International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 3 | ISSN : 2456-3307

Privacy Preserving In Cloud Using Dual Protection

C. Bhagya Lakshmi[1], K. Gomathi[1], P. Manisha[1], Mrs. L. Sharmila[2]
[1]UG Student, CSE , Alpha College of Engineering, Chennai, Tamilnadu, India
[2]Asst.Professor, CSE, Alpha College of Engineering, Chennai, Tamilnadu, India

ABSTRACT

Security in cloud is one of the important factor in cloud, here we can preserve our data into cloud storage. More number of clients will like to store their data to PCS (public cloud servers) along with the rapid development of cloud computing. To save local data storage costs the data is outsourced to cloud servers by cloud storage services. Multiple verification tasks can be performed efficiently by by the auditor from different users and the cloud-stored data can be updated dynamically. It allows the clients to check whether their data which has been outsourced is kept intact without downloading the whole data. In our system own auditing is used based on the token generation. Using this key generation technique compare the key values from original keys we can find out the changes about the file. We are using novel public verification scheme for the cloud storage using in distinguish ability obfuscation, which requires a lightweight computation on the auditor and delegate most computation to the cloud. Apart from storing ,the content will be encrypted in the cloud server. If someone tries to hack at the cloud end,it is impossible to break the two different blocks. The security mechanism of our scheme is under the strongest security model. They needto decrypt the files first and then combine the splitted files from three different locations. This is not possible by anyone. With the file owner permission anyone can download the files from the server. At the time of download,key will be generated (code based key generation) and it will be sent to the file owner. We can download the file need to use the key for verification and some other users want to download file owner permission is necessary.

Keywords : PCS, Cloud Servers, Cloud Storage Services, Key Generation Technique

I. INTRODUCTION

Cloud storage services is used to outsource the users data to cloud servers and access the outsourced data remotely from a variety of places and devices (e.g. Drop box [1], One Drive [2], and Google Drive [3]). Such services provide users with an efficient and flexible way to manage their data without deploying and maintaining the local storage device and service [4], [5], [6], [7]. Some recent reports indicate that more than 79% of organizations attempt to utilize data outsourcing and such increasing demand of the cloud storage service leads to the growing number of cloud storage providers[8],[9],[10]. While people enjoy the desirable benefits from the cloud storage service, critical security concerns in data outsourcing have been raised seriously. The most important security concern is data integrity. Since users do not own their data physically when once the data is outsourced to cloud servers, they always worry about the data integrity, i.e. whether their data remains faultless on the cloud servers. Checking for integrity on users data can be performed by a cloud server, however a good integrity report is always generated by the cloud server for good reputation even if some data are damaged or missing [11], [12], [8]. Users should be proficient to prevent the server from cheating. In addition, an external contender may warp users' data

on the cloud servers for financial or political reasons [13], [14]. So an efficient and secured verification method is often required for the users to ensure the integrity of their data. Some data verification schemes rely on users themselves to execute the verification [11], [12], [15], [16]. This means that a user have some additional communication and computation costs for data integrity verification. As a result of this, the user is required to bear heavy communication and verification burden to retrieve and use the data. To reduce the verification burden on users, a public verification paradigm has been proposed. An external and independent auditor is employed to verify the data integrity on behalf of users periodically[17], [6], [18],[19],[20], [21]. As a result, users will be free from the verification burden while the auditor needs to be equipped with strong computation capability for verification. In existing public verification schemes, the computation overhead of verification by the auditor linearly increases with the size of the verified data set. If the verification is required to be executed frequently for multiple users' data sets, the auditor will need a huge computation capability to accomplish the verifications and the verification delay will be huge. The deployment of such auditor is indeed a difficult problem. Therefore, for reducing the computation overhead and delaying on the auditor side has a significant value to make the verification scheme efficient and practical. Some public verification schemes achieve batch verification, where multiple delegated verification tasks from different users can be performed simultaneously by the auditor [18], [19]. However, the batch verification overhead in the current schemes is still linearly increasing with the number of users. Consequently, if the auditor is equipped with a constrained device and the verifications are required to be executed frequently, the verification may incur a huge delay and become a bottleneck in applications. For example, for the public verification scheme [19], even though the auditor is equipped with an Intel Core 2 processor running at 1.86 GHz, 2,048 MB of RAM, let the size of the verified data set be 300, when the number of verification tasks (i.e. the number of users) increases to 100, the verification delay is about 30 seconds. And if the number of verification tasks increases to 1000, the verification delay is about 300 seconds.

II. PROPOSED SYSTEM

A. Overview

An efficient distributed scheme with data in the cloud is been made. Here we are using the erasure code technique for distribute the data to cloud locations and access the data from cloud. User can register and login into their account. Provided an option to store, share and access the data from cloud storage. Here we are using the double ensured scheme for storing data into the cloud. First is your data or file splited into multiple parts and it will store into different cloud server locations. Each and every file generates the key-code for auditing. Then second is each and every splited file will encrypt before store into different locations. The shared users can edit the file in the cloud with file owner's permission. That file eligible of own public auditing. Search and download the files, at the time of download user should use the security key. As an authentication success it will be decrypt and combine to get the original data from cloud. Moreover, we design a novel public verifiable authenticator, which is generated by a couple of keys and can be regenerated using partial keys. Thus, our scheme can completely release data owners from online burden. In addition, we randomize the encode coefficients with a pseudorandom function to preserve data privacy. Extensive security analysis shows that our scheme is provable secure under random oracle model and experimental evaluation indicates that our scheme is highly efficient and can be feasibly integrated into the regenerating code- based cloud storage.

B. Architecture Diagram

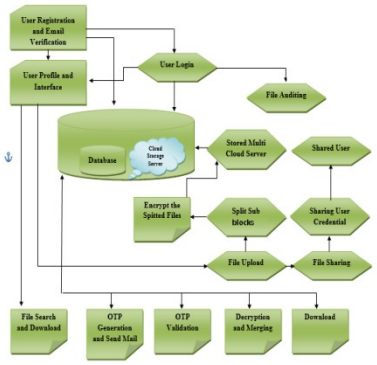

Figure 1

c.Algorithms and techniques

a) *Steps for Secure Erasure Code Technique:*

Step 1: Given a signal of m blocks, recode to n

- Blocks where n > m
- Optimal: reconstruct signal given any m unique blocks

Step 2: Suboptimal: Reconstruct signal using (1+e)m unique blocks

Rate r=m/n, and storage overhead is 1/r.

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).

Step 3: A cloud storage system, consisting of a collection of storage servers, provides long-term storage services over the Internet. Storing data in a third party's cloud system causes serious concern over data confidentiality.

Step 4: General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data.

Step 5: 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 checksum is computed and appended to the k source values:

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

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

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

b) DES Algorithm

Encryption has become a part and parcel of our lives and we have accepted the fact that data is going to encrypted and decrypted at various stages. However, there is not a single encryption algorithm followed everywhere. There are a number of algorithms existing, and I feel there is a need to understand how they work. So this text explains a number of popular encryption algorithms and makes you look at them as mathematical formulas

c) Data Integrity Checksum Algorithm:

Data integrity refers to maintaining and assuring the accuracy and consistency of data over its entire life-cycle, Data integrity is the opposite of data corruption, which is a form of data loss. The overall intent of any data integrity technique is the same: ensure data is recorded exactly as intended (such as a database correctly rejecting mutually exclusive possibilities,) and upon later retrieval, ensure the data is the same as it was when it was originally recorded. In short, data integrity aims to prevent unintentional changes to information. Data integrity is not to be confused with data security, the discipline of protecting data from unauthorized parties.

A checksum or hash sum is a small-size datum from a block of digital data for the purpose of detecting errors which may have been introduced during its transmission or storage. It is usually applied to an

installation file after it is received from the download server. By themselves checksums are often used to verify data integrity, but should not be relied upon to also verify data authenticity.

III. CONCLUSION

A privacy-preserving public auditing system for data storage security in computing. We utilize the homomorphism linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the server during the efficient auditing process, which not only eliminates the burden of user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multiuser setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency.

IV. REFERENCES

[1]. Prof H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 3, pp. 312-325, 2016.

[2]. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proceedings of ESORICS. Springer, 2009, pp. 355-370.

[3]. Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, 2011.

[4]. M.Sookhak,A.Gani,H.Talebian,A.Akhunzada,S. U.Khan,R.Buyya, and A. Y. Zomaya, "Remote data auditing in cloud computing environments: A survey, taxonomy, and open issues," ACM Computing Surveys, vol. 47, no. 4, 2015.

[5]. H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: When qoe meets qop," IEEE Wireless Communications, vol. 22, no. 4, pp. 74-80, 2015.

[6]. A. Juels and B. S. K. Jr, "Pors: Proofs of retrievability for large files," in Proceedings of CCS. ACM, 2007, pp. 583-597.

[7]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of CCS. ACM, 2007, pp. 598-609.

[8]. J. Ni, Y. Yu, Y. Mu, and Q. Xia, "On the security of an efficient dynamic auditing protocol in cloud storage," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 10, pp. 2760-2761, 2013.

[9]. Y. Zhang, C. Xu, J. Zhao, X. Zhang, and J. Wen, "Cryptanalysis of an integrity checking scheme for cloud data sharing," Journal of Information Security and Applications, vol. 23, pp. 68-73, 2015.

[10]. E. Shi, E. Stefanov, and C. Papamanthou, "Practical dynamic proofs of retrievability," in Proceedings of CCS. ACM, 2013, pp. 325-336.

[11]. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proceedings of INFOCOM. IEEE, 2010, pp. 1-9.

[12]. C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy preserving public auditing for secure cloud storage," IEEE Transactions on Computers, vol. 62, no. 2, pp. 362-375, 2013.

[13]. H. Shacham and B. Waters, "Compact proofs of retrievability," Journal of Cryptology, vol. 26, no. 3, pp. 442-483, 2013.

[14]. Y. Zhang, C. Xu, H. Li, and X. Liang, "Cryptographic public verification of data integrity for cloud storage systems," IEEE

Cloud Computing, vol. 3, no. 5, pp. 44-52, 2016.

[15]. R. C. Merkle, "Protocols for public key cryptosystems," in Proceedings of S & P. IEEE, 1980, pp. 122-134.

[16]. J. Katz and Y. Lindell, Introduction to Modern Cryptography. CRC Press, 2014.

[17]. C. Gentry, A. Lewko, A. Sahai, and B. Waters, "Indistinguishability obfuscation from the multilinear subgroup elimination assumption," in Proceedings of FOCS. IEEE, 2015, pp. 151-170.

[18]. D. Boneh and M. Zhandry, "Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation," in Proceedings of CRYPTO. Springer, 2014, pp. 480-499.

[19]. C. Baun, I. Damg˚ard, and C. Orlandi, "Publicly auditable secure multiparty computation," in Proceedings of SCN. Springer, 2014.