JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 923-938 (2017)
DOI: 10.6688/JISE.2017.33.4.4

An Enhanced Cloud Data Storage Auditing Protocol Providing Strong Security and Efficiency for Smart City

JIAN SHEN[1,2,3,4], DENGZHI LIU[1,3], QI LIU[1,3], DEBIAO HE[5,*] AND XINGMING SUN[1,3]
[1]*Jiangsu Engineering Center of Network Monitoring*
[2]*Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology*
[3]*School of Computer and Software*
Nanjing University of Information Science and Technology
Nanjing, 210044 P.R. China
[4]*State Key Laboratory of Information Security, Institute of Information Engineering*
Chinese Academy of Sciences
Beijing, 100864 P.R. China
[5]*State Key Laboratory of Software Engineering, School of Computer*
Wuhan University
Hubei, 430072 P.R. China
E-mail: {s_shenjian; liudzdh}@126.com; {qrankl; hedebiao; sunnudt}@163.com

Cloud-assisted applications have been developed in recent years, especially in servicing the smart city. Cloud computing is an emerging technology, which combines many redundant and distributed servers to provide various applications and services. On the one hand, the users can outsource their data to the cloud and enjoy the services that are provided by the cloud anytime and anywhere. On the other hand, outsourcing the data to the cloud will reduce the local storage burden and alleviate the hardware maintenance. However, the cloud server is semi-trusted, which possibly threaten the data's security. In addition, because the data owner will not physically possess the data after outsourcing the data to the cloud, it is required to check the storage correctness and data integrity. Cloud auditing protocols are proposed to solve this problem, but the efficiency and security of previous protocols are not well. In this paper, we proposed an enhanced auditing protocol, which can reduce the verification cost greatly and offer enough security assurance. Besides, our protocol is able to resist the forgery attack in the proof verification. The security and performance analysis shows that the proposed protocol can efficiently support storage verification and privacy preserving.

Keywords: cloud computing, security, cloud auditing protocol, storage verification, privacy preserving

1. INTRODUCTION

With the advancement of Urbanization, more and more applications need novel technologies to enhance their services and also expand its application areas, such as the government services, pollution control and citizens life services in smart city. Cloud computing is an emerging technologies which are based on the Internet. Hence, the cloud-assited application is a potential domain in smart city. Due to the advantages of the cloud computing, such as rapid resource elasticity, on-demand self service, universal access and pay-as-you-go mode [1, 2], cloud computing has aroused broad attention in both industry and academic field in recent years. Cloud computing centralizes the

Received June 29, 2016, revised August 7, 2016, accepted September 8, 2016.
Communicated by Zhe Liu.
* The corresponding author.

distributed and redundant resources. Therefore, the capabilities of its storage and computation can be regarded as infinite. With the limitations of storage capacity and processing capacity of local devices, the users would prefer to outsource data to the cloud. Nowadays, the cloud storage has become the main application of the cloud services [1, 3, 4].

Due to the cloud service provider's semi-trusted feature and its tendency for profit, the consumers are concerned about the security of cloud computing. Cisco and Intel indicate that the security will become an impediment to the development in cloud computing [5]. The security problem is an important factor in limiting the development of cloud computing [6-8]. An investigation of cloud usage from [9] revealed that only about one fifth of those surveyed are willing to select the cloud to store their personal information. No matter in which domain, the security problem is always an important factor in the development of cloud computing. Generally, key agreement protocol and management protocol as well as authentication protocol can solve most of the security problems in security community [10-19]. However, in cloud computing, the security problems are not just the authentication or key management issues. Using the cloud to store the data, the consumers cannot ensure whether their data in the cloud is integrated or not [20]. After outsourcing the data to the cloud and then discarding the local copy of the data, the user will not possess the data physically, which will increase the difficulty of verifying the data integrity. Since the user side's storage capability is limited, retrieving all of the data to local side is impractical. Hence, developing an efficient integrity verification scheme is necessary, so that the whole user's data is not required in verification process.

The main verification schemes on cloud data are divided into two aspects: provable data possession (PDP) [20, 21] and proof of retrievability (POR) [22, 23]. However, the high computing requirements of PDP and POR are unsuited to cloud user whose computing capability is limited. Moreover, in order to ensure the fairness and impartiality of the audit results, it is necessary to extend the traditional cloud data integrity verification scheme to support the third party auditing and public auditing [24-26]. The third party auditor (TPA) is an independent entity, it has expertise and capabilities that the cloud users don't have. TPA can be on behalf of the data owner to verify the correctness and completeness of the data in the cloud without the copy of the data. Moreover, using TPA to audit the data will not bring additional computation and communication burden to the data owner [27]. In recent cloud security researches, there were already many public and third party auditing schemes [28, 39]. These schemes can support the public remote auditing as well as privacy preserving.

In this paper, we proposed an enhanced cloud data storage auditing protocol, which can ensure the data intergrity and storage correctness. Besides, our protocol is more efficient than the related protocols [28, 29, 31] and can resist the forgery attack. The contributions can be summarized as follows:

(1) We design an efficient and secure auditing protocol, which can be used in the public cloud auditing to verify the storage correctness and data integrity.
(2) The proposed protocol can resist the forgery attack. Additionally, our protocol can decrease the computation overhead in the verification phase without influencing the security in the verification phase.
(3) In order to meet the actual requirements, we further extend our protocol to support the batch auditing in the circumstance of multi-owner and multi-cloud.

The rest of the paper is organized as follows. In Section 2, we briefly discuss some related works. Section 3 we review the technique preliminaries and state the definations that used in the protocol. In Section 4, we introduce the proposed protocol in detail, followed by the security analysis and the performance analysis are also described in Sections 5. Finally, we conclude this paper in Section 6.

2. RELATED WORK

With the aid of the third-party auditor (TPA), the public auditing protocol has become a hot spot in cloud computing security. The public auditing protocol in [28, 29, 32, 35] can support public auditing and batch auditing. Wang *et al.* in [30] proposed a flexible distributed storage integrity auditing mechanism. By using the homomorphic token and distributed erasure-coded data, the protocol can locate the data error and identify the misbehaving server. Yang *et al.* in [31] improved the protocol to support the batch auditing in multi-cloud and multi-user environment. A novel public verification mechanism in [33] was proposed, which utilized the multi-signature to support multi-owner data auditability. The auditing protocol in [34] can support the fine-grained data updates and the protocol in [39] can achieve both data integrity and deduplication. In [36] Liu *et al.* utilized a novel enhanced Merkle hash tree (MHT) in the protocol, which can support multiple replicas update. In order to ensure the scalability, Wang *et al.* in [35] proposed a protocol that can support user revocation. Moreover, this protocol can re-sign the data block without retrieving the whole data to the user side by utilizing the proxy re-signatures.

3. PRELIMINARIES AND DEFINATIONS

Before the detailed introduction of the proposed protocol, it is essential to briefly introduce some basic techniques that will be used in this paper. Moreover, the system model and its every entity's introductions are given to construct the cloud environment. Finally, we declare the design goals that the protocol needs to satisfy.

3.1 Preliminaries

Bilinear Maps: We assume $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are the multiplicative groups of prime order p, $\mathcal{G}_1$ and $\mathcal{G}_2$ are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. The bilinear map can be denoted as $\hat{e}$: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. For instance, $x \in \mathbb{G}_1$, $y \in \mathbb{G}_2$ and a, $b \in \mathbb{Z}_p^*$, $\hat{e}(x^a, y^b) = \hat{e}(x, y)^{ab}$. The bilinear property of the bilinear paring can be described that for x, $r \in \mathbb{G}_1$, $y \in \mathbb{G}_2$, $\hat{e}(x \cdot r, y) = \hat{e}(x, y) \cdot \hat{e}(r, y)$. In this map the $\hat{e}$ can be efficiently computed by an algorithm. In addition, it also possesses the property of non-degenerate that can be denoted as $\hat{e}(x, y) \neq 1$.

3.2 System Model

We define a cloud data storage auditing framework to implement the auditing protocol. The framework includes three entities: ***CUs***, ***TPA*** and ***CSP***. ***CUs*** can outsource their own data to the cloud, and of course, they can retrieve the data that they required. ***CUs*** delegate the auditing tasks to the ***TPA***, so they don't have to care about the heavy

and complicated auditing works. The system model of this paper is shown as Fig. 1 and the detailed introduction of every entity can be learned as follows.

CUs is a group of cloud users, which includes various terminal devices, such as personal computer devices, laptops as well as personal wireless electronic devices. The functions of them are various to consumers, but a same feature of them is that compares with the infinite cloud space, the ***CUs'*** storage space is limited. The purpose of accessing to the cloud is not just to expand storage amount. The more important reason is that the cloud can provide many various real-time applications and services.

TPA is an independent third party, which can audit cloud data when needed. Comparing with the ***CUs***, ***TPA*** possesses more capabilities in auditing the data. In other words, ***TPA*** has expertise that ***CUs*** do not, it can check the cloud data integrity on behalf of ***CUs*** without retrieving all of the data to the local side. Moreover, ***TPA***'s evaluation results can help the cloud service provider (***CSP***) to improve its services, which can form a win-win relationship between ***CUs*** and ***CSP***.

CSP manages many storage and application servers. It provides services to the ***CUs*** through the Internet. ***CSP*** cannot be fully trusted, which means that it will try its best to collect the consumers' secret information when it complies with the security protocol. In order to protect the sensitive information of the ***CUs***, the data will be outsourced to the cloud in an encrypted form. Above all, ***CSP*** receives the storage verification challenges from the ***TPA*** in the auditing protocol. Afterwards, it will generate a proof according to the security parameters and then send the correctness proof to the ***TPA***.

3.3 Design Goals

According to the real requirements and the related research works [28, 29, 40], the functions that the auditing protocol needs to accomplish can be summarized as follows:

(1) Public Auditability: The protocol should enable the TPA to verify the correctness and integrity of the data in the cloud without retrieving the whole data to the local side. Moreover, the auditing process should satisfy the requirements that it will not bring additional burden to the user side.

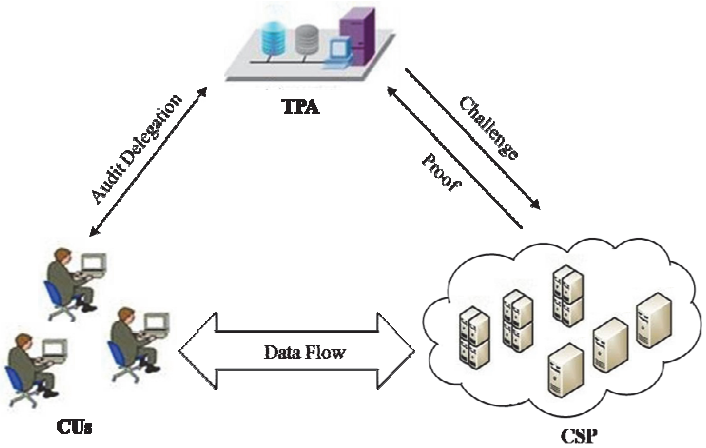

Fig. 1. The system model.

(2) Storage Correctness: In order to avoid the unnecessary security issue that a cheating server tries to pass the audit, the protocol should ensure only the server that stores the data can participant in the auditing process.
(3) Privacy Preserving: In the auditing process, the protocol should prevent the TPA from obtaining the CUs' data content.
(4) Batch Auditing: According to the real-world cloud environment, it is normal that multiple users and multiple servers execute auditing tasks simultaneously. Hence, it is essential to ensure that the auditing protocol should support the batch auditing.

4. THE PROPOSED SCHEME

In this section, the detailed construction of the protocol is introduced. First of all, we state an overview of the protocol that roughly introduces the construction. Secondly, the technology and the data structure for supporting the protocol will be summarized in the construction initialization. Thirdly, the detailed introduction of the protocol is given. Finally, in order to meet the needs of the real-world applications and actual requirements, we extend the protocol to support the batch auditing.

4.1 Overview

The auditing protocol needs to support the privacy preserving, so we utilize the random parameters to blind the sensitive information. Moreover, the TPA cannot obtain the information that contains the user's data. The proposed auditing protocol consists of five algorithms:

(1) *KeyGen* $(\lambda)\rightarrow(pk, sk, pk_t, sk_t)$: This *KeyGen* algorithm is to generate public and secret key pair. Its input is a security parameter λ, which is decided by the requirements of the system security. In order to ensure the security of the data, the output of this algorithm includes two secret-public key pairs. $\{pk, sk\}$ is to increase the security of bilinear pairing and $\{pk_t, sk_t\}$ is to gurantee the secret of the tag.
(2) *TagGen* $(M, sk_t)\rightarrow T$: This *TagGen* algorithm is to generate the file's tag, the input of this algorithm is the corresponding encrypted file set M and the tag secret key sk_t, the output is a tag set T.
(3) *ChallGen* $(Info_M)\rightarrow C$: According to the consumers' verification requirements, this algorithm is to generate a challenge information C, which generally includes file identity, file version and other security parameters.
(4) *ProofGen* $(M, T, C)\rightarrow P(P_T, P_M)$: This algorithm is to respond to the TPA's challenge, taking as input the file block, tag set T and challenge C, it outputs the proof which is consisted of tag proof P_T and file proof P_M.
(5) *VerifyProof* $(C, P, pk, pk_t)\rightarrow 0/1$: Upon receiving the proof from the cloud server, TPA executes *VerifyProof* algorithm to generate the auditing result. According to the calculation TPA will determine whether the storage is correct or not.

Note here that the algorithms *KeyGen* and *TagGen* are executed by the ***CUs***, the data owner decide the security parameters according to their data. The algorithms

ChallGen and *VerifyProof* are executed by the ***TPA***, and the *ProofGen* is executed by the ***CSP***.

4.2 Construction Initialization

In order to support the data storage auditability without the whole data in the cloud, the proposed protocol resorts to the bilinear property of the bilinear pairing, which can satisfy the public auditing feature and the TPA can verify the correctness proof but not decrypt it to obtain the detailed content. We assume that the encrypted file will outsource to the cloud that is divided into n blocks, the structure of file blocks can be shown as Fig. 2. The sub-block $M_i \in \mathbb{Z}_p$, where p is a large prime.

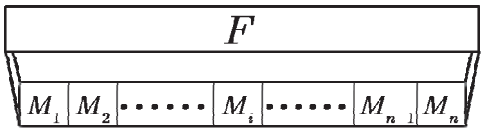

Fig. 2. The file block structure.

4.3 Auditing Protocol

These five algorithms will be introduced in detail in this part. We assume that every step of all the algorithms can correctly generate or calculate the parameters for requirement. The framework of the proposed auditing protocol is shown as Fig. 3.

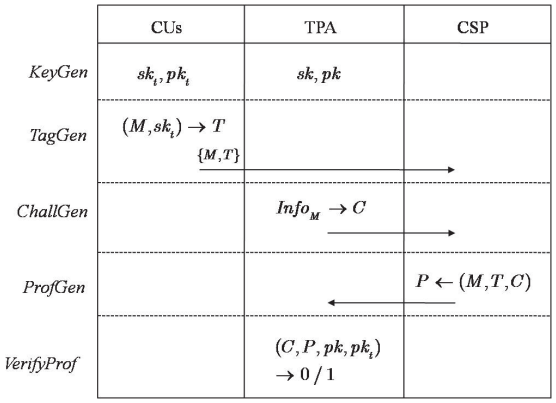

Fig. 3. The framework of the proposed auditing protocol.

(1) $KeyGen(\lambda) \to (pk, sk, pk_t, sk_t)$

The algorithm inputs an implicit security parameter λ and then it randomly chooses α_i, x, r from $\mathbb{Z}_p^*$, p is a large prime order. Then it will compute $sk_t=1/\alpha_i$, $pk_t = \mathcal{G}^{\alpha_i}$, $sk=x^r$, $pk=\mathcal{G}^r$, where pk_t, $pk \in \mathbb{G}$ and $\mathcal{G}$ is a generator of $\mathbb{G}$. Hence, the global parameters include $\{pk_t, pk, \mathcal{G}, p\}$. Here, the sk_t is designed to protect the tag proof and the pk_t is used to verify the tag proof, the detailed descriptions can be found in below.

(2) *TagGen*(M, sk_t)$\rightarrow T$

In order to avoid the leakage of the sensitive information, every auditing file block will be denoted as a tag. The *TagGen* algorithm randomly chooses δ_i and v_i from $\mathbb{Z}_p^*$, we assume $M_i = \delta_i^{vi}$ and then for every file block it computes the tag as Eq. (1).

$$t_i = (\delta_i^{\sum_{i=1}^{n} v_i})^{sk_t} \tag{1}$$

Hence, the tag set T can be denoted as $\{t_i\}_{1\le i\le n}$.

(3) *ChallGen*($Info_M$)$\rightarrow C$

On the one hand, the cloud consumer needs to verify the data correctness and integrity in the cloud. Only if the cloud receives a challenge from the TPA, the cloud will carry out the operations. On the other hand, the cloud cannot generate the correct proof without some parameters from the TPA. Hence, TPA should send a challenge to the cloud. In the purpose of facilitating the verification, we define a challenge, which is consisted of i, v_i, x, pk. That is to say, the challenge can be denoted as C=$\{i, v_i, x, pk\}$.

(4) *ProofGen*(M, T, C)$\rightarrow P(P_T, P_M)$

Upon receiving the challenge C, the cloud will calculate the proof according to the C as well as tag set and data block. Then, this algorithm will output the proof, which consist of tag proof P_T and file proof P_M. The denotation of them can be shown as Eqs. (2) and (3).

$$P_T = (sk \cdot \prod_{i=1}^{n} \delta_i^{v_i})^{sk_t} \tag{2}$$

$$P_M = \prod_{i=1}^{n} \hat{e}(M_i^{1/v_i}, \mathcal{G}^{v_i}) \tag{3}$$

(5) *VerifyProof* (C, P, pk, pk_t)$\rightarrow$0/1

This *VerifyProof* algorithm is executed by the TPA. Through the verification Eq. (4) holds or not, the TPA can determine whether the storage is correct.

$$P_M \cdot \hat{e}(x, pk) \stackrel{?}{=} \hat{e}(P_T, pk_t) \tag{4}$$

We assume the parameters that every entity received have not been modified or lost in above process, we can verify the proof equation by the property of bilinear paring. The left-hand side of Eq. (4) can be elaborated as follows:

$$\begin{aligned} & P_M \cdot \hat{e}(x, pk) \\ &= \prod_{i=1}^{n} e\left(M_i^{1/v_i}, \mathcal{G}^{v_i}\right) \cdot \hat{e}\left(x, \mathcal{G}^r\right) \\ &= \prod_{i=1}^{n} e\left(\delta_i, \mathcal{G}\right)^{v_i} \cdot \hat{e}\left(x, \mathcal{G}\right)^r \end{aligned}$$

$$= \hat{e}(\delta_i, \mathcal{G})^{\sum_{i=1}^{n} v_i} \cdot \hat{e}(x, \mathcal{G})^r$$
$$= \hat{e}(\delta_i^{\sum_{i=1}^{n} v_i}, \mathcal{G}) \cdot \hat{e}(x^r, \mathcal{G})$$
$$= \hat{e}(x^r \cdot \delta_i^{\sum_{i=1}^{n} v_i}, \mathcal{G})$$

The right-hand side of the Eq. (4) can be elaborated as follows:

$$\hat{e}(P_T, pk_t)$$
$$= \hat{e}((x^r \cdot \prod_{i=1}^{n} \delta_i^{v_i})^{1/\alpha_i}, \mathcal{G}^{\alpha_i})$$
$$= \hat{e}(x^r \cdot \delta_i^{\sum_{i=1}^{n} v_i}, \mathcal{G}).$$

Hence, the left-hand side equals the right-hand side, Eq. (4) holds. In other words, the protocol is effective for cloud data integrity verification.

4.4 Extending to Batch Auditing

Aforementioned protocol is a basic single consumer to single server. However, in real applications many users enjoy the cloud services simultaneously. There have been a lot of researches supporting batch auditing [28, 29]. Moreover, the cloud includes many distributed servers, which is inevitable that multi-servers provide services to the consumers. Supporting multi-servers batch auditing already has been existed in some researches [31, 41]. Thus, we extend the protocol to support the batch auditing for multi-users and multi-servers in the following introduction. The framework of the extended batch auditing protocol can be shown as Fig. 4.

(1) $KeyGen(\lambda) \rightarrow (pk_{os}, sk_{os}, pk_{os,t}, sk_{os,t})$

As mentioned in above algorithm, the *KeyGen* algorithm needs to generate some security parameters for the system setup. The input of it is still an implicit security parameter λ, the algorithm chooses $\alpha_{os,i}$, x_{os}, $r_{os} \in \mathbb{Z}_p^*$, where o, s denote the data owner and server respectively. Let $sk_{os,t} = 1/\alpha_{os,i}$, $pk_{os,t} = \mathcal{G}^{\alpha_{os,i}}$, $sk_{os} = x_{os}^{r_{os}}$, $pk_{os} = \mathcal{G}^{r_{os}}$, where $pk_{os,t}$, $pk_{os} \in \mathbb{G}$.

(2) $TagGen(M_{os}, sk_{os,t}) \rightarrow T_o$

This *TagGen* algorithm randomly chooses $\delta_{os,i}$ and $v_{os,i}$ from $\mathbb{Z}_p^*$, and the $M_{os,i} = \delta_{os,i}^{v_{os,i}}$. The tag of the file which is belonged to data owner o and will be outsourced to the server s can be computed as Eq. (5).

$$t_{os,i} = (\delta_{os,i}^{\sum_{o \in O, s \in S} \sum_{i=1}^{n} v_{os,i}})^{sk_{os,t}} \tag{5}$$

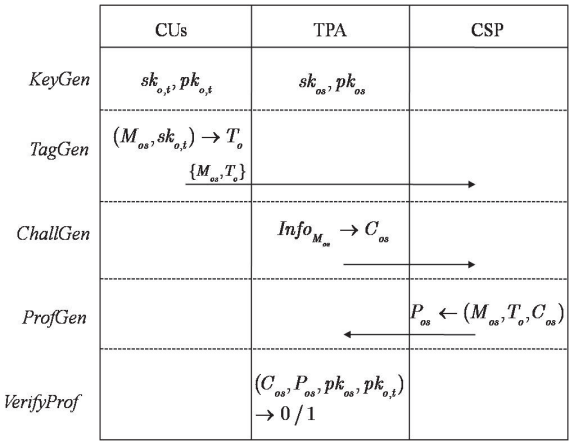

Fig. 4. The framework of the batch auditing protocol.

Similarly, the tag set T can be denoted as $\{t_{os,i}\}_{1\le i\le n, o\in O, s\in S}$, where O, S are the set of the data owners and servers respectively.

(3) *ChallGen* ($Info_{Mos}$)$\rightarrow C_{os}$

In order to adapt the batch auditing, the challenge information should be adjusted accordingly. Hence, we define the batch auditing challenge as $C_{os}=\{i_{os}, v_{os.i}, x_{os}, pk_{os}\}$.

(4) *ProofGen*(M_o, T_o, C_{os})$\rightarrow$*Pos*(P_{Tos}, P_{Mos})

This algorithm is to generate the batch proof, the proof of the data that is belonged to the owner o, the data will be outsourced to the server s. Therefore, the batch proofs of the tag and data can be computed as Eqs. (6) and (7) respectively.

$$P_{Tos} = (sk_{os} \cdot \prod_{o\in O}\prod_{s\in S} \delta_{os,i}^{v_{os,i}})^{sk_{os,t}} \tag{6}$$

$$P_{Mos} = \prod_{s=1}^{S}\prod_{o\in O} \hat{e}(M_{os,i}^{1/v_{os,i}}, \mathcal{G}^{v_{os,i}}) \tag{7}$$

(5) *VerifyProof*(C_{os},P_{os},pk_{os}, $pk_{o,t}$) $\rightarrow$0/1

TPA verifies the Eq. (8), if it holds, the system can confirm that this protocol supports the batch auditing.

$$\prod_{s\in S} P_{Mos} \cdot \prod_{s\in S} \hat{e}(x_{os}, pk_{os}) \stackrel{?}{=} \prod_{s\in S} e(P_{Tos}, pk_{os,t}) \tag{8}$$

We also utilize the property of bilinear paring to verify the batch proof equation, of course, we assume that all parameters have been transmitted and calculated correctly. The left-hand side of Eq. (8) expands as:

$$\prod_{s\in S} P_{Mos} \cdot \prod_{s\in S} \hat{e}(x_{os}, pk_{os})$$
$$= \prod_{s\in S} (\prod_{s=1}^{S} \prod_{o\in O} \hat{e}(M_{os,i}^{1/v_{os,i}}, \mathcal{G}^{v_{os,i}})) \prod_{s\in S} e(x_{os}, \mathcal{G}^{r_{os}})$$
$$= \prod_{s\in S} (\prod_{s=1}^{S} e(\delta_{os,i}, \mathcal{G})^{\sum_{i=1}^{n} v_{os,i}}) \cdot \prod_{s\in S} \hat{e}(x_{os}, \mathcal{G})^{r_{os}}$$
$$= \prod_{s\in S} \hat{e}(\delta_{os,i}, \mathcal{G})^{\sum_{o\in O, s\in S} \sum_{i=1}^{n} v_{os,i}} \cdot \prod_{s\in S} e(x_{os}^{r_{os}}, \mathcal{G})$$
$$= \prod_{s\in S} \hat{e}(x_{os}^{r_{os}} \cdot \delta_{os,i}^{\sum_{o\in O, s\in S} \sum_{i=1}^{n} v_{os,i}}, \mathcal{G})$$

The right-hand side of the Equation:

$$\prod_{s\in S} \hat{e}(P_{T_{os}}, pk_{os,t})$$
$$= \prod_{s\in S} \hat{e}((x_{os}^{r_{os}} \cdot \prod_{o\in O} \prod_{s\in S} \delta_{os,i}^{v_{os,i}}, pk_{os,t})$$
$$= \prod_{s\in S} \hat{e}((x_{os}^{r_{os}} \cdot \delta_{os,i}^{\sum_{o\in O, s\in S} \sum_{i=1}^{n} v_{os,i}})^{1/\alpha_{os,i}}, \mathcal{G}^{\alpha_{os,i}})$$
$$= \prod_{s\in S} \hat{e}((x_{os}^{r_{os}} \cdot \delta_{os,i}^{\sum_{o\in O, s\in S} \sum_{i=1}^{n} v_{os,i}}, \mathcal{G})$$

Hence, the left-hand side equals the right-hand side, Equation holds. We can summarize that this protocol can also support batch auditing.

5. SECURITY AND PERFORMANCE ANALYSIS

In order to evaluate the proposed protocol, we analyze its security from two aspects, which includes storage verification and privacy preserving. Moreover, we give a comparison of the protocol with some related researches in the performance analysis.

5.1 Security Analysis

5.1.1 Storage verification

Verifying the storage integrity and correctness is the primary task of the auditing protocol. From the above verification of the proof, as long as the security parameters have not been modified or lost, the left-hand side of the proof equation will equal the left-hand side. Hence, the integrity of the data in the cloud is verifiable.

The storage correctness means that the user's data was stored in the cloud server. Here, we can define Theorem 1.

Theorem 1: As long as the *TPA* can verify the integrity of the data which the data owner outsourced, the data owner can determine that the user's data has been properly stored.

In the algorithm *ProofGen*, if the cloud server does not possess the user's data, it cannot generate a correct proof according the challenge *C* and tag set *T*. Hence, in the algorithm *VerifyProof*, the verification proof equation cannot be proven.

In [31] the *DP* can be modified by an adversary, so that the adversary can pass the verification, at the same time the TPA doesn't know whether the *DP* is modified or not. As mentioned in [42], the comment paper suggests a solution that adding a signature can solve this problem. However, adding the signature will increase the cost of computation and communication. Our proposed protocol avoids this issue by adjust the encryption and security parameter. The P_T is encrypted by the sk_t and P_M is multiple bilinear paring multiplication that it is not easy to be modified. Even if it is modified, the proof will be influenced that it can't pass the verification algorithm.

5.1.2 Privacy preserving

The user's data is blind to TPA, in other words, the auditor cannot obtain the user's data in the auditing process. We assume that the TPA is corrupt and it tries to acquire the data during auditing. From the above protocol descriptions in section 4, TPA possesses *sk*, *pk*, *C*, *P*, the information about the data is blinded in P_T and P_M. Due to $M_i = \delta_i^{v_i}$ that was defined in algorithm *TagGen*, the TPA tries to decrypt the P_T to get the $\delta_i^{v_i}$. However, the $\delta_i^{v_i}$ was encrypted by the data owner, so the TPA cannot obtain the $\delta_i^{v_i}$ from P_T. In P_M the data block M_i was contained in the bilinear paring operation. To the best of our knowledge, the formula is almost unsolvable. Therefore, obtaining the completed file is very hard. Based on the above analysis, we can determine that the proposed protocol possesses the performance of privacy preserving.

Table 1. Comparison of the computation.

Scheme	*GenProof*	*VerifyProof*
[28]	1*P.*+(2c+1)*Mul.*+(c+1)*Exp.*+1*H.*	2*P.*+(c+1)*Mul.*+(c+3)*Exp.*+(c+1)*H.*
[29]	2c*Mul.*+c*Exp.*+(c+1)*H.*	4*P.*+(c+1)*Mul.*+(c+2)*Exp.*+(2c+2)*H.*
[31]	c*P.*+ 3c*Mul.*+2c*Exp.*	2*P.*+2c*Mul.*+(c+1)*Exp.*+(c+1)*H.*
Our Scheme	c*P.*+(2c+1)*Mul.*+(3c+1)*Exp.*	2*P.*

5.2 Performance Analysis

We simplify the process of verification without affecting the security and privacy preserving. Thus, our protocol is more efficient. Moreover, we further extend this protocol to support the batch auditing, which can audit the storage of multi-cloud and multi-user simultaneously. Here, we define *P.*, *Mul.*, *Exp.* and *H.* as the computation cost of bilinear pairing operation, multiplication, exponentiation and one-way hash function respectively, and c denotes the number of the corresponding operations in an algorithm. The comparisons of the protocol with the related researches can be shown as Table 1. From Table 1 we can determine that our protocol possesses less computation operations. In other words, our protocol is more efficient than the three related protocols in executing the security assurance in the cloud application circumstance.

6. CONCLUSIONS

In this paper, we proposed an enhanced cloud data storage auditing protocol, which can support strong security and efficiency. In order to satisfy the requirements of the real-world applications, we further extend our protocol to support the batch auditing in multi-user and multi-cloud environment. Observed from the security and performance analysis, we can conclude that this protocol can meet the security requirements of the storage verification and privacy preserving. It is worth noting that our protocol can resist the forgery attack. Compared with the related protocols, our protocol can highly reduce the computation cost without influencing the design goals of the auditing protocol. We believe that our protocol can be used in servicing the real-world applications, such as the government confidential services, citizens' sensitive data protection and other important security assurance services in smart city.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation of China under Grant No. 61672295, No. 61300237 and No. U1405254, the State Key Laboratory of Information Security under Grant No. 2017-MS-10, the 2015 Project of six personnel in Jiangsu Province un-der Grant No. R2015L06, the CICAEET fund, and the PAPD fund.

REFERENCES

1. P. Mell and T. Grance, "Draft NIST working definition of cloud computing," http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf, Vol. 53, 2009, p. 50.
2. K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, Vol. 16, 2012, pp. 69-73.
3. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, Vol. 53, 2010, pp. 50-58.
4. J. Shen, D. Liu, C. Lai, Y. Ren, and X. Sun, "A secure identity-based dynamic group data sharing scheme for cloud computing," *Journal of Internet Technology*, Vol. 18, 2016, DOI: 10.6138/JIT.2017.18.4.20160415.
5. J. Bradley, J. Macaulay, A. Noronha, and H. Sethi, "The influence of the cloud to it consumption pattern," http://www.cisco.com/web/about/ac79/docs/re/IT-Consumption_PoV/.
6. J. Shen, D. Liu, J. Wang, J. Shen, Q. Liu, and X. Sun, "A secure cloud-assisted urban data sharing framework for ubiquitous-cities," *Pervasive and Mobile Computing*, 2017, DOI: 10.1016/j.pmcj.2017.03.013.
7. F. Shahzad, "State-of-the-art survey on cloud computing security challenges, approaches and solutions," *Elsevier Procedia Computer Science*, Vol. 37, 2014, pp. 357-362.

8. J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, 2017, DOI: 10.1109/TIFS.2017.2705620.
9. R. Bhadauria, R. Chaki, N. Chaki, and S. Sanyal, "A survey on security issues in cloud computing," *International Journal of Engineering and Technology*, Vol. 5, 2011.
10. Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, and A. Alelaiwi, "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dynamics*, Vol. 83, 2016, pp. 2085-2101.
11. J. Shen, S. Moh, and I. Chung, "Identity-based key agreement protocol employing a symmetric balanced incomplete block design," *Journal of Communications and Networks*, Vol. 14, 2012, pp. 682-691.
12. D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks," *Information Sciences*, Vol. 321, 2015, pp. 263-277.
13. Q. Jiang, J. Ma, X. Lu, and Y. Tian, "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer-to-Peer Networking and Applications*, Vol. 8, 2015, pp. 1070-1081.
14. J. Shen, S. Moh, and I. Chung, "Enhanced secure sensor association and key management in wireless body area networks," *Journal of Communications and Networks*, Vol. 17, 2015, pp. 453-462.
15. D. He, S. Zeadally, N. Kumar, and J. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Systems Journal*, 2016, DOI: 10.1109/JSYST.2016.2544805.
16. Q. Wang, M. He, M. Du, S. S. M. Chow, R. W. F. Lai, and Q. Zou, "Searchable encryption over feature-rich data," *IEEE Transactions on Dependable and Secure Computing*, Vol. PP, 2016, DOI: 10.1109/TDSC.2016.2593444.
17. J. Shen, H. Tan, S. Moh, I. Chung, and J. Wang, "An efficient rfid authentication protocol providing strong privacy and security," *Journal of Internet Technology*, Vol. 17, 2016, pp. 443-455.
18. Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *International Journal of Communication Systems*, Vol. 28, 2015, pp. 383-393.
19. S. Hu, Q. Wang, J. Wang, Z. Qin, and K. Ren, "Securing SIFT: privacy-preserving outsourcing computation of feature extractions over encrypted image data," *IEEE Transactions on Image Processing*, Vol. 25, 2016, pp. 3411-3425.
20. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proceedings of ACM Conference on Computer and Communications Security*, 2007, pp. 598-609.
21. G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," *Advances in Cryptology-ASIACRYPT*, 2009, pp. 319-333.
22. A. Juels and B. S. Kaliski, "PORs: proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, 2007, pp. 584-597.
23. H. Shacham and B. Waters, "Compact proofs of retrievability," *Journal of Cryptology*, Vol. 26, 2013, pp. 442-483.

24. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of IEEE INFOCOM*, 2010, pp. 1-9.
25. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Proceedings of the 14th European Symposium on Research in Computer Security*, pp. 355-370.
26. Y. Zhu, H. Wang, Z. Hu, G. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in *Proceedings of ACM Symposium on Applied Computing*, 2011, pp. 1550-1557.
27. C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," *IEEE Network*, Vol. 24, 2010, pp. 19-24.
28. C. Wang, S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, Vol. 62, 2013, pp. 362-375.
29. Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, 2011, pp. 847-859.
30. C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," *IEEE Transactions on Services Computing*, Vol. 5, 2012, pp. 220-232.
31. K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, 2013, pp. 1717-1726.
32. B. Wang, B. Li, and H. Li, "Oruta: privacy-preserving public auditing for shared data in the cloud," *IEEE Transactions on Cloud Computing*, Vol. 2, 2014, pp. 43-56.
33. B. Wang, H. Li, X. Liu, F. Li, and X. Li, "Efficient public verification on the integrity of multi-owner data in the cloud," *Journal of Communications and Networks*, Vol. 16, 2014, pp. 592-599.
34. C. Liu, J. Chen, L. T. Yang, X. Zhang, C. Yang, R. Ranjan, and R. Kotagiri, "Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, 2014, pp. 2234-2244.
35. B. Wang, B. Li, and H. Li, "Panda: public auditing for shared data with efficient user revocation in the cloud," *IEEE Transactions on Services Computing*, Vol. 8, 2015, pp. 92-106.
36. C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "MuR-DPA: top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud," *IEEE Transactions on Computers*, Vol. 64, 2015, pp. 2609-2622.
    J. Liu, K. Huang, H. Rong, H. Wang, and M. Xian, "Privacy-preserving public auditing for regenerating-code-based cloud storage," *IEEE Transactions on Information Forensics and Security*, Vol. 10, 2015, pp. 1513-1528.
37. J. Li, X. Tan, X. Chen, D. S. Wong, and F. Xhafa, "OPoR: enabling proof of retrievability in cloud computing with resource-constrained devices," *IEEE Transactions on Cloud Computing*, Vol. 3, 2015, pp. 195-205.

38. J. Li, X. Tan, X. Chen, D. S. Wong, and F. Xhafa, "Secure auditing and deduplicating data in cloud," *IEEE Transactions on Computers*, Vol. 3, 2015, pp. 195-205.
39. D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, 2015, pp. 1-10.
40. Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multi-cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23, 2012, pp. 2231-2244.
41. J. Ni, Y. Yu, Y. Mu, and Q. Xia, "On the security of an efficient dynamic auditing protocol in cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, 2014, pp. 2760-2761.

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 Nanjing University of Information Science and Technology, Nanjing, China. His research interests include public key cryptography, secure data sharing and data auditing in cloud.

Dengzhi Liu received the B.S. degree and the M.E. degree from Nanjing University of Information Science and Technology in 2014 and 2017, respectively. He is currently working toward the Ph.D. degree in the School of Computer and Software, Nanjing University of Information Science and Technology. He focuses on the security and privacy issues in cloud Computing. His current research interests include applied cryptography, network and data security, and cloud computing security.

Qi Liu received his BSc degree in Computer Science and Technology from Zhuzhou Institute of Technology, China in 2003, and his MSc and Ph.D. in Data Telecommunications and Networks from the University of Salford, UK in 2006 and 2010. His research interests include context awareness, data communication in MANET and WSN, and smart grid. His recent research work focuses on intelligent agriculture and meteorological observation systems based on WSN.

Debiao He received the Ph.D. degree in Applied Mathematics from Wuhan University, Wuhan, China, in 2009. He is currently an Associate Professor with the State Key Laboratory of Software Engineering, School of Computer, Wuhan University. His main research interests include cryptography and information security, particularly cryptographic protocols.

Xingming Sun received his BS in Mathematics from Hunan Normal University, China, in 1984, MS in Computing Science from Dalian University of Science and Technology, China, in 1988, and Ph.D. in Computing Science from Fudan University, China, in 2001. He is currently a Professor in School of Computer and Software, Nanjing University of Information Science and Technology, China. His research interests include network and information security, digital watermarking, and data security in cloud.