A Review on User Privacy Preserving and Auditing for Secure Data Storage System in Cloud

Dimple Bedmutha
MET BKC Adgaon, Nashik,
Savitribai Phule Pune University,
Maharashtra India

P. M. Yawalkar
MET BKC Adgaon, Nashik,
Savitribai Phule Pune University,
Maharashtra India

ABSTRACT
Cloud Storage permits users to remotely store their data and also provides users with on-demand self service from a shared pool of configurable and computable resources and that can be rapidly provisioned and realized with minimal management efforts or service provider interaction [2]. Despite of its advantage, outsourcing storage prompts a number of interesting challenges. One of the important factors that need to be taken into consideration is to assure the user about the correctness of his outsourced data. Also, without worrying for the need to verify its correctness, cloud user should be able to use the cloud storage. Thus, enabling public verifiability for cloud storage system is of critical importance so that cloud user can resort to an external audit party i.e. third party auditor (TPA) to check the correctness of outsourced data. For TPA to be secure and effective, the auditing process should not introduce no new vulnerabilities that violate users' data privacy and no additional online burden to cloud user. In this paper, a secure data storage system that supports user privacy preserving and auditing is being proposed.

Keywords
User privacy preserving, cloud computing, TPA

1. INTRODUCTION
Cloud computing as a service over the internet is the provision of dynamically scalable and virtualized resources. Storing data remotely to the cloud in flexible on –demand way brings in ample of features [2]:

1. Agility which improves with ability of user to re-provision technological infrastructure resources.

2. Improvements for systems over utilization and efficiency are often only 10–20% utilized.

3. Using web services as the system interface performance can be monitored and consistent as well as loosely coupled architectures is being constructed.

4. Because of the increased security-focused resources, on centralization of data, etc., security could be improved; but there are still chances about leakage of sensitive data. However, when the data is distributed over a larger number of devices, complexities of security is drastically increased. Moreover, access of user to security audit logs may seem to be difficult or may be impossible. Users' desire regarding the avoidance of loss of control over information security and to retain control over the infrastructure motivated installations of private cloud.

5. Cloud computing applications need not to be installed on individual user's desktop and are accessible from any corner of the world, so maintenance of cloud applications is easier.

In order to solve security issues it has become necessary to verify the data integrity at untrusted servers. For example, cloud service providers (CSP) may discard the data that has been rarely accessed, or might even hide data loss incidents in order to maintain a reputation [3], [11], [12]. Even though outsourcing data to cloud is economically affordable for long-term large scale data storage, but it fails to provide immediate guarantee on availability and data integrity. Direct adoption of traditional cryptography is not sufficient to assure the data security protection. User auditability ensures data integrity of remotely stored data under different system and security models [3], [5], [6], [13]. User auditability allows third party (external auditors) to ensure the integrity of data on behalf of user. Users rely on TPA for storage security of their data and do not want this auditing process to bring in new vulnerabilities of unauthorized information leakage that leads to violation of data security [10]. Without a properly designed auditing protocol, encryption alone cannot prevent data from being the hands of external parties during the auditing process.

In order, to achieve a privacy-preserving external party auditing protocol to be independent of data encryption techniques, the following two requirements has to be fulfilled: 1) The audit of cloud data storage should be efficiently performed by TPA should efficiently audit cloud data storage without having the need to demand for local copy of data, and should not bring in any on-line burden to the user; 2) The third party auditor should not introduce any new vulnerabilities that will violate privacy preserving guarantee.

2. RELATED WORK
G. Ateniese et al. [3] proposed a Provable Data Possession (PDP) model which helps client who had outsourced their data that on untrusted server to make audit that the server has maintained their original data without retrieving it. For auditing homomorphic authenticators scheme is being utilized and suggests random samples a few blocks of the file. For the public verifiability the scheme demands for the linear combination of sampled blocks to be sent to third party auditor. However, the direct usage of these techniques is not suitable because the linear combinations of blocks may potentially reveal the user data information, and thus it fails to provide the guarantee of privacy being preserved. In case, if quantity of the linear combinations of the same data blocks is being collected during audit process, then by solving a system of linear equations TPA can derive the user's data content. Later, G. Ateniese et al. [4] constructed PDP technique which is highly efficient and secure is based on symmetric key cryptography and does not require bulky encryption. The Limitation of this technique is firstly that the number of updates to be made and challenges to be made by client is restricted and fixed a prior. Secondly, block insertions cannot

be made anywhere and only append-type insertions are allowed.

Juels et al. [5] proposed a Proof of retrievability protocol in which a server ensures a client that a outsourced file F is correctly present on server and the client can retrieve all of F with highest possible probability. The scheme two methods spot-checking and Error-correcting code which ensures possession and retrievability of data files on remote archive or backup service systems. The Limitation with this scheme is that the number of queries a client can make is limited and fixed a priori. This approach is suitable only with encrypted data. However, the quantity of audit challenges a user will perform could be a permanent priori, and public auditability isn't supported in their main scheme. Shacham et al. [6] designed the protocols based which uses homomorphic authenticators for file blocks, which makes use of block integrity values that can be efficiently aggregated in order to reduce bandwidth in PoR protocol and also to encode the file this scheme can make use of more efficient erasure code which is later transformed into an error-correcting code. Advantage of this scheme is that the unlimited number of very audits can be done, but still the solution remains static. To get dynamic solution, if even change of few bits is made to the contents of F it must propagate via error-correcting code, which introduces significant computation and communication complexity was proposed by Bowers in 2009. K. D. Bowers et al. [7], introduced High-Availability and Integrity Layer which is a distributed cryptographic system that allows multiple servers to ensure a client that a outsourced data file is correct and retrievable.

C.Wang et al. [12], designed an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of user's data in the cloud. It relies on erasure correcting code in the file distribution preparation in order to provide redundancies and guarantee the data dependability. The limitation of the scheme is that the numbers of challenges user's can perform against the server are limited and also, user has burden of storing pre-computed tokens locally. Q. Wang et al. [13], designed a scheme with explicit dynamic data support to ensure the correctness of user's data in the cloud. User can easily audit the correctness of his outsourced data with the help of challenge response protocol without much overhead. This scheme fails to address privacy concerns of users.

C. Erway et al. [8], designed a Dynamic Provable Data Possession method which assures a client that a cloud server maintains data file F in an informal sense. But it does not provide guarantee to client for retrieval of the file. Also the scheme doesn't support all dynamic operations. Curtmola et al. [9] tries to ensure data possession of multiple replicas over distributed storage system. They extend the PDP scheme in [3] is extended to cover multiple replicas without having need for being encoding individual replica separately, providing guarantee and that multiple copies of the distributed data are actually maintained.

The portions of work presented in paper [1], [10], [11] describes that data can be authenticated using two ways: one of the ways is to just upload the data blocks with their corresponding MACs to the cloud server along with corresponding secret key *sk* to the TPA. Blocks and their MACs can be retrieved randomly by TPA to verify the integrity of stored data via *sk.* Drawbacks of this solution: 1) its communication and computation complexity; 2) It is essential for TPA to have the knowledge of the data blocks for verification which violates privacy preserving guarantee. To overcome flaws of this scheme, other way is : For the whole data file *F*, cloud user select *s* message authentication code keys $\{sk_\tau\}_{1\le\tau\le s,}$ randomly and pre-computes *s* (deterministic) MACs, $\{\text{MAC}_{sk\tau}(F)\}_{1\le\tau\le s}$ and then forwards these verification metadata (keys and the MACs) to TPA. For each audit, the TPA can send the secret key *sk* to the cloud server and for comparison TPA requests for a fresh keyed MAC. This scheme is an improvement over the above scheme I, were TPA cannot see the data and hence it preserves privacy. However, it suffers from following drawbacks: 1) Job of the TPA is to maintain and update state between audits; 2) the number of times the particular file can be audited is limited by secret keys that must be fixed initially. Once all possible secret keys are used up, the user then has to download all the data from cloud to re-compute and re-publish new MACs to TPA.

3. PRIVACY PRESERVING AND PUBLIC AUDITING SCHEME

For cloud data storage security the public auditing scheme provides complete solution for checking integrity of outsourced data. In order to enable user-privacy preserving auditing for cloud data storage, mentioned protocol design should achieve the following performance and security guarantee:

1) Public verifiability: enables TPA to verify the correctness of the cloud data without retrieving the original copy of stored data.

2) Storage integrity: must ensure that cloud server should not be able forward the audit from TPA without indeed storing users' data intact.

3) Privacy-preserving: helps to ensures that from the information which is gathered during the auditing process the TPA should not be able to derive users' data content.

4) Batch auditing: enables TPA to securely and efficiently to cope with multiple auditing delegations from probably large number of different users at a time.

5) Lightweight: TPA should perform verification with minimum computation and communication overhead.

Overview of the scheme: Cong Wang et al. [1] proposed that user-preserving and auditing can be achieved by uniquely integrating homomorphic authenticator with random masking technique. In the proposed protocol, the linear combination of the sampled data blocks is being masked with randomness which is generated by a pseudo random function (PRF). By using random masking method, the TPA cannot be able get all the necessary information through which a correct group of linear equations can be build up and hence it cannot derive the user's original data blocks [3], [5] ,[6], even though large number of linear combinations of the same set of file blocks can be collected. Using to the algebraic property of the homomorphic authenticator, the data integrity validation of the block-authenticator pairs will remain unaffected by the randomness generated via a PRF, as illustrated in Fig. 1.Here HLA proposed in [6] is utilized which is based on the scheme of short signature as proposed in [15].

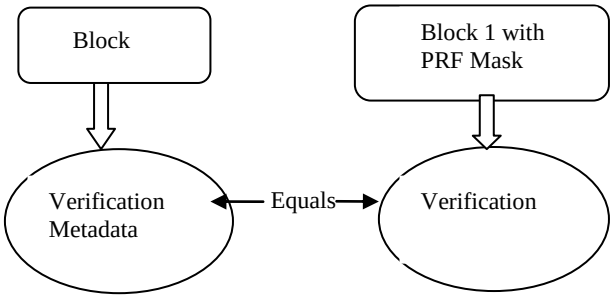

Fig. 1: Random mask by PRF

3.1 Algorithms Details

There are four algorithms in public auditing scheme: Key Generation, Signature Generation, Genproof and Verifyproof. User uses key generation algorithm to set up the scheme. Verification metadata is generated by the signature generation algorithm, where signature or identity of user is generated. Genproof algorithm is run on the cloud server to check the data storage correctness in the cloud, and for auditing the proof TPA uses to audit the proof of data storage integrity. Finally, VerifyProof is run by the TPA to verify the proof of storage correctness from the cloud server.

Public auditing scheme works in two phases namely: Setup and Verify

Setup Phase: The user initializes the public and secret parameters of the system are being initialized by the user by executing Key Generation, and pre-processes the data file *F* via Signature Generation in order to generate the verification metadata. At the cloud server user stores the data file F and delete its local copy, and forwards the verification metadata to TPA for verification to check for the correctness of stored data.

Verify Phase: The TPA sends an audit message to the cloud server in order to get an assurity that cloud server has properly retained the data file *F* at the time of the verification. By executing GenProof, cloud server will derive a response message for stored data file *F*. TPA uses verification metadata and verifies the response by executing VerifyProof algorithm**.**

3.2 Multiple Batch Auditing

TPA may handle multiple auditing delegations concurrently after receiving different users' requests. The auditing of individual tasks can be very inefficient for TPA. Given K auditing delegations from K different users on K distinct data files, it is more advantageous for TPA to make audit only once if these multiple tasks are batch together [14]. Thus by using signature aggregation technique and bilinear property, it is possible to aggregate K verification equations into a single equation, in order to achieve auditing of multiple tasks simultaneously.

3.3 Supports for Data Dynamics

The scheme explicitly and efficiently handles dynamic data operations of outsourced data in cloud. It is necessary to consider case the case, where a user may wish to perform various dynamic block-level operations of update, delete and append to modify the data file while maintaining the storage correctness assurance [13]. To support this user must be able to download all the data from the cloud servers and recomputed the whole parity blocks as well as verification tokens.

4. CONCLUSION

It has been reviewed that by utilizing homomorphic linear authenticator with random masking technique users are assured that TPA would not be able to learn any knowledge during auditing process about original data content stored on cloud. Also, the burden of users from auditing task is eliminates and users fear of outsourced data leakage is being alleviated. It has been considered that for better efficiency, TPA can perform multiple auditing tasks in batch manner as TPA is capable of handling concurrently multiple audits. Hence, privacy preserving and user auditing system provides the guarantee of cloud data correctness and availability.

5. ACKNOWLEDGEMENTS

The author is thankful to MET's Institute of Engineering Bhujbal Knowledge City Nasik, HOD of compute department, guide and parents for their blessing, support and motivation behind this work**.**

6. REFERENCES

[1] Cong Wang, Sherman S.-M. Chow, Qian Wang, Kui Ren, and Wenjing Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Transactions on Cloud Computing Year, 2013.

[2] http://en.wikipedia.org/wiki/Cloud_computing.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 598–609.

[4] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. of SecureComm'08, 2008, pp. 1–10.

[5] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.

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

[7] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Cryptology ePrint Archive, Report 2008/175, 2008, http://eprint.iacr.org/.

[8] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of CCS'09, 2009, pp. 213–222.

[9] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-replica provable data possession," in Proc. of ICDCS'08. IEEE Computer Society, 2008, pp. 411–420.

[10] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. of HotOS'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 1–6.

[11] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008, http://eprint.iacr.org/.

[12] C.Wang, Q.Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in Proc. of IWQoS'09, July 2009, pp. 1–9.

[13] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS'09, volume 5789 of LNCS. Springer-Verlag, Sep. 2009, pp. 355–370.

[14] A. L. Ferrara, M. Greeny, S. Hohenberger, M. Pedersen (2009), "Practical short signature batch verification", in Proceedings of CT-RSA, volume 5473 of LNCS. Springer-Verlag, pp. 309–324.

[15] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," J. Cryptology, vol. 17, no. 4, pp. 297–319, 2004.