Study and Implementation of Secure Storage Service in Cloud Computing

Mrs. Pramila Kailas Ahire, Prof. R. V. Patil

***Abstract*— Cloud storage provides users to easily store their data and enjoy the cloud applications need not install in local hardware and software system. In cloud computing, data is moved to a remotely located cloud server. Many users place their data in the cloud and so data integrity is very important issue in cloud storage. In this paper a flexible distributed storage auditing mechanism, using the homomorphism token and distributed erasure coded data and it support to ensure the correctness and availability of users' data in the cloud. It also proposes users to audit the cloud storage with very lightweight communication and computation cost. Considering the cloud data are dynamic in nature, this mechanism further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append.**

***Index Terms*— Cloud Computing, data dynamics, and Cloud Storage dependable distributed storage.**

I. INTRODUCTION

Cloud Computing is using hardware and software as computing resources to provide service through internet. Cloud computing provides various service models as platform as a service (PaaS), software as a service (SaaS), Infrastructure as a service (Iaas), storage as a service (STaaS), security as a service (SECaaS), Data as a service (DaaS) & many more[1] [2].

Cloud computing is the use of computing of sources that are delivered as a service over a network. Cloud enables user to store data. But data is stored at remote machine, so it poses new security risks. A major characteristic of the cloud services is that user's data are usually processed remotely in unknown machines that users do not operate. So, basic need is to provide security to cloud server [3].

Cloud computing has many advantages are: it can easily upload and download the data stored in the cloud without worrying about security. It can access the data from anywhere, any time on demand. Cost is low or pay per usage basis. Hardware and software resources are easily available without location independent [11].

Hence, the verification of cloud storage correctness must be conducted without explicit knowledge of the whole data files [9]. Meanwhile, cloud storage is not just a third party data warehouse. The data stored in the cloud may not only be accessed but also be frequently updated by the users [5], include insertion, deletion, modification, appending, etc. It is more advantages for individual users to store their data redundantly across multiple physical servers so as to reduce the data integrity and availability threats.

Manuscript received October, 2015.

***Pramila Kailas Ahire**, Computer Science and Technology, N.M.U., Jalgaon, Dhule, India.*

***R. V. Patil**, Computer Science and Technology, N.M.U., Jalgaon,, Dhule, India.*

The cloud prohibits the traditional cryptography technique for the purpose of data integrity. Th e data stored in the cloud not only accessed but frequently updated to the user [8], including block insertion, modification, append and delete operations. Cloud computing is a data centers running in distributed, simultaneous and cooperated manner [4]. It is more advantage for individual users to store their data across multiple servers to reduce data availability and data integrity threats. In this paper, it propose an effective and flexible distributed storage integrity auditing mechanism for dynamic data support to ensure the correctness and availability of data in cloud.[1]

A. System Model

Representative network architecture for cloud storage service architecture is illustrated in figure 1. Following different network entities can be identified.

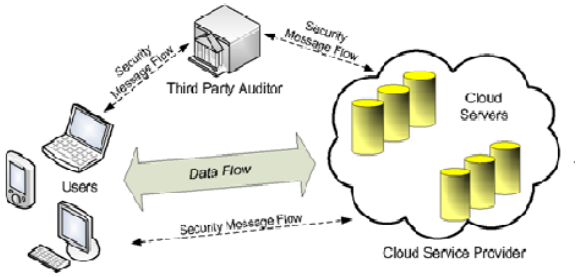


Fig 1: Cloud storage service architecture [1]

User: users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations [1].

Cloud Service Provider (CSP): a CSP, who has significant resources and expertise in building and managing distributed cloud storage servers, owns and operates live Cloud Computing systems.

Third Party Auditor (TPA): an optional TPA, who has expertise and capabilities that users may not have, is trusted to assess and expose risk of cloud storage services on behalf of the users upon request [1].

Owner
an entity, who has data to store to be stored in the cloud and release on the cloud for data storage and computation, can be either enterprise or individual customers[1].

File Retrieval and Error Recovery
The user can reconstruct the original file by downloading the data vectors from the first m servers, assuming that they return the correct response values. Notice that our verification scheme is based on random spot-checking, so the storage correctness assurance is a probabilistic one [1] [2] [11].

Dynamic Data Operations
The data stored in the cloud can be dynamic. User might have the requirement of performing various block-level operations such as update, delete and append to modify the existing data file. Storage correctness must also be assured at the same time. So for any

dynamic data operation, the user must first generate the file blocks using the secret key. The cloud service provider checks if dynamic operations are performed correctly using verification tokens [7].

The process is summarized below as follows:

(1) Update Operation
In cloud data storage, sometimes the user may need to modify some data block(s) stored in the cloud. In other words, for all the unused tokens, the user needs to exclude every occurrence of the old data block and replace it with the new one.

(2) Delete Operation
Sometimes, after being stored in the cloud, certain data blocks may need to be deleted. The delete operation we are considering is a general one, in which user replaces the data block with zero or some special reserved data symbol.

(3)Append Operation
In some cases, the user may want to increase the size of his stored data by adding blocks at the end of the data file, which we refer as data append. [1].

In cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Data redundancy can be employed with technique of erasure correcting code to further tolerate faults or server crash as users data grows in size and importance. However, to securely introduce such a TPA, any possible leakage of users' outsourced data towards TPA through the auditing protocol should be prohibited [12] [15].

B. System Goals

Our design goals can be summarized as the following [1] [10] [15]:

- **Public audit ability for storage correctness assurance:** to allow anyone, not just the clients who originally stored the file on cloud servers, to have the capability to verify the correctness of the stored data on demand.
- **Fast localization of data error:** to effectively locate the malfunctioning server when data corruption has been detected.
- **Dynamic data operation support:** to maintain the same level of storage correctness assurance even if users modify, delete or append their data files in the cloud.
- **Dependability:** to enhance data availability against Byzantine failures, malicious data modification and server colluding attacks, i.e. minimizing the effect brought by data errors or server failures.
- **Lightweight:** to enable users to perform storage correctness checks with minimum overhead [1] [12].
- **Block less verification:** no challenged file blocks should be retrieved by the verifier (e.g., TPA) during verification process for efficiency concern.

C. Need of System

It focuses on cloud data storage security, which has always been an important aspect of quality of service. The new scheme further supports secure and efficient dynamic operations on data blocks, including: data update, delete and append [2].

1. It is an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud.
2. Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, etc. To ensure storage correctness under dynamic data update is hence of paramount importance. However, this dynamic feature also makes traditional integrity insurance techniques futile and entails new solutions [1] [13].

Advantages

1. Compared to many of its predecessors, which only provide binary results about the storage state across the distributed servers, the challenge-response protocol in our work further provides the localization of data error.
2. Unlike most prior works for ensuring remote data integrity, the new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append [1] [15].

II. CURRENT DATA STORAGE CHALLENGES IN CLOUD

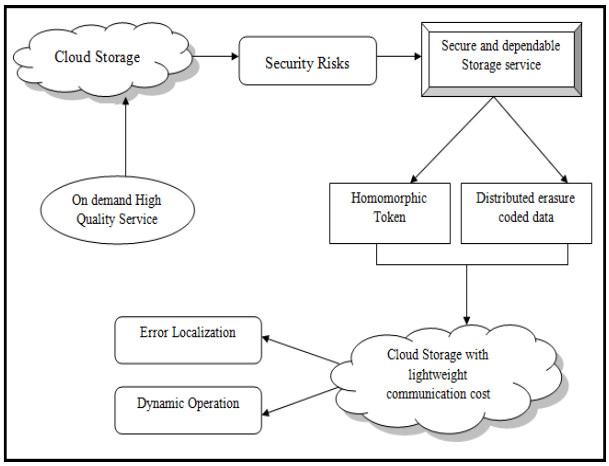


Fig 2: Functions of cloud storage service architecture

In cloud data storage system, user can upload or stores the data into cloud or use services from the cloud (Here it focused on file storage and retrieval operations). User stores data into set of cloud servers which are running in a distributed and cooperated manner. Data redundant techniques can be employed using erasure correcting code to protect from faults or server crashes. Users can perform manipulations on stored data like insert update and append through blocks. Block level updating and deletions are allowed with token checking.

A. File Distribution Preparation

It is well known that erasure-correcting code may be used to tolerate multiple failures in distributed storage systems. In cloud data storage, it rely on this technique to disperse the data file F redundantly across a set of $n = m+ k$ distributed servers. An (m, k) Reed-Solomon erasure-correcting code is used to create k redundancy parity vectors from m data vectors in such a way that the original m data vectors can be reconstructed from any m out of the $m+k$ data and parity vectors. By placing each of the $m+k$ vectors on a different server, the original data file can survive the failure of any k of the $m+k$ servers without any data loss, with a space overhead of k/m. For support of efficient sequential I/O to the original file, our file layout is systematic, i.e., the unmodified m data file vectors together with k parity vectors is distributed across m + k different servers [1].

B. Challenge Token Pre-computation

In order to achieve assurance of data storage correctness and data error localization simultaneously, our scheme entirely relies on the pre-computed verification tokens. The main idea is as follows: before file distribution the user pre-computes a certain number of short verification tokens on individual vector, each token covering a random subset of data blocks [1] [14]. When the user wants to make sure the storage correctness for the data in the cloud, he challenges

the cloud servers with a set of randomly generated block indices. Meanwhile, as all servers operate over the same subset of the indices, the requested response values for integrity check must also be a valid codeword determined by secret matrix P [1].

Algorithm: Token pre computation

1: procedure

2: Choose parameters l, n and function f,q:

3: Choose the number t of tokens;

4: Choose the number r of indices per verification;

5: Generate master key Kprp and challenge key K_{chal}

6: for vector G(j), j ← 1,n do;

7: for round i← 1, t do

8: Derive α_i= f K_{chal} and $k(i)_{prp}$ from. K_{prp}

9: Compute $V_i^{(j)} = \sum_{q=1}^{r} \propto_i^q * G(f)\left[\phi_{KRP}^i(q)\right]$

10: end for

11: end for

12: Store all the V i's locally.

13: end procedure

After token generation, the user has the choice of either keeping the pre-computed tokens locally or storing them in encrypted form on the cloud servers. In our case here, the user stores them locally to obviate the need for encryption and lower the bandwidth overhead during dynamic data operations.

C. Correctness Verification and Error Localization

Error localization is a key prerequisite for eliminating errors in storage systems. It is also of critical importance to identify potential threats from external attacks. However, many previous schemes, [3] do not explicitly consider the problem of data error localization, thus only providing binary results for the storage verification. This scheme outperforms those by integrating the correctness verification and error localization in our challenge-response protocol: the response values from servers for each challenge not only determine the correctness of the distributed storage, but also contain information to locate potential data error(s).

Once the inconsistency among the storage has been successfully detected, it can rely on the pre-computed verification tokens to further determine where the potential data error(s) lies in. Note that each response $R_i^{(j)}$ is computed exactly in the same way as token $v_i^{(j)}$, thus the user can simply find which server is misbehave by verifying the following n equations:

$$R_i^{(j)} =? v_i^{(j)}, j \in \{1, \ldots, n\}.$$

Algorithm: Correctness Verification and Error Localization

1. procedure CHALLENGE(i)
2. Recomputed $\propto_i = f_{k_{chal}}(i)$ and $k_{prp}^{(i)}$ from K_{PRP};
3. Send $\left\{ \propto_i, k_{prp}^{(i)} \right\}$ to all the cloud servers;
4. Receive from servers:

$$\left\{ R_i^{(j)} = \sum_{q=1}^{r} \propto_i^q * G^{(j)} \left[\varphi_{k_{prp}^{(i)}}(q) \right] \middle| 1 \le j \le n \right\}$$

5. For $(j \leftarrow m+1, n)$ do
6. $\left\{ R^{(j)} \leftarrow R^{(j)} - \sum_{q=1}^{r} f_{kj}(SI_{q,j}) \cdot \propto_i^q, I_q = \varphi_{k_{prp}^{(i)}}(q) \right\}$
7. end for
8. if $\left(\left(R_i^{(1)}, \ldots, R_i^{(m)} \right).P == \left(R_i^{(m+1)}, \ldots, R_i^{(n)} \right) \right)$ then
9. Accept and ready for the next challenge.
10. else
11. for (j ← 1, n) do
12. if $\left(R_i^{(j)} != v_i^{(j)} \right)$ then
13. Return server j is misbehaving.
14. end if
15. end for
16. end if
17. end procedure

D. Towards Third Party Auditing

As discussed in architecture, in case the user does not have the time, feasibility or resources to perform the storage correctness verification, he can optionally delegate this task to an independent third party auditor, making the cloud storage publicly verifiable. However, as pointed out by the recent work [9], to securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy. The persona of Third Party Auditor (TPA) is listed as follows: Reduce data owner's burden in managing the data. Ensure the client that the data stored in the cloud is intact and data integrity is maintained.

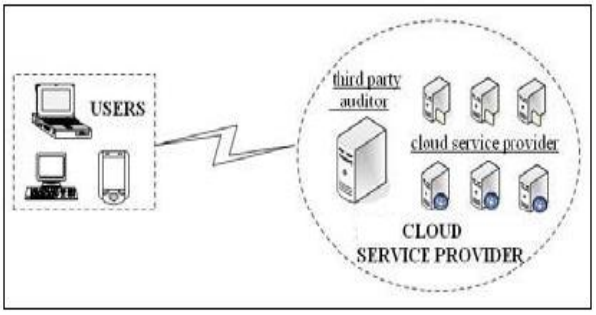


Fig 3: TPA with Service Provider

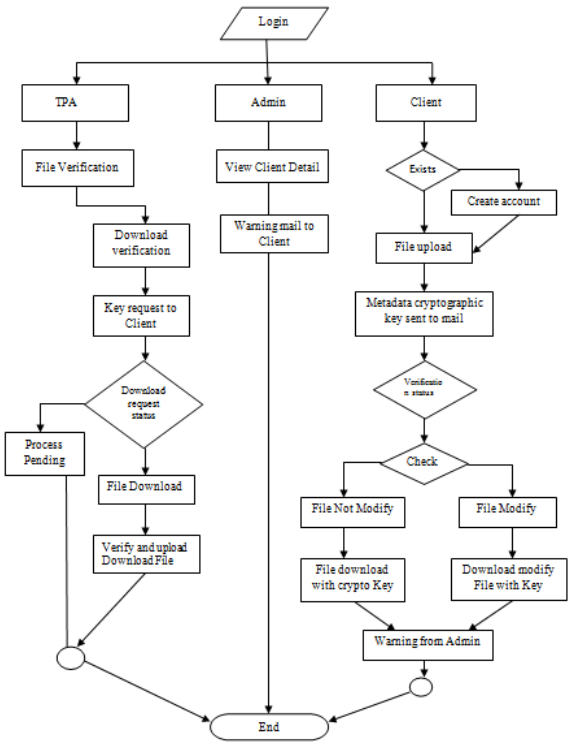
Namely, TPA should not learn user's data content through the delegated data auditing. Now it shows that with only slight modification, our protocol can support privacy-preserving third party auditing. The new design is based on the observation of linear property of the parity vector blinding process [2] [12].

Recall that the reason of blinding process is for protection of the secret matrix P against cloud servers. However, this can be achieved either by blinding the parity vector or by blinding the data vector (it assume $k < m$). Thus, if they blind data vector before file distribution encoding, then the storage verification task can be successfully delegated to third party auditing in a privacy-preserving manner [13].

III. SYSTEM DESIGN

It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

Flow Diagram



IV. PERFORMANCE EVALUATION

In this, the performance of the proposed storage auditing scheme is the cost of file distribution preparation as well as the token generation. As discussed, file distribution preparation includes the generation of parity vectors (the encoding part) as well as the corresponding parity blinding part. We consider two sets of different parameters for the (m, k) Reed-Solomon encoding, both of which work over. Fig 4 shows the total cost for preparing a file before outsourcing. In the figure, it set the number of data vectors and the number k is the dominant factor for the cost of both parity generation and parity blinding.

The following graph shown in Fig 4 is performance comparison with different file upload in which, number of files ranging from 0 to n are taken along x-axis and cost time taken for upload file ranging from 0 to 5 minutes are taken along y-axis. It can be inferred from the graph that as like Encryption; Decryption time taken by Reed Solomon Erasure Code is lesser than existing System which indicates proposed system works faster than the existing system.

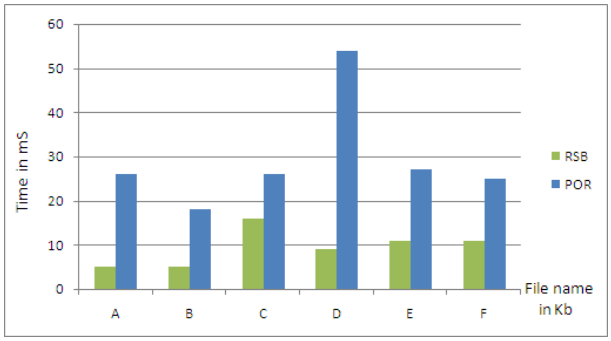


Fig 4: Performance comparison between different upload files

V. CONCLUSIONS

From all above discussion, it shows that it can provide security to data stored on cloud i.e. providing security to remotely stored data is possible. With the help of tokens generation and token matching we are providing security. It allows user to perform block operation i.e. append, delete, modify as well as to give challenge to uploaded to check correctness of data. It investigates the problem of data security in cloud data storage, which is essentially a distributed storage system. The integrity and availability of cloud storage service is achieved by flexible distributed scheme that will supports dynamic operations including block update, delete, append and insert and use the FNT based Reed Solomon erasure correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. This scheme also achieves the integration of storage correctness insurance and data error localization through utilizing the homomorphic token with distributed verification of erasure coded data

REFERENCES

[1] C. Wang, Q. Wang, K. Ren, Ning Cao and W. Lou, "Towards Secure and Dependable Storage Services in loud Computing", IEEE Transactions on Cloud Computing, Vo. 5 , no 2, April-June 2012,pp. 1-14.
[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in Proc. of IWQoS'09, July 2009, pp.1–12.
[3] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.
[4] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability:Theory and implementation," in Proc. of ACM workshop on Cloud Computing security (CCSW'09), 2009, pp. 43–54.
[5] 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, Alexandria, VA, October 2007, pp. 598–609.
[6] 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, volume 5789 of LNCS. Springer- Verlag, Sep. 2009, pp. 355–370.
[7] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of CCS'09, 2009, pp. 213–222.
[8] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," IEEE Network, vol. 24, no. 4, July/Aug. 2010,pp. 19-24.
[9] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Audit ability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.
[10] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for storage security in cloud computing," in Proc. of IEEE INFOCOM'10, San Diego, CA, USA, March 2010.
[11] Shingare Vidya Marshal, "Secure Audit Service by Using TPA for Data Integrity in Cloud System" IJITEE, Volume-3, Issue-4, September 2013.
[12] Tinku Abey Koshy and S Prema, "Third Party Auditor for Secure Cloud storage" IJISET, Vol. 1 Issue 3, May 2014, pp. 210-213.
[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE INFOCOM, pp. 525-533, 2010.
[14] K.Meenakshi and Victo Sudha George, "Cloud Server Storage Security Using TPA", IJARCST, Vol. 2 Issue Special 1 Jan-March 2014,pp. 295-299.
[15] 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, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, System pp. 50-58, 2010

