Privacy Preserving For Scalable and Efficient for Dynamic Audit

M.Suganya*, S.Dhanalakshmi.,** *M.E.,*
**Department of computer science and Engineering,*
A.R.J college of Engineering,
Mannargudi-India.
suganmahalingam2692@gmail.com
***Assistant professor,*
Department of computer science and Engineering,
A.R.J college of Engineering,
Mannargudi-India.
dhana8891@gmail.com

***Abstract*- Cloud computing consists a collection of computers and servers that are publicly accessible via the Internet. User accesses the data's and will pay as per user basis. Cloud computing has four essential characteristics: elasticity and the ability to scale up and down, self-service provisioning and automatic reprovisioning, application programming interfaces (APIs), billing and metering of service usage in a pay-as-you-go model. While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users‟ outsourced data. Data integrity and storage efficiency are two important requirements for cloud storage. Proof of Retrievability (POR) and Proof of Data Possession (PDP) techniques assure data integrity for cloud storage. Proof of Ownership (POW) improves storage efficiency by securely removing unnecessarily duplicated data on the storage server. The cloud storage service (CSS) relieves the burden for storage management and maintenance. Fragment Structure, random sampling and index table is used to construct the Audit service. These techniques are supported provable updates to cloud outsourced data. The third party auditing allow to save time and computation resources with reduced online burden of the user. In this work, a method based on Probabilistic query and periodic verification for improving the performance of audit services and also audit system verifies the integrity.**

Keywords: **Cloud computing, Proof of retrievability, integrity, auditing.**

I. INTRODUCTION

Cloud is not only for storage purpose but can also share across multiple users. The integrity of cloud leads to doubt. Cloud computing is a recent trending in IT that moves computing and data away from desktop and portable PCs into large data centres. It leads to applications handover as services over the Internet as well as to the actual cloud infrastructure — namely, the hardware and systems software in data centres that provide these services.

Considering the large size of the outsourced electronic data and the client's constrained resource capability, the core of the problem can be generalized as how can the client find an efficient way to perform periodical integrity verification without the local copy of data files. In order to overcome this problem, many schemes have been proposed under different system and security models [1], [2], [3], [4], [5], [6], [7],[8] . In all these works, great efforts have been made to design solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and irretrievability of data, etc. According to the role of the verifier in the model, all the schemes available fall into two categories: private verifiability and public verifiability.

A. Our Contributions

Our contribution can be summarized as follows: POR, a new POR scheme with two independent cloud servers. Particularly, one server is for auditing and the other for storage of data. The cloud audit server (CAS) is not required to have high storage capacity. Different from the previous work with auditing server and storage server, the user is relieved from the computation of the tags for files, which is moved and outsourced to the cloud audit server. Furthermore, the cloud audit server also plays the role of auditing for the files remotely stored in the cloud storage server.

- Develop a strengthened security model by considering the reset attack against the storage server in the upload phase of an integrity verification scheme. It is the first POR model that takes reset attack into account for cloud storage system.
- Present an efficient verification scheme for ensuring remote data integrity in cloud storage. The proposed scheme is proved secure against reset attacks in the strengthened security model while supporting efficient public verifiability and dynamic data operations simultaneously.

II. RELATED WORK

Recently, much research effort has been devoted largely to ensure the security of cloud computing [10], [15], [16], [17] and remotely stored data [1], [2], [3], [18]. Ateniese et al. [1] defined the "provable data possession" (PDP) model for ensuring possession of files on untrussed storages. They also proposed the first proof-of-storage scheme that supports public verifiability. The scheme utilizes RSA-based homomorphic tags for auditing outsourced data, such that a

linear combination of file blocks can be aggregated into a single block and verified by employing homomorphic property of RSA. However, the data owner has to compute a large number of tags for those data to be outsourced, which usually involves exponentiation and multiplication operations.

For the first time, Erway et al. [12] explored constructions for dynamic provable data possession. This scheme is essentially a fully dynamic version of the PDP solution. In particular, to support updates, especially for block insertion, they tried to eliminate the index information in the "tag" computation in Ateniese's PDP model [1]. However, any update on the stored file F, even few blocks, will result in the inevitable updates of rank and interval information of all nodes along the path from the updated blocks to the top leftmost node, thus introducing significant computational complexity and losing desirable efficiency. In our solution, it propose an efficient remote data verification scheme simultaneously supporting public verifiability and fully dynamic data operations for POR systems.

As an extension of [9], this paper firstly formally defines the system model and security model for the cloud storage. Furthermore, it also presents the detailed security analysis and efficiency analysis for POR in this paper under the new security model.

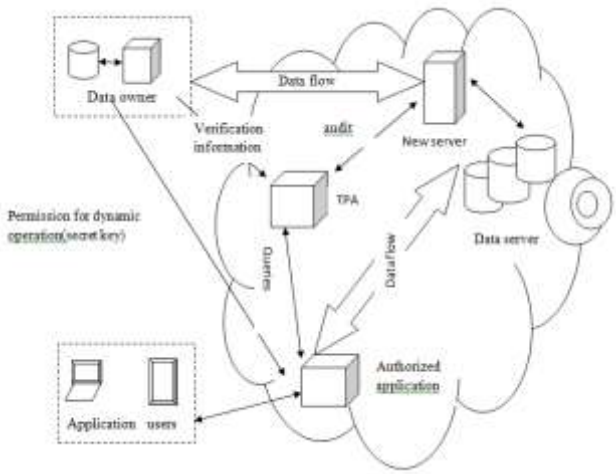


Fig. 1. Cloud data storage architecture.

III. PROBLEM STATEMENT

A. System Model

Representative network architecture for cloud data storage is illustrated in Fig. 1. Three different network entities can be identified as follows: Client. An entity that has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation can be either individual Consumers or organizations.

B. Cloud storage server

An entity, which is managed by cloud service provider, has significant storage space and computation resource to maintain client's data. The CSS is required to provide integrity proof to the clients or cloud audit server during the integrity checking phase.

C. Cloud audit server

A TPA, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request In this system, the cloud audit server also generates all the tags of the files for the users before uploading to the cloud storage server.

1) Design Goals:

Our design goals can be summarized as the following: (1) Public verifiability: to allow anyone, not just the clients originally stored the file, to have the capability to verify correctness of the remotely stored data; (2) Low computation overhead at the client side: to upload data to the cloud server while supporting verifiability, the data owner does not have heavy additional computation; (3) Dynamic sdata operation support: to allow the clients to perform block level operations on the data files while maintaining the same level of data correctness assurance; (4) Stateless verification: to eliminate the need for state information maintenance at the verifier side between audits and throughout the long term of data storage.

IV. CONSTRUCTION OF POR SCHEMES CLIENT MODULE

An entity that has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation can be either individual consumers or organizations.

A. Cloud server module

An entity, which is managed by Cloud Service Provider (CSP) has significant storage space and computation resource to maintain the clients' data. In the cloud paradigm, by putting the large data files on the remote servers, the clients can be relieved of the burden of storage and computation. As clients no longer possess their data locally, it is of critical importance for the clients to ensure that their data are being correctly stored and maintained. That is, clients should be equipped with certain security means so that they can periodically verify the correctness of the remote data even without the existence of local copies the clients may interact with the cloud servers via CSP to access or retrieve their prestored data.

B. public auditing module

Public audit ability also allows clients to delegate the integrity verification tasks to TPA while they themselves can be unreliable or not be able to commit necessary computation resources performing continuous verifications. Public audit ability allows anyone, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no private information. Then, clients are able to delegate the evaluation of the service performance to an independent third party auditor (TPA). In the cloud, the clients themselves are unreliable or may not be able to afford the overhead of performing frequent integrity checks. Thus, for

practical use, it seems more rational to equip the verification protocol with public audit ability, which is expected to play a more important role in achieving economies of scale for Cloud Computing. Homomorphism authenticators are un-forgeable verification metadata generated from individual data blocks in TPA for MHT, which can be securely aggregated in such a way to assure an auditor that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator.

C. Batch auditing module

To extend our scheme to support scalable and efficient public auditing in Cloud Computing we perform batch auditing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA. As cloud servers may concurrently handle multiple verification sessions from different clients. Batch auditing not only enables simultaneously verification from multiple-client, but also reduces the computation cost on the TPA side. To support efficient handling of multiple auditing tasks, further explore the technique of bilinear aggregate signature to extend our main result into a multiuser setting, where TPA can perform multiple auditing tasks simultaneously.

D. Data dynamics

Data dynamics via the most general forms of data operation, such as block modification, insertion, and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. To achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the classic Markel Hash Tree construction for block tag authentication. Hence, supporting data dynamics for privacy-preserving public risk auditing is also of paramount importance. Now we show how our main scheme can be adapted to build upon the existing work to support data dynamics, we can adopt this technique in our design to achieve privacy-preserving public risk auditing with support of data dynamics.

E. Key generation

The owner generates a public/secret key pair (pk, sk) by himself or the system manager, and then sends his public key pk to TPA. Note that TPA cannot obtain the client‟s secret key sk; secondly, the owner chooses the random secret.

F. Tag generation

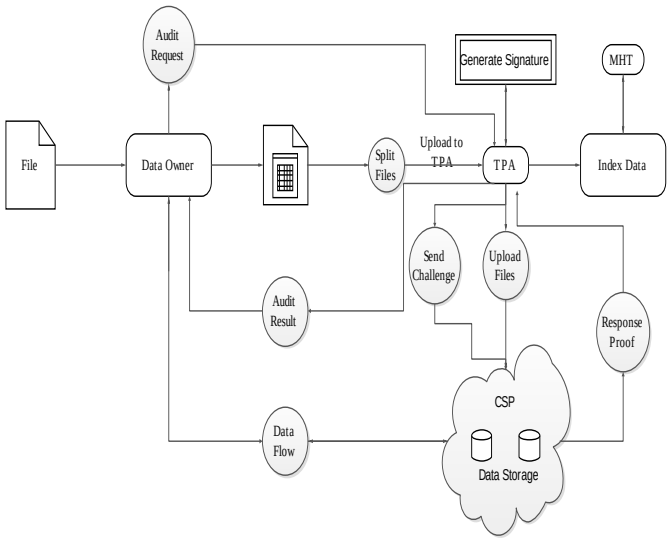
The client (data owner) uses the secret key sk to pre-process a file, which consists of a collection of n blocks, generates a set of public verification parameters and index-hash table that are stored in TPA, and transmits the file and some verification tags to CSP.

G. Periodic sampling audit

TPA (or other applications) issues a “Random Sampling” challenge to audit the integrity and availability of outsourced data in terms of the verification information stored in TPA[9].

The AAs should be cloud application services inside clouds for various application purposes, but they must be specifically authorized by DOs for manipulating outsourced data. Since the acceptable operations require that the AAs must present authentication information for TPA, any unauthorized modifications for data will be detected in audit processes or verification processes. Based on this kind of strong authorization-verification mechanism, we assume neither CSP is trusted to guarantee the security of stored data, nor a DO has the capability to collect the evidence of CSP‟s faults after errors have been found [14].

V. DATA FLOW DIAGRAM



VI. RESULTS

To ensure the security, dynamic data operations are available only to DOs or AAs, who hold the secret key sk. Here, all operations are based on data blocks. It is necessary for TPA and CSP to check the validity of updated data. First, an AA obtains the public verification information from TPA [12][13]. Second, the application invokes the Update, Delete, and Insert algorithms, and then sends to TPA and CSP, respectively. Next, the CSP makes use of an algorithm Check to verify the validity of updated data. Note that the Check algorithm is important to ensure the effectiveness of the audit. Finally, TPA modifies audit records after the confirmation message from CSP is received and the completeness of records is checked.

VII. CONCLUSION

Dynamic audit services for untrusted and outsourced storages. It also presented an efficient method for periodic sampling audit to enhance the performance of TPAs and storage service providers. Our experiments showed that our solution has a small, constant amount of overhead, which minimizes computation and communication costs. As a conclusion, both the experimental results demonstrate that privacy-preserving cost intermediate data sets can be saved significantly through our approach over existing ones where all data sets are encrypted. In this paper, I have proposed an approach that

identifies which part of intermediate data sets needs to be encrypted while the rest does not, in order to save the privacy preserving cost. A tree structure has been modeled from the generation relationships of intermediate data sets to analyze privacy propagation among data sets.

In accordance with various data and computation intensive applications on cloud, intermediate data set management is becoming an important research area. Privacy preserving for intermediate data sets is one of important yet challenging research issues, and needs intensive investigation. With the contributions of this paper, it are planning to further investigate privacy aware efficient scheduling of intermediate data sets in cloud by taking privacy preserving as a metric together with other metrics such as storage and computation. Optimized balanced scheduling strategies are expected to be developed toward overall highly efficient privacy aware data set scheduling.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "*Provable data possession at untrusted stores*," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 598–609.

[2] A. Juels and B. S. Kaliski Jr, "*Pors: Proofs of retrievability for large files*," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 584–597.

[3] H. Shacham and B. Waters, "*Compact proofs of retrievability*," in Proc. 14th Int. Conf. Theory Appl. Cryptol. Inf. Security, 2008, pp. 90– 107.

[4] K. D. Bowers, A. Juels, and A. Oprea, "*Proofs of retrievability: Theory and implementation*," in Proc. ACM Workshop Cloud Comput. Security, 2009, pp. 43–54.

[5] M. Naor and G. N. Rothblum, "*The complexity of online memory checking*," J. ACM, vol. 56, no. 1, pp. 2:1–2:46, Feb. 2009.

[6] E.-C. Chang and J. Xu, "*Remote integrity check with dishonest storage server*," in Proc. 13th Eur. Symp. Res. Comput. Security, 2008, pp. 223–237.

[7] M. A. Shah, R. Swaminathan, and M. Baker. (2008). Privacy-preserving audit and extraction of digital contents, Cryptology ePrint Archive, Report 2008/186 [Online]. Available: http://eprint.iacr. org/Sens. Netw., vol. 8, no. 1, pp. 9:1–9:24, Aug. 2011.

[8] L. V. M. Giuseppe Ateniese, R. D. Pietro, and G. Tsudik, "*Scalable and efficient provable data possession*," in Proc. Int. Conf. Security Privacy Commun. Netw., 2008, pp. 46–66.

[9] J. Li, X. Tan, X. Chen, and D. S. Wong, "*An efficient proof ofretrievability with public auditing in cloud computing*," in Proc.5th Int. Conf. Intell. Netw. Collaborative Syst., 2013, pp. 93–98.

[10] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "*Fine-grained accesscontrol system based on attribute-based encryption*," in Proc. Eur.Symp. Res. Comput. Security, 2013, pp. 592–609.

[11] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia. (2008).Dynamic provable data possession, Cryptology ePrint Archive,Report 2008/432 [Online]. Available: http://eprint.iacr.org/

[12] R. C. Merkle, "*Protocols for public key cryptosystems*," in Proc.IEEE Symp. Security Privacy, 1980, pp. 122–133.

[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "*Privacy-preserving publicauditing for data storage security in cloud computing*," in Proc.IEEE Conf. Comput. Commun., 2010, pp. 525–533.

[14] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "*Enabling publicverifiability and data dynamics for storage security in cloudcomputing*," in Proc. 14th Eur. Symp. Res. Comput. Security, 2009,pp. 355–370.

[15] J. Li and K. Kim, "*Hidden attribute-based signatures without anonymityrevocation*," Inf. Sci., vol. 180, no. 9, pp. 1681–1689, 2010.

[16] J. Li, C. Jia, J. Li, and X. Chen, "*Outsourcing encryption of attribute-based encryption with mapreduce*," in Proc. 14th Int. Conf.Inf. Commun. Security, 2012, pp. 191–201.

[17] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "*New algorithms ofoutsourcing modular exponentiations*," in Proc. Eur. Symp. Res.Comput. Security, 2012, pp. 541–556.

