ISSN: 2321-7782 (Online)
Volume 2, Issue 1, January 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Paper
Available online at: www.ijarcsms.com

Ensuring Data Storage Security using Cloud Computing

Hemant T. Dhole[1]
B.E.Comp
Department of Computer Engineering
Jaihind College of Engineering, Kuran
Pune - India

Praful C. Papade[2]
B.E.Comp
Department of Computer Engineering
Jaihind College of Engineering, Kuran
Pune - India

Sachin B. Bhosale[3]
Prof.
Department of Computer Engineering
Jaihind College of Engineering, Kuran
Pune - India

Abstract: ***Cloud computing has been envisioned as the next-generation architecture of IT enterprise. 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 management of the data and services may not be fully trustworthy. This unique attribute, however, poses many new security challenges which have not been well understood. In this article, we focus on cloud data storage security, which has always been an important aspect of quality of service. To ensure the correctness of users' data in the cloud, we propose an effective and flexible distributed scheme with two salient features, opposing to its predecessors. By utilizing the homomorphism token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server (s). Unlike most prior works, the new scheme further supports secure and efficient dynamic operations on data blocks, including: data update, delete and append. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.***

Keywords: Byzantine failure, Homomorphic token, distributed scheme, cloud.

I. INTRODUCTION

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers.

Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. Recent downtime of Amazon's S3 is such an example. From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging

security threats for number of reasons. Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging. Secondly, 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, reordering, 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. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. Individual user's data is redundantly stored in multiple physical locations to further reduce the data integrity threats. Therefore, distributed protocols for storage correctness assurance will be of most importance in achieving a robust and secure cloud data storage system in the real world. However, such important area remains to be fully explored in the literature.

II. LITERATURE SURVEY

A. *Existing System*

From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging security threats for number of reasons.

a) Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging.

b) Secondly, 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, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance.

These techniques, while can be useful to ensure the storage correctness without having users possessing data, can't address all the security threats in cloud data storage, since they are all focusing on single server scenario and most of them do not consider dynamic data operations. As an complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. Again, none of these distributed schemes is aware of dynamic data operations. As a result, their applicability in cloud data storage can be drastically limited.

B. *Proposed System*

In this paper, we propose an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphism token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s).

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) 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.

c) Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

III. ARCHITECTURE

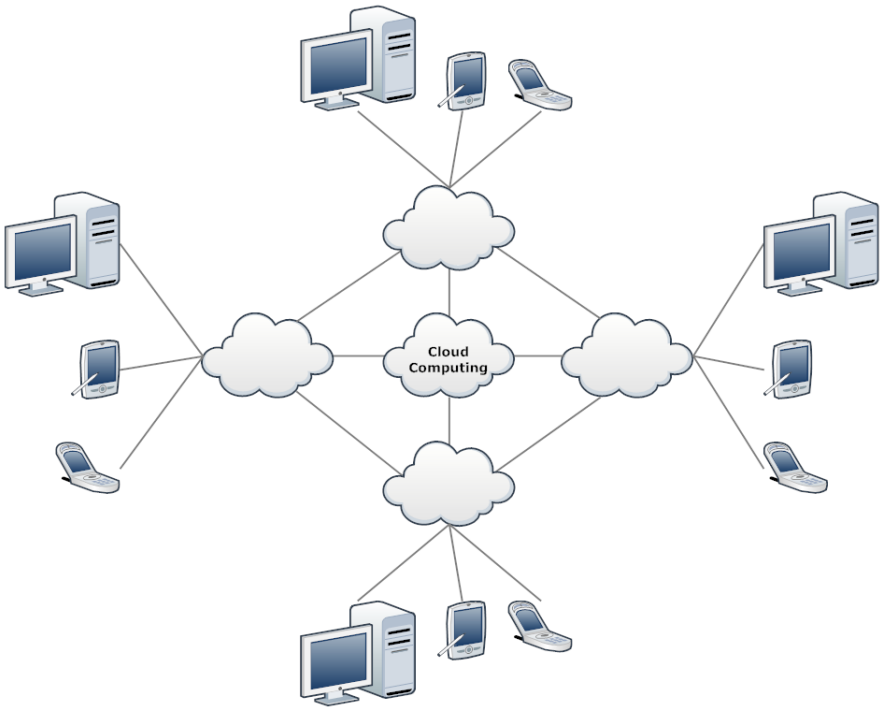

Fig. 3.1: Cloud Computing Architecture

Fig. Shows 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 user's data grows in size and importance. Thereafter, for application purposes, the user interacts with the cloud servers via CSP to access or retrieve his data. In some cases, the user may need to perform block level operations on his data. The most general forms of Data Flow Cloud Service Provider Users Cloud Storage Servers Security Message Flow Security Message Flow Security Message Flow Optional Third Party Auditor Fig. 1: Cloud data storage architecture these operations we are considering are block update, delete, insert and append. As users no longer possess their data locally, it is of critical importance to assure users that their data are being correctly stored and maintained. That is, users should be equipped with security means so that they can make continuous correctness assurance of their stored data even without the existence of local copies. In case those users do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the tasks to an optional trusted TPA of their respective choices. In our model, we assume that the point-to-point communication channels between each cloud server and the user is authenticated and reliable, which can be achieved in practice with little overhead. Note that we don't address the issue of data privacy in this paper, as in Cloud Computing, data privacy is orthogonal to the problem we study here.

IV. ALGORITHM

Algorithm shows the operation of the Autonomous Reconfiguration System.

(1) Homomorphic encryption:

It is a form of encryption which allows specific types of computations to be carried out on ciphertext and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext. For instance, one person could

add two encrypted numbers and then another person could decrypt the result, without either of them being able to find the value of the individual numbers.

This is a desirable feature in modern system architectures. Homomorphic encryption would allow the chaining together of different services without exposing the data to each of those services, for example a chain of different services from different companies could 1) calculate the tax 2) the currency exchange rate 3) shipping, on a transaction without exposing the unencrypted data to each of those services.[1] Homomorphic encryption schemes are malleable by design. The homomorphic property of various cryptosystems can be used to create secure voting systems,[2] collision-resistant hash functions, private information retrieval schemes and enable widespread use of cloud computing by ensuring the confidentiality of processed data.

There are several efficient, partially homomorphic cryptosystems, and a number of fully homomorphic, but less efficient cryptosystems. Although a cryptosystem which is unintentionally homomorphic can be subject to attacks on this basis, if treated carefully homomorphism can also be used to perform computations securely.

(2)Token Precomputation algorithm:

Procedure

1. Choose parameters F , fL and secret vector V

2. Choose number of blocks to be taken (normally fixed block size)

3. X= F + fL + V

Compute key

4. for i=1 to n

5. file Token =file Token +(Σ ni =1 split(Xi))

blocks dynamically. Once user has been sent the requested file to TPA. TPA monitors whether he is authenticated user or not for accessing the file.TPA maintains the file details and tokens (if

TPA is not present user will have the details) but not an entire file, TPA requests the file by passing the pre-computed token stored in the database for each block. If this token is same as it is present in cloud server, cloud server will send the requested blocks. We can easily check whether the file blocks were damaged or not by computing tokens dynamically as follows:

When TPA challenges or requests a block with block indices, cloud server receives this input and it computes the token of that particular block and sends the short signature to TPA. Upon receiving the signature TPA verifies it with the existing token signature. The result of two tokens are same means the block remains same without any effect, otherwise TPA assumes block was modified and it generates a message to cloud server to perform block recovery operation using distributed schemes and erasure coded techniques.

V. METHODOLOGY

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

A. Main Modules

a) Client Module: In this module, the client sends the query to the server. Based on the query the server sends the corresponding file to the client. Before this process, the client authorization step is involved. In the server side, it checks the client name and its password for security process. If it is satisfied and then received the queries form the client and search the corresponding files in the database. Finally, find that file and send to the client. If the server finds the intruder means, it set the alternative Path to those intruders.

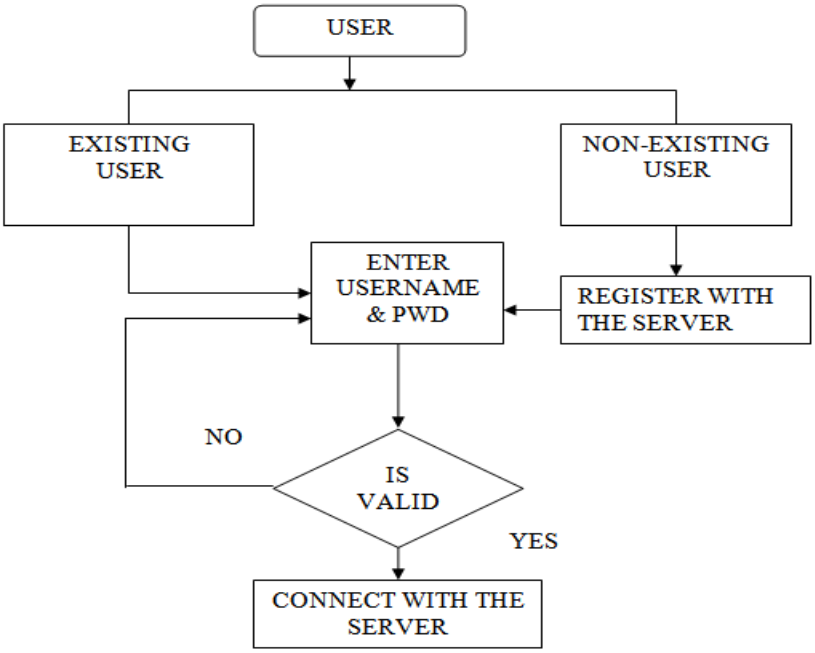

Fig. 5.1: Client Module

b) System Module: Representative network architecture for cloud data storage is illustrated in Figure 1. Three different network entities can be identified as follows:

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.

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.

Cloud data storage Module: Cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in a simultaneous, the user interacts with the cloud servers via CSP to access or retrieve his data. In some cases, the user may need to perform block level operations on his data.. users should be equipped with security means so that they can make continuous correctness assurance of their stored data even without the existence of local copies. In case that users do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the tasks to an optional trusted TPA of their respective choices. In our model, we assume that the point-to-point communication channels between each cloud server and the user is authenticated and reliable, which can be achieved in practice with little overhead.

c) Cloud Authentication Server: The Authentication Server (AS) functions as any AS would with a few additional behaviors added to the typical client-authentication protocol. The first addition is the sending of the client authentication information to the masquerading router. The AS in this model also functions as a ticketing authority, controlling permissions on the application

network. The other 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

d) Unauthorized data modification and corruption module: One of the key issues is to effectively detect any unauthorized data modification and corruption, possibly due to server compromise and/or random Byzantine failures. Besides, in the distributed case when such inconsistencies are successfully detected, to find which server the data error lies in is also of great significance

e) Adversary Module: Security threats faced by cloud data storage can come from two different sources. On the one hand, a CSP can be self-interested, untrusted and possibly malicious. Not only does it desire to move data that has not been or is rarely accessed to a lower tier of storage than agreed for monetary reasons, but it may also attempt to hide a data loss incident due to management errors, Byzantine failures and so on.

Weak Adversary: The adversary is interested in corrupting the user's data files stored on individual servers. Once a server is comprised, an adversary can pollute the original data files by modifying or introducing its own fraudulent data to prevent the original data from being retrieved by the user.

Strong Adversary: This is the worst case scenario, in which we assume that the adversary can compromise all the storage servers so that he can intentionally modify the data files as long as they are internally consistent. In fact, this is equivalent to the case where all servers are colluding together to hide a data loss or corruption incident.

VI. ADVANTAGE

a) Storage and Scalability

b) Backup and Disaster Recovery

c) Cost Efficiency

d) Server operates Automatically, No need to handle to person.

VII. APPLICATIONS

a) Data Storage security.

b) Banking.

VIII. CONCLUSION

A. *Concluding Remarks*

In this paper, we investigated the problem of data security in cloud data storage, which is essentially a distributed storage system. To ensure the correctness of users' data in cloud data storage, we proposed an effective and flexible distributed scheme with explicit dynamic data support, including block, update, delete, and append. We rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. By utilizing the homomorphism token with distributed verification of erasure coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the distributed servers, we can almost guarantee the simultaneous identification of the misbehaving server(s). Through detailed security and performance analysis, we show that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks.

References

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

2. H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. of Asiacrypt '08, Dec. 2008.

3. 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/.

4. 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.

5. G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. of Secure Comm'08, pp. 1– 10, 2008.

6. T. S. J. Schwarz and E. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc.

7. Amazon.com, "Amazon Web Services (AWS)," Online at http://aws. ama-zon.com, 2008. N. Gohring, "Amazon's S3 down for several hours.

8. http://www.pcworld.com/businesscenter/article/142549/amazo s s3 down for several hours.html, 2008. K. D. Bowers, A. Juels, and A. Oprea.

9. Availability and Integrity Layer for Cloud Storage," CryptologyePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.

10. L. Carter and M. Wegman, "Universal Hash Functions," Journal of Computer and System Sciences , vol. 18, no. 2, pp. 143–154, 1979.

11. J. Hendricks, G. Ganger, and M. Reiter, "Verifying Distributed Erasurecoded Data," Proc. 26th ACM Symposium on Principles of Distributed Computing, pp. 139–146, 2007.

AUTHOR(S) PROFILE

Mr. Hemant T. Dhole currently pursuing his B.E degree in Computer Engineering from Jaihind College of Engineering, Kuran (University of Pune). Also received the Diploma in Information Technology from Jaihind Polytechnic, Kuran (MSBTE) in 2011.

Mr. Praful C. Papade currently pursuing his B.E degree in Computer Engineering from Jaihind College of Engineering, Kuran (University of Pune). Also she received the Diploma in Computer Engineering from Jaihind Polytechnic, Kuran (MSBTE) in 2011.

Prof. Sachin B. Bhosale currently pursuing his M.Tech degree in Computer Science and Engineering from Ellenki College of Engineering,Sidhipeth (JNTU). Also he received the Diploma in Computer Engineering from Jaihind Polytechnic, Kuran (MSBTE) in 2009.

