Secure Cloud Storage by Dynamic-Hash-Table based Public Auditing

Meghana D[1], Nithya E[2]

[1](M. Tech Student, Department of CSE, Dr. Ambedkar institute of technology, Bangalore, India
[2](Assistant Professor, Department of CSE, Dr. Ambedkar institute of technology, Bangalore, India

Abstract - Cloud storage is an increasingly popular application of cloud computing, which can provide on-demand outsourcing of data services for both organizations and individuals. However, users may not fully trust the cloud service providers (CSPs) so that it is difficult to determine whether the CSPs meet their legal expectations for data security. Therefore, it is critical to develop efficient auditing techniques to strengthen data owners' trust and confidence in cloud storage. In this paper, we present a public auditing scheme for secure cloud storage based on dynamic hash table (DHT), which is a new two-dimensional data structure located at a third parity auditor (TPA) to record the data property information for dynamic auditing. Differing from the existing works, the proposed scheme migrates the authorized information from the CSP to the TPA, and thereby significantly reduces the computational cost and communication overhead. Meanwhile, exploiting the structural advantages of the DHT, our scheme can also achieve higher updating efficiency than the existing schemes. In addition, we extend our scheme to support privacy preservation by combining the homomorphic authenticator based on the public key with the random masking generated by the TPA, and achieve batch auditing by employing the aggregate BLS signature technique. We formally prove the security of the proposed scheme, and evaluate the auditing performance by detailed experiments and comparisons with the existing ones.

Key Words: **Cloud storage, Cloud security, Public Auditing, Dynamic Hash Table, Homomorphic Authenticator, Batch Auditing, BLS signature technique.**

1. INTRODUCTION

Cloud storage is an important branch of cloud computing , whose goal is to provide powerful and on demand out sourcing data services for users exploiting highly virtualized infrastructures. Due to the low cost and high performance of cloud storage, a growing number of organizations and individuals are tending to outsource their data storage to professional cloud services providers(CSP), which buoys the rapid development of cloud storage and its relative techniques in recent years. However, as a new cutting edge technology, cloud storage still faces many security challenges. One of the biggest concerns is how to determine whether a cloud storage system and its provider meet the legal expectations of customers for data security. This is mainly caused by the following reasons. First, cloud users (data owners), who outsource their data in clouds, can no longer verify the integrity of their data via traditional techniques that are often employed in local storage scenarios. Second, CSPs, which suffer Byzantine failures occasionally, may choose to conceal the data errors from the data owners for their own self interest. Whatever more severe, CSPs might neglect to keep or even deliberately delete rarely accessed data that belong to ordinary customers to save storage space. Therefore, it is critical and significant to develop efficient auditing techniques to strengthen data owners trust and confidence in cloud storage, of which the core is how to effectively check data integrity remotely.

So far, many solutions have been presented to overcome this problem, which can be generally divided into two categories: private auditing and public auditing. Private auditing is the initial model for remote checking of data integrity in which the verification operation is performed directly between data owners and CSPs with relatively low cost. However, it cannot provide convincing auditing results, since the owners and CSPs often mistrust each other. Moreover, it is not advisable for the users to carry out the audit frequently, since it would substantially increase the overhead that the users may not afford. Thus, the public auditing scheme is presented, in which the checking work is normally done by an authorized third party auditor (TPA). Compared with the former, the latter can offer dependable auditing results and significantly reduce unnecessary burden by introducing an independent TPA. Thus, it is more rational and practical, and popularly believed to be the right direction of future development. In the public auditing, however, some vital problems are as follows:

- **Privacy preserving**: data privacy protection (DPP) has always been an important topic for cloud storage. In the public auditing, the core of this problem is how to preserve uses' privacy while introducing a TPA. Although exploiting data encryption prior to outsourcing is an approach to mitigate the privacy concern in cloud storage, it cannot prevent data leakage during the verification process. Thus, it is important for the cloud auditing to include a privacy-preserving mechanism independent to data encryption.
- **Batch auditing**: to enhance the efficiency and enable the scalability of public auditing, the TPA should deal with multiple auditing tasks from various users in a fast and cost-efficient manner, therefore support the batching auditing.
- **Dynamic auditing:** it is well known that a

cloud storage system is not just a data warehouse, the users often need to update the data dynamically motivated by various application requirements. Therefore, it is significant for cloud storage auditing to support data dynamics.

In view of these problems, this paper presents a public auditing scheme (DHT-PA) using a new data structure called dynamic hash table (DHT). Exploiting the DHT, our schem can achieve dynamic auditing. Moreover, because DHT-PA migrates the authorized information from the CSP to the TPA, its computational costs and communication overhead are significantly smaller.

In addition, this paper extend DHT-PA to achieve privacy preserving by combining the homomorphic authenticator based on the public key with random masking generated by the TPA. Furthermore, we employ the well known BLS (Boneh-Lynn-Shacham) signature and bilinear maps to achieve batch auditing. Specifically, our contribution in this work can be summarized as follows:

1. We present a novel public auditing scheme, which can completely support three vital functions, i.e., dynamic data auditing, privacy protection and batch auditing.

2. We design a new data structure named DHT to record data properties for auditing in the TPA, and by virtue of it, achieve rapid auditing and efficient data updating.

3. We formally prove the security of proposed scheme, and evaluate its auditing performance by concrete experiments and comparisons with state-of-the-art schemes. The results demonstrate that the proposed scheme can effectively achieve secure auditing in clouds, and outperforms the previous ones in computation complexity, storage costs and communication overhead.

2. BACKGROUND

In recent years, cloud storage auditing has attracted increasing attention. One of the earliest related work is proof of retrievability (PoRs) presented by Juels[8] in 2007, which can check the correctness of data stored on the CSP and ensure data's retrievability with the use of error-correcting code. However, PoRs is a typical private auditing solution, and does not support auditing by the third party. In the same year, Ateniese[9] first presented an original public auditing scheme, provable data possession (PDP), which employs homomorphic tags based on RSA and can remotely check the integrity of outsourced data by randomly sampling a few blocks from the file. As above mentioned, compared with the private auditing, the public auditing can provide dependable verification results and greatly reduce users unnecessary overhead by introducing an independent TPA. Thus, Public auditing is believed to be more practical and promising.

Besides, there are some other significant concerns for cloud storage auditing, such as, privacy protection, batch auditing and dynamic auditing. To address the data-leakage concern, a privacy-preserving auditing protocol was presented. By integrating the homomorphic authenticator with the random masking, this protocol can guarantee that the TPA could not obtain any knowledge on the data content stored in the cloud servers during the whole verification process. Particularly, the authors observantly pointed out that privacy protection is indispensable for achieving the public auditability. Moreover, Wang extended their privacy-preserving auditing protocol into a multiuser setting to support batch verification for better efficiency. Later, Zhu et al proposed a cooperative PDP (CPDP) scheme exploiting the homomorphic verifiable response and hash index hierarchy to achieve batch auditing in multi-clouds scenarios. Further, Yang et al. presented another public auditing scheme for both multi-clouds and multi-users without introducing any trusted organizer.

Essentially, the batch auditing for multi-clouds is a task of the CSP that organizes the auditing information from different cloud servers. However, in terms of the batch verification performed by the TPA, the difficult point is how to effectively handle multiple audit requests from different users . A practical solution for this problem is to first aggregate the different data block tags produced by various users and then verify them as a whole which is also adopted in this work to achieve batch auditing.

To achieve dynamic data auditing, Erway extended the original PDP model by introducing a rank-based authenticated skip list, and presented a dynamic provable data possession (DPDP) scheme. Their foremost contribution is to demonstrate a general pattern for dynamic data auditing, i.e., incorporating dynamic data structures with verification algorithms. Later, Wang et al[11]. presented another classic public auditing scheme for dynamic auditing using Merkle Hash Tree (MHT), which simultaneously supports privacy-preserving and batch verification. However, both the above schemes would incur heavy computational costs of the TPA and large communication overhead during the updating and verification processes . Thus, Zhu et al. proposed another public auditing scheme (IHT-PA) based on Index Hash Table (IHT). Compared with the former ones, this scheme organizes the data properties for auditing using the IHT, and stores them in the TPA instead of the CSP. Consequently, it can reduce the computational costs and communication overhead. Nevertheless, its updating operations (particularly, the insertion and deletion ones) are inefficient, because they would induce the adjustment of average N/2 elements in the IHT, where N denotes the number of all blocks, due to the sequence structure of the IHT. Moreover, the operations would inevitably change the sequence numbers of some blocks, and finally cause the recalculations of their tags, which would induce more extra computational costs of the CSP and unnecessary communication overhead. Therefore, in this paper, we are motivated to design a new data structure, DHT, to achieve more efficient data updating and auditing.

3. OUR APPROACH

In this work, we concentrate on the design of an effective public auditing scheme based on the DHT illustrated in Fig. 1, which involves the following three entities: User, who stores a great quantity of data files in the cloud, can be an individual or a organization; Cloud Service Provider (CSP), who manages and coordinates a number of cloud servers to offer scalable and on-demand outsourcing data services for users; and Third Party Auditor (TPA), who can verify the reliability of the cloud storage services (CSS) credibly and dependably on behalf of the users upon request. Users can be relieved of the burden of storage and computation while enjoying the storage and maintenance service by outsourcing their data into the CSP. However, due to the loss of local possession of the data, they are keen to ensure the correctness and integrity of their data periodically. To obtain a convincing answer as well as alleviate the users burden potentially induced by the frequent verification, the TPA is involved to check the integrity of the users data stored in the cloud. However, in the whole verification process, the TPA is not expected to be able to learn the actual content of the user's data for privacy protection.

We assume the TPA is credible but curious. In other words, the TPA can perform the audit reliably, but may be curious about the user's data. In addition, the CSP is considered to be dishonest. That is to say, the CSP may choose to hide the fact of some data being corrupted motivated by self-interest. Specially, the CSP may launch the following attacks to the TPA:

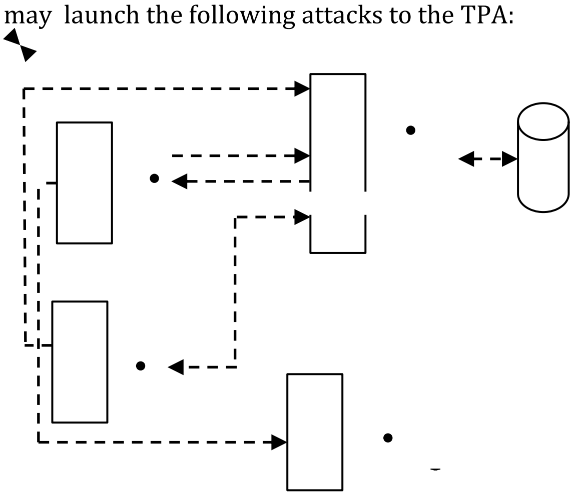

Fig 1 Architecture of DHT

- **Forge attack.** The CSP may forge the data blocks and/or their tags to deceive the verifier.
- **Replacing attack.** The CSP may want to pass the verification by replacing a required block and its tag, which have been corrupted, with another block and its corresponding tag.
- **Reply attack.** The CSP may attempt to pass the verification using the proof generated from the previous ones or other former information.

To enable secure and efficient public auditing for cloud storage, our scheme is designed to achieve the following objectives:

1. Public auditing: anyone (not only the users) is allowed to have the capability to verify the correctness and integrity of the users data stored in the cloud.
2. Storage correctness: the CSP, which does not correctly store user's data as required, cannot pass the verification.
3. Blockless verification: no data block needs to be retrieved by the TPA during the verification process.
4. Dynamic data auditing: dynamic data operations should be supported while the efficient public auditing is achieved.
5. Privacy preserving: the TPA cannot derive any actual content of users data from the received auditing information.
6. Batch auditing: the TPA can handle multiple auditing tasks from various users in a fast and cost-efficient manner.
7. Lightweight: the verification should be performed with the minimum communication and computation overhead.

3.1 Dynamic Hash Table (DHT)

The structure of the IHT is like a one-dimensional array, which contains index number, block number, version number and random value. The IHT-based scheme can also reduce the computational costs and communication overhead by storing the data properties for auditing using the IHT in the TPA instead of the CSP. Unfortunately, due to the sequence structure of the IHT, updating operations (particularly, insertion and deletion) on the IHT are inefficient, since they will lead to the adjustment of average $N/2$ elements, where N is the total number of all blocks. Moreover, during the insertion or deletion processes, the block numbers (Bi) of some blocks will be inevitably modified, which thereby will cause the regeneration of their corresponding block tags. That is obviously inefficient, and would cause more extra computational costs of users and unnecessary communication overhead. Therefore, we are motivated to design a new data structure, dynamic hash table (DHT), for better auditing efficiency. The DHT, like the IHT, is employed by the TPA to track the latest version information (VI) of the user' data for auditing. However, differing from the IHT, the DHT is a two-dimensional data structure, as illustrated in Fig. 2. In the DHT, there are two kinds of basic elements, namely, file elements and block elements. Each file element consists of the index number of the give n file, the File identifier (and a pointer indicating its first block element, which is stored in an array-like structure. Each file is organized using a linked list with the corresponding file element as the header node. Each block element is one node of the corresponding file list, including the current version of the given block its time stamp and a pointer indicating the next node. Accordingly, the operations on the DHT are divided into two categories: file operations and block operations, which both include search, insertion, deletion, and modification. Generally speaking, the block operations parallel those of the common linked list. To be specific, the search of a block is to locate the required element through visiting nodes from the first one in sequence; the insertion of a block after (before) an existing block is to first keep track of the given node and insert the new node after it; the deletion of a block is to first keep track of the required node and remove it from the current linked list. The search of a file is to locate the file element according to its index, and the other file operations would involve the manipulations on both file elements and block elements. Specifically, the insertion of a file involves inserting a file element into the file arrays and constructing a linked list that consists of corresponding block elements; the deletion of a file is to delete the linked list of the given file and its file element; the modification of a file is to update both the file element and related block elements. Taking the advantages of linked lists, the DHT significantly outperforms the IHT in the insertion and deletion of blocks. Further, the insertion and deletion of a block are unable to cause the change of other VI records in the DHT. That is to say, the block tags, which include the hash values of the VI records, would not be influenced. Therefore, compared with the scheme based on the IHT, our scheme can effectively reduce the computational costs of the CSP and communication overhead in the updating process. Moreover, although the cost more time than the IHT, it is too negligible to induce any material impact on the whole verification time. We will demonstrate the conclusion in the following text, and further prove that the verification time of our scheme is substantially smaller than that of the one based on the IHT.

4. CONCLUSIONS

In this paper, A novel public auditing scheme for secure cloud storage using dynamic hash table (DHT), which is a new two-dimensional data structure used to record the data property information for dynamic auditing is presented. Differing from the existing works, DHT scheme migrates the auditing metadata excerpt the block tags from the CSP to the TPA, and thereby significantly reduces the computational cost and communication overhead. Meanwhile, exploiting structural advantages of the DHT, this scheme can also achieve better performance than the state-of-the-art schemes in the updating phase. In addition to this, for privacy preservation, DHT scheme introduces a random masking provided by the TPA into the process of generating proof to blind the data information. Moreover, DHT scheme further exploits the aggregate BLS signature technique from bilinear maps to perform multiple auditing tasks simultaneously, of which the principle is to aggregate all the signatures by different users on various data blocks into a single short one and verify it for only one time to reduce the communication cost in the verification process. We formally prove the security of our scheme, and evaluate the auditing performance by detailed experiments and comparisons with the existing ones. The results demonstrate that DHT scheme can effectively achieve secure auditing in clouds, and induce significantly fewer costs of storage, communication and computation than the previous schemes.

In the future, we try to include different audit strategies that are to be designed for various types of cloud data because as, we would like to point out that no single method can achieve perfect audits for all types of cloud data, just as no standard has a universal validity.

ACKNOWLEDGMENT

Our sincere Thanks to Department of Computer Science and Engineering , Head of the Department , Our college Dr. Ambedkar Institute of Technology for supporting to carry out this project work.

REFERENCES

[1] H. Dewan and R. C. Hansdah. 'A Survey of Cloud Storage Facilities', Proc. 7th IEEE World Congress on Services, pp. 224-231, July 2011.
[2] C. Wang, Q. Wang, K. Ren, N. Cao and W. Lou.'Toward Secure and Dependable Storage Services in Cloud Computing', IEEE Trans. Service Computing, vol. 5, no. 2, pp. 220-232, 2012. [3] K. Ren, C. Wang and Q. Wang. "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.
[4] J. Ryoo, S. Rizvi, W. Aiken and J. Kissell. 'Cloud Security Auditing: Challenges and Emerging Approaches', IEEE Security & Privacy, vol. 12, no. 6, pp. 68-74, 2014.
[5] C. Wang, K. Ren, W. Lou and J. Li. "Toward Publicly Auditable Secure Cloud Data Storage Services", IEEE network, vol. 24, no. 4, pp. 19-24, 2010.
[6] 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. on Parallel and Distributed Systems, vol.
22, no. 5, pp. 847-859, 2011.
[7] F. Sebé, J. Domingo-Ferrer, A. Martínez-Ballesté, Y. Deswarte and J.-J. Quisquater, "Efficient Remote Data Possession Check-ing in Critical Information

Infrastructures," IEEE Trans. Knowledge Data Eng., vol. 20, no. 8, pp. 1034-1038, 2008.
[8] A. Juels and B.S. Kaliski Jr., "PoRs: Proofs of Retrievability for Large Files," Proc. ACM Conf. Computer and Communications Security (CCS '07), pp. 584-597, 2007.
[9] G. Ateniese, R.B. Johns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson and D. Song, ''Provable Data Possession at Untrusted Stores,'' Proc. 14th ACM Conf. on Comput. and Commun. Security (CCS), pp. 598-609, 2007.
[10] K. Yang and X. Jia. Data Storage Auditing Service in Cloud Computing: Challenges, Methods and Opportunities. World Wide Web, vol. 15, no. 4, pp. 409-428, 2012
[11] C. Wang, Q. Wang, K. Ren and W. Lou, ''Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,'' Proc. IEEE INFOCOM, pp. 1-9, 2010.
[12] C. Wang, S. M. Chow, Q. Wang, K. Ren and W. Lou, Privacy-Preserving Public Auditing for Secure Cloud Storage, IEEE Trans. on Computers, vol. 62, no. 2, pp. 362-375, 2013. [13] 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, 2012.
[14] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," IEEE Trans. on Parallel and Distributed Systems, vol. 24, no. 9, pp.1717-1726, 2013.
[15] C. C. Erway, A. Küpçü, C. Papamanthou and R. Tamassia."Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security, pp. 213-222, 2009.
[16] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu and S. S.Yau, "Dynamic Audit Services for Outsourced Storage in Clouds," IEEE Trans. on Services Computing, vol. 6, no. 2, pp. 227–238, 2013.