ijournals International Journal of Software & Hardware Research in Engineering
ISSN No: 2347-4890 Volume 2 Issue 1/ January 2014

A Dynamic Public Auditing Security Scheme To Preserve Privacy In Cloud Storage

Betzy K. Thomas[1] ; M. Newlin Rajkumar[2]
PG Scholar[1] ; Assistant Professor[2]
[1,2]Department of Computer Science and Engineering,
Anna University Regional Centre, Coimbatore, Tamil Nadu, India

ABSTRACT

The introduction of third party auditing in cloud computing reduces the burden of the users to check the integrity of the data stored in the cloud. In cloud computing, the outsourced data are not only accessed but also updated frequently by users for various application purposes. Thus supporting data dynamic operations in cloud including block level operations of modification, deletion and insertion is of vital importance. In this paper, we propose a new idea to support data dynamics by using the fractal tree representation along with privacy preserving public auditing protocol. We first identify the difficulties and security problems associated with the schemes used in prior works and then implement a new scheme for providing dynamic operations.

General Terms

Data Storage, privacy preserving, public auditing, data dynamics, data integrity

Keywords

Cloud Computing, Third Party Auditing, Fractal Tree, Merkle Hash Tree

1. INTRODUCTION

The cloud computing is a latest technology which provides various services through the internet. It is a subscription-based service where you can obtain networked storage space and computer resources. The cloud server allows the users to remotely store their data on the cloud. This helps the user to get rid of the difficulties of local data storage. Cloud data storage has many advantages over local data storage. Users can upload their data on the cloud and can access those data from anywhere at anytime. Also, users doesnot have to worry about the storage and maintenance of cloud data. Other advantages of cloud computing are improved computer performance, low cost, unlimited storage space, reliable data storage, easy availability of hardware and software resources etc..

Although there are several advantages for cloud computing, it also brings several security threats toward users outsourced data. Because cloud computing is a relatively new computing technology, there is a great deal of uncertainty about how security can be achieved. Many researchers had proposed different algorithms to resolve this security problem. Although outsourcing data to the cloud is economically attractive, it doesnot guarantee data integrity and confidentiality. As users no longer physically possess the storage of their data, traditional cryptographic methods cannot be adopted to provide security. While it is easy to check the data integrity by simply downloading the data from the cloud, downloading large amounts of data just for checking data integrity is a waste of communication bandwidth. Therefore, the researchers in cloud used a third party auditor(TPA) on behalf of the cloud client, to verify the integrity of the data stored in the cloud.

Another major concern among the researchers is to support dynamic data operations for cloud data storage applications. But unfortunately, dynamic data updates has received little attention so far. The different dynamic operations include data modification, insertion , deletion and appending.

Data Modification: One of the most frequently used dynamic operations in the outsourced data. It refers to the replacement of specified blocks with new ones.

Data Insertion: Insertion operation changes the logical structure of the file. It refers to inserting new blocks after some specified positions in the file.

Data Deletion: Deletion is the opposite of insertion. It refers to deleting a specified block.

Data Appending: Appending means adding a new block at the end of the outsourced data.

2. EXISTING SYSTEM

2.1 Privacy Preserving Public Auditing Protocol

Existing researches close to our work can be found in the areas of integrity verification and access control of outsourced data. Privacy preserving public auditing scheme allows data integrity to be verified without possession of the actual data file. Public auditability allows an external party in addition to user himself to verify the correctness of the data stored in the cloud. Most of these schemes uses a third party auditor(TPA) for this purpose. The TPA checks the integrity of the data on behalf of the users.

Privacy preserving public auditing protocol utilizes the technique of public-key based homomorphic linear authenticator(HLA) which enables the TPA to perform auditing without demanding the local copy of the data. Homomorphic authenticators are unforgeable metadata generated from individual data blocks. This technique drastically reduces the communication and computational overhead. By integrating the HLA with random masking technique, the TPA couldnot learn any information about the data content stored in the cloud during the auditing process. With the establishment of this technique in Cloud Computing, the TPA can concurrently handle multiple auditing tasks upon requests from different users.

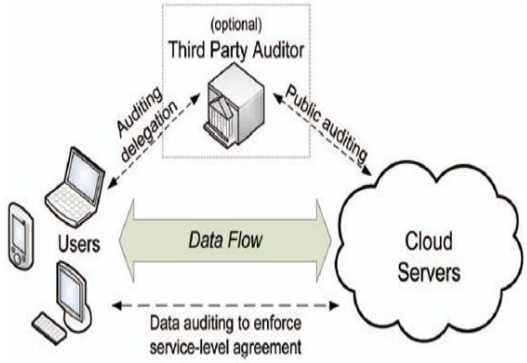

Fig 1: Existing System Architecture

2.2 Dynamic Data Operation

This scheme can handle dynamic data operations including data modification, insertion, deletion etc for cloud data storage. To achieve this, Merkle Hash Tree construction is used. Merkle tree is a tree where the value associated with a node is a one-way function of the values of the node's children. It is constructed as a binary tree where the leaves in the MHT are the hashes of authentic data values. Merkle tree finds a wide range of applications in cryptography and other security systems due to their simplicity and versatility.

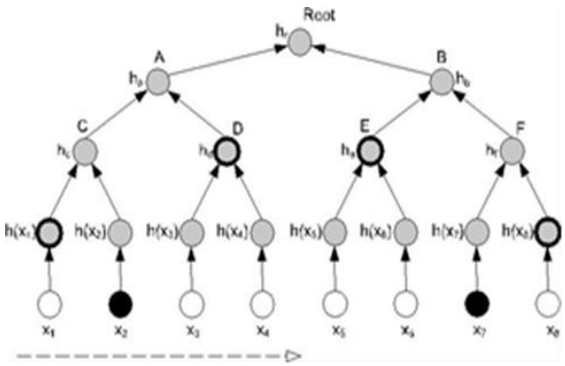

Fig 2: Merkle hash tree authentication of data elements

2.2.1 Data Modification:

Data modification refers to the replacement of specified blocks with new ones. Assume the outsourced data file F consists of a finite ordered set of blocks $m_1, m_2, \ldots, m_n$. Suppose the client wants to modify the i^{th} block m_i to m_i'. At first, the client generates $\sigma i' = (H(m_i').u_{mi'})\alpha$. Next, he generate an *update* request message (M,i,m_i',σ_i') and sends it to the CSS. When the CSS receives the request it runs *ExecUpdate(F,ϕ,update).* Specifically, the server 1) replaces m_i with m_i' and outputs F'; 2) replaces σi with $\sigma i'$; 3) replaces $H(m_i)$ with $H(m_i')$ in the MHT construction and generates the new root R' as shown in figure 3.

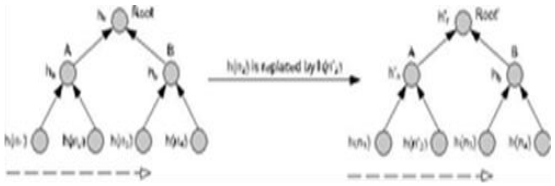

Fig 3: Example of MHT update under block modification operation

2.2.2 Data Insertion

Data insertion refers to inserting new blocks after some specified positions in the file. The protocol operations are similar to that of data modification operation.

2.2.3 Data Deletion

Data deletion is the opposite of data insertion. A single block deletion operation refers to deleting the specified block and moving all the latter blocks one block forward. When the server receives the *update* request for deleting block m_i, it delete m_i from its storage space, delete the leaf node $h(H(mi))$ in the MHT and generate the new root metadata R' as shown in figure 4.

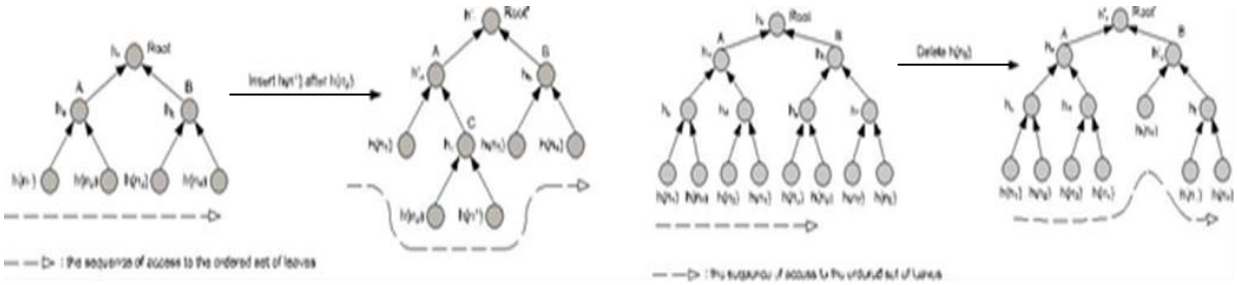

Fig 4: Example of MHT update under block insertion and block deletion operation

Although hash functions are very efficient, too many secret leaf values would need to be authenticated for each digital signature. For medium - size trees the computational cost is very high which can be reduced by reducing the space or time cost. This led to the concept that secure signature/authentication protocols are realizable, even if theoretic algorithms were not available. This construction roughly speeds up the signing operation inherent in Merkle's algorithm at a cost of requiring more space.

2.3 Disadvantages of MHT

- Limited number of signatures.
- Both time and space requirements of hash traversal technique is maximum.
- It is inefficient algorithm that generates a sequence of leaves along with their associated authentication paths.

3. PROPOSED SYSTEM

3.1 Objective

The objective is how to modify Merkle's scheduling algorithm to achieve various tradeoffs between storage and computation. The improvement is achieved by means of a careful choice of what nodes to compute, retain, and discard at each stage. The use of this technique is transparent to a verifier, who will not need to know how a set of outputs were generated, but only that they are correct. By reducing the time or space cost, we found that for medium - size trees the computational cost can be made sufficiently efficient for practical use.

3.2 System Overview

We use the algorithm of privacy-preserving public auditing system for data storage security in cloud computing .We extend our work to achieve efficient data dynamics using the Fractal tree representation. The data owner has a file F consisting of m blocks to be outsourced to a CSP where storage fees are specified according to the used storage space. For confidentiality, the owner can encrypt the data before sending to the cloud server. After outsourcing, the owner can interact with the CSP to perform block-level operations on the file. These operations include: modify, insert, append and delete specific blocks.

We have proposed a cloud based storage scheme which supports outsourcing of dynamic data where the owner is capable of not only archiving and accessing data stored by the CSP but also updating and scaling on the remote servers. The proposed scheme enables the authorised users that they are receiving the most latest version of the outsourced data.

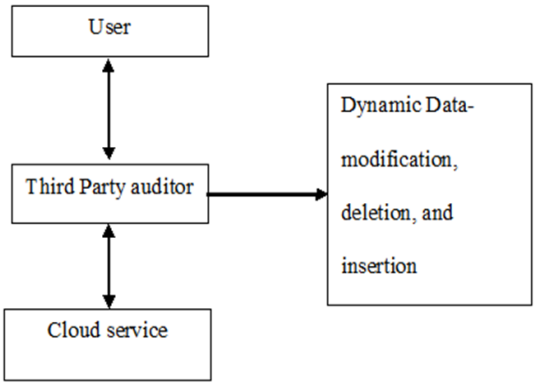

Fig 5: Proposed System

3.3 Fractal Tree Construction

The main idea of the fractal traversal is, to split up the merkle tree in subtrees and to save and compute these subtrees, instead of single nodes. Each subtree t has the same height h. Each root of one subtree is signed with a signature of the superior subtree. The signature generation phase consists of two phases: output phase and update phase. In the output phase, the signature and the authentication path is outputted. In the update phase, the subtrees with the next authentication node $Exist_i$ are set and the sub-

trees Desire$_i$ are computed.We use a slightly modified treehash algorithm to generate the subtree Desire$_i$ which is given below.

1. Set leaf =0.
2. Output: Authentication path for leaf with index leaf.
3. Next Subtree: For each i for which Existi is no longer needed, i.e. i∈{1,2,...,L−1} such that leaf = 2hi −1 (mod 2hi) :
   - Remove pebbles in Exist$_i$.
   - Rename tree i Desire as tree Exist$_i$.
   - Create new, empty tree i Desire (if leaf +2hi <2H).
4. Grow Subtrees: For each i∈{1,2,...,L−1}: Grow tree i Desire by applying two units to modified TREEHASH (unless i Desire is completed) starting from leaf with index 2ih.
5. Increment leaf and loop back to step 2 (while leaf <2H −1).

4. RELATED WORKS

Dynamic data have attracted attentions in the recent literature on efficiently providing the integrity guarantee of remotely stored data. Ateniese et al. [11] is the first to propose a partially dynamic version of the PDP scheme, using only symmetric key cryptography but with a bounded number of audits. In [4], Wang et al. implemented a similar support for partially dynamic data storage in a distributed scenario with additional feature of data error localization. In a subsequent work, Wang et al. [3] proposed to combine BLS-based HLA with MHT to support fully data dynamics. Erway et al. [5] developed a skip list based scheme to enable provable data possession with full dynamics support. However, verification in both the protocols requires a linear combination of sampled blocks as the input, like the designs in [9], [12], and thus does not support privacy-preserving public auditing.

5. CONCLUSION

In this paper, we proposed a framework that provides public auditability and data dynamics simultaneously for remote data integrity checking in cloud computing. We utilize the privacy preserving public auditing protocol for providing storage security in the cloud. This protocol uses the homomorphic linear authenticators and random masking technique to achieve data integrity without actually accessing the data stored in the cloud. Inorder to achieve efficient data dynamics, we use fractal tree representation for block tag authentication. Performance analysis shows that this technique is highly efficient and secure.

6. ACKNOWLEDGMENTS

We thank our Supervisor for all his valuable comments and active support.

7. REFERENCES

[1]. C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Storage Security in Cloud Computing,” IEEE Trans on Computers, vol 62, no. 2, Feb 2013

[2]. C. Wang, Q. Wang, K. Ren, and W. Lou, “Towards Secure and Dependable Storage Services in Cloud Computing,” IEEE Trans. Service Computing, vol. 5, no. 2, 220-232, Apr.-June 2012.

[3]. 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 and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[4]. C. Wang, K. Ren, W. Lou, and J. Li, “Towards Publicly Auditable Secure Cloud Data Storage Services,” IEEE Network Magazine, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[5]. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, “Dynamic Provable Data Possession,” Proc. ACM Conf. Computer and Comm. Security (CCS ’09), pp. 213-222, 2009.

[6]. P. Mell and T. Grance, “Draft NIST Working Definition of Cloud Computing,” http://csrc.nist.gov/groups/SNS/cloudcomputing/index.html, June 2009.

[7]. M.A. Shah, R. Swaminathan, and M. Baker, “Privacy-Preserving Audit and Extraction of Digital Contents,” Cryptology ePrint Archive, Report 2008/186, 2008.

[8]. A. Juels and J. Burton, S. Kaliski, “PORs: Proofs of Retrievability for Large Files,” Proc. ACM Conf. Computer and Comm. Security (CCS ’07), pp. 584-597, Oct. 2007.

[9]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores,” Proc. 14th ACM Conf. Computer and Comm. Security (CCS ’07), pp. 598-609, 2007

[10]. Cloud Security Alliance, “Security Guidance for Critical Areas of Focus in Cloud Computing,” http://www.cloudsecurityalliance.org, 2009.

[11]. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” Proc. Int’l Conf. Security and Privacy in Comm. Networks (SecureComm ’08), pp. 1-10, 2008.

[12]. H. Shacham and B. Waters, “Compact Proofs of Retrievability,”Proc. Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (Asiacrypt), vol. 5350, pp. 90-107, Dec. 2008.

[13]. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the Clouds: A Berkeley

View of Cloud Computing," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009.

[14]. G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage from Homomorphic Identification Protocols," Proc. 15th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT), pp. 319-333, 2009.

[15]. Cloud Security Alliance, "Top Threats to Cloud Computing,"http://www.cloudsecurityalliance.org, 2010.

[16]. M. Arrington, "Gmail Disaster: Reports of Mass EmailDeletions," http://www.techcrunch.com/2006/ 12/28/ gmail-disasterreportsof-mass-email deletions /, 2006.

[17]. R. Curtmola, O. Khan, and R. Burns, "Robust Remote Data Checking," Proc. Fourth ACM Int'l Workshop Storage Security and Survivability (Storage SS '08), pp. 63-68, 2008.