New Data Storage Auditing Protocols

Xinpeng Zhang[1,2], Chunxiang Xu[1]
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China
[2]Logistic Information Center, Joint Logistics Department, Chengdu Military Region, Chengdu 610015,China

Wei Sai, Ying Li, Tao Zhou
Logistic Information Center, Joint Logistics Department, Chengdu Military Region, Chengdu 610015, China

Abstract— **Cloud storage is a kind of cloud computing services that allows users to store their data in a remote cloud. Because of the loss of data control, data owners will concern that their data will be misused or unauthorized access by other users, in addition, they also worry their data may be lost in the clouds. Therefore, verify the authenticity of the data has become a key issue of data stored on the untrusted server.**

Shacham and Waters [17] give two Data storage audit protocols with full security proofs against arbitrary adversaries in the strongest secure model, but the server needs to give back a linear combination of the blocks that will leak audit data to the auditor. In order to improve the agreement of Shaham and waters, we use a hash function and blind technique to construct a public's privacy audit protocol.

Keywords- data storage auditing; cloud computing; blind technique, hash function

I. INTRODUCTION

Cloud computing [1], as a popular, emerging computing model currently that allows users to store their data in a remote cloud, so as to enjoy the on-demand services. The cloud is very convenient for users who want to access to data stored in the cloud at anytime, anywhere or using any device, it can be quickly deployed at very high efficiency and minimal administrative overhead.

However, the data stored in the cloud will bring some security risks. For example, data owners worry that their data will be accessed by unauthorized users or their data will be lost by misuse or access in the cloud. Verify the authenticity of data has become a key issue of data stored on the untrusted server. It appears in the peer-to-peer storage systems [2,3], Network File System [4,5], Long-term archiving [6], web-service object store [7], and database systems [8]. This system prevents distortion or modify data storage server checks the authenticity of the access data.

Note that it is not a useful method that just simply download the data as integrity verification because of the costs of input / output and network file transfer file in the network. In addition, it is often not enough to detect data corruption when the verifier access the data, because it may be too late for recover the lost or corrupted data. Taking into account the large size of outsourced data and limited resources of the user's ability, it is difficult and expensive for cloud computing users to audit the accuracy of the data [9].

There is a common method to check the integrity of the data with checking on retrieval, which means after access their data, data owners will check the integrity of data. [10] However, this will lead to heavy input and output overhead and high communication costs from cloud server due to the data retrieval operation. So it is desirable to own a storage audit services to ensure data owners' data is correctly stored in the cloud. However, data owners are reluctant because of the heavy overhead and cost of such audit services. In fact, this is not fair for either side of the cloud service provider or owner of the auditing data, neither of them can ensure fair and honest audit results [11]. Third-party audit is an inevitable choice for the storage of the audit. Third-party auditors who have the expertise and ability to do a more efficient job, can convince both sides of cloud service providers and data owners.

Therefore, to fully guarantee the security of data in the cloud and save the computing resources of users, it is critical to make cloud data storage capacity of public audit, so that users can turn to third-party auditor (TPA). According to the results of the auditor, the auditing report can be published by TPA, it will not only help users to assess the risks of cloud data they subscribe from the service, but also conducive to cloud service providers to improve their cloud-based service platform [12]. Overall, the public audit risk from this new agreement play an important role in the cloud storage auditing protocols; where the user needs to assess the risks and gains the trust in the Cloud .

Up to now, in order to ensure the integrity of the data stored remotely ,there have been proposed several protocols of public auditing capacity in different systems and security model [13] [14], [15]. Public auditing allows an external party, in addition to the user to verify the accuracy of stored data remotely. However, most of these programs [13], [14], [15] does not support the user's external auditors, namely, data privacy protection, they could potentially reveal the auditor about the users' data information. This serious drawback has greatly affected the security of these protocols in the cloud. From the perspective of the protection of data privacy, users, who owns the data, rely on their data storage security TPA only, and do not want the information from unauthorized

disclosure of this audit process to introduce new security vulnerabilities in their data [16].

Shacham and Waters [17] give two Data storage audit protocols with full security proofs against arbitrary adversaries in the strongest secure model , but the server needs to give back a linear combination of the blocks that will leak audit data to the auditor. In order to improve the agreement of Shaham and waters, we use a hash function and blind technique to construct a public's privacy audit protocol.

Organization of the paper: In section 2, we give a brief introduction of the concepts of preliminaries and introduce two system models. We introduce Shacham and Waters's protocols in section 3. Then we introduce our improvement in section 4, then give its simple security analysis in section 5. Finally, we conclude our work.

II. PRELIMINARIES

We give a brief introduction of the bilinear pairings and system model that will be used later.

A. *Bilinear pairings[18]*

Let G_1 and G_2 be cyclic groups of prime order p with the multiplicative group action and g is a generator of G_1, and $e: G_1 \times G_1 \to G_2$ is a map ,which has the following three properties:

1.Bilinearity: for any all $a,b \in Z_p^*$, $e(g^a, g^b) = e(g,g)^{ab}$

2. Non-degeneracy: for any g, $e(g, g) \neq 1$.

3.Computability: For all $a, b \in G_1$,it exist an efficient algorithm to compute $e(a, b)$.

B. *System model*

If necessary, refer to [19] for more details about the system model and the threat model.

Data Owner Auditing: Check the integrity of their data stored remotely by the data owners . We call this type of auditing protocol as the data owner auditing (shown in Figure 1). Data owners auditing system model contains only a remote cloud server and data owners.

Figure 1: Data Owner Auditing

Third Party Auditing: However, preferably by a third-party auditing services in the cloud computing data storage, rather than the data owners, the system model includes three types of entities: data owner, cloud servers and third-party auditors, shown in Figure 2 and Figure 3.

Figure 2: Third Party Auditing: Initializtion

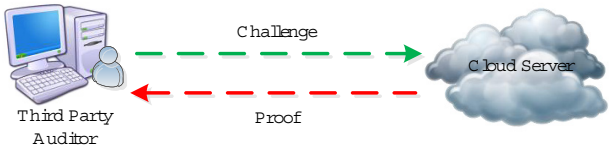

Figure 3: Third Party Auditing: Challenge and Proof

C. *Risk Model*

Smetimes, the cloud server is dishonest, the auditor can face replace attacks, replay attacks, and forgery attacks.

III. SHACHAM AND WATERS'S SCHEME

A. ***MAC-based batch verification***

Before the initialization of system the data owner first divides the encrypted data into n parts ,where p is a large prime , $m_1, m_2, ..., m_n \in Z_p$. Second, the data owner chooses randomly α , s from Z_p^* as secret numbers.

We define $s_i = G(s,i)$,where i is from 1 to m, and G is Pseudo-random number generator. The data owner computes $\sigma_i = \alpha m_i + s_i \bmod p$ for each part data i ,then gives both m_i (data parts) and s_i (corresponding MACs) to the server. The follows is security proof of their protocol.

The Auditor first chooses a challenge set Q randomly and coefficients $v_i (i \in Q)$ from Z_p randomly ,then transmits $\{(i, v_i)\}(i \in Q)$ to the cloud server as a challenge. Secondly, the cloud server computes and sends back the proof (σ, μ) to the Auditor, where $\sigma = \sum_{i \in Q} v_i \sigma_i$ and $\mu = \sum_{i \in Q} v_i m_i$. Unless $\sigma = \alpha\mu + \sum_{i \in Q} v_i s_i$,the Auditor will accepts the proof of protocol. In their protocol, the cloud server is asked to send $\mu = \sum_{i \in Q} v_i m_i$ to the auditor, which may include all linear combinations of blocks to be challenged. Since the auditor has full understanding of all coefficients of $\{(i, v_i)\}$, after several choices , the auditor is possible

to recover the data parts after receiving enough linear combinations of data parts.

B. *RSA method*

In Shacham et al's protocols [17], after giving the data message M, data owner uses the erasure code to get M' at first, and then divides M' into *n* parts and *s* blocks $(m_{ij})(1 \le i \le n, 1 \le i \le s)$ for each part. Define $H: \{0,1\}^* \to Z_N^*$ as a full domain hash function. Define *e* as the public key and *d* as the private key. For each data part m_i , the data owner calculates the tags $\sigma_i = (H(name \| i) \cdot \prod_{j=1}^{s} u_j^{m_{ij}})^d \bmod N$,where u_j are random chosen elements from Z_N^*. Both the data $\{m_{ij}\}$ and the tags $\{\sigma_i\}$ are given to the cloud server. The date owner gives to the auditor the data message and the public key *e*.

In order to the auditing of data storage, the auditor selects a subset of data parts *Q* randomly and generates coefficients v_i for each chosen data part m_i . After that the auditor give $\{(i, v_i)\}(i \in Q)$ to the cloud server as a challenge. After receiving the challenge, the cloud server calculates and gives back the tag proof $\sigma = \prod_{i \in Q} \sigma_i^{v_i} \bmod N$ together along with the data proof $u_j = \sum_{i \in Q} v_i m_{ij}$, where the sum is calculated without the modular reduction in Z_N^*.

The auditor will accepts the proof of protocol by the equation $\sigma^e = \prod_{i \in Q} H(name \| i)^{v_i} \cdot \prod_{j=1}^{s} u_j^{\mu_j} \bmod N$.

IV. OUR IMPROVEMENT

In Shacham et al's protocols, the cloud server needs to send back to the auditor about all the linear combinations of data parts. According to our analysis , the Auditor is possible to recover the data . To improve their protocols, we use two ways to achieve privacy-preserving public auditing protocols.

We choose MAC-based batch verification as study, the similar analysis to the RSA method.

A. *Hash function*

Before the initialization of system the data owner first divides the encrypted data into *n* parts ,where *p* is a large prime , $m_1, m_2, ..., m_n \in Z_p$. Second, the data owner chooses randomly α and *s* from Z_p^* as secret numbers.

We define $s_i = G(s, i)$,where *i* is from 1 to *m*, and *G* is Pseudo-random number generator. The data owner also let a hash function: $H: \{0,1\}^* \to Z_p$. The data owner computes $\sigma_i = \alpha H(m_i) + s_i \bmod p$ for each part data *i* ,then gives both m_i (data parts) and s_i (corresponding MACs) to the server. The follows is security proof of their protocol. The Auditor first chooses a challenge set *Q* randomly and coefficients $v_i (i \in Q)$ from Z_p randomly ,then transmits $\{(i, v_i)\}(i \in Q)$ to the cloud server as a challenge.

Secondly, the cloud server computes and sends back the proof (σ, μ) to the Auditor, where $\sigma = \sum_{i \in Q} v_i \sigma_i$ and $\mu = \sum_{i \in Q} v_i H(m_i)$.

Unless $\sigma = \alpha\mu + \sum_{i \in Q} v_i s_i$,the auditor will accepts the proof of protocol. In our protocol, m_i is hidden by a hash function.

B. *Blind technique*

Because $\mu = \sum_{i \in Q} v_i m_i$ is the reason to leak message to auditor. So we make a simple modify. The cloud server computes and sends back the proof (σ, μ) to the Auditor, where $\sigma = \sum_{i \in Q} v_i \sigma_i$ and $\mu = g^{\sum_{i \in Q} v_i m_i}$, where *g* is a public parameter. The auditor will accept the proof unless $g^{\sigma} = \mu^{\alpha} g^{\sum_{i \in Q} v_i s_i}$. We can see m_i is hidden by an exponential function.

V. SECURITY ANALYSIS

A. *Theorem: The auditor cannot get any information from both using hash function and blinding technique protocols.*

Proof: First, we consider the protocol with hash function. Since the auditor has full understanding of all coefficients of $\{(i, v_i)\}$, after several choices, the auditor can get some linear equations related $H(m_i)$, and then get $H(m_i)$ after solving the linear equations. However, due to the irreversible nature of the hash function, the auditor cannot obtain m_i , otherwise, the auditor find a collision in the hash function.

Second, we consider the protocol with blind technique. The protocol refers to exponential function; discrete logarithm problem is intractable problems. m_i is on the part of exponent, so the auditor cannot get any information about m_i , otherwise, it could be used to solve the discrete logarithm problem.

VI. CONCLUSION

In this paper, the auditing problems for data storage in cloud computing is studied, and we use hash function and blind technique to construct privacy-preserving public auditing protocols to enhance Shacham and Waters's protocols, which reduce the risk of leaking the data to the Auditor.

ACKNOWLEDGMENTS

This work is supported by The Weaponry Equipment Pre-Research Foundation, the PLA General Armament Department (NO.9140A04020311DZ02)and Science and Technology on Communication Security Laboratory Foundation (NO.9140C110301110C1103).

REFERENCES

[1] Mell P, Grance T. The NIST definition of cloud computing[J]. National Institute of Standards and Technology, 2009, 53(6): 50.

[2] Kubiatowicz J, Bindel D, Chen Y, et al. Oceanstore: An architecture for global-scale persistent storage[J]. ACM Sigplan Notices, 2000, 35(11): 190-201.

[3] Muthitacharoen A, Morris R, Gil T M, et al. Ivy: A read/write peer-to-peer file system[J]. ACM SIGOPS Operating Systems Review, 2002, 36(SI): 31-44.

[4] Li J, Krohn M N, Mazieres D, et al. Secure untrusted data repository (SUNDR)[C]//OSDI. 2004, 4: 9-9.

[5] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: Scalable secure file sharing on untrusted storage. In Proc. of FAST, 2003.

[6] Maniatis P, Roussopoulos M, Giuli T J, et al. The LOCKSS peer-to-peer digital preservation system[J]. ACM Transactions on Computer Systems (TOCS), 2005, 23(1): 2-50.

[7] Yumerefendi A R, Chase J S. Strong accountability for network storage[J]. ACM Transactions on Storage (TOS), 2007, 3(3): 11.

[8] Maheshwari U, Vingralek R, Shapiro W. How to build a trusted database system on untrusted storage[C]//Proceedings of the 4th conference on Symposium on Operating System Design & Implementation-Volume 4. USENIX Association, 2000: 10-10.

[9] Brunette G, Mogull R. Security guidance for critical areas of focus in cloud computing v2. 1[J]. Cloud Security Alliance, 2009: 1-76.

[10] Kallahalla M, Riedel E, Swaminathan R, et al. Plutus: Scalable Secure File Sharing on Untrusted Storage[C]//Fast. 2003, 3: 29-42.

[11] Wang C, Ren K, Lou W, et al. Toward publicly auditable secure cloud data storage services[J]. Network, IEEE, 2010, 24(4): 19-24.

[12] Shah M A, Swaminathan R, Baker M. Privacy-Preserving Audit and Extraction of Digital Contents[J]. IACR Cryptology ePrint Archive, 2008, 2008: 186.

[13] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, nd D. Song, "Provable data possession at untrusted stores," Cryptology Print Archive, Report 2007/202, 2007, http://eprint.iacr.org/.

[14] Wang Q, Wang C, Li J, et al. Enabling public verifiability and data dynamics for storage security in cloud computing[M]//Computer Security–ESORICS 2009. Springer Berlin Heidelberg, 2009: 355-370.

[15] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc.of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107.

[16] Shah M A, Baker M, Mogul J C, et al. Auditing to Keep Online Storage Services Honest[C]//HotOS. 2007.

[17] Shacham H, Waters B. Compact proofs of retrievability[M] // Advances in Cryptology ASIACRYPT 2008. Springer Berlin Heidelberg, 2008: 90-107.

[18] D.Boneh, M.Franklin, Identity Based Encryption from the Weil Pairing[J]. Journal of Computing, 2001, 32(3): 586-615.

[19] Yang K, Jia X. Data storage auditing service in cloud computing: challenges, methods and opportunities[J]. World Wide Web, 2012, 15(4):409-428.