ISSN:0975-9646
Chayashree G/ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015, 3659-3662

Ensuring Secure Data Communication Cloud Environment

Chayashree G
Asst. professor ISE Dept.
GSSSIETW, Mysore, India

***Abstract*—Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' physical possession of their outsourced data, which inevitably poses new security risks toward the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, a flexible distributed storage integrity auditing mechanism, utilizing the homomorphism token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result ensures strong cloud storage correctness guarantee.**

Index Terms*— Cloud, AES, Erasure, integrity *(key words)

I. INTRODUCTION

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. The project is focused 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, an effective and flexible distributed scheme with two salient features is proposed, 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.

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 centres 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 centres. 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) [2] 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 (CSP) for the availability and integrity of their data [3].

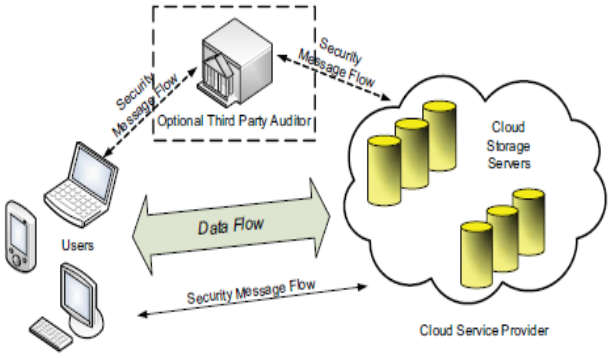

Fig 1 Cloud Architecture

II. PROBLEM STATEMENT

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. 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. 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.

III. RELATED SOLUTIONS

Cong Wang, Qian Wang and Kui Ren- Ensuring Data Storage Security in Cloud computing

In this they have addressed the security issues associated in cloud data storage and have explored many security issues, whenever a data vulnerability is perceived during the storage process a precision verification across the distributed servers are ensured by simultaneous identification of the misbehaving nodes through analysis in term of security malfunctioning, it is proved that their scheme is effective to handle certain failures, malicious data modification attack, and even server colluding attacks. This new technology opens up a lot of new security issues leading to unexpected challenges which is of dominant importance as security is still in its infancy now many research problems are yet to be solved and identified.

Balachandra Reddy Kandukuri, Ramakrishna Paturi V, Dr.Atanu Rakshit- Cloud security Issues

Security Content Automation Protocol (SCAP) and the benefits it can provide to cloud and tools for system security such as patch management and vulnerability management software, use proprietary formats, nomenclatures; measurements, terminology and content. It has been mentioned that the lack of interoperability causes delays in security assessment was addressed.

Siani Pearson-Taking account of Privacy when Designing Cloud Computing Services CLOUD'09

It has been described about the overview of privacy issues within cloud computing and a detailed analysis on privacy threat based on different type of cloud scenario was explained, the level of threat seem to vary according to the application area. Their work has stated the basic guidelines for software engineers when designing cloud services in particular to ensure that privacy are not mitigated. The major focus of their schemes rests on the privacy risks, analysis on privacy threats, privacy design patterns and accountability with in cloud computing scenario.

Meiko Jensen, Jorg Schwenk, Nils Gruschka, Luigi Lo Iacono- On technical security issues in cloud computing

In it clearly stated about the issues associated in choosing a security mechanisms or security frameworks in the Cloud computing context and given a brief outline on flooding attacks. Also they have given an idea about, the threats, their potential impact and relevance to real-world cloud environment. It is well understood from their investigation, a significant pace for improving data security in cloud is to initial intensification of the security competence of both web applications and frameworks.

IV. SYSTEM DESIGN

The proposed system provides a secure layer for ensuring secure data communication in cloud environment. The proposed system will be having three modules and they are:

1. Data owner
2. Broker and
3. Cloud storage

Data owner: Data owner is an entity, who has data to be stored in the cloud and relies on the cloud for data storage and computation, can be either enterprise or individual customer. Data owner will be authenticating using regular login form, ie, username and password. User will be provided with uploading the file, downloading the file, calculating the cost of uploading data into the cloud and can audit the cloud data which he has uploaded.

Broker: Broker is an entity, which is an intermediate between the data owner and the cloud storage. Broker can also be called as Cloud Service Provider. It provides data storage services and has significant storage space and computation resources. Broker will do all security issues for the secure data communication between users and the cloud storage. Architecture of broker module and the security layer are shown below.

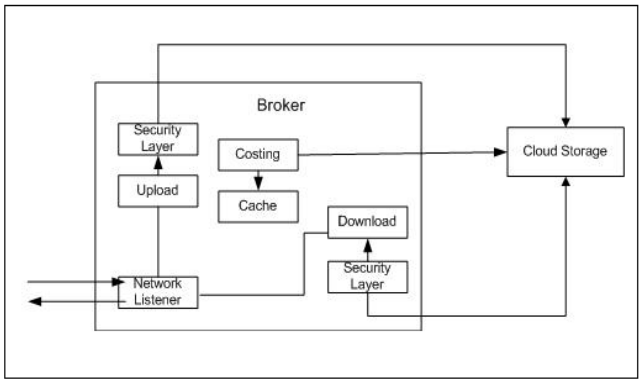

Fig 2 Broker Module

Secure layer contains eraser share split module, block hash generator and hash repository to store the hash key. When a file has been uploaded the eraser code will split the file into number of shares and encrypt the data within it. These encrypted data will be reassembled and further sent to block hash generator to generate hash key for each of the shares given to it. The hash keys generated will be stored for further integration check in the future.

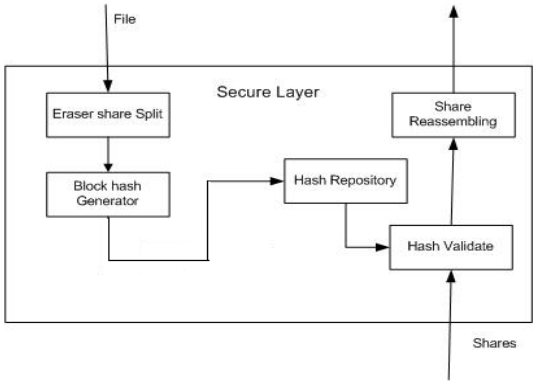

Fig 3. Secure Layer

Following are the algorithms used in our project for security. AES algorithm is used for authentication during login and Eraser code algorithm for splitting data and then encrypting it.

1) AES algorithm

i. **Key Expansion** - round keys are derived from the cipher key using Rijndael's key schedule

ii. **Initial Round** AddRoundKey - each byte of the state is combined with the round key using bitwise xor

iii. **Rounds**

a) SubBytes - a non-linear substitution step where each byte is replaced with another according to a lookup table.

b) ShiftRows - a transposition step where each row of the state is shifted cyclically a certain number of steps.

c) MixColumns - a mixing operation which operates on the columns of the state, combining the four bytes in each column.

d) AddRoundKey

iv. **Final Round** (no MixColumns)

a) SubBytes

b) ShiftRows

c) AddRoundKey

v. **Key generation**

This module handles key generation by the server side. The server generates unique keys for users once they authenticate themselves with the server. The key is generated using instances of AES key generator class. This key is then transferred to the cloud client via the LAN Wi-Fi connection which receives and stores a copy for it for decrypting purpose. The key is a 16 byte or a 128 bit key.

An Example of a key generated is: 8xRER4LyFiU3Hs9a40xExQ== after the key generation and encryption, the cipher text is sent to the client, the client uses the reverse process of the AES encryption. Decryption to obtain the original plaintext that was transferred by the server. Hence the client receives intended file in a secure manner over the LAN.

2) Erasure code algorithm

An **erasure code** is a forward error correction (FEC) code for the binary erasure channel, which transforms a message of k symbols into a longer message (code word) with n symbols such that the original message can be recovered from a subset of the n symbols. The fraction $r = k/n$ is called the code rate, the fraction k'/k, where k' denotes the number of symbols required for recovery, is called reception efficiency.

Optimal erasure codes have the property that any k out of the n code word symbols are sufficient to recover the original message (i.e., they have optimal reception efficiency). Optimal erasure codes are maximum distance separable codes (MDS codes). Optimal codes are often costly (in terms of memory usage, CPU time, or both) when n is large. Except for very simple schemes, practical solutions usually have quadratic encoding and decoding complexity. Using FFT techniques, the complexity may be reduced to $O(n \log(n))$; however, this is not practical.

1) Parity check

Parity check is the special case where $n = k + 1$. From a set of k values $\{v_i\}_{1 \le i \le k}$, a check-sum is computed and appended to the k source values:

$$v_{k+1} = -\sum_{i=1}^{k} v_i .$$

The set of $k + 1$ value $\{v_i\}_{1 \le i \le k+1}$ is now consistent with regard to the check-sum. If one of these values, v_e, is erased, it can be easily recovered by summing the remaining variables:

$$v_e = -\sum_{i=1, i \ne e}^{k+1} v_i .$$

V. CONCLUSION & FUTURE SCOPE

This project 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, the proposed system provides an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. It relies 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, it can be almost guarantee the simultaneous identification of the misbehaving server(s). Through detailed security and performance analysis, that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks.

The data storage security in Cloud Computing, an area full of challenges and of paramount Importance, is still in its infancy now, and many research problems are yet to be identified. The most promising one that believe is a model in which public verifiability is enforced. Public verifiability, allows TPA to audit the cloud data storage without demanding users' time, feasibility or resources. Besides, along with our research on dynamic cloud data storage, there is a future plan to investigate the problem of fine-grained data error localization.

REFERENCES

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

[2] 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.

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

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

[5] 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/.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] 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/.
[11] L. Carter and M. Wegman, "Universal Hash Functions," *Journal of Computer and System Sciences*, vol. 18, no. 2, pp. 143–154, 1979.
[12] J. Hendricks, G. Ganger, and M. Reiter, "Verifying Distributed Erasurecoded Data," *Proc. 26th ACM Symposium on Principles of Distributed Computing*, pp. 139–146, 2007.
[13] 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.
[14] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," *Proc. of IEEE INFOCOM*, 2009.
[15] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple- Replica Provable Data Possession," *Proc. of ICDCS '08*, pp. 411–420, 2008.
[16] 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/.
[17] 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.