International Journal of Recent Trends in Engineering & Research
ISSN (ONLINE) : 2455 – 1457
IMPACT FACTOR : 4.101

# Public Auditing for a Secure Cloud Storage using Dynamic Hash Table

**S. N. Saliya [1], A. H. Hingmire [2]**
[1,2] *Asst.professor, Computer Department, RSCOE, Pune*

**Abstract**: Cloud storage is a service model used to store digital data in which the physical storage contain multiple servers sometimes multiple locations, and the physical environment is typically hold and supervised by a hosting company typically called as cloud service providers(CSP) . These CSP are responsible for keeping the data available and accessible, and provide security to physical environment, ensures that they are running properly. People and organizations usually buy or lease storage capacity from the CSP to store user, organization, or application data. While storing data in cloud, security is one of the major challenges faced by cloud service provides. When users are storing data in cloud storage they will not completely trust cloud service providers hence it is very important to develop an efficient auditing scheme to acquire data owners trust and confidence related to data security. In this paper we designed a public auditing scheme for securing data residing on cloud using dynamic hash table(DHT).DHT is a two dimensional data structures sited at a third party auditor(TPA) to keep track of data property information for dynamic auditing. This proposed scheme transfer authorized data from cloud service provider to third party auditor with reduced computational cost and communication overhead. The advantages of using dynamic hash table are excellent updating efficiency. For privacy preservation the proposed system use public key with random masking generated by the third party auditor. The system efficiently achieves secure auditing for cloud storage with reduced communication overhead and computational cost.

**Keywords**: cloud storage, cloud security, dynamic auditing, dynamic hash table

## I. INTRODUCTION

Cloud computing is an emerging technology in which large pool of systems are connected in private or public networks. This computing paradigm will provide dynamically scalable infrastructure for application, data and file storage. By using this new technology, the cost of computation, hosting of application, content storage and delivery is reduced drastically. The cloud computing is based on the idea of “reusability of IT capabilities”. The difference between cloud computing and traditional computing like grid computing, distributed computing, utility computing, and autonomic computing is to expand horizons across organizational boundaries.

Cloud storage [1] is a term used for managed data storage through hosted network called as cloud service providers. Various types of cloud storage systems are available to support both personal and business purpose. Basic cloud storage facility will allow users to upload individual files or folders from personal computers to the cloud server. So users can access their data from cloud server to other device, can enable remote access to the files, also create backup if their originals are lost. However as a new technology cloud storage faces lot of security challenges. The main concern is whether cloud service providers can meet expectation of customer related to data security [2], [3]. Suppose users are placing their data in cloud data center; the client will lose their direct control over their data. The Cloud Service Provider (CSPs) is responsible for ensuring the security of stored data of clients by firewalls and virtualization. But when data is secured using these mechanism it will not provide 100 percentage securities due to the vulnerabilities in the network. So, cloud computing paradigm needs secure auditing methods [4] to store and manage data for preserving data confidentiality and privacy so it can gain customers trust.

DOI: 10.23883/IJRTER.2017.3529.MMYCV

In order to overcome security issues generally two types of auditing schemes are using: Private auditing and public auditing [5], [12]. Private auditing mechanism allows user to challenge the cloud service provider to check the integrity of his data. Here verification operation directly happens between CSP and user with reduced cost. But it cannot provide credible audit result. Public auditing scheme allows data owner i.e. user to delegate a component who is expertise in it called as Third Party Auditor(TPA) and most promisable and trustworthy next to Provider. This method allows TPA to check the integrity of data without knowing content. By public auditing scheme users burden will be reduced since auditing is doing by third party auditor.

*Table 1. Performance comparison for data integrity verification schemes*

| Table | | | | | | |
|---|---|---|---|---|---|---|
| Performance Comparison of Auditing Schemes of Cloud Storage | | | | | | |
| Schemes | Communication Overhead | Computation Costs | | | | Detection Probability |
| | | Verification | | Updating | | |
| | | Auditor | CSP | User/TPA | CSP | |
| PoRs [8] | O(1) | O(1) | O(1) | __ | __ | $1-(1-t)^c$ |
| PDP[9] | O(1) | O(1) | O(1) | __ | __ | $1-(1-t)^c$ |
| CPDP[13] | O(c+s) | O(c+s) | O(c+s) | __ | __ | $1-(1-t)^{c.s}$ |
| DAP[14] | O(c) | O(c) | O(c.s) | O(n) | O(w) | $1-(1-t)^{c.s}$ |
| DPDP (skip list) [15] | cO(logn) | cO(logn) | cO(logn) | wO(logn) | wO(logn) | $1-(1-t)^c$ |
| DPDP (MHT) [6] | cO(logn) | cO(logn) | cO(logn) | wO(logn) | wO(logn) | $1-(1-t)^c$ |
| IHT-PA[16] | O(c+s) | O(c+s) | O(c+s) | O(n) | O(w) | $1-(1-t)^{c.s}$ |
| DHT-PA | O(c) | O(c) | O(c)(O(c.s)) | O(w) | O(w) | $1-(1-t)^c$( 1-(1-t)c.s) |

Some of the problem needs to be addressed even while using public auditing scheme.

- Privacy Preserving: Data Privacy Protection is an important factor of cloud storage. In public auditing scheme the question is how to provide user's privacy as auditing is done by third party auditor. Only encrypting data before uploading into cloud storage is not a complete solution, it cannot completely avoid data leakage. So in order to reduce data leakage there should be some privacy preserving mechanism in addition to data encryption [7], [11].
- Dynamic Auditing: The cloud users frequently required to update the data dynamically on cloud according to application requirement. So it is very important for cloud storage auditing to support dynamic auditing [6].
- Batch auditing: Batch auditing mechanism is used for multi user environment. Third Party Auditor should supports auditing for multiple users simultaneously to enhance efficiency and scalability of public auditing.

In order to address these problem this paper presents a new public auditing mechanism that uses dynamic hash table. Advantages of using dynamic hash table is dynamic auditing can be done efficiently as well as it supports batch auditing and enable privacy preserving mechanism. Dynamic hash table transfer only authorized data from cloud service provider to third party auditor, its computational cost and communication overhead is much less compared to other data structure.

For preserving privacy this system uses homomorphic encryption based on public key cryptography and masking generated by third party auditor. By using Homomorphic encryption it can convert data from original form into cipher text and that can be handled with as if it were still in its original form. So the contribution of this paper is as follows.

- Presenting a novel public auditing scheme based on homomorphic authenticator which can support dynamic auditing, batch auditing as well as privacy preservation.
- This method introduces a new data structure dynamic hash table to enable dynamic auditing for cloud storage efficiently and with reduced cost.
- For privacy Preservation it uses homomorphic encryption based on public key cryptography and masking generated by third party auditor.

## II. RELATED WORK

One of the previous paper related to cloud storage security is “proof of retrievability (POR)” by Juels et al.[8]. They presented a theoretical framework for the design of PORs. It guaranteed to improve the previously proposed POR It supports a fully Byzantine adversarial model, carrying only the restriction fundamental to all PORs that the adversary’s error rate be bounded when the client seeks to extract F. Techniques support efficient protocols across the full possible range of , up to non-negligibly close to 1. But the problem with this invention is it is a private auditing scheme and does not support third party auditing [9].

To avoid data leakage on cloud storage, Wang et al. [5] presented an auditing protocol for preserving privacy. This system combines random masking with homomorphic cryptography, So Third party auditor is able to audit the cloud data storage without using local copy of data efficiently with no additional burden also the third party auditing scheme will not introduce no new vulnerabilities towards user data privacy.

Zhu et al. [13] presented a cooperative PDP scheme focus on homomorphic verifiable response and hash indexing method to achieve batch auditing. After that Yang et al. [14] introduced another public auditing mechanism for multi-clouds and multi-user without introducing any trusted organizer, but the main focus is how it will handle multiple audit requests from different users. The solution for this problem is aggregate the different data block tags generated by different users and then verify them as a whole. The same techniques used in this paper for batch processing.

Erway et al. [15] extended the PDP Model by adding a new rank based authentication mechanism with the help of skip list and proposed a dynamic provable data possession scheme. Further, Wang et al. [6] presented another public audit scheme using Merkle hash tree, which supports both batch auditing and privacy preserving. But disadvantages of these two schemes are its communication overhead and computational cost is quite large for updating and verification process.

Later, Zhu et al. [16] presented another public auditing scheme based on indexed hash table. This scheme organizes data properties for auditing in index hash table and storing them in TPA instead of CSP. This method is it will reduce communication overhead and computational cost. Nevertheless updating operation in index hash table is inefficient because due to sequence structure they would induce the adjustment of average N/2 elements in the IHT. Therefore in this paper we introduce dynamic hash table based public auditing for efficient updating and auditing. The performance comparison of various auditing scheme is given in the Table 2.

*Table 2. Performance comparison of various auditing schemes*

| Table | | | | |
|---|---|---|---|---|
| Function Comparison of Auditing Schemes of Cloud Storage | | | | |
| Schemes | Public Auditing | Privacy Protection | Dynamic Auditing | Batch Auditing |
| PoRs [8] | No | __ | No | No |
| PDP[9] | Yes | No | No | No |
| CPDP[13] | Yes | Yes | Yes | Yes |
| DAP[14] | Yes | Yes | Yes | Yes |
| DPDP (skip list) [15] | No | __ | Yes | No |
| DPDP (MHT) [6] | Yes | Yes | Yes | Yes |
| IHT-PA[16] | Yes | Yes | Yes | NA |
| DHT-PA | Yes | Yes | Yes | Yes |

## III. PROPOSED SCHEME

This scheme is designed to enable secure, trusted and efficient public auditing for cloud storage. This scheme mainly contains three modules, TPA, CSP and Dynamic Hash Table. Architecture of system is shown in Fig. 1. The objectives of this scheme is

- Public auditing: Enable the capability of verifying the correctness and integrity of data stored on cloud
- Storage correctness: The CSP that does not store data correctly on cloud will not pass the verification test
- Dynamic auditing: Dynamic up gradation of stored data should be supported
- Batch Auditing: TPA should be able to handle multiple auditing tasks.
- Privacy preservation: TPA not allowed to learn actual content of users data stored in cloud storage

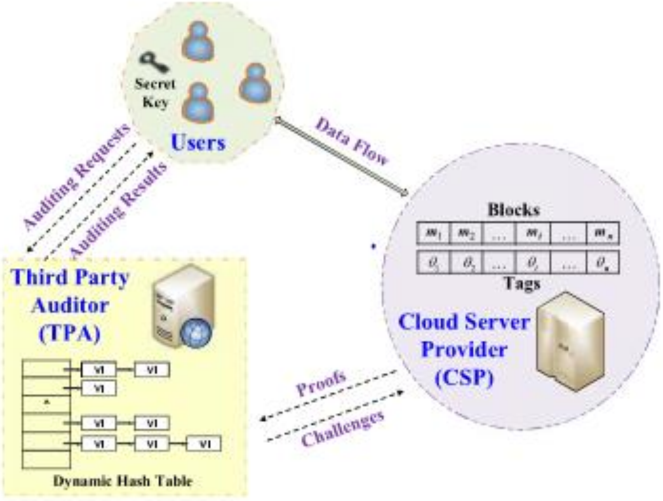
Secret Key
Users
Auditing Requests
Auditing Results
Data Flow
Blocks
$m_1$ $m_2$ ... $m_i$ ... $m_n$
$\theta_1$ $\theta_2$ ... $\theta_i$ ... $\theta_n$
Tags
Third Party Auditor (TPA)
Cloud Server Provider (CSP)
vi
Dynamic Hash Table
Proofs
Challenges

*Fig. 1. System architecture*

### Dynamic Hash Table

It is an authentic and efficient data structure used to achieve dynamic auditing for cloud storage. The PDP based on skew list [15] and MHT-based public auditing scheme [6] is in efficient due to their large computation cost and communication overhead. Later work public auditing based on index hash table structure is shown in Table 3. Considerably reduced computation cost and communication overhead but the problem with this scheme is due to its sequence structure updation operation is inefficient.

*Table 3. Structure of Index Hash Table*

| Table | | | | |
|---|---|---|---|---|
| Index Hash Table | | | | |
| No. | Bi | Vi | Ri | |
| 0 | 0 | 0 | 0 | ← Used to Head |
| 1 | 1 | 2 | $r'_1$ | ←□Update |
| 2 | 2 | 1 | $r_2$ | |
| 3 | 4 | 1 | $r_3$ | ←□Delete |
| 4 | 4 | 1 | $r_5$ | |
| 5 | 5 | 2 | $r'_5$ | ←□Insert |
| . | . | . | . | |
| . | . | . | . | |
| . | . | . | . | |
| N | N | 1 | $r_n$ | |
| n+1 | n+1 | 1 | $r_{n+1}$ | ←□Append |

DHT is like IHT employed by TPA to track latest updated information of users data for auditing. It is a two dimensional structure, illustrated in Fig. 2. In DHT there are two types of elements File elements and block elements.

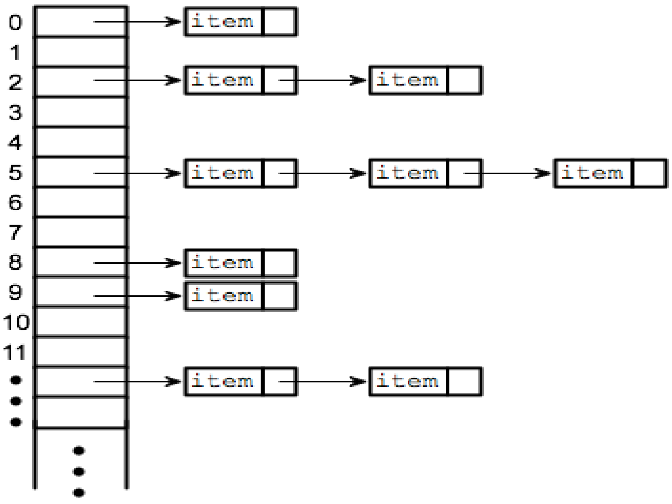

*Fig.2. Structure of dynamic hash table*

File elements consists of index number(NO$_i$) of the given file(F$_i$), file identifier(ID$_i$) and a pointer to first block element which is stored in an array liked structure. Each file is structured using a linked list with corresponding file element as a header node. Each block element is one node of the respective file list including current version of the block. Like same way the operation of DHT is divided into File operation and Block operation, which consist of searching, inserting, deletion and updation. Dynamic auditing scheme involves two phases setup phase and verification phase. Workflow of the dynamic verification is as shown in Fig. 3.

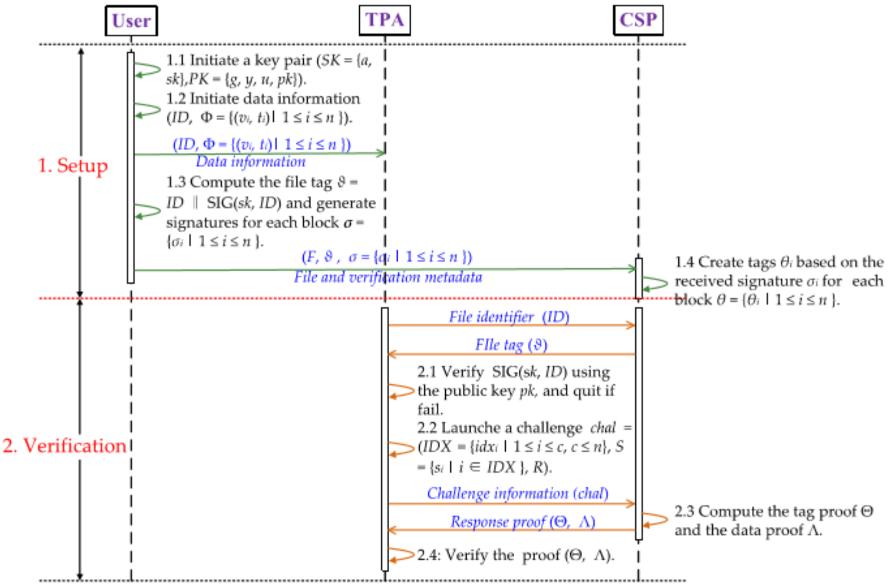
User
TPA
CSP
1. Setup
1.1 Initiate a key pair (SK = {a, sk},PK = {g, y, u, pk}).
1.2 Initiate data information (ID, Φ = {(vᵢ, tᵢ)| 1 ≤ i ≤ n }).
(ID, Φ = {(vᵢ, tᵢ)| 1 ≤ i ≤ n })
Data information
1.3 Compute the file tag ϑ = ID ∥ SIG(sk, ID) and generate signatures for each block σ = {σᵢ | 1 ≤ i ≤ n }.
(F, ϑ , σ = {σᵢ | 1 ≤ i ≤ n })
File and verification metadata
1.4 Create tags θᵢ based on the received signature σᵢ for each block θ = {θᵢ | 1 ≤ i ≤ n }.
2. Verification
File identifier (ID)
FIle tag (ϑ)
2.1 Verify SIG(sk, ID) using the public key pk, and quit if fail.
2.2 Launche a challenge chal = (IDX = {idxᵢ | 1 ≤ i ≤ c, c ≤ n}, S = {sᵢ | i ∈ IDX }, R).
Challenge information (chal)
Response proof (Θ, Λ)
2.3 Compute the tag proof Θ and the data proof Λ.
2.4: Verify the proof (Θ, Λ).

***Fig. 3. Workfolw of dynamic verification and privacy preserving***

The setup phase contains four steps such as key initiation, data information initiation, signature generation, tag generation.

- Key Initiation: The user generates a key pair (SK= (a,sk), PK= (g,y,u,pk), where (sk,pk) is a random key pair of user signature, a€ $Z_p$ is a random number, g and u are the random elements of a multiplicative cyclic group.
- Data Information Initiation: The user sends information of user including data ID, set of all block, version number, time stamp of the block $m_i$ to the TPA. After receiving the data information TPA will add this information to dynamic hash table.
- Signature Generation: For each block $m_i$ the CSP generate one signature with the public key u.
- Tag Generation: For each block $m_i$ the CSP generate one tag based on the signature received from the previous step.

The verification phase contains file identifier check, challenge, proof generation, proof check.

- Identifier Check: The TPA first retrieves the file tag and then verifies the signature by using user’s public key pk. If they pass the test it will retrieve file identifier then go to challenge phase otherwise TPA will quite the verification process.
- Challenge: The TPA initiate a challenge by sending the challenge information to CSP.
- Proof Generation: After receiving the challenge by TPA cloud service provider generates data storage correctness proof contains tag proof as well as data proof and will send the proofs to TPA.
- Proof Check: TPA needs to do the verification operation now so it will compute hash value for challenged data blocks. This information will be checked against the information received from previous step.

## IV. PERFORMANCE EVALUATION

In this section performance evaluation of dynamic hash table based public auditing is done using various points and compares it with other schemes.

- Communication Cost: The communication cost of various schemes is given in the Table 4. From which we can understand that communication cost of first two schemes are more (logn times) compared to other scheme during updation and verification process.

***Table 4: Communication cost comparison of various scheme***

| Schemes | Communication Cost | |
|---|---|---|
| | Verification | Updating |
| DPDP(MHT) | cO(logn) | O(logn) |
| DPDP(Skip List) | cO(logn) | O(logn) |
| DAP | O(c) | O(1) |
| Index Hash Table based Public Auditing | O(c) | O(1) |
| Dynamic Hash Table Based Public Auditing | O(c) | O(1) |

- Storage Cost: DAP [14], IHT-PA [16] and Dynamic hash table based public auditing transfer the metadata which has been stored in data structure to TPA in order to reduce the communication overhead but other will not do this. In that public auditing using dynamic hash table have less storage cost due to their structural advantage.
- Computational Cost: For Analysing computational cost compares it with index hash table based public auditing and dynamic auditing protocol because other scheme involve more metadata than this also communication cost also large. Users processing time to handle same data is considerably less in DHT based scheme. Search time of DHT based scheme decreased by increase in number of blocks.

## V. CONCLUSIONS

When users are storing data on cloud storage the security is the major concern. In public key auditing, auditing is done by TPA user will not trust CSP for integrity and security of data. SO this scheme proposed to ensures privacy preservation, dynamic auditing and batch auditing. In this scheme the data auditing is done by TPA without knowing actual content of data. CSP will try to meet user's legal expectation and their trust. It uses a two dimensional structure to record the data and transfer auditing metadata to TPA from CSP so advantages of using this scheme is support of dynamic auditing, privacy preservation and batch auditing with reduced computational cost and communication overhead.

## REFERENCES

I. C.Wang , Q. Wang, K. Ren, N.Cao "Toward secure and dependable storage services in cloud computing", IEEE Tras. Serv. Comput., vol. 5, no. 2, pp.220-232, April-June 2012

II. Hrishikesh Dewan ; R. C. Hansdah "A Survey of Cloud Storage Facilities", 2011, IEEE World Congress on Services, 4-9 July 2011, ISSN: 2378-3818

III. Kui Ren, Cong Wang, Qian Wang "Security Challenges for the Public Cloud" IEEE Internet Computing, Volume: 16, Issue: 1, Jan.-Feb. 2012, ISSN: 1089-7801

IV. Jungwoo Ryoo, Syed Rizvi, William Aiken, "Cloud Security Auditing: Challenges and Emerging Approaches" IEEE Security & Privacy, Volume: 12, Issue: 6, Nov.-Dec. 2014 ,

V. Kui Ren, Cong Wang, Wenjing Lou "Toward publicly auditable secure cloud data storage services" IEEE Network ( Volume: 24, Issue: 4, July-August 2010 ), ISSN: 0890-8044

VI. Qian Wang , Cong Wang , Kui Ren "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing", IEEE Transactions on Parallel and Distributed Systems ( Volume: 22, Issue: 5, May 2011 ), ISSN: 1045-9219.

VII. Francesc Sebé, Josep Domingo-Ferrer, Antoni Martinez-Balleste, "Efficient Remote Data Possession Checking in Critical Information Infrastructures", IEEE Transactions on Knowledge and Data Engineering ( Volume: 20, Issue: 8, Aug. 2008 ), ISSN: 1041-4347.

VIII. Ari Juels and Burton S. Kaliski Jr., “PORs: Proofs of Retrievability for Large Files”, 14th ACM conference on Computer and communications security, ISBN: 978-1-59593-703-2

IX. Giuseppe Ateniese, , R. Burns, R. Curtmola, J. Herring, L. Kissner, and D.Song, “Provable data possession at untrusted stores”, 14th ACM conference on Computer and communications security, Pages 598-609 , ISBN: 978-1-59593-703-2.

X. K. Yang and X. Jia. “Data storage auditing service in cloud computing: Challenges, methods and opertunities”, World Wide Web, Vol 15, No 14 pp. 409-428, 2012

XI. Cong Wang, Qian Wang, Kui Ren, “Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing “, INFOCOM, 2010 Proceedings IEEE, 14-19 March 2010, DOI: 10.1109/INFCOM.2010.5462173.

XII. Cong Wang, Qian Wang, Kui Ren, S.M. Chow, “Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing”, INFOCOM, 2010 Proceedings IEEE, vol. 62, no. 2, pp.362-375, 2013.

XIII. Yan Zhu, Hongxin Hu, Gail-Joon Ahn ” Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage”, IEEE Transactions on Parallel and Distributed Systems ( Volume: 23, Issue: 12, Dec. 2012 ), ISSN: 1045-9219

XIV. Kan Yang, Xiaohua Jia, “An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing”, IEEE Transactions on Parallel and Distributed Systems ( Volume: 24, Issue: 9, Sept. 2013 ), ISSN: 1045-9219.

XV. C. C. Erway, A. Kupcu, C. Papamanthau and R Tamassia ‘Dynamic provable data possession’ in Proc. 16th ACM conf. Commuter Communiction Security 2009 PP213-222

XVI. Yan Zhu, Gail-Joon Ahn, Hongxin Hu, “Dynamic Audit Services for Outsourced Storages in Clouds”, IEEE Transactions on Services Computing ( Volume: 6, Issue: 2, April-June 2013 ), ISSN: 1939-1374.