International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

# Privacy Preserving Auditing Protocol Using Cryptography for Cloud Storage Systems

**Anuradha Appasaheb Jagadale[1], Shilpa Gite[2]**

[1]S.I.T. , Symbiosis International University, Pune, Maharashtra, India

[2] Professor, S.I.T. , Symbiosis International University, Pune, Maharashtra, India

**Abstract:** ***Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. Now days lots of data owners outsourcing their data on cloud servers. Due to the data outsourcing, this new paradigm of data hosting service also introduces new security challenges, which requires an auditing service to check the data integrity in the cloud. This work studies the problem of ensuring the integrity of cloud data storage. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the data owner, to verify the integrity of the data stored in the cloud. Some existing integrity checking methods can only be useful for static data and, thus, cannot be applied to the auditing service since the data in the cloud can be dynamically updated. Thus, secure dynamic auditing protocol is desired to ensure data owners that their data is correctly stored in the cloud storage systems. In this paper, we design an auditing framework for cloud data storage systems and propose privacy-preserving and dynamic auditing protocol. Also, proposed auditing protocol doesn't require any trusted organizer to support batch auditing for multiple clouds. The analysis and results show that our proposed auditing protocols are secure and efficient, especially it reduce the computation overhead of the auditor.***

**Keywords:** Storage auditing, dynamic auditing, privacy-preserving auditing, batch auditing, cloud computing.

## 1. Introduction

Cloud Service Provider (CSP) provides an important service of cloud computing [1], which allows data owners (consumers) to move their data from local computing systems to the cloud storage. More and more owners start to store the data in the cloud [2]. This new data storage paradigm in "Cloud" brings about many challenging design issues which have profound influence on the security and performance of the overall system. Owners would worry that the data could be lost in the cloud storage; this is because data loss may happen in any storage system, no matter what high degree of security measures cloud service providers would take [3]-[5]. Sometimes, cloud service providers might be dishonest. They could delete the data which has not been accessed or rarely accessed to save the storage space and claim that the data are still correctly stored in the cloud. Therefore, data owners need to be convinced that the data are correctly stored in the cloud storage. Owners can check the data integrity based on two-party storage auditing protocols [6]-[9]. In cloud storage system, it may be inappropriate to let any side of cloud service providers or data owners conduct such auditing, because none of them guaranteed to provide unbiased auditing result. In this scenario, *third party auditing* is the choice for auditing stored data in cloud computing. A third party auditor (auditor) that has expertise and capabilities can do a more efficient work and convince both cloud service providers and owners.

For the third party auditing protocol in cloud storage systems the auditing protocol should have the following properties:

1) *Confidentiality* The auditing system should keep owner's data confidential against the auditor.
2) *Dynamic Auditing* The auditing protocol should support the dynamic updates of the data in the cloud storage.
3) *Batch Auditing* The auditing protocol should also be able to support the batch auditing for multiple owners and multiple clouds.

Recently, several remote integrity checking protocols were proposed to allow the auditor to check the data integrity on the remote server [10]-[18]. We can find that many of them cannot support the data dynamic operations or they are not privacy preserving, so that they cannot be applied to cloud storage systems.

In [13], the authors proposed a dynamic auditing protocol that can support the dynamic operations on the data which is stored on the cloud, but this method may leak the data content to the auditor. In [14], the authors extended their dynamic auditing system to be privacy preserving and support the batch auditing for multiple owners. But, due to the large number of data tags, this auditing technique may incur a heavy storage overhead on the server. In [15], Zhu et al. proposed a cooperative provable data possession scheme that can support the batch auditing for multiple clouds and extended it to support the dynamic auditing in [16]. However, their proposed scheme cannot support the batch auditing for multiple owners. Because the parameters for generating the data tags used by each owner are different, and that's why, they cannot combine the data tags from multiple owners to conduct the batch auditing. Also their scheme applies the mask technique to ensure the data privacy which requires an additional trusted organizer to send a commitment to the auditor during the multicloud batch auditing. However, such additional organizer is not practical in cloud storage systems.

In this paper, we propose secure dynamic auditing protocol, which can meet the above listed properties. To solve the data privacy problem, proposed method is to save an encrypted data in cloud storage by using the encryption techniques,

such that the auditor cannot decrypt it but can verify the correctness of the data. Without using the mask technique and this method does not require any trusted organizer during the batch auditing for multiple clouds.

## 2. Existing System

In this scheme [19], author proposed an efficient and secure dynamic auditing protocol. For solving the data privacy problem, method used is to generate an encrypted proof with the challenge stamp by using the Bilinearity property of the bilinear pairing. Because of that the auditor cannot decrypt it but can verify the correctness of the proof. This scheme doesn't use the mask technique, so it does not require any trusted organizer during the batch auditing for multiple clouds. Also, in this method, the server computes the proof as an intermediate value of the verification, so that the auditor can directly use this intermediate value to verify the correctness of the data.

The author's contribution can be summarized as follows:

- Designing a framework for cloud storage systems and propose a privacy-preserving and efficient storage auditing protocol. This protocol ensures the data privacy by using cryptography method. This auditing protocol incurs less communication cost between the auditor and the server.
- Extending this storage auditing protocol to support the data dynamic operations, this is efficient and secure in the random oracle model.
- Further extending this storage auditing protocol to support batch auditing for not only for multiple owners of data but also for multiple clouds. Here multicloud batch auditing does not require any additional trusted organizer.

To improve the performance of an auditing system, author applied the data fragment technique and homomorphic verifiable tags. The data fragment technique can reduce number of data tags, such that it can reduce the storage overhead and improve the system performance.

This auditing protocol consists of the following algorithms: KeyGen $(\lambda) \rightarrow (skh, skt, pkt)$, TagGen $(M, skt, skh) \rightarrow T$, Chall $(Minfo) \rightarrow$ C, Prove $(M, T, C) \rightarrow$ P, Verify (C, P, *skh, pkt, Minfo*) $\rightarrow$ 0/1.

This system works as the owner generates the keys and the tags for the data. After storing the data on the server, the owner asks the auditor to conduct the auditing to check the correctness of data. For securing the dynamic auditing, author introduced an index table (ITable) which records the information of the data. The ITable consists of four different components: Index, Bi, Vi, and Ti. The Index denotes the current block number of data block in the data component. Bi denotes the original block number of data block, and Vi denotes the current version number of data block. Ti is the time stamp used for generating the data tag. This ITable is managed by the auditor and created by the owner. Once the owner done with the data dynamic operations, it sends an update message to the auditor for updating the ITable that is stored on the auditor. Proposed system by author is efficient and secure. It protects the data privacy against the auditor by combining the cryptography method with the bilinearity property, rather than using the mask technique. It also supports the multi owner and multi cloud auditing.

## 3. Proposed System

### 3.1 Definition of a System Model

In this section, we describe the system model and give the definition of privacy preserving auditing protocol.

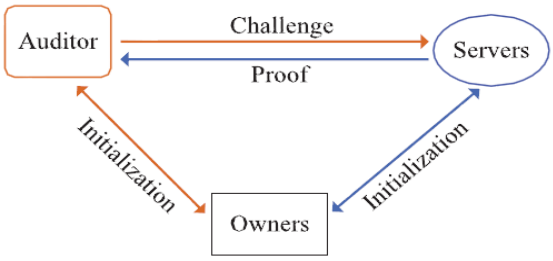


**Figure 1**: System model of data storage auditing

We consider an auditing system for cloud storage as shown in Fig. 1, which involves data owners (owner), the cloud service provider (CSP), and the third-party auditor (auditor). The owner, it can be individual owner or any organization that creates the data and store their data in the cloud. The cloud service provider, it has significant storage space to store the owners' data and provides the data access to them. The auditor, it is a trusted third-party that has expertise and capabilities that cloud users do not have and it provides the data storage auditing service for both the owners and servers. The auditor can be a trusted system, which can provide unbiased auditing result for both data owners and cloud servers.

In the figure above we prepared a model in which User i.e. owner, CSP and TPA are shown. Registered users can get logged in to the system and can store data on cloud. The user asks the CSP to provide service where CSP authenticate the client and provide a storage service. In this scheme, AES algorithm is used where client encrypt and decrypt the file. And the auditor performs the dynamic auditing or on demand auditing on the user request.

### 3.2 Techniques for Auditing Protocol

In this section, we present some techniques applied in the design of our privacy-preserving auditing protocol. Then, we describe the detailed construction of our auditing protocol for cloud storage systems.

#### 3.2.1 Overview of Our Solution

Suppose there is a file F has. Each data has its physical meanings and can be updated dynamically by the data owners. For public data, the data owner does not need to encrypt it, but for private data, the data owner needs to encrypt it with its corresponding key. The main challenge in the design of auditing protocol is the data privacy problem. This is because for encrypted data, the auditor may obtain keys through any special channels and may be able to decrypt

the data. To solve the data privacy problem, our method is to generate an encrypted data and then store it in the cloud storage, such that the auditor cannot decrypt it, but the auditor can verify the correctness of the data without decrypting it.

### 3.2.2 Framework for Our Privacy-Preserving Auditing Protocol

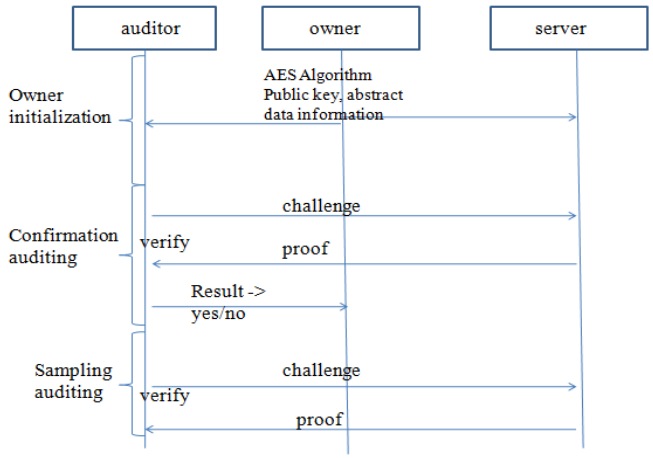


**Figure 2**: Framework for privacy preserving auditing protocol

As illustrated in Fig. 2, our storage auditing protocol consists of three phases: owner initialization, confirmation auditing, and sampling auditing. During the system initialization, the owner generates the keys for the data.
After storing the data on the server, the owner asks the auditor to conduct the confirmation auditing to make sure that their data is correctly stored on the server. Once confirmed, the owner can choose to delete the local copy of the data. Then, the auditor conducts the sampling auditing periodically to check the data integrity.

Phase 1: Owner initialization. The owner runs the AES algorithm to generate the pair of private-public tag key. Then, the owner sends each data component and its corresponding public key to the server together with the set of parameters. The owner then sends the public tag key and the abstract information of the data which includes the username, file name and file description to the auditor.

Phase 2: Confirmation auditing. In our auditing construction, the auditing protocol only involves two-way communication: Challenge and Proof. During the confirmation auditing phase, the owner requires the auditor to check whether the owner's data are correctly stored on the server. The auditor conducts the confirmation auditing phase as

1) The auditor runs the challenge algorithm to perform auditing for all the data records for particular user and sends to the server.
2) Upon receiving the challenge from the auditor, the server runs the prove algorithm Prove to generate the result and sends it back to the auditor.
3) When the auditor receives the proof from the server, it runs the verification algorithm Verify to check the correctness of data and extract the auditing result.

The auditor then sends the auditing result to the owner. If the result is true, the owner is convinced that its data are correctly stored on the server, and it may choose to delete the local version of the data.

Phase 3: Sampling auditing. The auditor will carry out the sampling auditing periodically by challenging the data records of user. The frequency of taking auditing operation depends on the service agreement between the data owner and the auditor. Similar to the confirmation auditing in Phase 2, the sampling auditing procedure also contains two-way communication as illustrated in Fig. 2.

### 3.2.3 Secure Dynamic Auditing

In cloud storage systems, the data owners will dynamically update their data. As an auditing service, the auditing protocol should be designed to support the static data as well as dynamic data. There are three types of data update operations that can be used by the owner: modification, insertion, and deletion. However, the dynamic operations may make the auditing protocols insecure. To prevent this, we introduce the table to record the abstract information of the data. This table consists of four components: username, file name, file description and public key for corresponding file. This table is managed by the auditor. When the owner completes the data dynamic operations, it sends an update message to the auditor for updating the table that is stored on the auditor. After the confirmation auditing, the auditor sends the result to the owner for the confirmation that the owner's data on the server and the abstraction information on the auditor are both up-to-date. This completes the data dynamic operation.

### 3.2.4 Batch Auditing for Multicloud

Some of the data owners may store their data on more than one cloud servers. To ensure integrity of data in all the clouds, the auditor will send the auditing challenges to every cloud server that hosts the owner's data and verify all the records from them. To reduce the communication cost of the auditor, it is desirable to combine all these requests together and do the batch auditing.

### 3.3 Cryptography at user level

The user cannot trust on TPA so the cryptography is required at user level. Before storing the data into the cloud storage the user generates two large random numbers as a key i.e. public key and private key. After generation of keys by data owner he will encrypt the file F to F' by using the private key and one copy of public key is given to the auditor. While storing the data to the cloud storage it is stored by using username, and file name. Auditor maintains the table containing abstract information of data. After storing data to the cloud storage, auditor performs his task. Auditor will perform the dynamic auditing or on demand auditing. Auditor asks the CSP for challenge and CSP replies with set of public keys, then auditor matches the set of public keys

given by CSP and its own copy stored in table. If the username, file name and public keys are matched the data is not tampered in cloud storage. If the data is tampered corresponding public keys don't get matched. At the end auditor will generate the auditing report containing number of records audited for particular user, auditing time and server computing time. Auditor generates the mail containing all these details and sends it to the owner of data, also the status indicating that data is tampered or not.

## 4. Result Analysis

We implemented AES-based instantiations in Windows7. Our experiment is conducted using Java on a system with an Intel Dual core processor running at 1.73 GHz, 2GB RAM. Algorithms AES is implemented using FlexiCoreProvider with Eclipse. Initially we created one CSP, data owner and TPA. Data owner stores data on CSP and request auditor to perform auditing. TPA performs the auditing for requested owner & audits the all records of that owner. As a respons to auditing request auditor generates the auditing report containing the status whether the user data is tampered or not, server computation time in ms and auditing time in ms. To achieve results we took a file range from 100 to 1000 KB. Results were obtained after taking number of trials for different users. Fig. 3 shows the auditing performance graph. In our observation we find that as the number of records challenged for auditing increases, corresponding auditing time and server computation time also get increased. We also find that our scheme is privacy preserving, as we had converted the file into encrypted format.

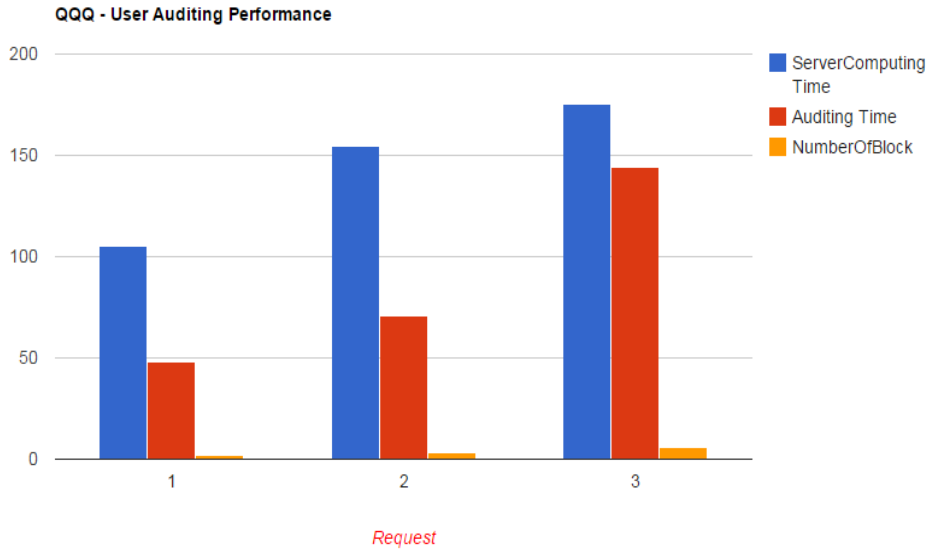


**Figure 3**: User auditing performance

## 5. Conclusion

Cloud Computing is an emerging infrastructure paradigm for storing data. But as market grows the threat on data also grows. In this paper, we have proposed the privacy and integrity preserving dynamic auditing protocol. It protects the data privacy against the third party auditor by using the cryptographic techniques. Here, TPA plays very important role, Privacy-preserving to ensure that TPA maintains the correctness of the cloud data on demand without deriving the user's data contents from the information collected during auditing process. It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted. Proposed scheme also supports the data dynamic modifications and batch auditing for multiple clouds.

## References

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

**[2]** Armbrust,M., Fox, A.,Griffith,R., Joseph, A.D.,Katz, R.H.,Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)

[3] Bairavasundaram, L.N.,Goodson, G.R., Pasupathy, S., Schindler, J.: An analysis of latent sector errors in disk drives. In: Proceedings of the 2007 ACMSIGMETRICS International conference on measurement and modeling of computer systems (SIGMETRICS'07), pp. 289–300. ACM (2007)

[4] Kher, V., Kim, Y.: Securing distributed storage: challenges, techniques, and systems. In: Proceedings of the 2005 ACM workshop on storage security and survivability (StorageSS05), pp. 9–25. ACM (2005)

[5] Schroeder, B., Gibson, G.A.: Disk failures in the real world: What does an mttf of 1, 000, 000 hours mean to you. In: Proceedings of the 5th USENIX conference on file and storage technologies (FAST'07), pp. 1–16. USENIX (2007)

[6] Filho, D.L.G., Barreto, P.S.L.M.: Demonstrating data possession and uncheatable data transfer. IACR Cryptology ePrint Archive **2006**, 150 (2006)

[7] Juels, A., Jr., Kaliski, B.S.: PORS: proofs of retrievability for large files. In: Proceedings of the 14th ACM conference on computer and communications security (CCS'07), pp. 584–597. ACM (2007)
[8] Sebé, F., Domingo-Ferrer, J., Martínez-Ballesté, A., Deswarte, Y., Quisquater, J.J.: Efficient remote data possession checking in critical information infrastructures. IEEE Trans. Knowl. Data Eng. **20**(8), 1034–1038 (2008)
[9] Yamamoto, G., Oda, S., Aoki, K.: Fast integrity for large data. In: Proceedings of the ECRYPT workshop on software performance enhancement for encryption and decryption, pp. 21–32. ECRYPT, Amsterdam, the Netherlands (2007)
[10] G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J. Peterson, and D.X. Song, "Provable Data Possession at Untrusted Stores," Proc. ACM Conf. Computer and Comm. Security, P. Ning, S.D.C. di Vimercati, and P.F. Syverson, eds., pp. 598-609, 2007.
[11] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, J. Pieprzyk, ed.,pp. 90-107, 2008.
[12] C.C. Erway, A. Ku¨ pc¸u¨ , C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. ACM Conf. Computer and Comm. Security, E. Al-Shaer, S. Jha, and A.D. Keromytis, eds., pp. 213-222, 2009.
[13] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.
[14] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE INFOCOM, pp. 525-533, 2010.
[15] Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231-2244, Dec. 2012.
[16] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S. Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds," Proc. ACM Symp. Applied Computing, W.C. Chu, W.E. Wong, M.J. Palakal, and C.-C. Hung, eds., pp. 1550-1557, 2011.
[17] K. Zeng, "Publicly Verifiable Remote Data Integrity," Proc. 10th Int'l Conf. Information and Comm. Security, L. Chen, M.D. Ryan, and G. Wang, eds., pp. 419-434, 2008.
[18] G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage from Homomorphic Identification Protocols," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, M. Matsui, ed., pp. 319-333, 2009.
[19] Kan Yang, and Xiaohua Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 9, September 2013.

## Author Profile

**Anuradha Appasaheb Jagadale** received B.E. in Computer Science and Engineering from Bharati Vidyapeeth College Of Engg., Kolhapur, India in 2011. Currently pursuing M. Tech in Computer Science and Engg. from Symbiosis Institute of Technology, Symbiosis International University, Pune, India. Area of intrestest Parallel and Distributed Systems.