ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st& 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

A Flexible Auditing Mechanism for Storages in Cloud Computing

Saranya C, Vinoth P

PG Scholar, Dept of Computer Science Engineering, Mepco Schlenk Engineering,

College Sivakasi, India.

Assistant Professor, Dept of Computer Science Engineering, Mepco Schlenk Engineering,

College Sivakasi, India.

Abstract-In recent years, Cloud Computing has achieved greater heights. Cloud storage is more preferable than normal local storage, as when we use global internet based storage, it is easy to use wherever we go. But these cloud infrastructures are prone to threats like Cloud Service Provider may behave unfaithful towards the users data and also users may faulty perform improper operations that leads to loss of data. The solution for the above problem is that a Flexible Auditing Mechanism was proposed which allows a Third Party Auditor (TPA) to verify the integrity of data stored in the cloud .It uses the fragment structure and secure tag generation method for periodic sampling audit process .It also supports audit for dynamic operations. This audit service can provide public auditability without downloading raw data and protect privacy of the data. Therefore, through this audit system the dynamic data operations and timely anomaly detection can be supported at low storage and communication cost.

Keywords-Storage Auditing, data integrity, cloud storage systems.

I.INTRODUCTION

Cloud computing is an upcoming technology where the computing resources are shared across internet for usage, that is we can pay as we use. There is no need for any local storage .The storage in clouds has become a popular way of storing and maintaining the data. The integrity of stored data should be preserved. But cloud infrastructures are prone to threats. These threats can be internal threat, caused by virtual machine failure or external threat, caused by system holes. This damages the integrity of data. The next issue is that the cloud service provider would behave unfaithful towards the cloud data. They may discard the data that are rarely accessed or not accessed to save storage space and claim that the owner's data are correctly stored in the cloud. The other issue is that users may perform improper operations that is they may accidently delete the data and claim the cloud service provider is responsible for the data loss. So, it is necessary for the service provider to have a flexible auditing service to check the correctness of the data stored in the cloud.

Audit involves the tracking of all activities that includes data accesses, security violations, and attacks and so on. However, traditional cryptographic schemes lack to support integrity verification without the local copy of the data. If the audit services are designed to download the whole data which incurs storage and communication cost.
This situation should be managed so, the main objectives of the flexible auditing services are:

- *Efficient Auditability.*
  To have a third party auditor (TPA) to verify the correctness of the data without downloading the whole data.
- *Dynamic Auditability.*
  To support auditing for dynamic operations done by the user.
- *Periodic detection of errors and losses.*
  To perform a periodic sampling audit on the data stored in the cloud.
- *Forensic Evidences.*
  To provide forensic evidences for the anomalies those occur in the cloud.
- *Lightweight Audit Tasks.*

To incur the less cost computation audit tasks.

To achieve these objectives, a flexible auditing service was introduced to provide efficient integrity verification. This audit system supports dynamic data operations and timely anomaly detection. The techniques used are fragment structure, random sampling and index-hash table.

The rest of the paper is organized as follows: Section 2 describes the research background and related work. Section 3 describes the audit system architecture and the techniques used. Section 4 describes the definition and the algorithms. Section 5, we present the performance of the experimental results. Finally, we conclude the paper in Section 6.

II.BACKGROUND AND RELATED WORK

Traditional schemes based on the hash functions and signature methods can't work without downloading the whole data. That results in expensive communication especially for large-size files. As the work of auditing the cloud data are really expensive and complex. This auditing work of verifying the correctness of the data is moved from cloud service provider (CSP) to the third party auditor (TPA).TPA has the capability for periodically auditing the data.This service gives data assurance and acts as a digital forensics.

Many researchers proposed new notations for public auditability like the notions of proof of retrievability (POR)[2] and PDP[5]. Anyone can use these schemes to prove the correctness of the data and offer a publicly accessible remote interface to check large amount of data.

Some audit services solutions are there. Xie et al. [10] proposed a content comparability method but it is not suitable for irregular data. Wang et al. [11] proposed a privacy preserving property for outsourced data. In this scheme, a file is directly split into n blocks and a verification tag is generated for each block. Here, the size of the block is equal to the cryptosystem.For,160 bits which is 20 bytes. That is,1M bytes file is split into 50,000 blocks and generates 50,000 tags , this storage of tags is 1M bytes. So, it is inefficient to construct a system using this scheme.

To overcome this problem, a fragment structure technique is used to reduce the extra storage. Using this scheme, the data can be dynamically updated by the clients. The block operations like modification, deletion and insertion.

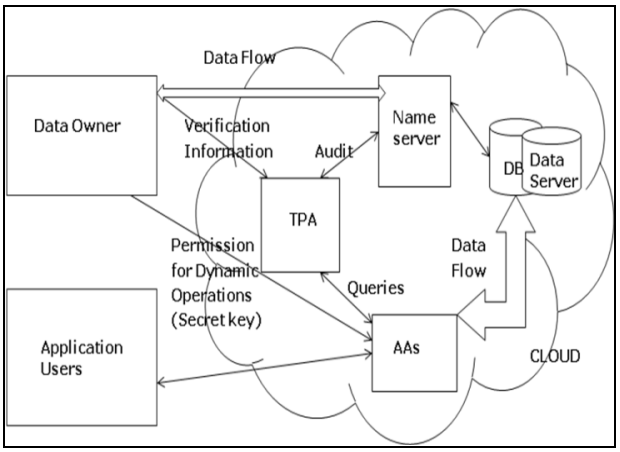


Fig. 1.Architecture of Cloud Audit Service.

In the existing schemes, security issues arise a lot. Hence, to develop a secure mechanism for dynamic audit services, a flexible scheme is required.

III.ARCHITECTURE AND TECHNIQUES

Audit system architecture for outsourced data in clouds is shown in Fig. 1.This architecture the audit architecture consists of five entities.

- Cloud Service Provider(CSP)
  CSP provides data storage service and has enough storage space and computation resources.
- Authenticated Applications(AA)
  AA has the right to access and manipulate the stored data
- Data Owner (DO)
  DO have a large amount of data to be stored in the cloud.
- Third Party Auditor(TPA)
  TPA has capabilities to manage or monitor the outsourced data under the delegation of DO.
- Application User's
  Cloud User's enjoy various cloud application services via the AA's.

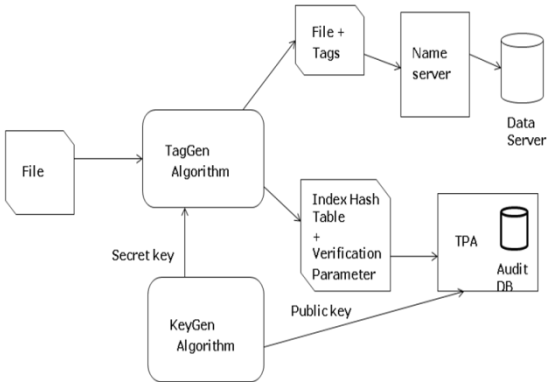
  TPA is reliable and independent through the following audit functions:
  - TPA should be able to make regular checks on the integrity and availability of the delegated data at appropriate intervals.
  - TPA should be able to organize, manage, and maintain the outsourced data instead of DO
  - TPA support dynamic data operations for AAs.
  - TPA should be able to take the evidences for disputes about the inconsistency of

data in terms of authentic records for all data operations.

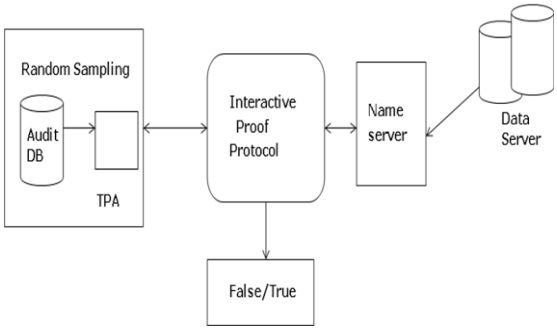
To achieve these functions, our audit services has 3 processes:

- *Tag Generation And User's Verification*: The client (DO) uses a secret key s_k to pre-process a file, which consists of a collection of n blocks, generates a set of public verification parameters (PVP's) and IHT that are stored in TPA, transmits the file and some verification tags to CSP, and may delete its local copy(see Fig.2a);
- *Periodic Sampling Audit:* By using an interactive proof protocol of retrievability, TPA issues a "random sampling" challenge to audit the integrity and availability of the outsourced data in terms of verification information (involving PVP and IHT) stored in TPA(see Fig. 2b);and
- *Audit For Dynamic Data Operations:* An AA, who holds a DO's secret key λ, can manipulate the outsourced data and update the associated IHT stored in TPA. The privacy of λ and the checking algorithm ensure that the storage server cannot cheat the AAs and forge the valid audit records(see Fig. 2c);

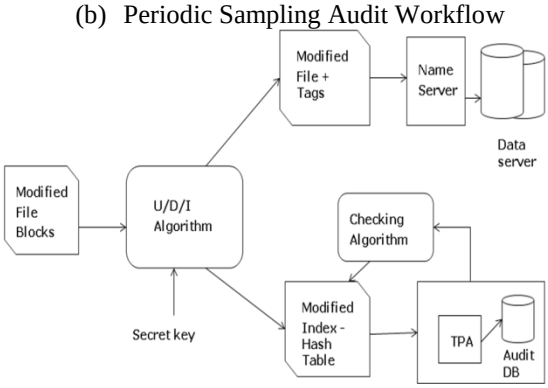
The Authorized Application's provides the cloud application services, must be authorized by the DOs. Therefore, any AAs must give TPA the authorized information to perform the dynamic operations on the data. If any unauthorized applications occur then the auditing service should detect it. So, this is a strong authentication-verification mechanism.



(a) Tag Generation and User Verification Workflow



(b) Periodic Sampling Audit Workflow



(c) Dynamic operations Audit Workflow

Fig. 2. Three processes of the audit system.

The entire goal of this audit structure is to provide the creditability in cloud. So, the burden of the DO and CSP are greatly reduced. The entire auditing service is handled by a Trusted Third Party Auditor (TTPA).There is no additional cost for using this mechanism.

This process involves the following algorithms: *KeyGen, TagGen, Update, and Delete, Insert* algorithms.

The following techniques are required to construct a this algorithms. The techniques are Fragment structure, Bilinear Mapping and Index table (ITable).

3.1 Fragment Structure and Secure Tags

To maximize the performance the audit system, a general fragment structure is used.A file F is split into *n* data components $\{m_1,\ldots,m_n\}$.Each data block m_i is further split into s_i sectors.

1. Select the maximum no. of sectors s_{max} among all the sector numbers s_i .
2. m_i has s_{max} sectors
   Set $m_{ij}=0$ for $s_i < j <_ s_{max}$
3. Size of each sector is constant and equal to security parameter *p.*
4. Calculate the number of data blocks.
5. Encrypted data component is

$$M = \{m_{ij}\} i \in [1,n],\ j \in [1,s]$$

where
s = number of sectors in each data block.
p= security parameter.

3.2 Bilinear Mapping

Let G_1,G_2 and G_T be multiplicative groups with same prime order p. Bilinear map is given by

$$e: G_1 \times G_2 \rightarrow G_T$$

Let g_1 and g_2 be the generators of G_1 and G_2 . . Keyed secure hash function

$$h: \{0,1\}^* \rightarrow G_1$$

1.3 Index table (ITable)

ITable is used to record the abstract information of the data .It consists of 4 components

Index -Current block number of data block m_i..

B_i -Original block number of data block m_i .

V_i -Current version number of data block m_i .

T_i-Timestamp used for generating the data tag .

TABLE 3ITable of the Abstract Information of Data M

Index	B_i	V_i	T_i
1	1	1	T1
2	2	1	T2
…	…	…	…

IV.ALGORITHMS FOR AUDIT SYSTEM

The algorithms for tag generation process:

1. *Key Generation Algorithm* - KeyGen(**λ)**

Algorithm KeyGen (λ)
//Input: λ security parameter
//Output: sk_h secret hash key
// sk_t secret public tag key
// pk_t public tag key
begin
Choose 2 random numbers sk_t , sk_h ∈ Z_p
Compute $pk_t = g_2^{sk_t} \in G_2$
end

2. *Tag Generation Algorithm* -TagGen(M, sk_t, sk_h)

Algorithm TagGen (M, sk_t, sk_h)
//Input: M encrypted file
// sk_h secret hash key
// sk_t secret public tag key
//Output: T set of tags
begin
Choose s random values x_1 , x_2,… x_s ∈ Z_p
Compute $u_j = g_1^{x_j} \in G_1 \forall j \in [1, s]$

for each data block mi
Compute $t_i = (h(sk_h, W_i).\prod_{j=1}^{s} u_j^{m_{ij}})^{sk_t}$
end for
end
where
W_i = FID || i .
FID = identifier of data.
i = block number of m_i.

3. *Challenge Algorithm* - Chall (M_{info})

Algorithm Chall (M_{info})
// Input: M_{info} Abstract information of data
// Output: *C* Challenge
begin
Select some data blocks and construct challenge set Q
for each chosen block m_i
Generate random numbers v_i ∈ Z_p^*
Compute challenge stamp

$$R = (pk_t)^r$$

end for
return $C=(\{i,v_i\}_{i \in Q},R)$
end
where r = randomly chosen number $r \in Z_p^*$

4. *Proof Algorithm* - Prove (M,T,C)

Algorithm Prove (M,T,C)
//**Input:** M File
// T Tag from auditor
// *C* Challenge
//**Output:** P= (TP,DP) Proof
// TP Tag Proof
// DP Data Proof
begin
Generate tag proof $TP = \prod_{i \in Q} t_i^{v_i}$
Compute sector linear combination
Generate data proof $MP_j = \sum_{i \in Q} v_i.m_{ij}$
$DP = \prod_{j=1} e(u_j, R)^{MP_j}$

End

5. *Verification Algorithm* - Verify (C ,P, sk_h, pk_t ,M_{info})
   - The output of this algorithm is 0 means then , the integrity of the cloud data is compromised.

Algorithm Verify (C ,P, sk_h, pk_t , M_{info})
// Input: T tag from auditor
// *C* challenge
// sk_h secret hash key
// pk_t public tag key
// M_{info} abstract information of data
// Output: 0 or 1
begin
Compute identifier hash values h=(sk_h,W_i).
Compute challenge hash

$$H_{chal} = \prod_{i \in Q} h(sk_h, W_i)^{rv_i}$$

Verifies the proof by verification equation

$$DP.e(H_{chal}, pk_t) = e(TP, g_2^r)$$

```
  if both the values are equal then
        return 1
  else
        return 0
  end if
end
```

V.CONSTRUCTION OF THE AUDIT SCHEME

This scheme includes a 2-move interactive proof protocol, which also provides privacy protection property to ensure the confidentiality of secret data.

5.1 Notations

Let be a keyed hash family of functions H_k : $H = \{H_\kappa\}$
The collision resistance hash function SHA1 is used, where the probability is over random choice is made . This hash function can be obtained from

$$\{0,1\}^* \rightarrow \{0,1\}^n$$

hash function of BLS signatures.Bilinear pairings proposed by Boneh and Franklin [11] is used . Let G and G_T be two multiplicative groups using elliptic curve conventions with a large prime order p. The function e be a computable bilinear map e : G x G-> G_T with the following properties: 1) Bilinearity2) Nondegeneracy and 3) Computability. The file is encrypted using collision resistance algorithm by using the bilinear mapping and then the tags are generated for this file which is fragmented.

5.2 Construction for Dynamic Auditing

Data update. There are three types of data update operations that can be used by the owner: modification, insertion, and deletion. For each update operation, there is a corresponding algorithm in the dynamic auditing to process the operation and facilitate the future auditing, defined as follows:
Data update. There are three types of data update
Operations that can be used by the owner: modification, insertion, and deletion. For each update operation, there is a corresponding algorithm in the dynamic auditing to process the operation and facilitate the future auditing, defined as follows:

$$Modify(m_i^*, sk_t, sk_h) \rightarrow (Msg_{modify}, t_i^*)$$

The modification algorithm takes as inputs the new version of data block m_i^*,the secret tag key sk_t, and the secret hash key sk_h. Itgenerates a new version number V_i^*, new time stamp T_i^*,and calls the TagGen to generate a new data tag t_i^* for data.

$$Insert(m_i^*, sk_t, sk_h) \rightarrow (Msg_{insert}, t_i^*)$$

The insertion algorithm takes as inputs the new data block m_i , the secret tag key sk_t, and the secret hash key sk_h. It inserts a new data block mi before the ith position. It generates an original number B_i, a new version number V_i , and a new time stamp T_i . Then, it calls the TagGen to generate a new data tag t_i for the new data block m_i. It outputs the new tag t_i and the update message
Then, it inserts the new pair of data block and tag (m_i^*,t_i^*)on the server and sends the update message Msg_{insert} to the auditor. $Msg_{insert} = (i, B_i^*, V_i^*, T_i^*)$

$$Delete(m_i) \rightarrow Msg_{delete}$$

The deletion algorithm takes as input the data block m_i. It outputs the update message
$Msg_{delete} = (i, B_i, V_i, T_i)$
It then deletes the pair of data block and its tag (m_i, t_i)from the server and sends the update message Msg_{delete} to the auditor.

VI CONCLUSION

In this paper, a flexible auditing mechanism was proposed by using Bilinearity property of bilinear pairing and also supports audit for dynamic operations. This enhances the performance of TPAs.

REFERENCES

[1] Yan Zhu, Gail-Joon Ahn, Hongxin Hu, Stephen S. Yau, Ho G. An, and Chang-Jun Hu, "Dynamic Audit Services for Outsourced Storages in Clouds", *IEEE Transactions on Services Computing, Vol. 6, No. 2, April-June 2013*.

[2] 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.

[3] M. Mowbray, "The Fog over the Grimpen Mire: Cloud Computing and the Law," *Technical Report HPL-2009-99, HP Lab.*, 2009.

[4] A.A. Yavuz and P. Ning, "BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems," *Proc. Ann. Computer Security Applications Conf. (ACSAC), pp.* 219-228, 2009.

[5] G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J. Peterson, and D.X. Song, "Provable Data Possession at Untrusted Stores," *Proc. 14th ACM Conf. Computer and Comm.Security, pp.* 598-609, 2007.

[6] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," *Proc. Fourth Int'l Conf. Security and Privacy in Comm. Netowrks (SecureComm), pp.* 1-10,2008.

[7] C.C. Erway, A. Ku¨ pc¸u¨ , C. Papamanthou, and R. Tamassia,"Dynamic Provable Data Possession*," Proc. 16th ACM Conf. Computer and Comm. Security, pp.* 213-222, 2009.

[8] H. Shacham and B. Waters, "Compact Proofs of Retrievability," *Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology Advances in Cryptology,(ASIACRYPT '08), J. Pieprzyk, ed., pp.* 90-107, 2008.

[9] A.R. Yumerefendi and J.S. Chase, "Strong Accountability for Network Storage," *Proc. Sixth USENIX Conf. File and Storage Technologies (FAST), pp*. 77-92, 2007.

[10] M. Xie, H. Wang, J. Yin, and X. Meng, "Integrity Auditing of Outsourced Data," *Proc. 33rd Int'l Conf. Very Large Databases (VLDB), pp*. 782-793, 2007.

[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.*