ISSN:2231-0711
www.ijcset.net

Data Integrity Maintenance in Dynamic Cloud Data Storage

[1]N.Pandeeswari , [2]P.Ganesh Kumar,[3]R.Sivakami

Dept. of IT, PSNA College of Engineering and Technology
Dindigul, TamilNadu, India

[1]ponds_it@yahoo.co.in
[2]msy2ganeshkumar@gmail.com
[3]rsivakami@psnacet.edu.in

Abstract: **Cloud computing has been often viewed as a next generation of IT enterprise. The cloud makes the users to store the data remotely in cloud storage and reduces the burden of storage and maintenance. And it also moves the application software and databases to the centralized large storage, where the management of data and services are not fully trustable. This brings many new security challenges on storing the data remotely without having any backup .While the prior works, often lacks with the efficient integrity verification and dynamic data operations. In this work, it is proposed to verify the integrity of data and allowing dynamic data operations such as block insertion, modification and deletion. It is proposed to use homomorphic tokens based on RSA-Signatures to verify the integrity of remotely stored data and in this the Merkle –hash tree is used for authenticating the data. Further this scheme is extended to do batch auditing i.e., doing multiple auditing tasks simultaneously. The performance and security analysis shows that the proposed scheme is highly efficient and provably secure.**

Key Words: ***data integrity, dynamic data operations, homomorphic authentication.***

I. INTRODUCTION

Cloud computing means "internet computing". The internet is viewed as clouds, hence the term cloud computing for computation done through internet. With cloud computing the users can store and access their data through internet with out worrying about the local maintenance and management of data. Over time many internet based companies have come to realize that only a small amount of data storage is being used. Amazon's simple storage solution (S3) and Amazon Elastic Compute cloud are the best known facilities. The increasing network bandwidth and reliable yet flexible network connections makes it possible even possible that clients can now subscribe high quality services from data and software that reside solely on remote data centers. Although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are facing both internal and external threats for data integrity. Users no longer possess their data locally, so traditional cryptographic primitives for the purpose of data security protection can not be adopted directly. Now there was a problem that how to efficiently verify the correctness of the outsourced cloud data without the local copy of data files becomes a big challenge for data storage security in cloud computing. Downloading the data for verification is an expensive process. It makes the process much slower. Besides it is often insufficient to detect the data corruption when accessing the data, as it might be too late for recover the data loss or damage. To fully ensure the data security and save the cloud user's computation resources, it is important to enable external auditing [2] for cloud storage that who has expertise and capabilities that the users do not, to audit the outsourced data when needed. The security is achieved with signing the data. The security is achieved by signing the data blocks. The sign is performed with RSA-signatures.ThirdParty Auditor [2] (TPA) performs auditing task for each user. This increases the auditing time and computation overhead. Earlier works performs auditing for static data .This paper supports dynamic data operations on cloud data such as block insertion block modification and block deletion.

II. PROBLEM STATEMENT

A. System model

A representative architecture for cloud storage is illustrated in fig 1.Three different entities are present here.

Clients: Client is the entity who stores and accesses the data on cloud storage; it can be either any individual or any other organization.

Cloud servers: Server is the entity for providing service to the clients through Cloud service Provider who is having storage space and computation resources.

Third party auditor: an entity, who has expertise and knowledge of doing verification upon request by the client

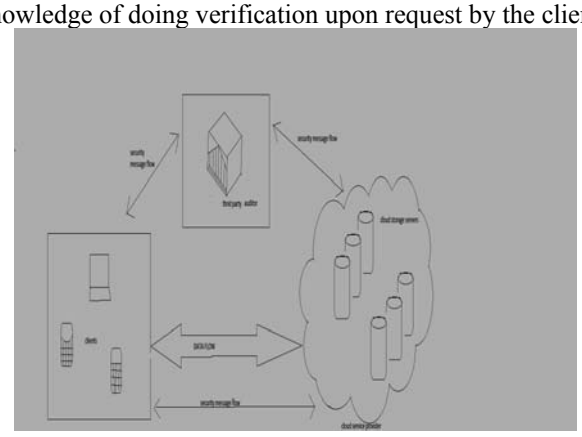

Fig .1. Representative model for cloud storage

In this cloud paradigm, the client stores the large amount of data files in cloud without having any local copies. As the clients are not having their data locally, it is of critical

importance to verify the integrity of data files and to ensure the storage correctness of the data being stored. To do this the clients has to frequently verify the data files. Without the proper resources and time for the clients, this task of verifying the storage correctness delegates to the Third party auditor, who is not the client. The clients stores and access the data in cloud storage through CSP. The clients perform block level operations such as block insertion, block modification and block deletion.

B. Design goals

1. External auditablity for verification of integrity of data and storage correctness: Allow anyone to do verification who is not the client, and who is capability of doing verification.
2. Support dynamic data operations: To enable the dynamic data operations such as insertion, modification and deletion. Design efficiently the integration of auditablity and dynamic data support.

III. EXISTING SYSTEM

In PDP (Provable data possession) model for ensuring possession of files on untrusted servers uses RSA[3] based homomorphic tags; Public auditability is supported but this doesn't support dynamic nature of data operations. Dynamic version of PDP [5] also doesn't support dynamic data operations. In POR model the public auditability is not supported. In the improved version of POR (Proof of retrievability) [4] also only for static data files. Dynamic version of PDP implemented using rank based authenticated skip list (DPDP) [7] is the first one that supports dynamic version of cloud storage. Still the efficiency is not clear. The technique to ensure dynamic cloud storage and public auditability using BLS [1] signatures supports dynamic nature of cloud storage and verifying the integrity of data. Since BLS signatures does not support variable sized blocks and also does not consider about data privacy.

IV. PROPOSED SCHEME

With considering the design goals in mind the proposed scheme supports dynamic data operations and allows a Third party auditor to verify the storage of data. It is important to support dynamic storage while maintaining the storage correctness. The proposed system creates efficient and dynamic data operations for storage assurance and integrity maintenance of the stored data. For verification RSA-signatures are used.

The proposed work can be summarized as:

1. To enable external auditing system for data integrity verification in cloud computing.
2. To support dynamic data storage especially blocks insertion which is not supported in existing systems. Further extend auditing to do batch auditing, in which multiple auditing task will be done simultaneously.
3. Compare the performance of our scaleable and efficient auditing task with other tasks.

A. Notation and Preliminaries

1. Merkle hash tree: A merkle hash tree is an authentication structure [8] to store a set of elements safely and efficiently without any damage and alteration. The construction of MHT uses binary tree implementation; where the leaves in MHT are the hash values of authenticate data values. The verifier with authentic h_r requests for $\{x_2,x_7\}$and requires the authentication of received blocks. The prover provides the verifier with the auxiliary authentication information $\Omega_2=<x_1,h_d>$ and $\Omega=<h(x_8),h_e>$.

The verifier verifies x_2, x_7 by first computing $h_c=(h(x_1) \| h(x_2))$ and $h_f=(h(x_7) \| h(x_8))$; $h_a=(h_c\|h_d)$,$h_b=(h_e\|h_f)$ and $hr=(h_a,h_b)$ .the verifier checks the hr is same as the authentic one.

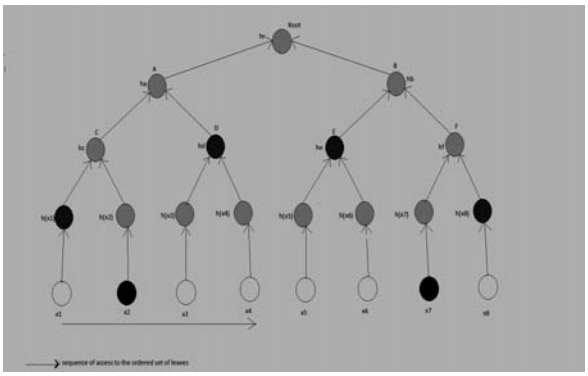

Fig .2. Sample merkle hash tree.

B. Basic Solutions

The outsourced data file F consists of a finite ordered set of blocks $m_1,m_2…m_n$.One way to ensure data integrity is to precompute MACs for entire data file. During auditing process each time data owner has to show the secret key to the cloud server. The number of verification is limited by the number of secret keys. Once, if the keys are exhausted, the data owner has to recomputed MAC; which is impractical. Moreover public auditablity is not supported by doing verification with private keys.

Another solution is to use signatures to obtain public auditablity.The data owner precomputes the signature of each block mi (I € [1, n]) and sends both F and signatures to the cloud server for storage. This algorithm could result in a large communication overhead and greatly affects system efficiency.

C.The Proposed Solution

To effectively support public auditability without having to retrieve the data blocks they use homomorphic authenticator technique [3], [4]. This design uses RSA-based homomorphic authenticator [3].The main idea behind the scheme is the file f is divided into n blocks, where mi€Z_p $m_1,m_2…m_n$ and P is a large prime and uses the cryptographic hash function h.

1. RSA-Key generation: 1. randomly select two large distinct prime's p, q of the same bit length.
2. Compute n=pq and Φ (n) = (p-1) (q-1).
3. Select an arbitrary integer e, 1<e<Φ (n), such that gcd (e, Φ (n)) =1.
4. Compute d, 1<d<Φ (n), such that ed=1 mod Φ (n).
5. Public key is (n,e) ;Private key is d.

2. Signature generation: Given f= $(m_1,m_2…m_n)$ data blocks . Then the client computes signature σ_i for each block m_i (i=1, 2... n)

1. Compute M=H (m); where H is the cryptographic hash function such as SHA-1.
2. Use private key d to compute $\sigma i = M^d \bmod n$.

Denote the set of signatures by Φ= {σi}, 1<=i<=n.

The client then generates a root R based on the construction of Merkle Hash tree.The client signs the root by

sig (H(R)) $=S=M_R^d \bmod n$.

3. Signature verification process: 1. Obtain the authentic public key (n, e).

2.Compute M=H(m).

3.Compute sig(H(R)); $M'=S^e \bmod n$= sig' H(R)

4. Verify M=M'.

The server computes μ and σ where both the data blocks and the corresponding signature blocks are aggregated into a single block respectively.

Prover will also provide the verifier with a small amount of auxiliary information.

$\{\Omega_i\}$ $S_1<i<S_c$.The prover responds the verifier with proof p= $\{\mu, \sigma \{H(m_i), \Omega_i\}\}$ $S_1<=i<=S_c$,sig (H(R)).

Upon receiving the responses; the verifier generates root R using $\{H(m_i), \Omega_i\}$ $S_1<=i<=S_c$.

V. Dynamic Data Operation with Integrity Assurance

The proposed scheme can efficiently handle fully dynamic data operations including data modification, insertion and deletion for cloud data storage. The file F and sign Φ have already been generated and properly stored at server.

A. Data Insertion

Data insertion [1] is to insert new m^* blocks at some specified positions in the data file F.The client generates the corresponding signature σ^*.Then generate an update request message update= (I,i, m^*, σ^*) and sends to the server. After receiving the message, the server runs Exec Update (F, Φ, Update).

1. The server stores m^* and adds a leaf h (H (m^*)) and outputs F'.
2. Generate new root R'.
3. Adds the sign σ^* into the signature set and outputs Φ'.

B. Data Modification

Data modification [1] is the most frequently used operations in cloud data storage. Client wants to modify the i^{th} block m_i to m_i' .The clients generate the signature $\sigma i'$ for the new modified block. He constructs an update request message update= (M, I, mi', σ_i') and sends to the server.

The server runs Exec Update (F, Φ, Update)

The server

1. replaces the block m_i with m_i' and outputs F'.
2. replaces σ_i with σ_i'.
3. replaces H (m_i) with H (m_i').

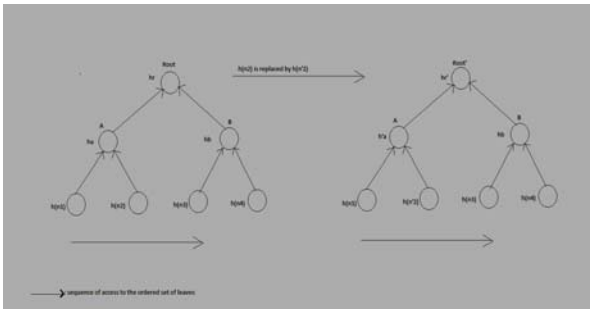

Fig.3. Example for Data modification in MHT

C.Data deletion

Data deletion [1] process is just the reverse process of data insertion.Any request to delete the data m_i deletes the data by deleting the leaf h(m_i).

Then create new Root h_r'. This has been explained in the fig.4.

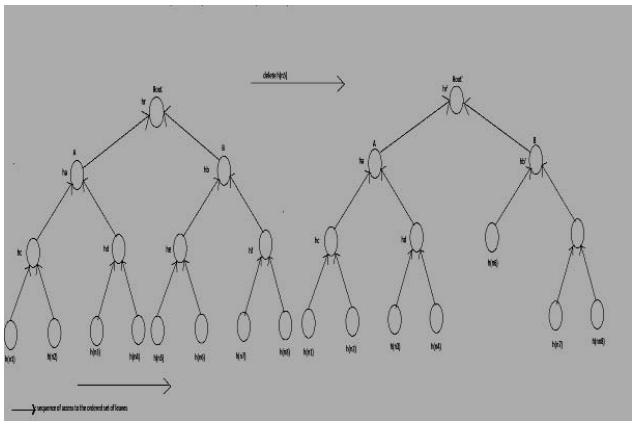

Fig.4. Example for data deletion in MHT

VI. **BATCH AUDITING FOR MULTI-CLIENT DATA**

As cloud servers may concurrently handle multiple verification sessions from different clients, given k signs on k distinct data files from k clients [1].

The Proposed scheme based on RSA allows the creation of signatures aggregation on various blocks. It supports the aggregation of multiple signs by distinct signers on distinct messages into a single short signature and reduces communication cost and providing efficient verification for authenticity of all messages.

VII. Security and Performance Evaluation

A. Security Analysis

A Signature scheme is said to be secure against chosen message attacks. The adversary cannot forge and he may have low control over the contents of message in the existential forgery. The problem of inverting the RSA function (i.e., finding e^{th} roots modulo n) is intractable. The RSA signature on hash tree provides more security.

B.Performance Evaluation

The experiment is conducted [1] using C on a system with an Intel core-2 processor running at 2.4 GHz, 2048 MB RAM. Algorithm SHA-1 is implemented for Hash tree construction. All results represent the mean of 10 trials.

TABLE 1A
PERFORMANCE COMPARISON RATE P=99%. UNDER 1GB FILE. THE BLOCK SIZE FOR RSA BASED INSTANTIATION AND DPDP IS 4KB.

	RSA based instantiation	**BLS based instantiation**	**DPDP model**
Tolerance rate ρ	99%	99%	99%
Server computation time (ms)	6.45	13.81	14.13
Verifier computation time (ms)	806.01	779.10	782.56
Communication cost (KB)	239	223	280

TABLE 1B
PERFORMANCE COMPARISON RATE P=97 % .THE BLOCK SIZE CHOSEN FOR RSA BASED INSTANTIATION AND DPDP IS 16KB.

	RSA based instantiation	BLS based instantiation
Tolerance rate ρ	97%	97%
Server computation time (ms)	2.11	4.55
Verifier computation time (ms)	284.17	210.47
Communication cost (KB)	80	76

Table1A. and Table 1B. lists the performance metric for 1 GB file under various erasure code rate ρ while maintaining high detection probability (99%) and (97%) respectively of file corruption.

Due to smaller block size (i.e., 20 bytes) the RSA based scheme slower when compared to BLS based scheme. The communication cost is very much low at the verifier side when compared to BLS scheme. The communication cost of DPDP is the largest amount among the three in practice. Moreover, RSA based scheme can be used for variable sized blocks. The communication cost grows almost linearly as the block size increases; which is mainly caused by the increasing in size of the verification block.

Batch auditing [1] not only enables simultaneous verification from multiple clients but also reduces the communication cost on TPA side. Given total clients are k in the system, the batch auditing helps reduce the number of expensive pairing operations 2k; as k+1 as required in individual auditing. The following same experiment setting as p=99% and 97% batch auditing saves TPA's computation overhead for about 5% and 14% respectively. To maintain detection probability of 99% the random sample size is 460 whereas to maintain detection probability of 97%; the random sample size is 152.

VIII. CONCLUSION

The proposed scheme investigates the problem of data integrity verification in cloud storage. To achieve the assurances of cloud data integrity and availability, the proposed scheme introduced an third party auditor for verification and the proposed scheme supports the dynamic data operations. To achieve efficient data dynamics; the merkle hash tree is used; RSA signature scheme with hash tree provides more security. Considering the time consumption, communication cost and variable block size; the proposed scheme is highly efficient and resilient to Byzantine failure. The performance evaluation shows that the proposed scheme is highly efficient and provably secure. Here it is shown that multiple tasks reduce the TPA computation time.

REFERENCES

[1].Q.Wang , C.Wang, K.Ren, W.Lou, Jin Li ," Enabling Public auditability and data dynamics for storage security in cloud computing " in Proc. Of ESORICS'09 published in IEEE. May 2011. pp.847 -859.

[2].Q.Wang, C.Wang, J.Li, K.Ren and W.Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing ", in Proc. Of ESORICS'09.Saint Malo, France: Springer – Verlag, 2009.pp-355-370

[3].G.Ateniese, R.Burns, R.Curtmola, J.Herring, l/Kissner, Z.Peterson, and D.Song,"provable data possession at untrusted stores." in Proc. Of CCS'07.New York, NY, USA ACM, 2007, pp. 598-609

[4].H.shacham and B.Waters, "Compact proofs of retrievability,"in.Proc.ofASIACRYPT'08.Melbourne,Australia: springer-Verlag, 2008,pp.90-107.

[5].G.Ateniese, R.D.Pietro, L.V.Mancini, and G.Tsudik, "Scalable and efficient provable data possession, "in proc.of SecureComm'08.Newyork, NY, USA ACM, 2008, pp.1-10

[6].C.Wang , Q.Wang, K.Ren ,and W.Lou, " Ensuring data storage security in cloud computing ,"in proc .of IWQoS'09, Charleston, South Carolina,USA,2009

[7].C.Erway, A.Kupcu, C.Papamanthou, and R.Tamassia, "Dynamic provable data possession," in Proc. Of CCS '09. Chicago, IL, USA ACM, 2009

[8].R.C.Merkle , "protocols for public key cryptosystems ,Proc .of IEEE symposium on security and Privacy 80, pp.122 -133,1980