Distributed scheme to authenticate data storage security in cloud computing

B. Rakesh[1], K. Lalitha[2], C. Sushama[3], H. Parveen Sultana[4], O. Rajitha[5]
[1,2,3,5]Department of CSSE, Sree Vidyanikethan Engineering College (Autonomus), Tirupati, India
[4]SCOPE, VIT University, Vellore, India

Article Info

Article history:

Received Jan 10, 2019
Revised Feb 28, 2019
Accepted Mar 6, 2019

Keywords:

Cloud computing
Cloud storage security
Homomorphic token

ABSTRACT

Cloud computing is the revolution in current generation IT enterprise displaces database and application software to the large data centers for IT services. In this paper, we designed and simulated an adaptable and efficient scheme to guarantee the correctness of user data stored in the cloud and also with some prominent features. Homomorphic token is used for distributed verification of erasure–coded data. By using this scheme, we can identify misbehaving servers. In spite of past works, our scheme supports effective and secure dynamic operations on data blocks such as data insertion, deletion and modification. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, cloud computing moves the application software and databases to the large data centers, where the data management and services may not be absolutely truthful. This effective security and performance analysis describes that the proposed scheme is extremely flexible against malicious data modification, convoluted failures and server clouding attacks.

Corresponding Author:

B. Rakesh,
Department of CSSE,
Sree Vidyanikethan Engineering College (Autonomus),
Tirupati, India.
Email: bandarupallirakesh@gmail.com

1. INTRODUCTION

Cloud computing is the revolution in current generation IT enterprise [1]. Cloud computing displaces database and application software to the large data centers, where the management of services and data may not be predictable, where as the conventional solutions, for IT services are under proper logical, physical and personal controls [2]. This aspect attribute, however comprises different security challenges which have not been well understood [3]. It concentrates on cloud data storage security which has always been an important aspect of quality of service (QOS) [4].

The Cloud Computing provides the service over the internet by using the computing resources i.e. hardware and software [5]. By using software as a service, users can access software and database in the business model [6]. The management of infrastructure and platform is done by using Cloud providers. Sometimes, software as a service is referred to as on-demand software. Because of that Cloud providers charged on pay per use basis. So this causes proponents to get the computing resources in Outsourcing manner [7]. That's why the maintenance of hardware and software is being reduced for users [8]. Introducing of Cloud Computing avoids direct management of hardware and it causes great convenience to users. Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3) are the best examples for initiators of Cloud Computing [9]. In the Cloud Computing the responsibility of local machines is eliminating due to computing platform shift [10]. Here in the Cloud Computing, the data is integrated so that's why the users are showing kindliness towards the cloud service providers.

Providing security to the data is a very important factor which has always been an important factor of quality of service [11]. Cloud Computing is exhibiting new challenges due to many reasons. In Cloud Computing the very important thing is that data should be under the control of user. But by using the traditional cryptographic techniques the users lose the control of data. So traditional Cryptographic primitives cannot be adopted directly [12]. If we use the traditional Cryptographic primitives directly, we will have conduct the verification for correctness of data stored in the cloud without having the proper knowledge about the whole data [13]. It is even more challenging that to provide the continuous assurance of the data safety, if we store different categories of data stored by different users.

The users are updating the data continuously stored in the cloud. The updation of the data includes insertion, deletion, modification, appending, reordering etc. During the dynamic data updation it is very important task that to assure the storage correctness of data [14]. Techniques which are used can be useful to give assurance tof storage correctness without possessing data of user, where it cannot addresses all the security threat of data storage in cloud, since they all concentrating on single server situation and a large portion of them don't consider dynamic operations on data [15]. As a balancing approach, researchers have also proposed protocols which are distributed for ensuring storage correctness data throughout multiple servers or peers. Again, in these no distributed schemes is conscious about operations on dynamic data. As a result, their applicability on cloud data storage can be considerably limited [16].

In this paper, we present a viable and flexible distributed scheme with dynamic data support explicitly to verify the user’s data correctness in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability [17]. This development radically diminishes the communication and storage overhead when contrasted with the conventional techniques of replication-based file distribution [18]. Utilization of homomorphic token with distributed verification of erasure-coded data, this scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been identified during the verification of storage correctness, this scheme can localize the data errors concurrently, i.e., the identification of the misbehaving server(s) [19].

a. 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.
b. Dislike past works for guaranteeing data integrity remotely, on data blocks the new scheme supports efficient and secure dynamic operations, which includes update, delete and append.
c. The highly efficient security and performance analysis shows that the proposed scheme is extreemly efficient and rigid against Byzantine failure, server colluding attack, and even malicious data modification attacks.

1.1. Cloud data storage architecture

The network architecture for cloud data storage is as represented in the Figure 1. The architecture is consisting of three different network modules as represented below:

a. Users: There are different categories of users. The users consisting of both organizations and individual users. The users are having the data to store in the cloud. The users must have the trust on the cloud computation.
b. Cloud Service Provider (CSP): A cloud service provider has the knowledge and important resources in building and managing distributed cloud storage servers. Cloud service providers poses and operates live Cloud Computing systems.
c. Third Party Auditor (TPA): An optional TPA, who has expertise and capabilities that users may not have, illstrusted to assess and expose risk of cloud storage services on behalf of the users upon request. 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. Redundancy of data redundancy can be engaged with technique of erasure-correcting code to tolerate server crash or faults as data of the user grows in importance and also size.

After that for the purpose of application, the user interacts with the cloud servers via Cloud Service Provider(CSP) to retrieve or access his data. In some cases, block level operations has been performed by user on his data.

Most general styles are block insert, delete, update and append are different operations. As users now not possess their knowledge for data locality, it’s of critical importance to guarantee users that their data are being properly stored and maintained [20]. That is, users ought to be equipped with security means that in order that they'll build continuous correctness assurance of their data stored in cloud even without existence of copy of data locally.

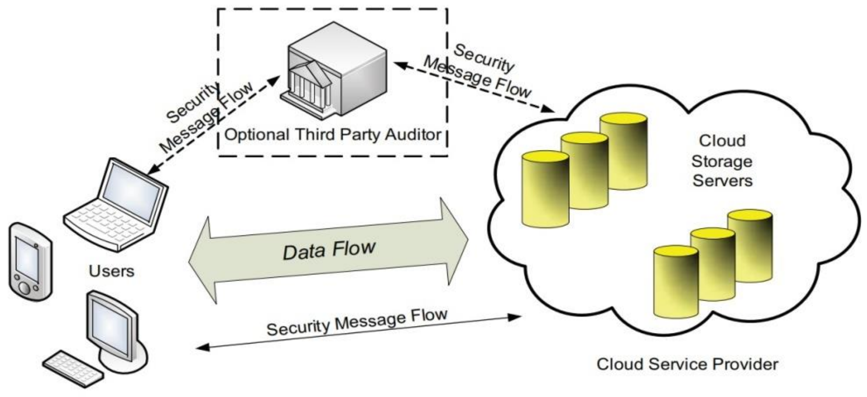

Figure 1. Cloud data storage architecture

Figure 2 is representing cloud server data security architecture, which consists of four modules. In case if the users don't necessarily have the time, possibility of resources to check their data, they can hand over the tasks to an optional trusted Third Party Auditor (TPA) of their relevant choices. In this model, we presume that the point-to-point communication channels between each user and the cloud server is authenticated and trustworthy, which can be achieved in practice with little overhead [21]. Note that we don't address this type of data privacy issue here, as in Cloud Computing, data privacy is orthogonal to the problem what we study here.

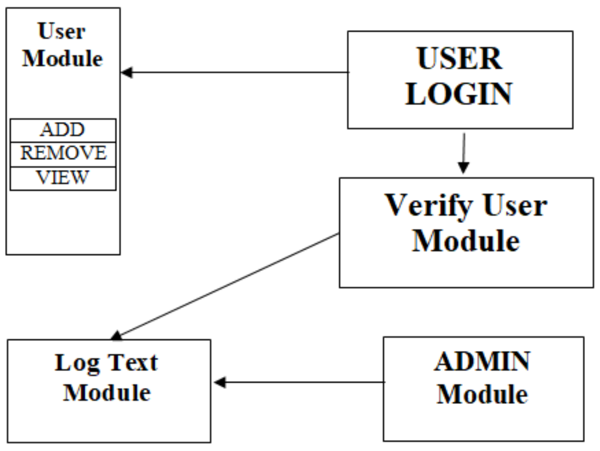

Figure 2. Cloud server data security architecture

2. KEY GENERATION ALGORITHM

2.1. Key generation(μ)

Step :

a. Start
b. Obtain parameters (g, h, p).
c. G1 can be generated by g and h.
d. G2 can be generated by prime number p.
e. Set μ=(g, h, ẽ, G1, G2, p, f)
f. f:Zp*{0,1} Zp*Is a one-way hash function
g. User A selects three parameters a1; a2; a3 Zp*
h. PKA=( ),ska=(a1, a2, a3).
i. Stop.

2.2. Share key generation algorithm

Saharekeygen(SKA,t,m)
Step:

a. Start
b. Select m key servers
c. Share secret key SKA to key server ksi
d. SKA,i=(fA, 1(i), fA, 2(i)) where 1≤i ≤m.
e. Stop.

2.3. Encryption algorithm

Encryption(PKA, T, m1, m2,... mk)
Step:

a. Start
b. Divide message in to m1,m2,…mk
c. Calculate cipher text c1,c2,... ck by using

$$Ci=(O,\alpha i,\beta,\gamma i)=(0,gr2,T,m1,e(ga1,Tr1)$$

d. Stop.

2.4. Key recovery algorithm

Key Recover(SKA,i1, SKA,i2, SKA,i3, SKA,i4 ..., SKA,in)
Step:

a. Start
b. User searches for first component a1.
c. if (a1 found) , then No need of key recovery.
d. if (a1 not found)
   Recovered by using

$$a_1 = \sum_{s \in T} (f_{A,1}(s) \prod_{s \in T} \left(\frac{-s'}{s - s'} \right) mod\, p$$

e. Stop

2.5. Re key generation algorithm

ReKeyGen(PKA, SKA, ID, PKB)
Step:

a. Start
b. Encrypt the data by using SKA.
c. Select a random number e from Zp*.
d. Compute rekey for proxy reencryption by using

$$RK_{A->B}^{ID} = ((h^{b_z})^{a_{1}(f(a_3ID)+e)}, h^{a_1 e})$$

e. Stop.

2.6. Proxy re-encryption algorithm

ProxyReEncryption()
Step:

a. Start
b. Calculate RK,C’.
c. ReEncrypted symbol can be computed by using

$$C'' = \left(1, \alpha, h^{b_2 a_1}(f(a_3 ID) + e)\gamma, \tilde{e}(\alpha, h^{a1e})\right)$$

d. Stop

3. IMPLEMENTATION

3.1. Client module

In this client module, the server receives query from the client. Based on the query sent by user to the server it sends the corresponding file to the client. Before doing this process, the client should authorize. At the server side; it verifies the client name and its password for security process. If it is verified successfully and received the queries form the client and search those corresponding files in the database. Finally, it finds that files and sends to the client. If the server finds the intruder means, it set the alternative Path to those intruders .

3.2. System module

Representative network architecture for cloud data storage is illustrated in Figure 1. There are 3 network entities which can be identified as below.

3.2.1. 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.

3.2.2. Cloud Service Provider (CSP)

A Cloud Service Provider, who has considerable resources and expertise in managing and building distributed servers of cloud storage, owns an operative live Computing systems of cloud.

3.2.3. 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 up on request.

3.3. Cloud data storage module

This is where an user stores his data by using CSP into a group of cloud servers, which are running concurrently, then the user interacts with the cloud servers via CSP to retrieve or access his data . In some cases, the user may need to perform block level operations on his own data. So, users ought to be equipped with security means that in order to build continuous correctness verification of their data stored in cloud even without existence of copy of data locally . If in case the use don’t have time and feasibility of resources to monitor their data, they can hand over the tasks to optional trusted Third Party Auditor(TPA) of their respective choices. In this model, we presume that the point-to-point communication channels between user and the cloud server, if user is authenticated and reliable, which can be achieved with little overhead in practice.

3.4. Cloud Authentication Server

The Authentication Server (AS) which functions as any Authentication Server would with a few more behaviors added to the typical protocol which named as client-authentication protocol. First, sending information of the client authentication to the masquerading router . The Authentication Server also functions as authorizing permission control and tickets on the application network. There is one more optional function that should be supported by the AS is the updating of client lists, causing a reduction in authentication time or even the removal of the client as a valid client depending upon the request.

3.5. Unauthorized data modification and corruption module

One of the important issue is to detect any unauthorized modification of data and corruption effectively, possibly due to server conciliation and/or random Byzantine failures. Also, in the distributed case when that type of inconsistencies are successfully detected, to find the data error lies in which server is also of great significance.

3.6. Adversary module

Threats for Security faced by data storage in cloud can come from two different resources. First, a CSP can be self-interested, and not trusted and possibly malicious. Its desire is to not only move data which is not been or is rarely used to a lower tier of storage than accepted for monitoring reasons, but there is one more chance also attempt to hide a data loss incident which is caused due to Byzantine failures or management errors and so on.

Second, there may be a chance for existence of adversary who is an economically motivated, it has capability to compromise a number of cloud data storage servers subsequently in different time intervals and it is able to modify or delete user's data for certain period while remaining undetected by CSPs. Consider two diiferent types of adversary models with different capability levels.

3.6.1. Weak adversary

This adversary is interested in mortifying the data files of user stored on individual servers. If one server is comprised, this adversary can corrupt the original data files by introducing or modifying by its own deceptive data to avoid the original data which is retrieved by user.

3.6.2. Strong adversary

This adversary is the worst case scenario, where we expect that the adversary can compromise all the other servers which are used for storage so that he/she can intentionally change the data files as long as they are internally reliable. In fact, this is equivalent to the situation where all the servers are colluding together to conceal a data loss or corruption occurrence.

4. CONCLUSION AND FUTURE WORK

Here in this paper, we examine data security problem of data storage in cloud, which is essentially in a distributed storage system. To verify the correctness of users' data of data storage in cloud, here we proposed an effective and supple distributed scheme with dynamic data support explicitly, including block update, append, and delete. Here we depends on erasure-correcting code for the preparation of distribution of file to provide parity vector of redundancy and guarantees the dependability of data. By using the homomorphic token of erasure coded data with distributed verification, our system achieves that integration of storage correctness assurance and localization of data errors, that is whenever data corruption has been detected during the correctness verification of data storage in the cloud across the distributed servers. Now, we can almost guarantee that the concurrent identification of the misbehaving servers. Through this detailed performance and security analysis, we proved that our scheme is highly resilient and efficient to malicious data modification attack, Byzantine Failures and server colluding attacks.

We believes that storage security for data in Cloud Computing is an area with full of challenges and have vital importance, is still in its immaturity now, and there are many research problems are yet to be identified in this area. We imagine many possible directions for future research in this area. The most hopeful one we believe that is a model in which public verifiability is compulsory. Public verifiability, supported in allows the Third Party Auditor (TPA) to audit the data storage in cloud without demanding users' time, resources or feasibility . An interesting question here is if we can build a scheme to achieve both storage correctness assurance and public verifiability of dynamic data. Besides, with this research on dynamic data storage in cloud, we also plan to examine the problem of fine-grained error localization of data.

REFERENCES

[1] Cong Wang, Qian Wang, and Kui Ren, "Ensuring Data Storage Security in Cloud Computing," *2009 17th International Workshop on Quality of Service,* Charleston, SC, 2009, pp. 1-9.

[2] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," *Proc. of SecureComm '08,* pp. 1–10, 2008.

[3] Amazon.com, "Amazon Web Services (AWS)," Online at http://aws.amazon.com, 2008.

[4] A. Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," *Proc. of CCS '07,* pp. 584–597, 2007.

[5] H. Shacham and B. Waters, "*Compact Proofs of Retrievability*," Proc. of Asiacrypt '08, Dec. 2008.

[6] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Cryptology ePrint Archive, Report 2008/175, 2008, http://eprint.iacr.org/.

[7] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.

[8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," *Proc. Of CCS '07,* pp. 598–609, 2007.

[9] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple- Replica Provable Data Possession," *Proc. of ICDCS '08,* pp. 411–420, 2008.

[10] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," *Proc. of the 2003 USENIX Annual Technical Conference (General Track),* pp. 29–41, 2003.

[11] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating Data Possession and Uncheatable Data Transfer," Cryptology ePrint Archive, Report 2006/150, 2006, http://eprint.iacr.org/.

[12] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," *Proc. 11th USENIX Workshop on Hot Topics in Operating Systems (HOTOS '07),* pp. 1–6, 2007.

[13] T. S. J. Schwarz and E. L. Miller, “Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage,” *Proc. of ICDCS ’06*, pp. 12–12, 2006.
[14] N. Gohring, “Amazon’s S3 down for several hours,” Online at http://www.pcworld.com/businesscenter/article/142549/amazons s3 down for several hours.html, 2008.
[15] K. D. Bowers, A. Juels, and A. Oprea, “HAIL: A High-Availability and Integrity Layer for Cloud Storage,” Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.
[16] L. Carter and M. Wegman, “Universal Hash Functions,” *Journal of Computer and System Sciences*, vol. 18, no. 2, pp. 143–154, 1979.
[17] J. Hendricks, G. Ganger, and M. Reiter, “Verifying Distributed Erasurecoded Data,” *Proc. 26th ACM Symposium on Principles of Distributed Computing*, pp. 139–146, 2007.
[18] J. S. Plank and Y. Ding, “Note: Correction to the 1997 Tutorial on Reed-Solomon Coding,” University of Tennessee, Tech. Rep. CS-03- 504, 2003.
[19] Q. Wang, K. Ren, W. Lou and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," *IEEE INFOCOM 2009*, Rio de Janeiro, 2009, pp. 954-962.
[20] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, “MR-PDP: Multiple- Replica Provable Data Possession,” *Proc. of ICDCS ’08*, pp. 411–420, 2008.
[21] D. L. G. Filho and P. S. L. M. Barreto, “Demonstrating Data Possession and Uncheatable Data Transfer,” Cryptology ePrint Archive, Report 2006/150, 2006, http://eprint.iacr.org/.