EMPHERICAL APPROACH IN DEVELOPING CLOUD STORAGE AUDITING PROTOCOL FOR SECURITY KEY UPDATES

PROF.KALAM NARREN

PROF.V.VINAY KRISHNA

Dr.CHANDRA MOHAN

[1]Research scholar, [2,3]Professor

[1,2,3] CSE Department

[1,3]JJT University, Jhunjhunu, Rajasthan, [2]Institute of Aeronautical Engineering, Hyderabad.

ABSTRACT

Key-exposure resistance is an important problem in cyber defense. In the cloud storage auditing the key exposure problem was well analyzed and studied. In the existing solution the client will be updating the secret keys in every period of time so it is very difficult for the client particularly those who are in limited computation resources. In the work, we focus on how key updates are transparent to the client and propose a new concept called cloud storage auditing with verifiable outsourcing of key updates. Key updates can be safely outsourced to some authorized party and thus the key update work burden to the client will be reduced. In this we leverage the third party auditor in many existing public auditing designs, The role of authorized party and make it in charge of both the storage auditing and the secure key updates for key-exposure resistance. In this design, TPA only needs to hold an encrypted version of the client's secret key while doing all these burdensome tasks on behalf of the client. The client only needs to download the encrypted secret key from the TPA when uploading new files to cloud. Besides, our design also equips the client with capability to further verify the validity of the encrypted secret keys provided by the TPA. All these salient features are carefully designed to make the whole auditing procedure with key exposure resistance as transparent as possible for the client. The security proof and the performance simulation show that this detailed design instantiations are secure and efficient.

Keywords- Decryption, Encryption, key update, Third party Auditor, Secret key

1.0 INTRODUCTION

Cloud computing technology is becoming more and more popular now a days. It can provide users with unlimited computing resource. people can outsource time consuming computation workloads to cloud without spending the extra capital on maintaining hardware and software. In

recent years, outsourcing computation has attracted much attention and been researched widely. It has been considered in many applications including scientific computations linear algebraic computations linear programming computations and modular exponentiation computations etc. Besides, cloud computing can also provide users with seemingly unlimited storage resource. Cloud storage is universally viewed as one of the most important services of cloud computing. Although cloud storage provides great benefit to users, it brings new security challenging problems.

One important security problem is how to efficiently check the integrity of the data stored in cloud. In recent years, many auditing protocols for cloud storage have been proposed to deal with this problem. These protocols focus on different aspects of cloud storage auditing such as the high efficiency the privacy protection of data the privacy protection of identities dynamic data operations the data sharing etc. The key exposure problem, as another important problem in cloud storage auditing, has been considered recently. The problem itself is non-trivial by nature. Once the client's secret key for storage auditing. The authorized party should only hold an encryption

The distributed storage benefit (CSS) eases the weight for capacity administration and upkeep. In any case, if such an essential administration is helpless against assaults or disappointments, it would convey hopeless misfortunes to the customers in light of the fact that their information or documents are put away in a dubious stockpiling pool outside the ventures. These security dangers originate from the accompanying reasons: it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third-party auditor (TPA) who has expertise and capable to audit the outsourced data when needed. Public audit ability allows an external party, in addition to the user himself, to verify the correctness of remotely stored data.

2.0 LITERATURE REVIEW

G. Ateniese (2007) Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. The large amount of data is stored in the cloud. To verify the integrity of a data which is stored on the cloud, the cloud storage auditing is used. Auditing is an integrity check in the cloud data base. It is an important checking in the cloud auditing protocols that are highly researched on recent years. Each protocols act as a different auditing mechanism.

The aim of introducing the protocol is to achieve high bandwidth and computation efficiency. Most of the auditing protocols are based on the assumption that the client's secret key for auditing is secure. The security is not fully achieved, because of the low security parameters of the client. If the auditing protocol is not secured means the data of the client will exposed inevitably.

G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik (2008) As Today's reality relies upon progressively refreshed data, the most ideal approach to store and refresh data is cloud storage benefit. The basic issue for putting away data in cloud storage is its security however every individual customer holds his/her own mystery key the key administration must be steady and is compelling to the client in various circumstances, so key upgrade of outsourcing is essential. The key redesigns can be dealt with by some approved controller known as TPA (Third Party Auditor) to diminish key overhaul trouble from client. It is the dependable of TPA presently, to spare key overhauls and makes key updates straightforward for customer. In existing arrangements, customer needs to refresh key without anyone else's input at occasional occasions which prompts issue for the individuals who need to focus on their principle part in the market or with the general population who have restricted assets.

F. Sebe, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater (2008) Cloud storage auditing is viewed as an important service to verify the integrity of the data in public cloud. Current auditing protocols are all based on the assumption that the client's secret key for auditing is absolutely secure. However, such assumption may not always be held, due to the possibly weak sense of security and/or low security settings at the client. If such a secret key for auditing is exposed, most of the current auditing protocols would inevitably become unable to work. In this paper, we focus on this new aspect of cloud storage auditing. We investigate how to reduce the damage of the client's key exposure in cloud storage auditing, and give the first practical solution for this new problem setting.

C. Wang, K. Ren, W. Lou, and J. Li (2010) The security issue of key exposure is one of the major problems in cloud storage auditing. To overcome this issue, initially the key-exposure resilience scheme had been proposed. However in this scheme, the data from the cloud can be illegally accessed later than the key-exposure time period using the same secret key hat had been provided for auditing the cloud data. An innovative paradigm called strong key-exposure resilient auditing for secure cloud storage which allows to set a particular time period for the key

exposure. This preserves the security of the cloud not only earlier but also later than the key exposure time period. The security proof and experimental results demonstrate that our proposed scheme achieves expected security without affecting its efficiency

3.0 METHODOLOGY

TPA only needs to hold an encrypted version of the client's secret key while doing all these burdensome tasks on behalf of the client. The client only needs to download the encrypted secret key from the TPA when uploading new files to cloud. our design also equips the client with capability to further verify the validity of the encrypted secret keys provided by the TPA. All these salient features are carefully designed to make the whole auditing procedure with key exposure resistance as transparent as possible for the client. The definition and the security model of this paradigm. The security proof and the performance simulation show that our detailed design instantiations are secure and efficient.

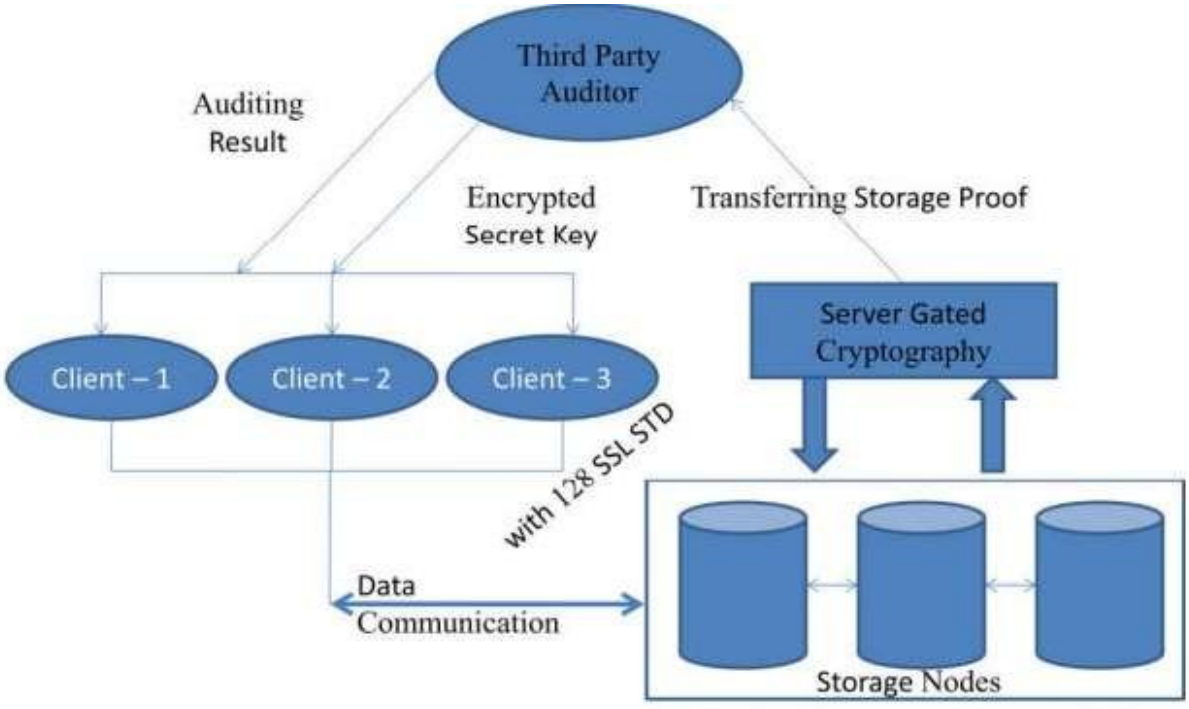

Figure Architecture Diagram

Data Producer and Retriever

The client is the owner of the files that are uploaded to cloud. The total size of these files is not fixed that is the client can upload the growing files to cloud in different time points. The cloud stores the client's files and provides download service for the client. A cloud storage auditing

protocol with secure outsourcing of key updates is composed by seven algorithms (Sys Setup, Ekey Update, Ver ESK, Dec ESK, Auth Gen, Proof Gen, Proof Verify), shown below:

1) SysSetup: the system setup algorithm is run by the client. It takes as input a security parameter k and the total number of time periods T , and generates an encrypted initial client's secret key ESK0, a decryption key DK and a public key PK. Finally, the client holds DK, and sends ESK0 to the TPA.

2) EkeyUpdate: the encrypted key update algorithm is run by the TPA. It takes as input an encrypted client's secret key ESKj : the current period j and the public key PK, and generates a new encrypted secret key ESKj+1 for period j + 1.

3) VerESK: the encrypted key verifying algorithm is run by the client. It takes as input an encrypted client's secret key ESKj : the current period j and the public key PK, if ESKj is a well-formed encrypted client's secret key, returns 1; otherwise, returns 0.

4) DecESK: the secret key decryption algorithm is run by the client. It takes as input an encrypted client's secret key ESKj , a decryption key DK, the current period j and the public key PK, returns the real client's secret key SK j in this time period.

5) AuthGen: the authenticator generation algorithm is run by the client. It takes as input a file F, a client's secret key SK j , the current period j and the public key PK, and generates the set of authenticators _ for F in time period j .

6) Proof Gen: the proof generation algorithm is run by the cloud. It takes as input a file F, a set of authenticators _, a challenge Chal, a time period j and the public key PK, and generates a proof P which proves the cloud stores F correctly.

7) Proof Veri f y: the proof verifying algorithm is run by the TPA. It takes as input a proof P, a challenge Chal, a time period j , and the public key PK, and returns "True" if P is valid; or "False", otherwise.

4.0 RESULTS

These experiments are run on a Linux server with Intel processor running at 2.70 GHz and 4 GB memory. A bilinear map is chosen that uses a super singular elliptic curve to achieve the fast pairing operations. The base field of this curve is 160 bits, the size of an element in Zq is 20 bytes, and the size of an element in group G1 is 128 bytes.

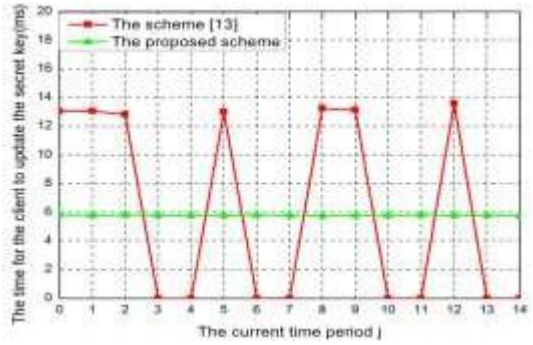

Figure the key update time in our proposed scheme and the scheme

In our experiments, the data file is set to 20M, which consists of 1,000,000 blocks. In our proposed scheme, the time for the client to update a secret key is independent of the time period because it needs one exponentiation and one multiplication in G1 in any time period. We compare the key update time between the both schemes in Fig. the key update time is related to the depth of the node corresponding to the current time period in the full binary tree.

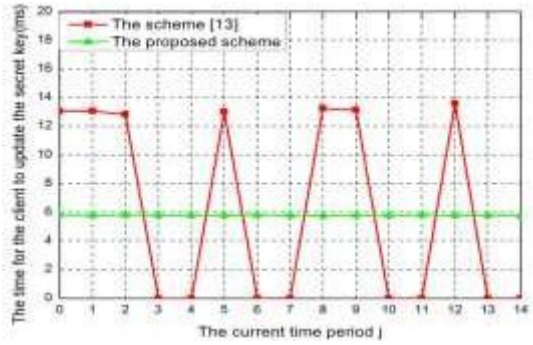

Figure The time of auditing processes different number of challenge blocks

When the node is the internal node, the key update time is about 12.6ms; when the node is the leaf node, the key update time is nearly 0ms. In our proposed scheme, the key update time is about 5.8ms in all time periods. It means the client can update the signing secret key using the same computational resource in each time period. The proposed scheme achieves stronger security without decreasing the efficiency of key updates for the client. In addition, our proposed scheme supports key updates for unlimited time periods. However, the lifetime of the file stored in cloud must be known and fixed for T time periods.

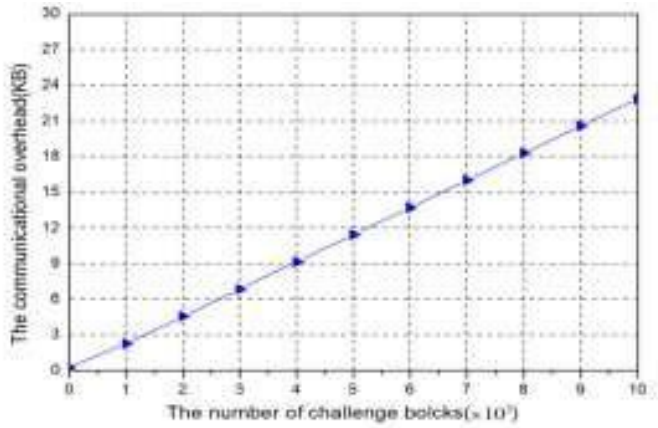

Figure the challenge overhead with different number of challenged blocks

When T time periods are over, the cloud storage auditing cannot work any longer. In our experiment, the number of challenged blocks varies from 100 to 1,000. From Fig.4, we can know the challenge generation process spends the least time, which is 0.26s at most; the proof generation process spends a little more time, varying from 0.18s to 1.79s; the proof verification process spends the most time, varying from 0.22s to 2.05s.

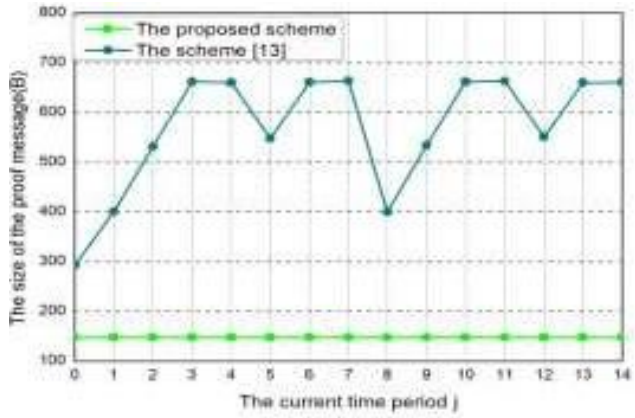

Figure the proof overhead with different of challenged blocks

The challenge overheads of the proposed scheme and the existing scheme, i.e., cloud storage auditing with key exposure resistance are Chal= {i ,vi}, where i is used to determine which blocks will be checked and vi is used to mix the challenged blocks. Therefore, the proposed scheme and the existing scheme have the same challenge overhead. In Fig, when the number of checked blocks varies from 100 to 1,000, the size of the challenge message varies from 2.29KB to 22.89KB. From Fig, we can see that the size of the proof message keeps about 148B in all the time periods.

5.0 CONCLUSION

In this worldview key overhauls can be securely outsourced to some approved gathering and along these lines the key-upgrade trouble on the customer will be kept insignificant. In particular, we influence the outsider inspector (TPA) in numerous current open examining outlines, let it assume the part of approved gathering for our situation and make it accountable for both the capacity reviewing and secure key upgrades for key-presentation resistance. As of late, key presentation issue in the settings of distributed storage examining has been proposed and concentrated on. In this worldview, key redesigns can be securely outsourced to some approved gathering, and subsequently the key-overhaul load on the customer will be kept insignificant. In particular, we influence the outsider evaluator (TPA) in numerous current open examining plans, let it assume the part of approved gathering for our situation, and make it accountable for both the capacity inspecting and the safe key upgrades for key-introduction resistance. Moreover, our plan additionally outfits the customer with capacity to facilitate confirm the legitimacy of the scrambled mystery keys gave by TPA. We formalize the definition and the security model of this worldview. While the client can further verify the validity of the encrypted secret keys when downloading them from the TPA. We give the formal security proof and the performance simulation of the proposed scheme. The security confirmation and the execution reenactment demonstrate that our point by point plan instantiations are secure and productive.

REFERENCES

[1] G. Ateniese et al., ―Provable data possession at untrusted stores,‖ in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 598–609.

[2] 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.

[3] F. Sebe, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-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.

[4] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, ―MR-PDP: Multiplereplica provable data possession,‖ in Proc. 28th IEEE Int. Conf. Distrib. Comput. Syst., Jun. 2008, pp. 411–420.

[5] H. Shacham and B. Waters, ―Compact proofs of retrievability,‖ in Advances in Cryptology—ASIACRYPT. Berlin, Germany: Springer-Verlag, 2008, pp. 90–107.

[6] C. Wang, K. Ren, W. Lou, and J. Li, ―Toward publicly auditable secure cloud data storage services,‖ IEEE Netw., vol. 24, no. 4, pp. 19–24, Jul./Aug. 2010.

[7] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, ―Efficient provable data possession for hybrid clouds,‖ in Proc. 17th ACM Conf. Comput. Commun. Secur., 2010, pp. 756–758.

[8] K. Yang and X. Jia, ―Data storage auditing service in cloud computing: Challenges, methods and opportunities,‖ World Wide Web, vol. 15, no. 4, pp. 409–428, 2012.

[9] 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.

[10]C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, ―Privacypreserving public auditing for secure cloud storage,‖ IEEE Trans Comput., vol. 62, no. 2, pp. 362–375, Feb. 2013.