Reducing Key Leakage by Outsourcing Key Updates in Cloud Storage

[1]Shilpa S.B, [2]Srinivasachar G

[1]PG student, [2]Assistant Professor
Computer Science and Engineering
Atria Institute of Technology, Bengaluru, India

***Abstract*— Key leakage impedance has been an important problem in numerous safety applications. To tackle this challenge, current results all need the client to revise his mystery key in each time point, which delivers new load to the client. To overcome from this problem, making key revise as clear as feasible for the client and propose a concept called reducing key leakage by giving key revise to certified party(TPA). In this concept outsourcing key revise operations to certified party, so that key revise problem on the client can be negligible. In this propose, TPA acts as certified party and he is in charge for both the files auditing and the safe key revise for key leakage impedance.**

***Index Terms*— Cloud storage, third party Auditor, key revise, outsourcing computation.**

I. INTRODUCTION

Cloud Computing is a high technical and communal reality and a rising technology. The cloud provides the delusion of endless computing resources. Cloud computing and storage solutions offer many capabilities to store up and process the data in confidentially owned with users and enterprises. Cloud storage is made up of numerous scattered resources. Although cloud storage serves many uses to users, it furnishes new safekeeping troubles. Fig. 1 shows the cloud data storage architecture. Cloud storage allows data owners to upload files to cloud so that clients will have rights to use those records from anywhere and allows data customers to access files from the cloud but it brings new security challenges.

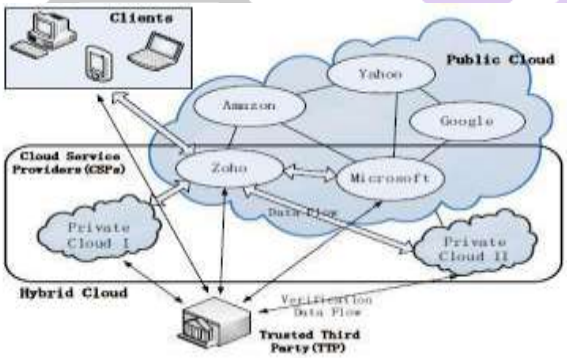

Figure 1 Architecture of cloud data storage

Notwithstanding the way that disseminated stockpiling gives immense favorable position to clients, it delivers fresh security problems. One fundamental safekeeping problem is that the methods by which to effectively verify the reliability of the data placed in cloud. In cutting edge years, a variety of assessing methods used for disseminated capacity has been introduced to tackle the problems. These methods concentrate on different category of circulated capacity inspecting.

The key leakage issue is the major problem in cloud storage auditing. Cloud storage auditing is to check the integrity of the files placed in cloud. If the client's mystery key is disclosed to cloud for auditing of files, the cloud can effortlessly shield the data loss events. In most modern duration, audit protocols in cloud storage have been engrossed more concentration. Using cloud storage audit protocol for key leakage impedance, the spoil of the client's secret key can be reduced by securely updating the user's mystery key in every time point. The client will update the encrypted mystery key in every time point and also client will audit the data saved in cloud are right. But it delivers new trouble for the client because client has to update the key in every time period, this might be the problem for some clients with less computation resources. So, it would be more attractive to make key revise as clear as feasible to the client. To accomplish this aim, need to outsource the key updates to authorised party.

To accomplish this objective, some requirements need to be satisfied. First, the client's actual secret key must not be known by the third party auditor (authorized party) who is responsible for performing outsourcing computations for key updates in cloud storage auditing. Otherwise, it will bring new security issues. Second, the authorized party is doing key updates computation he only knows the encrypted secret keys and key updates have to finish under the encrypted state only. Third, the client should be

able to pick up the actual secret key from the encrypted state. Lastly, when client gets encrypted secret key from the authorized party, client must be capable of verifying the validity of the encrypted secret key.

To accomplish the outsourcing of key updates, need to build a cloud storage space auditing protocol for outsourcing key revise to some authorized party that should assure above requirements. The entire existing audit protocols are constructed on the hypothesis may not always be held, because of the less safety environment at the client. Few existing audit protocols could surely become incapable to job when mystery key is uncovered for files audit.

In the proposed protocol called cloud storage audit for outsourcing key revise. In this protocol, key revise operations are achieved by an authorized party not by the client. While outsourcing key updates to authorized party, it relaxes the owners of data from the pressure of key updates and safeguarding. Once the client gets the encrypted mystery key from TPA, client can decrypts it whenever he wants to send files or data to cloud. With this, client can also authenticate the validity of the encrypted mystery key.

In the plan of cloud storage audit protocol for outsourcing key revise, the third party auditor who plays the character of authorized party in key updates, with this, TPA is also responsible to certify the integrity of the client files saved in cloud. Authorized party will never know the actual mystery key of the client for storage auditing. In this protocol, blinding technique with homomorphic property is used to encrypt the secret keys held by the third party auditor. The blinding technique allows key updates operations to be carried under the blinded version. This technique makes protocol protected and the decryption process efficient. Homomorphic encryption is a type of encryption which allows definite kinds of calculations to be passed out on ciphertexts and produce an encrypted outcome which, when decrypted, compare the outcome of operations accomplished on the plaintexts.

The third party auditor can be measured as an authorized party with more dominant computational capacity or a service in a different self-governing cloud. When the client stores file in cloud, the complete life span of the file which is saved in cloud is separated into T+1 time points. Each file is assumed to be partitioned into several chunks. Furthermore, each chunk is not partitioned into several sectors in the explanation of our protocol. At the end of each time point, the third party auditor updates the encrypted client's secret key according to the next time point for cloud storage auditing. But the public key will not get changed in entire time period. Whenever the client wants to upload new files to cloud, client asks third party auditor to generate secret key by sending key requirement to TPA.

When the TPA gets key requirements from the client, the third party auditor generates encrypted secret key and transmit the encrypted mystery key to the client. The third party auditor will complete key update operations under the encrypted state only. Once the client gets encrypted secret key, he can decrypts it to get his actual secret key, produces authenticators for files. When client gets encrypted secret key from the TPA, client will authenticate the validity of the encrypted secret key. The third party auditor will inspect (audit) the files in cloud are saved properly between it and the cloud at regular time by a challenge-response protocol.

In the composed System Setup calculation, the TPA just holds an underlying encoded mystery key and the client holds a decoding sort which is utilized to unscramble the encoded mystery key. In the outlined Key update calculation, homomorphic property makes the mystery key ready to be refreshed under encoded state and makes checking the scrambled mystery key conceivable. The VerESK calculation can make the client check the legitimacy of the encrypt mystery keys quickly.

The cloud is extraordinary for putting away non-touchy data, as schedules on stages like Evernote. In any case, obviously, putting away individual data some place "up in the cloud" makes many individuals watchful. In this project, public cloud is used. The open cloud is generally the internet. Facility providers make use of internet to construct resources, such as applications and storage. Amazon EC2, sun cloud is the examples of public clouds. In this project, Drive HQ public cloud is used.

II. **LITERATURE SURVEY**

Before you begin to format your paper, first write and save the Various methodologies are executed for secure outsourcing of calculations. Atallah [1] proposed a structure for veiling logical computations and their costs, numerical properties and levels of security. Creators demonstrated that there is nobody cloak procedure is proper for an expansive scope of logical estimations however there is a variety of cover systems available so that generally any logical calculation could be camouflaged at a satisfactory cost and with towering level of wellbeing. Creators likewise centered around an issue where the unraveling attempt is extensive contrasted with the issue mass and they additionally focused on scientific critical thinking and assorted numerical plans.

Zhang et al [2] introduced a scheme for a class of homomorphic functions which is called generic computation outsourcing scheme. To make a brief and protected generic calculation outsourcing system they apply the intrinsic property. There is no use of public key computations in this scheme. According to the input, output privacy and security definitions of verifiability they give the security proofs of this scheme. The security proof is never depends on any computational guesses rather than the security proof is data-theoretic.

Ateniese et al.[3] proposed a representation for "Provable data possession(PDP)" where it is more attractive to reduce the file chunk access, the calculation on the slave(server), and the client-slave(server) message. It also permits the client who has saved

files at an unprocessed server to authenticate that the server acquires the actual files with no access. The representation produces probabilistic evidence of acquiring by variety arbitrary set of chunks from the slave (server), which radically minimizes input and output costs.

C. Wang [4] projected a concept for safe cloud data storage space called public privacy-preserving auditing protocol. To ensure that the certified party(TPA) would never know the details about the files which placed on the cloud during the auditing course they apply the homomorphic linear authenticator and random masking, which reduce the difficulty of cloud customer from the costly auditing work and it also mitigates the users dread of their outsourced data leakage.

K. Yang and Jia [5] proposed secure audit protocol for cloud data storage space. Instead of using the mask techniques, they utilize the cryptography concept with the bilinearity farm of bilinear paring to guard the files confidentiality against the third party auditor. So, multiple-cloud group auditing protocol never require any extra arranger. Batch auditing can also be possible for multiple owners in batch auditing protocol. In this auditing scheme, by giving the lots of audit work from the auditor(TPA) to the server, communication cost and the computation cost of the auditor would be less.

B. Wang[6] proposed public auditing secrecy storing for mutual data in cloud storage. To build homomorphic authenticators, they apply the ring signatures so that the auditor(TPA) can be able to audit the reliability of mutual data. With this mechanism, TPA does not know the signer's identity on each chunk in shared data, but still TPA can be able to validate the reliability of the mutual data without acquiring the entire file.

B. Wang[7] proposed public auditing with efficient user repeal for shared data in cloud. In audit mechanism, when one person in the group is removed, cloud allows to re-sign chunks with proxy re-signatures which were signed by the removed user. In this scheme, without accessing the complete information from the cloud, the universal verifier can be able to check the truthfulness of the mutual files which has re-signed by the cloud.

M. Sookhak [8] proposed remote data auditing dynamically in cloud storage. To verify the truthfulness of the data saved in cloud and to minimize the computational burden on the client and server, this method employs algebraic signature properties for cloud storage system. Authors also designed divide and conquer table data organization, where owners of data can make modifications, deletion, insertion and can add computations at chunk stage without necessarily downloading the complete file.

Q.Wang[9] proposed a method called enabling open audit capacity and data passage for storage safety in cloud to achieve capable information sequence, they improve the current verification of capacity models by regulating the exemplary merkle hash tree (MHT) progress for piece marker validation and to bolster productive treatment of different reviewing assignments.

III. System Architecture

In our configuration there are three individuals in the framework: the client, the cloud and outsider reviewer (TPA) as shown in the Fig. 2. The client is the information owner of the documents which are saved in cloud. The document size is not settled, i.e. customer can transfer any measure of documents to cloud at various eras. The cloud can store any number of client's records and it additionally enables client's to download the documents from the cloud. The Third party evaluator (Authorized gathering), he is the person who will review the records spared in cloud for the client and he additionally refresh the encoded mystery keys of the client in each time focuses. At whatever point client needs to transfer the new information documents to cloud, he sends the key prerequisites to outsider inspector. At that point the TPA conveys the scrambled mystery key to the client. From that point onward, the client decodes it to take his genuine mystery key, figures set of authenticators for information documents, and transfers these information records alongside authenticators to cloud.

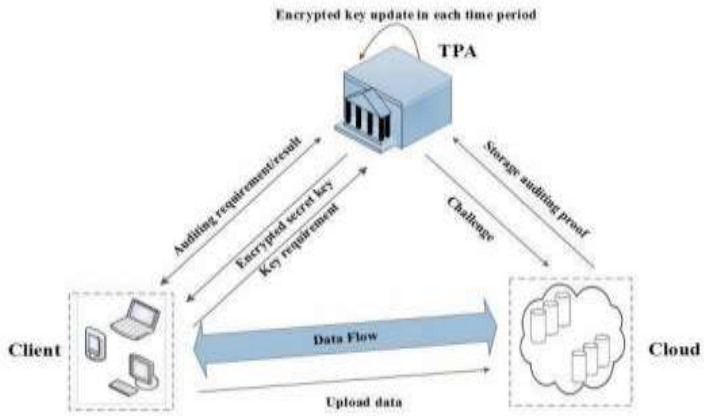

Figure 2: System Architecture

Functions

To reduce leakage problem in cloud storage auditing protocol by securely outsourcing key updates to authorized party is composed by seven functions. They are SysSetup, EkeyUpdate, VerESK, DecESK, AuthGen, Proof-Gen, ProofVerify. The description is given below.

1. SysSetup(k,t):The system setup is race by the client.
Input: safety parameter k, Time t. Read security parameter k as a user id, Read time t as a client registered time, Encrypt using triple DES(k, t). Generate ESK, DK, PK and Send ESK0 to TPA. Output: Encrypted secret key ESK0, Decryption key DK, Public key PK.
2. EkeyUpdate(ESKj, t): The encrypted key update algorithm is run by the TPA
Input: encrypted secret key ESK, TPA received time t. Output: Updated key ESKJ+1, for the current time. TPA reads the ESK. Generate ESKj+1.
3. VerESK(ESKJ+1, PK): The encrypted key verifying algorithm is race by the client.
In VerESK, checking the uniqueness of key. Input: Client ESKj+1, PK. Output: Verified result. Client reads ESKJ+1 and current time. Verify with ESKJ with the client public key PK, if(ESKJ==ESKJ+1) then return 1;else return 0.
4. DecESK(ESKJ, DK, PK, t): The secret key decryption algorithm is run by the client.
Input: input ESKJ, decryption key DK, public key PK. Output: Generate real client secret key SK. Read current encryption key, client DK, PK. Decrypt using triple DES(ESKJ,DK,PK). Return SKJ.
5. AuthGen(F,SKJ,PK):The authenticator generation algorithm is run by the client
Input: File F, client secret key SKJ, public key PK. Output: Generate authentication to a access the file System generates authentication for the file access.
6. ProofGen(F, a, chal, PK):The proof generation algorithm is run by the cloud.
Input: File F, authenticators a, a challenge chal, public key PK. Output: Generate proof p. Read file F and authenticator a, public key PK by cloud. Cloud generates the proof p for the files.
7. ProofVerify(P, Chal, PK):The proof verifying algorithm is race by the certified party.
Input: proof p, a challenge Chal, open key PK. Output: verified results by TPA.

Problem Statement

The key leakage impedance is the foremost problem in variety of cloud safety applications. All existing results all need the client to revise his mystery key in every time point, which delivers new pressure for the client. The proposed concept called reducing key leakage by outsourcing key revise to certified party. In this concept, outsourcing key revise operation to TPA so that we can reduce the key revise problem on the client and the certified party will also check the files saved in cloud are right.

Existing system Disadvantages

- Current courses of action all need the client to upgrade his mystery keys in every time, which delivers new pressure to the client.
- Third gathering have to carry the mixed interpretation of the client's mystery key while performing all the troublesome coursework for the advantage of client. A client simply can download the encoded mystery key from TPA while exchanging fresh records to cloud.

Proposed system Advantages

- In the cloud storage auditing protocol, key updates are outsourced to the TPA and are transparent for the client.
- The TPA only sees the encrypted version of the client's secret key, while the client can additionally check the validity of the encoded mystery keys when getting them from the TPA.

IV. Conclusion

Key updates outsourcing for cloud storage auditing with key-leakage impedance have studied and an effectual protocol for reducing key leakage by outsourcing key updates to cloud storage. In this protocol, outsourcing key updates to the TPA and making key updates transparent to the client. With this, the TPA can only hold the encrypted version of the data owner's mystery key and the validity of encrypted secret keys are verified by the client when downloading them from the third party auditor.

References

[1] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. E. Spafford, "Secure outsourcing of scientific computations," *Adv. Comput.*, vol. 54, pp. 215–272, 2002.
[2] F. Zhang, X. Ma, and S. Liu, "Efficient computation outsourcing for inverting a class of homomorphic functions," *Inf. Sci.*, vol. 286, pp. 19–28, Dec. 2014.

[3] G. Ateniese *et al.*, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 598–609.
[4] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.
[5] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1717–1726, Sep. 2013.
[6] B. Wang, B. Li, and H. Li Oruta, "Oruta: Privacy-preserving public auditing for shared data in the cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 43–56, Jan./Mar. 2014.
[7] B. Wang, B. Li, and H. Li, "Public auditing for shared data with efficient user revocation in the cloud," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2904–2912.
[8] M. Sookhak, A. Gania, M. K. Khanb, and R. Buyyac, "Dynamic remote data auditing for securing big data storage in cloud computing," *Inf. Sci.*, Sep. 2105, doi: 10.1016/j.ins.2015.09.004.
[9] Jia Yu, Kui Ren, *Fellow, IEEE*, and Cong Wang, *Member, IEEE*, "Enabling Cloud Storage Auditing With Verifiable Outsourcing of Key Updates", IEEE transactions on information forensics and security, vol. 11, no. 6, June 2016.
[10] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 847–859, May 2011.
[11] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1167–1179, Jun. 2015.
[12] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," *J. Syst. Softw.*, vol. 113, pp. 130–139, Mar. 2016
[13] D. Benjamin and M.J Atallah, "private and cheating-free outsourcing of algebraic computations," in Proc. 6th Annu. Conf. Privacy, Secur. Trust, 2008, pp.240-245.
[14] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 820–828.
[15] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," in *Proc. 17th Eur. Symp. Res. Comput. Secur.*, 2012, pp. 541–556.
[16]A. Juels and B. S. Kaliski, Jr., "PORs: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 584–597.
[17] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2008, pp. 90–107.
[18] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw.*, 2008, Art. ID 9.
[19] F. Sebe, J. Domingo-Ferrer, A. Martinez-balleste, Y. Deswarte, and J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1034–1038, Aug. 2008.
[20] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations," in *Proc. 5th ACM Symp. Inf., Comput. Commun. Secur.*, 2010, pp. 48–59.
[21] X. Chen, J. Li, X. Huang, J. Li, Y. Xiang, and D. S. Wong, "Secure outsourced attribute-based signatures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3285–3294, Dec. 2014.