Multimedia Tools and Applications (2020) 79:10851–10870
https://doi.org/10.1007/s11042-019-08558-1

Efficient data integrity auditing with corrupted data recovery for edge computing in enterprise multimedia security

Dengzhi Liu[1] · Jian Shen[1,2,3] · Pandi Vijayakumar[4] · Anxi Wang[1] · Tianqi Zhou[1]

Received: 12 April 2019 / Revised: 16 October 2019 / Accepted: 27 November 2019 /
Published online: 27 January 2020

Abstract

Enterprise multimedia can provide various services for enterprises and staffs, such as minutes of the meeting, staff training and internal news release. The format of enterprise multimedia data includes images, videos, documents, etc. The diversity and complexity of the multimedia data content increase the difficulty of storage and processing of the data. Using the nearest edge nodes to store and process the enterprise multimedia data can reduce the investment of the local storage hardware, and improve the utilization rate of peripheral dispersing devices. However, edge nodes are not fully trusted. The lack of service managers in edge computing determines that the stored enterprise multimedia data suffers from many security threats. To guarantee the security of the multimedia data in enterprises, an efficient data integrity auditing scheme is proposed for edge computing in this paper that can be used in enterprise multimedia security. Note that the technology of the homomorphic authenticator is used to construct the proposed scheme, which determines that our scheme can provide data integrity auditing with high efficiency. To reduce the financial loss caused by data corrupted for enterprise, the enterprise multimedia data are backed up in the remote cloud. Moreover, the data storage structure of One-way Linked Information Table (OLIT) is used to store the history multimedia data for enterprise, which provides high efficiency of the data recovery. In security analysis, the correctness proof is provided, and the data update auditing and the replay attack resistance are also analyzed. Performance analysis shows that our scheme has low computational cost that can be employed in the practical enterprise multimedia security.

Keywords Edge computing · Enterprise multimedia security · Data integrity auditing · Data recovery

✉ Jian Shen
s_shenjian@126.com

Extended author information available on the last page of the article

1 Introduction

The research of enterprise multimedia security was initially started from *Enterprise Security Workshop* in 2014, and then subsequently with ACM workshop in 2017 and 2018, respectively [8, 10]. From a business perspective, enterprises need to follow new technologies to innovate their production modes if they want to develop sustainably. However, security issues in enterprise multimedia constraints its further development. The security issues in enterprise multimedia arise from the combination of enterprises and network communication techniques, including IoT, big data and cloud computing [10]. To reduce the occurrence of privacy leakage issues and improve the service quality for enterprises, edge computing based enterprise multimedia is developed, which can improve the efficiency of production and management, and bring more financial benefits to enterprises [21].

With the development of enterprise information construction, the growth of multimedia data amount in enterprises shows a very high speed. On the one hand, the huge amount of enterprise multimedia data leads to a long time needed for staffs to retrieve useful data. On the other hand, the accuracy of the data retrieve is very low in huge enterprise multimedia data [4]. Hence, the security of enterprise data management and storage needs to be explored. That is to say, the multimedia security in enterprises needs to be ensured. The study of the multimedia security is a very important research direction in the field of information security. The multimedia security mainly includes five aspects, which are multimedia content confidentiality, multimedia content integrity, multimedia content availability, multimedia content controllability and non-repudiation. In the research of multimedia security, many schemes have been proposed by researchers [2, 6, 9, 12, 13, 16, 17, 19, 23, 30, 39]. Narang et al. in [23] provide a security gateway scheme that protects the copyright for the multimedia content and enhance the service security for multimedia data. Moreover, by using the technology of watermarking, a security property of anonymity to data users can be provided in Narang et al.'s scheme. In [6], multimedia data security assurance scheme is designed by Byun et al. using the watermarking and asymmetric cryptography, which can achieve the properties of confidentiality, integrity and nonrepudiation of multimedia data. Note that the multimedia data can be blinded in the encrypted digital data in [6], which can enhance the data transmission security for the multimedia data. In [9], Chen et al. proposed a real-time multimedia data transmission scheme based a chaotic system, which can compute the binary sequence for the system to highly improve the success of data transformation.

Edge computing is an effective technology for enterprise multimedia data storage [11]. Edge computing is traced back to cloud computing, which can realize the faster network service response and improve the utilization of user-centered peripheral devices [22]. Using edge computing can meet the industries' demands in real-time business, security and privacy protection [33]. Edge computing originated in the field of media, which refers to an open platform that uses the network, computation and storage to provide the nearest end services at the side near to the source of the data [22]. The applications in edge computing are initiated on the edge to generate faster network service response that can be used to satisfy the demands of industries in real-time business, intelligence and privacy protection. For the development of IOT, it means that many data and privacy controls will be implemented through local devices without being outsourced to the remote cloud by using edge computing. Moreover, the tasks will be processed at the local edge computing layer, which will greatly improve the processing efficiency and reduce the computational overhead in cloud servers [33]. In addition, because edge computing is more closer to users, the efficiency of the response to users will be high compared to cloud computing [22]. Note that security issues are the important obstacle to the further development of edge computing. Various outside security threats make the data storage no longer secure in edge computing [25, 32]. The sensitivity and the confidentiality [18, 29, 34, 35, 38, 42] of enterprise multimedia data determine that some sound security guarantee protocols need to be designed in multimedia security to improve the data storage security.

Note that the existing multimedia security schemes can deal with general security issues in practice without considering the problem of the data storage integrity. Moreover, the existing schemes cannot provide a mechanism of data recovery if the multimedia data are corrupted. In addition, the overhead of the computation and the communication in existing multimedia data security schemes is very high. Hence, a more efficient data integrity auditing with corrupted data recovery needed to be designed for edge computing in enterprise multimedia security.

1.1 Main contributions

An efficient data integrity auditing scheme with corrupted data recovery for edge computing in enterprise multimedia security is proposed in this paper. The contributions of the proposed scheme are three folds:

- The technology of the homomorphic authenticator is utilized to construct the proposed scheme, which can realize efficient data integrity auditing for edge computing in enterprise multimedia security. To improve the efficiency of multiple data blocks auditing, the proposed scheme is extended to support batch auditing.
- Our scheme supports the data recovery for the TPA if the stored data blocks cannot pass the auditing. Moreover, the data storage structure of One-way Linked Information Table (OLIT) is used to store the original data, which provides history data recovery for users.
- After the data recovery, the proposed scheme supports updated data auditing using the original security keys and parameters. In addition, the property of the replay attack resistance is provided. That is to say, the adversary cannot use the original data file authenticator to compute forged storage proofs in the proposed scheme.

1.2 Organization

This subsection presents the organization of the rest parts in this paper. Section 2 describes the related work about the proposed scheme. Section 3 provides the preliminaries that will be used. Section 4 provides the system model of the proposed scheme and the design goals. Section 5 introduces the proposed scheme in detail. Section 6 shows the evaluation results of our scheme. Finally, this paper is concluded in Section 7.

2 Related work

Many cryptography based multimedia security schemes have been studied by researchers in recent years [2, 16, 39]. In [2], Baheti et al. utilized the cyclic elliptic curve to design a symmetric encryption scheme, which can reduce the performance delay and improve the small key space in the multimedia system. Compared to previous schemes, the data encryption mechanism in [2] is more efficient in and can protect the sensitivity of secret keys. In order to optimize the multimedia data, Xiao et al. proposed an analysis model for multimedia data encryption [39]. In addition, Xiao et al. proposed a lightweight speed tunable video encryption to protect the video information in multimedia data. To support real-time encryption, Xiao et al. proposed an encryption control model, which supports selective encryption control and well encryption for multimedia data. In [16], Jung et al. used XMDR-DAI to design a data access control model, which can realize the multimedia data sharing and protect the sensitive information in multimedia data management. Note that the proposed scheme in [16] can also manage different types of multimedia content, and can well improve the reliability of multimedia data searching.

In the research of third party storage integrity checking, some studies have had a tremendous impact on academia and industries. To let users check the data stored in remote servers, the protocols of Provable Data Possession (PDP) [1] and Proof of Retrievability (POR) [15] are proposed by researchers in 2007. PDP and POR enables users to check the data storage without having to retrieve the corresponding data blocks. Specially, POR can provide retrievability of the stored data to users while data possession checking. To improve the efficiency of the stored data checking, a third party auditor (TPA) is used in the data storage checking system to achieve the stored data auditing on behalf of users [36, 37, 40]. In [37], a public auditing protocol is designed by researchers that supports privacy-preserving for users' data. Moreover, the tasks of the data auditing can be delegated to the TPA in [37]. In [36], the proposed data auditing protocol can provide data deletion, data update and data insertion by exploiting the MHT (Merkle Hash Tree). To provide the batch auditing for the system, Yang et al. utilized the properties of the bilinear pairing proposed to design a data auditing protocol and extend the designed protocol to execute the auditing for multiple servers and cloud users simultaneously [40].

Some excellent research results that have been proposed can be well used to solve various security issues in enterprise multimedia security. However, the proposed protocols in existing studies cannot be directly used in edge computing. Moreover, the mechanisms of data dynamics in existing schemes cannot support data recovery and updated data auditing, which is not suitable for edge computing in enterprise multimedia security.

3 Preliminaries

In this section, the related cryptographic tools in this paper are introduced. The presented technologies will be used to design the proposed scheme. First, the bilinear pairing is provided. Then, the homomorphic authenticator is briefly described.

3.1 Bilinear pairing

Bilinear pairing was first utilized to design the security protocols by Boneh et al. in the research of the Identity-based Encryption in 2001 [5]. After that, many security protocols are constructed based on bilinear pairing due to the high security of bilinear pairing [18, 20, 29, 36, 37, 40]. Suppose that there are three groups of G_1, G_2 and G_1 in the system. The symbol of p is the prime order of groups. The classical bilinear pairing is shown as follows: $e : G_1 \times G_2 \rightarrow G_T$. Two elements of $\mathcal{P}$ and $\mathcal{Q}$ are randomly selected from G_1. Then, randomly select a and b from $\mathbb{Z}_p^*$. Here, the symbol of $\mathbb{Z}_p^*$ is a set that contains elements of 1 to $p-1$ with modulo p. The symbols of g_1 and g_2 is the generator of gro G_T up G_1 and G_2, respectively. For any points $\mathcal{P}_1$ and $\mathcal{P}_2 \in G_1$, we have $e(\mathcal{P}_1 \cdot \mathcal{P}_2, \mathcal{Q}) = e(\mathcal{P}_1, \mathcal{Q}) \cdot e(\mathcal{P}_2, \mathcal{Q})$. The three characteristics of bilinear pairing are shown in the following:

- Bilinear: the bilinear property of bilinear pairing can is shown as $e\left(\mathcal{P}^a, \mathcal{Q}^b\right) = e(\mathcal{P}, \mathcal{Q})^{ab}$.
- Non-degenerate: the bilinear pairing of two generators cannot be equal to 1. That is to say, we have $e(g_1, g_2) \neq 1$.
- Computable: the bilinear pairing of $e(\mathcal{P}, \mathcal{Q})$ can be computed by an algorithm.

3.2 Homomorphic authenticator

The homomorphic authenticator [26] is a technology that is used in the construction of PDP [1] and POR [1]. The brief introduction of the homomorphic authenticator is shown as follows:

Suppose that one data file $DF = \{f_1, f_2, \cdots, f_n\}$ has n data blocks. Assume that two multiplicative groups are G_1 and G_2. The prime order of groups is p. Suppose that there is a bilinear pairing $e: G_1 \times G_1 \rightarrow G_2$. Randomly select a security parameter a from $\mathbb{Z}_p$, where $\mathbb{Z}_p$ denotes the set that has elements of 0 to $p-1$ with modulo p. Here, we can define that security parameter a is the system's secret key. The corresponding public key is calculated as $\tau = g^a$ using g, where g is the generator of G_1. First, the authenticator of the data file blocks is computed as $\delta_i = (H(f_i))^a$ using secret key a and the one-way hash function $H(\cdot)$. Assume that δ is the file authenticator set and s is the block set that needs to be checked. The prover can check the data storage via checking whether the verification equation $e\left(\prod_{i \in s} H(f_i), \tau\right) ?e(\delta, g)$ can hold.

4 The system model and design goals

This section provides the system model of the proposed scheme as well as the design goals. In the system model, all entities in the system are described. Moreover, the corresponding threat model is given. Then, the design goals that the designed scheme needs to realize are present.

4.1 The system model

In our scheme, there are four entities: Cloud Computing, Edge Computing, Enterprises and TPA. The system model of the proposed scheme is shown in Fig. 1. The detailed introduction of every entity is shown as follows.

(1) **Cloud Computing**. In the proposed scheme, cloud computing is used to backup the data file from edge computing. Cloud computing can provide unconstrained computing capability and storage resource to consumers. Cloud computing consists of many various distributed cloud servers. Cloud servers are idle computers and servers with no tasks execution in their memory at present. Note that the cloud servers in cloud computing can be connected through the wired or wireless network. A Cloud Service Provider (CSP) is

the manager of cloud servers [28]. Many services can be provided by cloud computing for cloud users, such as entertainment, data storage, real-time response services and computation outsourcing. In general, the purpose of CSP is to maximize financial profits. That is to say, cloud services are not free. Users should use the cloud service as the pay-as-you-use mode. Moreover, it is possible that the CSP may hide or conceal security events that affect its reputation. For instance, when some unused stored data is destroyed or deleted, the CSP will try to deceive the corresponding user that the data storage is normal so that the user can continue to use the cloud and pay for the service usage. From the aspect of the security, cloud computing is honest-but-curious to the stored data. That is to say, the cloud can store the data for users, but the cloud may try to extract the data for benefits [7, 14, 20, 31, 41]. Moreover, the infrequently used files may be discarded by the CSP to save the cloud storage.

(2) **Edge Computing**. Edge nodes in edge computing are the neighboring devices with limited storage space and computing capability. These edge nodes include wired infrastructures and wireless portable devices, which are distributed around the center and combined together through the network. The purpose of developing edge computing is to make full use of the idle resources of surrounding computing devices and respond quickly to local computing and instant storage requests. Edge computing can speed up the data processing and transmission for users. Moreover, edge computing can improve the speed of the data transmission, outsourcing and computation for users. Edge computing can store the data files for enterprise in our scheme. However, due to the limited number and low computing capacity of nodes, the storage space and the computing capability of edge computing is much less. In addition, the potential security threats of edge computing are similar to that of cloud computing.

(3) **Enterprises**. Enterprises are the users in the system. Compared to the TPA, the edge node and the cloud, the enterprise has limited storage space and computing capability. The multimedia data are generated by enterprises and uploaded to edge computing. For ease of description and understanding, enterprise is denoted by ‘user’ in the whole paper.

(4) **TPA**. In this paper, the TPA is responsible for checking the multimedia data in edge computing on behalf of enterprises. Note that the TPA can also recover the original data for enterprise in this paper. In general, the TPA is managed by an authorized department or government organization. The TPA belongs to the administrator or an agency set up by the government. Moreover, the TPA has expertise and capabilities for data auditing. In this paper, the TPA is a fully trusted entity compared to the other entities and it will never be colluded with adversaries to attack the system.

4.2 Design goals

Some goals should be provided in the design of our scheme in this paper. The design goals of the proposed scheme are listed in the following four points.

(1) **Data integrity auditing**. The proposed scheme should support the data integrity auditing. Moreover, the auditing tasks may be delegated to the TPA. Last but not least, to meet the requirement of high efficiency, the proposed scheme can be extended to support the batch auditing.

(2) **Data recovery**. If the data blocks are stored incompletely or corrupted by adversaries, the proposed scheme should provide a mechanism of original data recovery. The data recovery is executed by the TPA. Moreover, all history data can be retrieved by the TPA according to users' demands.

(3) **Update auditing**. The proposed scheme should support update auditing. That is to say, when the original data are recovered to edge computing from the cloud, the updated data blocks can be audited by using the original security keys and parameters.

(4) **Replay attack resistance**. The security property of the replay attack resistance needs to be provided in this paper. In other words, the adversary cannot generate a forged storage proofs using the original data file authenticator after the original data are retrieved to edge computing.

5 The proposed scheme

Note that seven main phases are included in our scheme. For ease of understanding, a high description is provided before the detailed introduction of our scheme.

5.1 High description

In the proposed scheme, the technology of homomorphic authenticator is used to construct the auditing mechanism for edge computing in enterprise multimedia security. Moreover, to reduce the financial loss caused by the data corrupted in enterprise, a data backup method is proposed, which can retrieve the history data for users if the stored data in edge computing are stored incompletely or corrupted. There are seven phases in the proposed scheme, which are *Key Generation*, *Data Uploading and Backup*, *Challenge Generation*, *Proof Generation*, *Auditing, Data Recovery* and *Batch Auditing*. The phase of *Key Generation* is mainly focused on generating and computing the security parameters for the construction of our scheme. Note that the input of this phase determines the length of security parameters and keys. In *Data Uploading and Backup*, the pre-processed data files are uploaded to edge nodes as well as the tags generated according to the identifier of the data file. Moreover, the stored data will also be outsourced to the remote storage center (Cloud Computing) by edge nodes for backup. In the phase of *Challenge Generation*, the data integrity auditing challenge can be generated by users and transmitted to the TPA for further storage auditing. The auditing challenge generated by users including the number of auditing data blocks and security parameters for storage proofs computing. Upon receiving the challenge, the edge node needs to compute the storage proofs according to the stored authenticators and data files in the phase of *Proof Generation*. After that, the storage proofs are sent back to the TPA. Finally, the TPA can check the data blocks storage proofs using the public keys. If the auditing phase outputs fail, the original data blocks can be recovered by retrieving from the backup. In the last phase, the basic auditing scheme is extended to batch auditing, which is more efficient for multiple users to audit multiple data blocks.

5.2 The detailed scheme

This subsection introduces the construction of the proposed scheme in detail. The main process is shown in Fig. 2.

(1) **Key Generation**. The input of this phase is a security parameter λ, which determines the length of the selected security keys and parameters in the scheme construction. The outputs of this phase are $\{G_1,G_2,e,H,g\}$. The main process of this phase is shown as follows. First, a security parameter a is randomly chosen by the user from $\mathbb{Z}_p$. Here, the parameter of p is a large prime order and security parameter$a$ is the secret key. Randomly select generator g from G_1. The bilinear map is denoted as e, where $e : G_1 \times G_1 \rightarrow G_2$. Then, the user computes $\tau = g^a$ as the public key. The singing keys of sk_s and pk_s are randomly selected by the user. The tag of the data file can be computed as $t_f = Id_f \| SSig_{sk_s}\left(Id_f\right)$, where Id_f is denoted as the identifier of the user's data file and $SSig$ is a signing algorithm. The original information of $SSig_{sk_s}$ can be recovered using pk_s. The data file of the user can be denoted as $DF = \{f_1, f_2, \cdots, f_n\}$. Here, the symbol of n is the number of data blocks in the whole data file. After that, the user randomly selects security parameter η from the group of G_1. The authenticator of the data file can be computed as $\delta_i = \left(H\left(Id_f \| T_i\right)\cdot\eta^{f_i}\right)^a \in G$, where T_i is the timestamp of this round. The public parameters of the system are $\{\eta,\tau,g,pk_s\}$.

(2) **Data Uploading and Backup**. The tag set $\{\delta_i\}_{1 \le i \le n}$ and data file DF are uploaded to the servers in edge computing. To avoid the data loss caused by server attacks or deliberate deletion by edge nodes, the uploaded data file will also be sent to the remote storage center, namely, cloud servers. The storage structure consists of the index table and

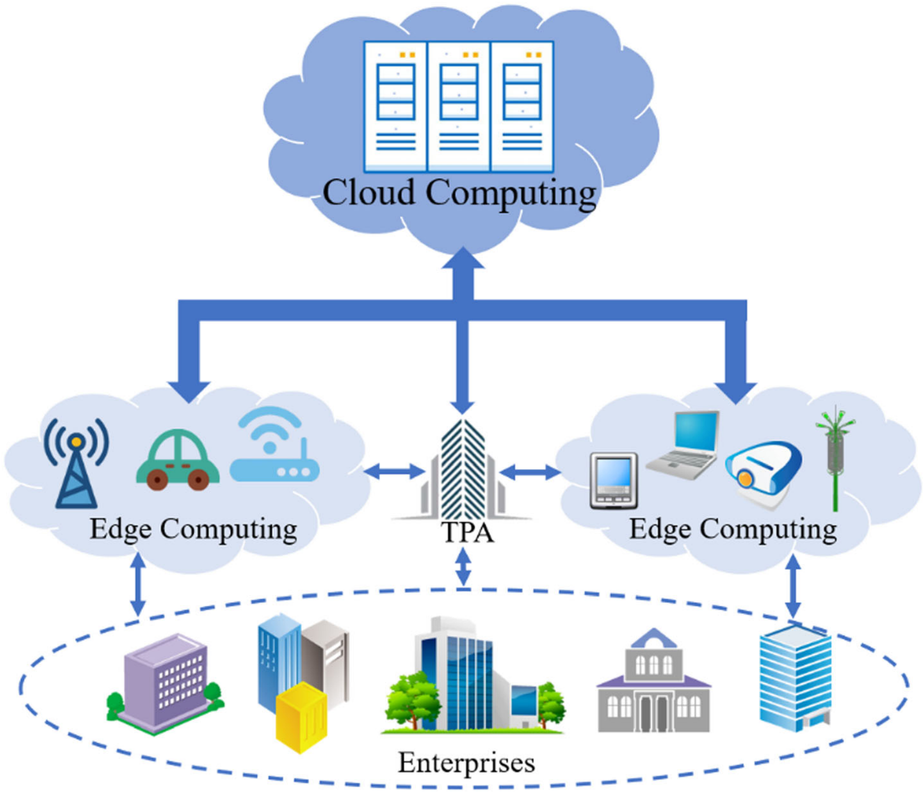


Fig. 1 The system model

storage pool. The storage structure is shown as Fig. 3. In the index table, the storage ID includes the edge node number and user ID information. The description in Fig. 3 demonstrates the data information, including the information of the data owner, data generation time, location and so on. The user can retrieve the original data files to the edge node according to the keyword search and storage ID search. The version number is the data version. The version number will be increased if the data are updated. Note that the new data will be appended after the last version of this data file. Moreover, the history data and the corresponding version number are also stored in the cloud for the further data usage. Because the data file stored in an edge node is not very large, the OLIT is used to store data blocks. Compared to the storage structure of DLIT (Doubly Linked Information Table) [27], the OLIT does not need to store and maintain a large number of pointers, which can highly save the storage space for the system.

(3) **Challenge Generation**. The storage integrity proofs are computed based on the request challenge from the user side. Before auditing the data, the tag t_f of the data needs to be retrieved from edge nodes by the TPA and then the correctness of the tag will be verified using pk_s. If the verification of the tag can be passed, the TPA will recover the data identifier of from the file tag. Assume that m data blocks stored in the edge node need to be audited, the user needs to generate an audit request firstly, and then m elements $\{s_1, s_2, \cdots, s_m\}$ are randomly chosId_fen by the TPA according to the audit request. After that, the TPA selects parameters c_i rand$Chal = \{i, c_i\}$omly from $\mathbb{Z}_p$ for every element s_i, where $1 \leq i \leq m \leq n$. The auditing challenge can be denoted as $Chal = \{i, c_i\}$.

(4) **Proof Generation**. In this phase, the storage proof can be generated in the cloud. Upon receiving $Chal = \{i, c_i\}$, the cloud server can compute the storage proofs as $p_t = \prod_{i\in[1,s_m]} \delta_i^{c_i}$ and $p_f = \sum_{i\in[1,s_m]} (c_i \cdot f_i)$ based on the stored tags and data files. Finally, the tag proof and the data proof are transmitted back to the TPA.

(5) **Auditing**. When the TPA gets the storage proofs, it can judge the correctness of the received proofs using public parameters η, g and public key τ. Then, the TPA computes an aggregated auditing parameter A_f as $A_f = \prod_{i\in[1,s_m]} e\left(\left(H\left(Id_f \| T_i\right)\right)^{c_i}, \tau\right)$ for the data file.

The proofs verification equation is shown as Eq. (1).

$$A_f \cdot e(\eta, \tau)^{p_f} ? e(p_t, g) \tag{1}$$

(6) **Data Recovery**. If the above basic auditing outputs a fail, the TPA can be sure that that one or more data blocks in the auditing challenge blocks have been corrupted. The TPA can send a data recovery request R_{s_m} to the user. On the one hand, if the user is agree with the TPA to retrieve the original data blocks, he/she will response 'Y'; otherwise, the TPA cannot retrieve the original data to the edge node side. On the other hand, the user can delegate the TPA to audit the blocks using binary search [3] until the corrupted or incomplete data blocks are located. Moreover, the user needs to update the authenticator as $\delta_i = \left(H\left(Id_f \| T_{i+1}\right) \cdot \eta^{f_i}\right)^a$ if the data file is retrieved to the edge node side.

(7) **Batch Auditing**. To reduce the computational overhead in the condition of multiple auditing tasks, we extend our scheme to support batch auditing. Suppose that k users delegate auditing tasks to the TPA, and every user audits s_m data blocks. The process of key generation and challenge generation is similar to that in the basic auditing. The detailed

introduction is omitted. Upon receiving the auditing challenge from the TPA, the cloud can calculate aggregated storage proofs as $p_{jf} = \sum_{i\in[1,s_m]}\left(c_{ji}\cdot f_{ji}\right)$ and $p_{jt} = \prod_{i\in[1,s_m]}\delta_{ji}^{\ c_{ji}}$. Note that p_{jf} and $p_{jt}$are user j‘s data file proof and tag proof. Then, p_{jf} and p_{jt} are transmitted back to the TPA. When receives the proofs, the aggregated data file information can be computed as $A_{jf} = \prod_{i\in[1,s_m]}\left(\prod_{j\in[1,k]} e\left(\left(H\left(Id_{jf}\|T_{ji}\right)\right)^{c_{ji}}, \tau\right)\right)$ by the TPA for user j‘s data file. Here, $1 \le j \le k$. The correctness of storage proofs can be verified by Eq. (2).

$$A_{jf}\cdot \prod_{j\in[1,k]} e(\eta,\tau)^{p_{jf}} ? \prod_{j\in[1,k]} e\left(p_{jt}, g\right) \tag{2}$$

6 Evaluation

This section presents the evaluation of the proposed scheme, including security analysis and performance analysis. First, the correctness proof of our scheme is provided as well as the

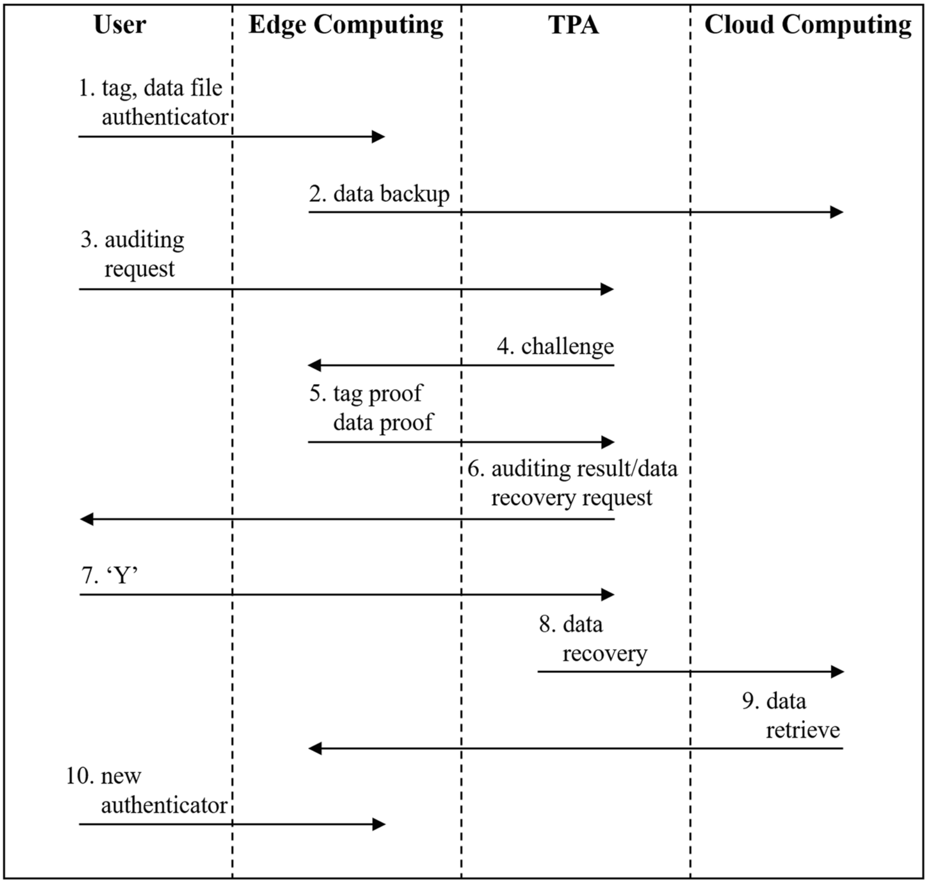
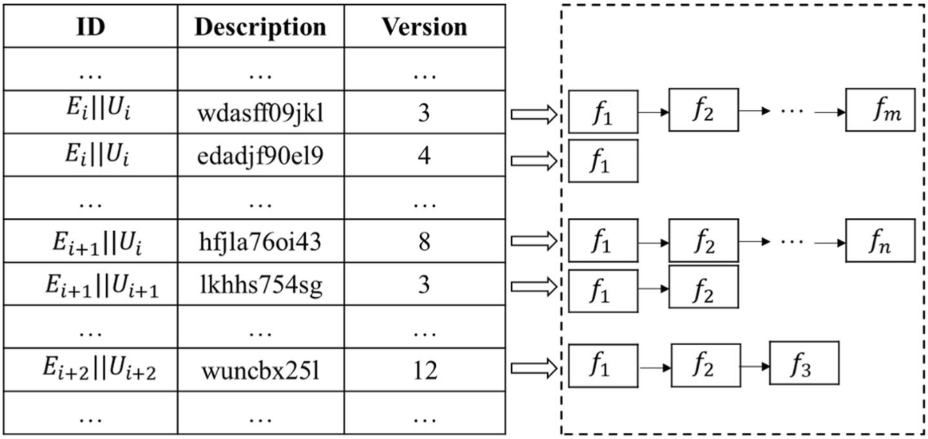


Fig. 2 The main process of our scheme



ID	Description	Version
…	…	…
$E_i \| \| U_i$	wdasff09jkl	3
$E_i \| \| U_i$	edadjf90el9	4
…	…	…
$E_{i+1} \| \| U_i$	hfjla76oi43	8
$E_{i+1} \| \| U_{i+1}$	lkhhs754sg	3
…	…	…
$E_{i+2} \| \| U_{i+2}$	wuncbx251	12
…	…	…

Fig. 3 The storage structure of the data file backup in the cloud

security properties in security analysis. Then, the performance of our scheme is analyzed in performance analysis.

6.1 Security analysis

Theorem 1 The proposed scheme is correct if all the security parameters are correctly generated and computed in the construction of the proposed scheme.

Proof Our scheme can be proved to be correct by proving the correctness of the basic auditing and the batch auditing. By elaborating Eq. (1) and Eq. (2), if the two equations can hold, the correctness of the basic auditing and the batch auditing can be proved. First of all, assume that all the security parameters and security keys in this paper are generated correctly. Equation (1) can be elaborated as follows:

$$\begin{aligned}
A_f \cdot e(\eta, \tau)^{p_f} &= \prod_{i\in[1,s_m]} e\left(\left(H\left(Id_f \| T_i\right)\right)^{c_i}, \tau\right) \cdot e(\eta, \tau)^{p_f} \\
&= e\left(\prod_{i\in[1,s_m]} \left(H\left(Id_f \| T_i\right)\right)^{c_i}, g^a\right) \cdot e\left(\eta^{\sum_{i\in[1,s_m]} (c_i \cdot f_i)}, g^a\right) \\
&= e\left(\prod_{i\in[1,s_m]} \left(H\left(Id_f \| T_i\right)\right)^{c_i}, g^a\right) \cdot e\left(\prod_{i\in[1,s_m]} \eta^{c_i \cdot f_i}, g^a\right) \\
&= e\left(\left(\prod_{i\in[1,s_m]} \left(H\left(Id_f \| T_i\right)\right)^{c_i} \cdot \prod_{i\in[1,s_m]} \eta^{c_i \cdot f_i}\right)^a, g\right) \\
&= e\left(\prod_{i\in[1,s_m]} \left(\left(H\left(Id_f \| T_i\right) \cdot \eta^{f_i}\right)^a\right)^{c_i}, g\right)
\end{aligned}$$

Because $\delta_i = \left(H\left(Id_f \| T_i\right) \cdot \eta^{f_i}\right)^a$ and the right-hand side of Eq. (1) can be elaborated as $e(p_t, g) = e\left(\prod_{i=1}^{s_m} {\delta_i}^{c_i}, g\right)$, Eq. (1) can hold. That is to say, the basic auditing is correct.

The correctness of Eq. (2) can be used to prove the correctness of the batch auditing. The elaboration of Eq. (2) is shown in the following:

$$\begin{aligned}
A_{jf}\cdot \prod_{j\in[1,k]} e(\eta,\tau)^{p_{jf}} &= \prod_{j\in[1,k]}\left(\prod_{i\in[1,s_m]} e\left(\left(H\left(Id_{jf}\|T_{ji}\right)\right)^{c_{ji}},\tau\right)\right)\cdot \prod_{j\in[1,k]} e(\eta,\tau)^{p_{jf}} \\
&= \prod_{j\in[1,k]} e\left(\prod_{i\in[1,s_m]}\left(H\left(Id_{jf}\|T_{ji}\right)\right)^{c_{ji}},g^a\right)\cdot \prod_{j\in[1,k]} e\left(\eta^{\sum\limits_{i\in[1,s_m]}\left(c_{ji}\cdot f_{ji}\right)},g^a\right) \\
&= \prod_{j\in[1,k]} e\left(\prod_{i\in[1,s_m]}\left(H\left(Id_{jf}\|T_{ji}\right)\right)^{c_{ji}},g^a\right)\cdot \prod_{j\in[1,k]} e\left(\prod_{i\in[1,s_m]}\eta^{c_{ji}\cdot f_{ji}},g^a\right) \\
&= \prod_{j\in[1,k]} e\left(\left(\prod_{i\in[1,s_m]}\left(H\left(Id_{jf}\|T_{ji}\right)\right)^{c_{ji}}\cdot \prod_{i\in[1,s_m]}\eta^{c_{ji}\cdot f_{ji}}\right)^a,g\right) \\
&= \prod_{j\in[1,k]} e\left(\prod_{i\in[1,s_m]}\left(\left(H\left(Id_{jf}\|T_{ji}\right)\cdot\eta^{f_{ji}}\right)^a\right)^{c_{ji}},g\right)
\end{aligned}$$

Similar to the description in the basic auditing proof, we can get that $\delta_{ji}=\left(H\left(Id_{jf}\|T_{ji}\right)\cdot\eta^{f_{ji}}\right)^a$. The Eq. (2) can continue to be elaborated as the following description:

$$\prod_{j\in[1,k]} e\left(\prod_{i\in[1,s_m]}\left(\left(H\left(Id_{jf}\|T_{ji}\right)\cdot\eta^{f_{ji}}\right)^a\right)^{c_{ji}},g\right) = \prod_{j\in[1,k]} e\left(\delta_{ji}{}^{c_{ji}},g\right) = \prod_{j\in[1,k]} e\left(p_{jt},g\right)$$

Hence, it can be summarized that the correctness of the batch auditing is our scheme can be proved.

From the elaboration result in Eqs. (1) and (2), it can be summarized that the proposed scheme is correct □.

Theorem 2 The proposed scheme supports update auditing after the data recovery.

Proof The corrupted data can be recovered by retrieving history data from the cloud. Suppose that the recovered data file is $DF^*=\{f_1^*,f_2^*,\cdots,f_n^*\}$. The corresponding data file authenticator is $\delta_i^*=\left(H\left(Id_f\|T_{i+1}\right)\cdot\eta^{f_i^*}\right)^a$. The auditing challenge can be sent to the cloud by the

Table 1 The computational cost

Phase of the Scheme	Computational Cost
KeyGen	$3T_{E.}+1TH_{.}+1\eta T_{Mul.}$
ProofGen	$nT_{E.}+2nT_{Mul.}+nT_{Add.}$
Auditing	$(n+1)T_{E.}+(2+n)T_{P.}+nT_{H.}+1T_{Mul.}$
Batch Auditing	$(2+k)nT_{E.}+(2+k)nT_{P.}+nkT_{H.}+(2n+1)T_{Mul.}+nT_{Add.}$

* $T_{E.}$, $T_{H.}$, $T_{Mul.}$, $T_{Add.}$, $T_{P.}$: The time required to execute the corresponding operations

* n: The number of auditing blocks

* k: The number of users

TPA, and the data storage proofs are computed as $p_t^* = \prod_{i\in[1,s_m]} {\delta_i}^{*c_i}$ and $p_f^* = \sum_{i\in[1,s_m]} \left(c_i \cdot f_i^*\right)$. Then, the TPA can check the correctness of the proofs of p_t^* and p_f^* using public parameters η, g and public key τ. That is to say, the recovered data can also be audited by using the previous public parameters and public keys. Hence, the proposed scheme supports update auditing after the data recovery.□.

Theorem 3 The property of the replay attack resistance is provided in the proposed scheme.

Proof When the data are recovered to the side of the edge node, the data file authenticator is updated by using new timestamp as $\delta_i = \left(H\left(Id_f \| T_{i+1}\right) \cdot \eta^{f_i}\right)^a$. That is to say, the adversary cannot use the original data file authenticator to forge storage proofs. Despite the adversary can get the timestamp of T_{i+1}, identifier Id_f and security parameter , he/she still cannot compute the correct authenticator for the replay attack. □.

6.2 Performance analysis

6.2.1 Theoretical analysis

The theoretical analysis of the proposed scheme is provided from comparation analysis. For ease of understanding, we utilize the symbols of $T_{E.}$, $T_{H.}$, $T_{Mul.}$, $T_{Add.}$, T_P to denote the computational time that the operations in the proposed scheme need to cost. The operations of symbols $T_{E.}$, $T_{H.}$, $T_{Mul.}$, $T_{Add.}$, T_P are time required that the system executes exponentiation, hash to point, multiplication, addition and pairing map. Table 1 shows the computational cost. We can see that the computational cost of *Key Generation* is $3T_{E.} + 1TH_{.} + 1T_{Mul.}$ in Table 1.

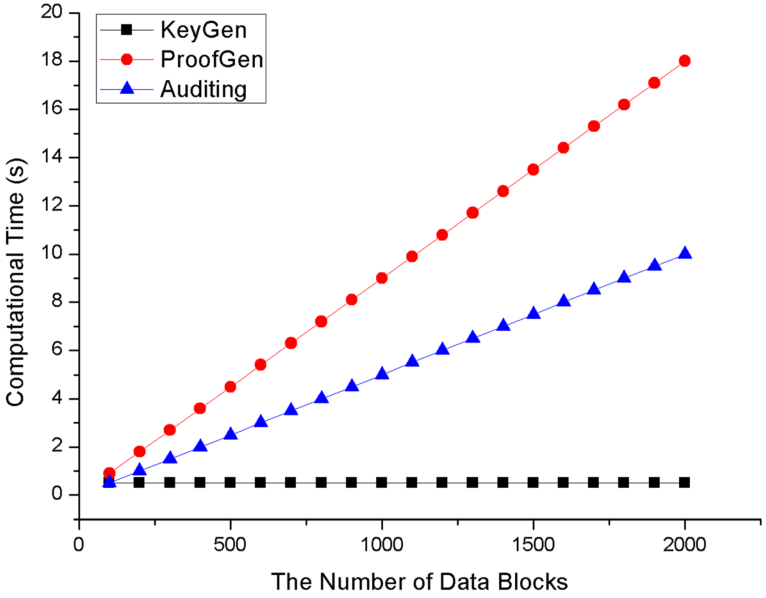


Fig. 4 The computational time of the basic scheme with the increase of the audited data blocks

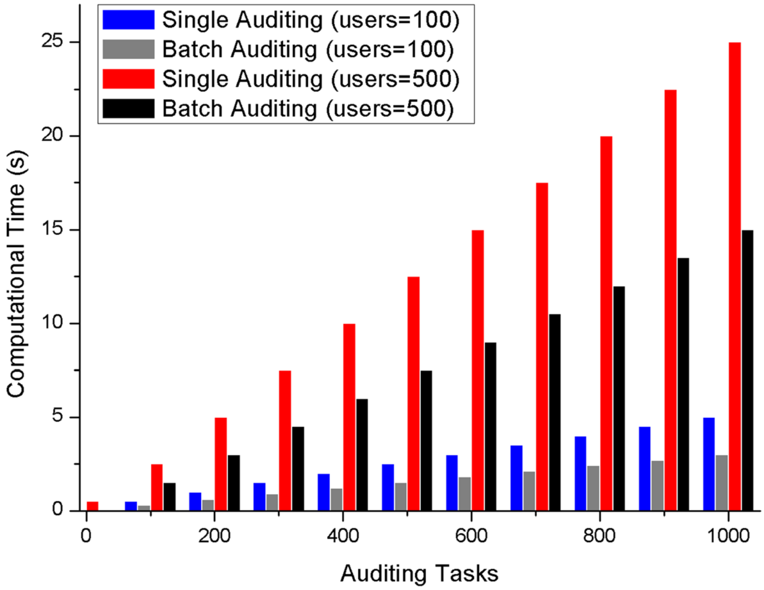


Fig. 5 The computational time of single auditing and batch auditing under different auditing tasks

Note that the computational cost in the phase of *Key Generation* is independent of the auditing blocks. The computational cost of *Proof Generation* and *Auditing* will be increased with the

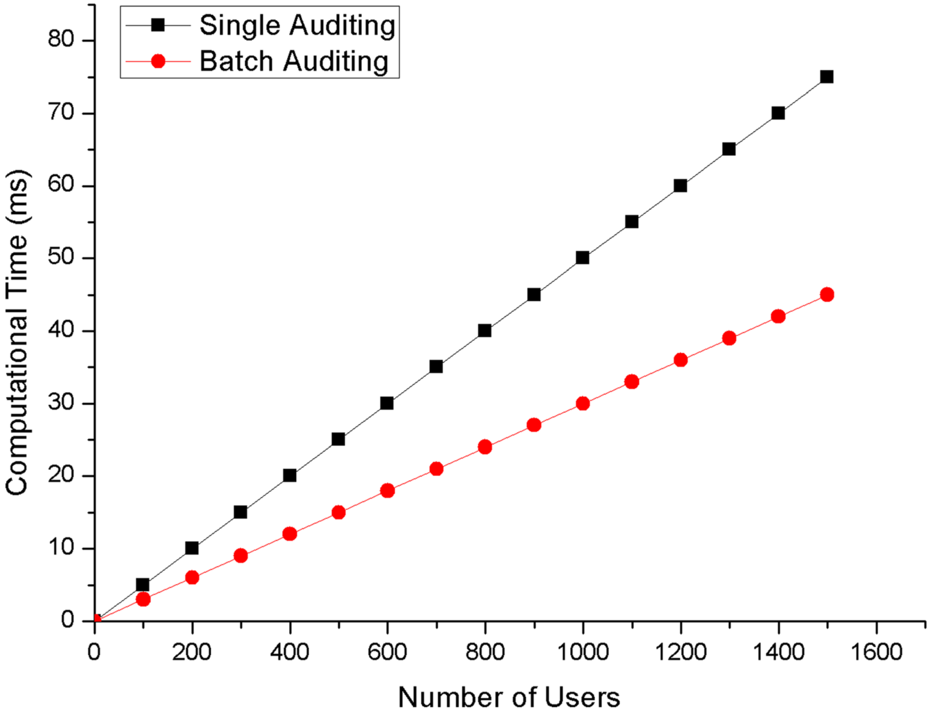


Fig. 6 The computational time of the single auditing and the batch auditing

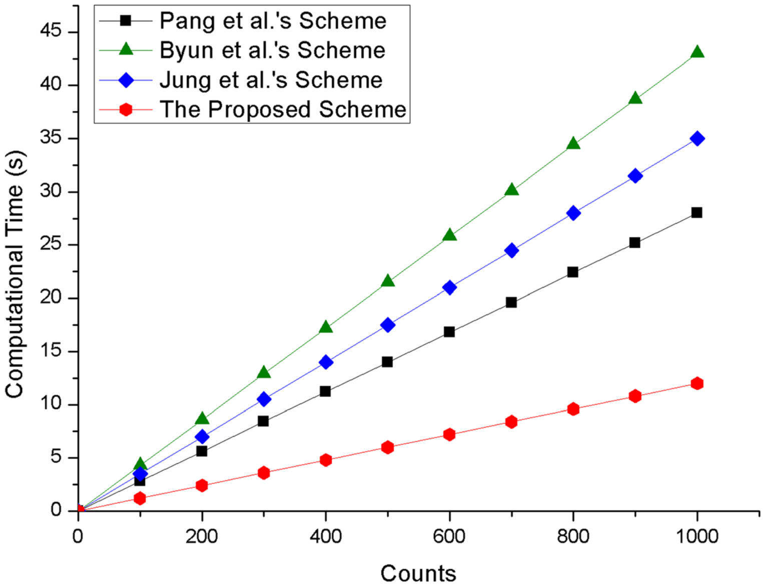


Fig. 7 The comparison of the computational cost at the user side

gorwth of the auditing block amount. With the increase of the number of auditing blocks and users, the computational cost of *Batch Auditing* will be large. Although there are some additional operations of hash to point in the phase of *Auditing*, the computational cost of hash to point is negligible compared to the operation of multiplication. Hence, it can be summarized that the phase of *Proof Generation* costs much computational time compared to *Auditing*.

6.2.2 Simulation analysis

An experimental platform is built based on the libraries of GMP[1] and PBC[2] to simulate cryptography schemes. The experimental computer is configured with Linux system, the memory is 8GB and the CPU is 2.60 GHZ. Note that C programming language is used to simulate the proposed scheme and similar schemes on the experimental platform in this paper.

In the simulation of the basic scheme, the number of the auditing blocks are 100 to 2000. The simulation result of the basic scheme is shown in Fig. 4. In Fig. 4, *KeyGen* denotes the phase of the key generation and *ProofGen* denotes the phase of the proof generation. The axis of x and y in Fig. 4 is the number of data blocks and computational time. Note that the computational time in *KeyGen* is a constant. With the increase of data blocks, *ProofGen* costs much computational time compared to *Auditing*. Moreover, the computational time of *ProofGen* is more than that of *KeyGen* when the number of data blocks is same. The comparison of the computational cost in the basic auditing and the batch auditing is also given in Fig. 5. The axis of x and y in Fig. 5 is the number of auditing tasks and computational time, respectively. The number of data blocks is defined as 100 in the simulation. Figure 5 also shows that the computational time is increased with the increase of the auditing tasks.

[1] The GNU Multiple Precision Arithmetic Library: http://gmplib.org/
[2] The Pairing-Based Cryptography Library: http://crypto.stanford.edu/pbc/

The batch auditing costs less computational time compared to the single auditing when the number of users is different. In addition, the computational time will also be increased accordingly if the number of users increases. Although the computational cost will be increased with the increase of the auditing tasks, the computational time is still acceptable.

To demonstrate the superiority of the batch auditing clearly, the computational time of the single auditing and the batch auditing is simulated under the different number of users. Figure 6 shows the simulation result. Note that the *x*-axis is the number of users and the *y*-axis is the corresponding computational time in the simulation in Fig. 6. Here, we define the number of data blocks that every user audits is 100. From Fig. 6, the computational time of the batch auditing is always less than that of the single auditing. Moreover, as the amount of users increases, the advantage of the batch auditing becomes more and more obvious. Therefore, it can be concluded that the batch auditing is very important in improving the efficiency of the system.

We also compare the user side's time cost in our scheme with that in the similar schemes [6, 16, 24]. Figure 7 shows the simulation result. In Fig. 7, the *x*-axis and the *y*-axis denotes computing counts and time cost, respectively. As the simulation results show in Fig. 7, our scheme and the three comparative schemes cost much computational time with the increase of counts. Moreover, Byun et al.'s scheme [6] costs much computational time compared to the proposed scheme and similar schemes [16, 24] when the counts are same. Note that under the condition of the counts 1000, our scheme costs about 12 s and the three comparative schemes [6, 16, 24] costs about 43 s, 35 s, 28 s. That is to say, with the increase of the computing counts, our scheme will be more efficient compared to the comparative schemes [6, 16, 24]. Hence, the proposed scheme in this paper can be executed with low computational cost compared to the similar schemes [2, 16, 24].

7 Conclusion

An efficient data integrity auditing scheme is proposed in this paper, which supports corrupted data recovery for edge computing in enterprise multimedia security. The technology of the homomorphic authenticator is utilized to construct the proposed scheme. To alleviate the user side's time cost, a fully trusted entity TPA is utilized to assist the data auditing in edge computing. To reduce the loss caused by data loss for users, a data recovery mechanism is provided, which can backup the data for users in the cloud based on the data structure of One-way Linked Information Table (OLIT). From the security analysis, we can get that our scheme can be proved to be correct. Moreover, the replay attack resistance and update auditing can be provided in the proposed scheme. In the performance analysis, the theoretical analysis and the simulation analysis are presented. From the performance analysis result, it can be summarized that our scheme can be executed with high efficiency, which determines that our scheme is practical and can be used in related application scenarios for edge computing in enterprise multimedia security.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Grants No. U1836115, No. 61672295, No. 61922045, No. 61672290, the Natural Science Foundation of Jiangsu Province under Grant No. BK20181408, the Foundation of State Key Laboratory of Cryptology under Grant No. MMKFKT201830, the Peng Cheng Laboratory Project of Guangdong Province PCL2018KP004, the CICAEET fund, and the PAPD fund, the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant SJKY19_0955.

References

1. Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, Song D (2007) Provable data possession at untrusted stores. ACM conference on computer and communications security pp 598-609
2. Baheti A, Singh L, Khan AU (2014) Proposed method for multimedia data security using cyclic elliptic curve, chaotic system and authentication using neural network. International conference on communication systems and network technologies pp 664-668
3. Bentley JL (2006) Multidimensional binary search trees in database applications. IEEE Trans Softw Eng SE-5(4):333–340
4. Bimonte S, Tchounikine A, Miquel M (2005) Towards a spatial multidimensional model. ACM international workshop on data warehouse and OLAP pp 39-46
5. Boneh D, Franklin MK (2001) Identity-based encryption from the weil pairing. In: International cryptology conference on advances in cryptology, pp 213–229
6. Byun S, Ahn B (2003) More on the multimedia data security for e-commerce. International conference on information technology: research & education pp 412-415
7. Chang V (2018) Data analytics and visualization for inspecting cancers and genes. Multimed Tools Appl 77(14):17693–17707
8. Chang V, Ramachandran M, Yen NN, Walters RJ, Wills G (2016) The second international workshop on enterprise security. In: Proceedings-IEEE 7th international conference on cloud computing technology and science, pp 16–17
9. Chen H, Guo J, Huang L, Yen J (2013) Design and realization of a new signal security system for multimedia data transmission. J Appl Signal Process 13:1291–1305
10. Duncan B, Whittington M, Chang V (2017) Enterprise security and privacy: why adding IoT and big data makes it so much more difficult. In: International conference on engineering and technology (ICET), pp 1–7
11. Han J, Stefanovic N, Koperski K (1998) Selective materialization: an efficient method for spatial data cube construction. Pacific-Asia conference on knowledge discovery and data mining pp 144–158
12. Hsia C-H (2018) Improved finger vein pattern method using wavelet-based for real-time personal identification system. J Imaging Sci Technol 62(3):30402-1–30402-8
13. Hsia C-H (2018) New verification method for finger-vein recognition system. IEEE Sensors J 18(2):790–797
14. Hsia C-H, Guo J-M, Wu C-S (2017) Finger-vein recognition based on parametric-oriented corrections. Multimed Tools Appl 76(23):25179–25196
15. Juels A, Kaliski BS (2007) Pors: proofs of retrievability for large files. ACM conference on computer and communications security pp 584–597
16. Jung K, Moon S, Kim J (2017) Data access control method for multimedia content data sharing and security based on XMDR-DAI in mobile cloud storage. Multimed Tools Appl 76:19983–19999
17. Li X, Tang S, Xu L, Wang H, Chen J (2017) Two-factor data access control with efficient revocation for multi-authority cloud storage systems. IEEE Access 5:393–405
18. Liu D, Shen J, Wang A, Wang C (2018) Lightweight and practical node clustering authentication protocol for hierarchical wireless sensor networks. Int J Sens Netw 27(2):95–102
19. Liu D, Shen J, Wang A, Wang C (2019) Secure real-time image protection scheme with near-duplicate detection in cloud computing. J Real-Time Image Process:1–10. https://doi.org/10.1007/s11554-019-00887-6
20. Liu D, Shen J, Chen Y, Wang C, Zhou T, Wang A (2019) Privacy-preserving data outsourcing with integrity auditing for lightweight devices in cloud computing. In: International conference on information security and cryptology, pp 223–239
21. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv Tutorials 19(3):1628–1656
22. Miao Y, You C, Zhang J, Huang K, Letaief KB (2017) A survey on mobile edge computing: the communication perspective. IEEE Commun Surv Tutorials 19(4):2322–2358
23. Narang MS, Grove PS, Kuushik S (2000) Multimedia security gateway protocol to achieve anonymity in delivering multimedia data using watermarking. IEEE international conference on multimedia & expo. pp 529-532
24. Pang H, Tan K (2004) Authenticating query results in edge computing. In: International conference on data engineering, pp 560–571
25. Satyanarayanan M (2017) The emergence of edge computing. Computer 50(1):30–39
26. Shacham H, Waters B (2008) Compact proofs of retrievability. In: International conference on theory and application of cryptology and information security, pp 90–107
27. Shen J, Shen J, Chen X, Huang X, Susilo W (2017) An efficient public auditing protocol with novel dynamic structure for cloud data. IEEE Trans Inf Forensics Secur 12(10):2402–2415

28. Shen J, Liu D, Lai C, Ren Y, Wang J, Sun X (2017) A secure identity-based dynamic group data sharing scheme for cloud computing. J Internet Technol 18(4):833–842
29. Shen J, Liu D, Bhuiyan MZA, Shen J, Sun X, Castiglione A (2017) Secure verifiable database supporting efficient dynamic operations in cloud computing. IEEE Trans Emerg Top Comput. https://doi.org/10.1109/TETC.2017.2776402
30. Shen J, Wang C, Lai J, Xiang Y, Li P (2019) CATE: cloud-aided trustworthiness evaluation scheme for incompletely predictable vehicular ad hoc networks. IEEE Trans Veh Technol. https://doi.org/10.1109/TVT.2019.2938968
31. Shen J, Liu D, Chen X, Li J, Kumar N, Vijayakumar P (2019) Secure real-time traffic data aggregation with batch verification for vehicular cloud in VANETs. IEEE Trans Veh Technol. https://doi.org/10.1109/TVT.2019.2946935
32. Shi W, Dustdar S (2016) The promise of edge computing. Computer 49(5):78–81
33. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: vision and challenges. IEEE Internet Things J 3(5):637–646
34. Tang S, Xu L, Liu N, Huang X, Ding J, Yang Z (2014) Provably secure group key management approach based upon hyper-sphere. IEEE Trans Parallel Distrib Syst 25(12):3253–3263
35. Tang S, Li X, Huang X, Xiang Y, Xu L (2016) Achieving simple, secure and efficient hierarchical access control in cloud computing. IEEE Trans Comput 65(7):2325–2331
36. Wang Q, Wang C, Ren K, Lou W, Li J (2011) Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans Parallel Distrib Syst 22(5):847–859
37. Wang C, Chow SM, Wang Q, Ren K, Lou W (2013) Privacy-preserving public auditing for secure cloud storage. IEEE Trans Comput 62(2):362–375
38. Wang C, Shen J, Lai C, Huang R, Wei F (2018) Neighborhood trustworthiness based vehicle-to-vehicle authentication scheme for vehicular ad hoc networks. Concurr Comput Pract Exp. https://doi.org/10.1002/cpe.4643
39. Xiao C, Wang L, Jie Z, Chen T (2016) A multi-level intelligent selective encryption control model for multimedia big data security in sensing system with resource constraints. International conference on cyber security and cloud computing pp 148-153
40. Yang K, Jia X (2013) An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Trans Parallel Distrib Syst 24(9):1717–1726
41. Zhang M, Yao Y, Li B, Tang C (2018) Accountable mobile e-commerce scheme in intelligent cloud system transactions. J Ambient Intell Humaniz Comput 9:1889–1899. https://doi.org/10.1007/s12652-017-0672-4
42. Zhou T, Shen J, Li X, Wang C, Tan H (2018) Logarithmic encryption scheme for cyber physical systems employing fibonacci q-matrix. Future Gener Comput Syst. https://doi.org/10.1016/j.future.2018.04.008

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dengzhi Liu received the M.E. degree from Nanjing University of Information Science and Technology in 2017. He is currently working toward the Ph.D. degree in the School of Computer and Software, Nanjing University of Information Science and Technology. His current research interests include applied cryptography, network and data security, and cloud computing security.

Jian Shen received the M.E. and Ph.D. degrees in Computer Science from Chosun University, Korea, in 2009 and 2012, respectively. Since late 2012, he has been a professor at Nanjing University of Information Science and Technology, Nanjing, China. His research interests include computer networking, and information security systems.

Pandi Vijayakumar received the BEng degree from Madurai Kamarajar University in 2002, the MEng degree in computer science and engineering from the Karunya Institute of Technology in 2005, and the PhD degree in computer science and engineering from Anna University, Chennai, in 2013. He was the former dean of University College of Engineering, Tindivanam and presently working as Assistant Professor in the department of Computer Science and Engineering. He is guiding many Ph.D scholars in the field of network and cloud security. He has published various quality papers in reputed journals like IEEE Transactions, Elsevier, Springer, IET, Taylor & Francis, Wiley, etc. His main thrust research areas are key management in network security and multicasting in computer networks.

Anxi Wang received his B.E. degree in 2016 and is currently working toward the M.E. degree at NUIST. He focuses on information security and incompletely predictable ad hoc networks. His research interests include information security, ad hoc networks and systems, and wireless sensor networks.

Tianqi Zhou received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2016. She is currently a postgraduate with the School of Nanjing University of Information Science and Technology, Nanjing, China. Her research interests include computer and network security, security systems and cryptography.

Affiliations

Dengzhi Liu[1] • **Jian Shen**[1,2,3] • **Pandi Vijayakumar**[4] • **Anxi Wang**[1] • **Tianqi Zhou**[1]

[1] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technologys, Nanjing, China

[2] State Key Laboratory of Cryptology, Beijing, China

[3] Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China

[4] Department of Computer Science & Engineering, University College of Engineering Tindivanam, Tindivanam, India