A Cost Effective Dynamic Auditing Scheme for Outsourced Data Storage in Cloud Environment

Esther Daniel
Dept.of CSE
Karunya University
Coimbatore
estherdaniell@gmail.com

N.A Vasanthi
Dept. of Information Technology
Dr.N.G.P Institute of Technology
Coimbatore
vasanti.au@gmail.com

***Abstract*— Cloud data storage outsourcing is beneficial as it eliminates the burden of initial infrastructure setup and maintenance costs. Nonetheless data security and integrity remains challenging due to lack of control and physical procession over the data by the data owners. In this paper we present an enhanced dynamic auditing method for data integrity verification. The proposed scheme combines Elliptic Curve Cryptography Diffe Hellman(ECCDH) signatures with a Improved Distributed Hash Table (IDHT) as data structure that efficiently supports dynamic auditing process. Our proposed data structure will incur minimum computation, communication and storage cost. The proposed scheme when compared with other state-of-the-art techniques proves to be cost effective in terms of computational and communication costs on the server and the auditor.**

Keywords*— *Cloud Environment, Storage Security, Data Integrity, Dynamic Auditing.

I. INTRODUCTION

In recent years, cloud computing is becoming more and more popular in both academic and commercial world. Cloud computing offers resources such as computing power, storage, online applications and networking infrastructures shared as services over the Internet[1]. Cloud storage is an important service of cloud computing, which allows data owners (DO) to outsource their data from their local computing systems to the remote cloud. It provides low-cost, scalable, location-independent platform to store and manage users' data. Therefore naturally more and more data owners start to use cloud storage services.

Cloud storage service offers a lot of advantages and possibilities and it takes the storage management and maintenance from the hands of data owners, but it also introduces new challenges and security threats from the external sources. Security threat can also arise from inside the cloud. There are several motivations for CSP to behave unfaithfully towards the cloud users. Therefore, the owners need to be certain that the data are correctly stored in the cloud. The owner should be notified promptly if any problems regarding to owner's data occur. Moreover in cloud storage system, none of the sides, namely data owner and cloud storage provider, is guaranteed to provide the auditing report in and unbiased manner. Therefore third-party auditing is an intuitive selection for cloud storage auditing. The efficient and dynamic auditing method utilizes a dynamic hash table data structure that allows the auditor to perform dynamic data operations effectively with minimum computational and communicational overhead on the TPA and cloud server.

This paper is organized in to following sections. Section 2 reviews few of the existing state of art techniques and section 3 briefs the system model of the proposed auditing scheme. Section 4 and 5 presents the protocol of the proposed auditing scheme and analyses its performance. Finally section 6 gives the conclusion and future work of this paper.

II. RELATED WORK

In latest years, there has been an exhaustive research on integrity auditing of cloud data storage. Ateniese et al. [3] were the first to propose a novel approach for public auditing in their provable data possession model. They used RSA based homomorphic linear authenticator scheme for auditing the outsourced data. Their scheme is lightweight with low overhead on the server but this scheme can handle only the static storage files which will not be suitable for dynamic operations. Tan, Shuang, et al.[4] provided a dynamic version of audit supporting batch auditing through which only required data blocks can be verified based on the request from the data owner. However only limited number of queries can be handled and does not support public auditing. Erway et al.[5] introduced DPDP based on authenticated skip lists and RSA tree. DPDP provides dynamic and improved detection. Due to the various processes running in the cloud, this method produces huge computation cost at the server side. Juels et.al.[6] introduced a POR protocol in which the verifier stores only a single key and requires access to a small portion of the file for its auditing. This scheme takes lot of preprocessing and encoding of the data file before the storage in CSP which causes high computational overhead. Shacham et.al [7] provided an improved POR scheme based on the BLS signatures which reduces the communication cost but does not support dynamic operations. The dynamic updating of data is a critical feature of data auditing protocol. This allows the DO

978-1-5090-5778-8/17/$31.00©2017 IEEE

to update, modify, insert, delete and append their outsourced data on the fly without downloading it. The Merkle Hash Tree (MHT) let the input data to be divided into separate block with same block size[8]. Hash code for each data block will be generated and the entire hashed block should be arranged in the same order as original file so that updating in the data will not result in error. Concatenate each pair of hashed block and hash the resulting data. In the next level of tree, repeat the same process again until all blocks are hashed. Move up to the next level if no root has been found and repeat the entire process until there is a single hash digest for the entire file. If any data block is updated, Instead of retrieving the whole data block and hash value, only the hash value of the specified block can be obtained based on the index number. The data block which is changed will be hashed and added to the tree directly. After replacing the hash value in the block, the tree can be calculated again in both DO and TPA. If a particular block is deleted in a file then the whole tree needs to be updated due to the change of position of data blocks. The data block which is next to the deleted node will be moved to left, hashed and concatenated with the previous node. The process will be repeated until there is a single hash digest for the entire file. However, they would incur heavy computational costs and increased communication overhead during the verification and updating processes at the TPA. Zhu et al.[11] introduces a data structure called index hash table (IHT) that allows the blocks to generate the hash value of each bock in the verification process. Its one-dimensional array structure contains index number, block number, version number and random value. This scheme stores the data abstract information in the TPA thus reducing the computational costs and communication overhead. But the single array sequence structure makes it inefficient to execute operations like delete and insert as these operations leads to more number of blocks to be adapted causing renewal of the corresponding tags thus mounting the communication and computations costs.A simple MAC based third-party auditing scheme [9] was more efficient than many other public key based scheme. Even though this scheme supports dynamic auditing, batch auditing and runs with low communication cost, this method has heavy storage overhead and computation cost at the TPA and CSP due to the generation of this MAC based value for its auditing purpose. Grounded on these research findings a requirement for efficient and dynamic auditing scheme which schedules and audits the dynamic data on the remote servers with reduced communication, computation and storage cost is evident. So we propose an effective auditing protocol based on Elliptic Curve Cryptography Diffie Hellman (ECCDH)and Improved Distributed Hash Table(IDHT) cryptographic primitives that guarantee reduced communication and computation cost with increased assurance level of data file remotely stored.

III. System Architecture

Our auditing scheme as shown in Fig.1 consists of the entities namely Data Owners (DO), Cloud Storage Providers (CSP), Third Party Auditor (TPA) and Approved User. Initially data owners stores data in the storage servers provided by the cloud service providers. After storing the data to Cloud, user no longer posses the physical control of the data, thus he needs to be able to verify, that his data are correctly stored and maintained on the server. Due to possibly large costs in terms of resources and expertise and for security reasons, data owner might ask TPA to perform the data auditing task, while keeping the data private from TPA itself. We assume that TPA is trustworthy and independent that has no motivation to conspire with either CSP or DO. It performs honestly during the whole auditing process, but it's inquisitive about the received data.

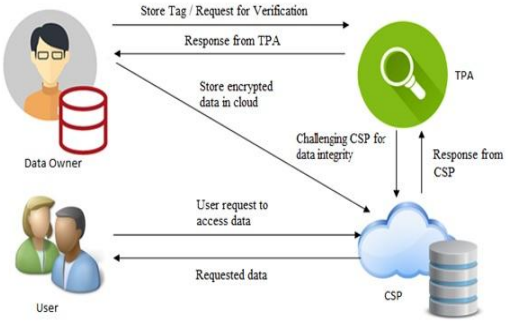

Fig 1: System Architecture

IV. Proposed Algorithm

We proposed techniques and algorithms of our dynamic auditing scheme with IDHT [10] by using the characteristics of the bilinear property and Elliptic Curve Cryptography Diffie Hellman (ECCDH) cryptographic primitives.

First the client get the data in the file format and using random generation of number the client generates the key to encrypt the data. After the key is generated the client encrypts the complete data using the key which was generated and using symmetric key encryption techniques. After the complete data is encrypted then the fragmentation process is done where the complete file is fragmented into Data Blocks of 4 KB. And then the tag is generated for each data block and this tag is used for auditing purpose. After the tag is generated the client sends the data to the server and the server should store the data in the cloud. And the tag generated will be send to the auditor to verify the data tag later with the server.

Now the server receives the data from the client and stores it in the cloud into the database in sectors in the table in the encrypted form. When the auditor asks for any proof from the server it generates the tag at that moment and send it to the auditor to verify.

The auditor receives the tag which was generated by the client and stores it in the table in the auditor database.

The Auditors Scheduler ensures continuous auditing at periodic intervals to ensure the integrity of the data. The auditor send the challenge to the server at several intervals of time and the server generates the proof and send it back to the auditor. The challenge is gathered from multiple clients and then processed in batches by the auditor to the server. Then the auditor checks for the integrity from the proof it received from the server and if the integrity is maintained then the auditor inform to the client that the data is secure.

If there is any misbehavior detected or corrupted block is identified the auditor will inform the Storage Client about it and also reports to the maintenance and monitoring manager for preventive measures to be enforced and SLA's to be updated to further prevent such inconsistencies and also to ensure prevention mechanism enabling the users ease of maintenance of data with high security in a remote storage cloud environment.

Algorithm 1: Auditing Protocol

Keygen(λ) –Select p, a, b and G belonging to Z_p whose order n is λ usually of 160 bit length. The user generate the private key Prk by selecting a random number $n_A < G$ .Compute the equivalent public tag key PuK $=n_{A*}G$ then compute the shared secret key Ssk $=n_{A*}$PuK

Taggen() –Randomly pick $\{r_i\}^n_{i=1}$ for signing the block. The encrypted block of File F and the corresponding tags are computed and stored in the cloud by the following equations.

$$CT(Bi)=(E(Bi,PrK)||r_i)^{ssk} \quad (1)$$
$$T_i = (Hash\ (CT(Bi))\ ,\ Bi,ts)^{ri} \quad (2)$$
$$T=\Pi^n_{i=1}\ T_i$$

Chal() – Select the sample set of blocks to be challenged 'cb' and send the challenge request to the CSP along with the timestamp 'ts' and challenge id 'c_{id}'

$$Chal()=\{cb_i||ts||c_{id}\ ,r_i\}\ ,\ \text{where i=1 to n}$$

Response() – The challenge query is extracted for various values and the response id generated by the following equation

$$Res()= \sum_{i=1}^{m}(Hash(CT(cbi)||ts||cid))^{ptk}$$

Verify() – The TPA first computes the hash values for all the challenged data blocks using the merkle hash tree and computes the root challenge hash and then verifies the response from the server matches the hash of the blocks challenged.

The distributed hash table works with the put/get interface that provides a lookup service of a specified key value pair from the widely distributed data at the storage servers. The improved distributed hash table (IDHT) constructs a data structure based on the hash values of the file with the container that holds the subsequent blocks of the file. The pointers in each block points to the next block stored in the bucket. The containers are considered as a bucket which stores the object id and the data in it.

Algorithm 2:IDHT Dynamic Auditing

Input: F_{id},$B_{id,}$,CT,V_i,ts
Output: Updated table at CSP/TPA
Procedure

```
If (request == Insert)
    {
     Select the position p_i in bucket
     Store the encrypted file block (CT)
     Display the pointer to point to the
    block
     Update the bucket and hash table address.
     }

If (request ==Append)
    {
    Select the block to which the append
    operation has to be done.
    Create block space adjacent to the previous
    block where the new data block is to be
    inserted
    Insert the encrypted data at the new address
    space and set the return pointer.
    Update the bucket and the hash table
    address
    }

If (request == Update)
    {
    Select the block to be modified
    Update the modified block CT replacing the
    previous one.
    Reset the hash table address and the bucket.
    }
If (request == delete){
    Delete the block
    Free the block space in memory
    Update the free hole in the bucket table
    }
```

End

V. Performance Analysis

For the implementation of the proposed protocol Java has been used. The experiments were carried out in the development environment of OpenStack called Dev Stack. For demonstration purposes of the implementation of our proposed scheme, we implemented multiple client-server architecture. For each entity, we run separate virtual machine. CSP's server and TPA's server are listening on agreed ports and waiting for Auditor's and DO's client connections respectively. Communication and data transfer were achieved via sockets. We have set the block sixe as 4 KB each block and the security parameter λ=160 bit. The

To authenticate the effectiveness of our system the auditing protocol was compared with MHT [8] and IHT [11] and the following results were analyzed. Due to continuous scheduled auditing scheme the probability of detection of misbehavior with 75%,85% and 95% assurance level fig 2 requires few blocks for challenge when compared with the protocol without the a TPA. Fig 3 gives the computation time taken for the updating operations like append, insert, delete and update modify is effective and moderate as the IDHT scheme of auditing is implemented

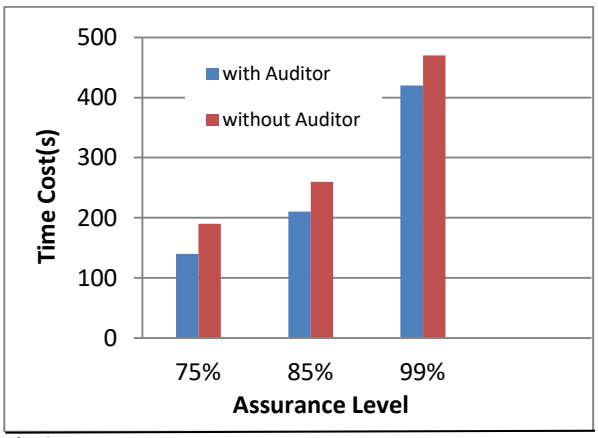

Fig 2: Assurance level with and without auditor

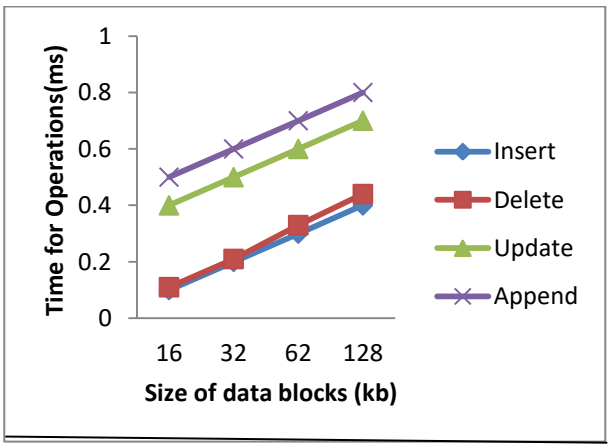

Fig 3: Computation Costs for Updating operations

The table 1 gives the communication costs of the MHT, IHT and IDHT schemes. This reveals that the verification and updating mechanisms shows a constant time complexity.

Table 1: Comparison of Communication Costs

Scheme	Verification	Updating
MHT[8]	$cO(logn)$	$O(logn)$
IHT[11]	$O(c)$	$O(1)$
IDHT	$O(c)$	$O(1)$

The table 2 reveals that the storage costs for the IDHT is reduced when compared with the MHT and IHT.

Table 2: Comparison of Storage Costs

Scheme	Verification	Updating
MHT[8]	$l\cdot(2_{logn+1}-1+n)$	-
IHT[11]	$n\cdot l$	$v\cdot n$
IDHT	$n\cdot l$	$v\cdot(n+1)$

Where n is the number of the data blocks; *l* is the length of every block;v is the size of the hash table

Fig 2 assures that there is need for auditor to achieve 99% of unaltered audit results.Fig.3 reveals that the computation cost for the dynamic auditing operations like insert, delete, append and modify is effective and moderate as the IDHT scheme of auditing is implemented

VI. Conclusion

The proposed protocol ensures privacy and integrity of the data by symmetric encryption and ECCDH. The Improved Distributed Hash Table enables efficient and fast storage and retrieval of data with improved level of integrity assurance. The scheme also minimizes the communication and computation cost of the TPA. Further this scheme can be enhanced to identify the de-duplication of the outsourced data and thus reduces the storage cost of this auditing protocol.

References

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica: A view of cloud computing,ACM ,pp. 50-58 (2010)

[2] Kher, V., Kim, Y.:Securing distributed storage: challenges, techniques, and systems. In:Proc of the ACM workshop on Storage security and survivability. pp. 9-25 (2005)

[3] Ateniese, Giuseppe, et al. "Remote data checking using provable data possession." ACM Transactions on Information and System Security (TISSEC) 14.1 (2011): 12.

[4] Tan, Shuang, et al. "An efficient method for checking the integrity of data in the cloud." China Communications 11.9 (2014): 68-81.

[5] Erway, C. Chris, et al. "Dynamic provable data possession." ACM Transactions on Information and System Security (TISSEC) 17.4 (2015): 15.

[6] Juels, Ari, and Burton S. Kaliski Jr. "PORs: Proofs of retrievability for large files." Proceedings of the 14th ACM conference on Computer and communications security. Acm, 2007

[7] Juels, Ari, and Burton S. Kaliski Jr. "PORs: Proofs of retrievability for large files." Proceedings of the 14th ACM conference on Computer and communications security. Acm, 2007.

[8] 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.

[9] Wan, Changsheng, et al. "Efficient privacy preserving third party auditing for ambient intelligence systems." Journal of Ambient Intelligence and Humanized Computing 7.1 (2016): 21-27.

[10] https://en.wikipedia.org/wiki/Distributed_hash_table

[11] Zhu, Yan, et al. "Dynamic audit services for outsourced storages in clouds." IEEE Transactions on Services Computing 6.2 (2013): 227-238.

[12] Esther Daniel,N.A Vasanthi, "An Efficient Continuous Auditing Methodology for Outsourced Data Storage in Cloud Computing", Proceedings of Advances in Intelligent Cyber Security and Computational Models,Springer,pp.461-468(2015)