August, 2015.

[22] Y. Zhang, J. Yu, R. Hao, C. Wang, K. Ren, Enabling efficient

user revocation in identity-based cloud storage auditing for shared big data, *IEEE Transactions on Dependable and Secure Computing*, Vol. 17, No. 3, pp. 608-619, May-June, 2020.

[23] Y. Wang, Q. Wu, B. Qin, W. Shi, R. H. Deng, J. Hu, Identity-based data outsourcing with comprehensive auditing in clouds, *IEEE Transactions on Information Forensics and Security*, Vol. 12, No. 4, pp. 940-952, April, 2017.

[24] J. Yu, H. Wang, Strong key-exposure resilient auditing for secure cloud storage, *IEEE Transactions on Information Forensics and Security*, Vol. 12, No. 8, pp. 1931-1940, August, 2017.

Biographies

Linghong Jiang received the B.E. degree in 2018 and is currently working toward the M.E. degree in Nanjing University of Information Science and Technology (NUIST), Nanjing, China. Her research interests include data auditing and cloud computing security.

Jian Shen received the M.E. and Ph.D. degrees in Computer Science from Chosun University, South Korea, in 2009 and 2012, respectively. Since late 2012, he has been a professor at the NUIST. His research interests include public cryptography, cloud computing and security, data auditing and sharing, and information security systems.

Sai Ji received his M.S. degree from the Nanjing Aeronautics and Astronautics University (NUAA), Nanjing, China, in 2006. He works as an Associate Professor at the NUIST. His research interests are in the areas of computer measurement and control, structural health monitoring, and WSNs.

Yihui Dong received the B.E. degree in 2018 and is currently working toward the M.E. degree in the NUIST, Nanjing, China. His research interests include computer networking, cloud computing security, and IoT security.

Tiantian Miao received the B.E. degree in 2018 and is currently working toward the M.E. degree in the NUIST, Nanjing, China. Her research interests include data access control and privacy preserving in cloud computing.