SIMPLE PROTOCOL TO RESTRICT CLIENT KEY EXPOSURE OVER CLOUD STORAGE SYSTEMS FOR AUDITING

A.H.N.V.L.Prasanna[#] K.Surya Ram Prasad[*] D. D. D. Suri Babu[**]

[#]Student, M.Tech (CSE) , DNR COLLEGE OF ENGINEERING AND TECHNOLOGY

[*] Asst. Professor, Dept. of CSE, DNR COLLEGE OF ENGINEERING AND TECHNOLOGY

[**]Head & Assoc. Professor, Dept. of CSE, DNR COLLEGE OF ENGINEERING AND TECHNOLOGY

Abstract:* Storage is a biggest change over internet to store information over cloud without depending on any client device. The only requirement is to have an internet connection. Keeping data securely in cloud is a big challenging task, many proposed algorithms exist for encrypting information. To challenging the data over cloud we need both public and private keys. For auditing information over cloud it needs private key from client side to audit the information. Once client changes the key the data resides in cloud needs to be updated along with the new key for encryption. In this paper, we propose a simple algorithm which detect changes of all data over cloud along with new key. We propose a novel protocol i.e., cloud contains a private key and it needs a public key every time when the user logging on so that we can provide security for client. Once client gets public key, they can upload the files and use the cloud but in case of downloading the existing files, cloud server needs timely based key called as timestamp key. By providing public key and time key we can view the files. In this approach we provide two level security and prevent from threaten sources. It will improve the performance when compared to existing protocol et. al (Enabling cloud storage auditing with key-exposure resistance) to generate a binary tree and traversing the tree in preorder traversal .

Keywords:CLOUD,SERVER,AUDITING, PRIVATEKEY,PUBLICKEY,TIMESTAMPKEY.

1. INTRODUCTION

Cloud computing is one of the revolution over a decade. Users have right to access information and works on heavy servers without their own hardware setup. It simply needs an internet connection. All these services are available and categorized as three ways:

i) Plateform as a Service (PaaS)
ii) Infrastructure as a Service(IaaS)
iii) Software as a Service(SaaS)

In this paper, we focus on Infrastructure as a Service (SaaS) i.e., storage as a service. Client can store the information as he stores like in their own hard drive but to ensure security, several protocols are used. Client can challenge the files with different tokens without remembering of all keys. He has a single private key and at the time of logging cloud generates a key dynamically. The combination of both this private and public key only can access the file resources .here we are considering a block less data. Cloud storage system stores user data securely with different encryption mechanism. To ensure data integrity we provide auditing on cloud storage systems for statistical information and integrity check. Auditing protocols are used to verify the privacy protection on client data. Third party Auditors are audit *et. al* [6] on behalf of the client and verify the integrity of dynamic data stored in the client. Verification is done for block modification, insertion and deletion based on homomorphic tokens. According to [5] remote data possession checking which allows unlimited number of file verification integrity having maximum time is based on the storage of the cloud user. In this paper we focus on cloud client to renew the key each time when it challenges the cloud for the same file meanwhile to access the file it needs timestamp key to download the file. This timestamp key is changed according to the present time and date. These keys are securely send to the cloud user's mail to ensure high security. The aim of this approach is the file can challenge with different keys. if any intruder can find the private key it is not possible to get a public key even it hacks by the intruder then he needs the timestamp key. This approach is more secure and it needs a simple challenge tokens to the cloud server. In case of auditing, TPA audit the files on cloud servers, so we maintain the one secure key to verify the data integrity by the TPAs. According to k-anonymity of data privacy, even the auditor is not able to view the actual data he can able to see only the statistical information and the total server usage. Data will be in encrypted format he can challenge only the encrypted blocks. In this technique the keys were generated to the client secure mail. If the public keys are exposed to the cloud that not only is sufficient to access because it also needs the private key.

2. Related Work

Users remotely store the data and enjoy on demand high quality services connected over internet from different sources with pay per usage [8].especially users use these services as local storage without worrying about the integrity and security. According to et.al [6] TPA will work on behalf of the client to audit the information over cloud. Dynamic operations like block updating , block insertion and block deletion can also be performed using client secret key resides in cloud server. Public auditing may lead to exposure of these keys and vulnerable actions like TPA itself may changes the key information. To support efficient handling of multiple auditing tasks [6] explore a technique called bilinear aggregate signature .This signature verification is complex and later Jia Yu and Kiu Ren [1]propose a new aspect of cloud storage auditing by reducing the damage of clients secret key exposure while auditing called as key exposure resilience. They propose a technique on binary tree structure with preorder traversal and also develop a novel approach on forward security by block less verifiability

3. System Model

The system model consists of three parties

i) Client
ii) Cloud Server
iii) TPA (Third party Auditor)

Client: Client is the end user who consumes cloud services. Client can upload and download his own files .while uploading the files to cloud several encrypted algorithms are running and saved the file in encrypted format. In this model at time of registration client will get its own private key. Private Key serves main purpose of all needs of the client. In addition to that ever time client gets public key to interact with the cloud for entire session

Cloud Server: In this model Cloud server provider provides infrastructure as a service to the users. It stores the client uploaded files in encrypted format and we can perform all CRUD operations as we did in our local disk space.

TPA: TPA are third party auditors to audit the cloud instead of clients. It verifies the files and gets all audit information.

Naïve Solution:

In this solution, the client still uses the traditional key revocation method. Once the client knows his secret key for cloud storage auditing is exposed, he will revoke this secret key and the corresponding public key. Meanwhile, he generates one new pair of secret key and public key, and publishes the new public key by the certificate update. The authenticators of the data previously stored in cloud, however, all need to be updated because the old secret key is no longer secure. Thus, the client needs to download all his previously stored data from the cloud, produce new authenticators for them using the new secret key, and then upload these new authenticators to the cloud.

4. Architecture

Cloud Client challenges the cloud with public and private key combinations. To access the files client challenges the cloud with public and timestamp key. This time stamp key behaves like a session key .it will expire after certain time period. In this paper, we propose auditing protocol which only needs single private key for specific auditor and audit all the cloud storage and verify the integrity of the files over the cloud. Data resides in cloud can be stored as block data or block less data. In this paper, we focus on only block less data over the cloud. This can be applicable to data blocks over cloud

5. Algorithm

1. Client setup registration key as private key P_{key}
2. For every client unique generates unique
   Private key P_{key} =UniqueGenerator()
3. To access the cloud client has to use both private key P_{key} and public key Pub_{key}
   I= LoginClound(P_{key} , Pub_{key})
4. If (**i** is true)

Upload: Dynamically generates timestamp keys (Tk1,Tk2..Tk5) for periodical sessions

Based on client key P_{key} and encrpypt the

File using encrypt(P_{key}, Pub_{key},T_{key})

Download: it needs session key of that period decrypt(P_{key},Pub_{key}, T_{key})

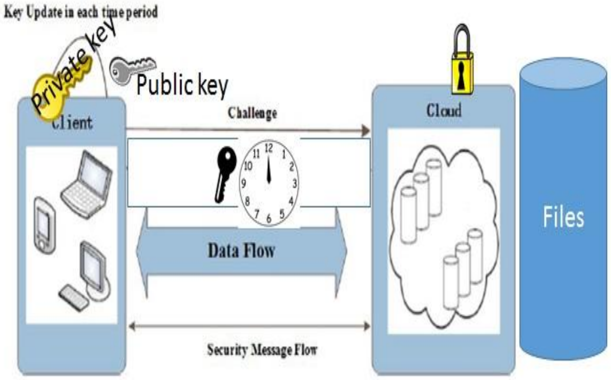

Fig. 1: Client Activity

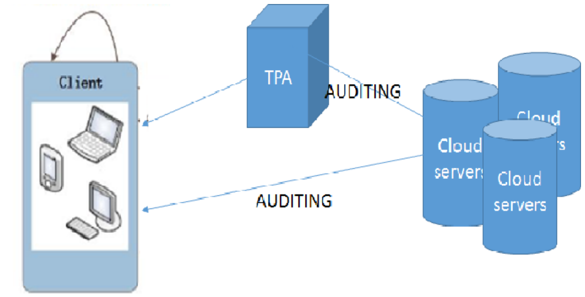

Fig. 2: Auditing the cloud server

4. Security Model

i) Naïve Solution Model(Traditional)

In this solution, client uses the traditional secret key to challenge the cloud and key was stored in the cloud server. if it was exposed in auditing client has to regenerate the new private key and update the keys in all residing files related to this key.it is very complex procedure and uses lot of resources and more time consuming approach and cloud itself contains the secret key, the auditors may modify the data with this secret key which is not known to the users.

ii) Moderate solution

In this solution, we can involve TPA to audit the cloud storage without considering the secret keys over cloud. Because in this approach TPA will have a single secret key which is different from users private and key. While auditing the cloud data auditor may get only statistical information and encrypted data .it is impossible to edit the information over cloud, the data is visible in the form of encryption. In this model, user has to generate a new public key every time when he logging to the cloud server. To access the cloud files he has to give the timestamp key which is also dynamic for session period

5. Conclusion

In this paper, we propose a novel algorithmic approach to challenge the files on cloud server with dynamic public keys and timestamp keys. To challenge the same file access, we are generating public keys with user private key. Auditing is done based on single key and the data which is resides on cloud is in the encrypted format .auditor will collect all statistical information of files and cloud storage details. Even if Client key is exposed, intruders may not succeed to get the user files from the cloud because it needs public key and time stamp key. This model is very simple and provides more security than complex cryptographic approaches

6. References:

1. *J. Yu, K. Ren, C. Wang and V. Varadharajan, "Enabling Cloud Storage Auditing With Key-Exposure Resistance," in* IEEE Transactions on Information Forensics and Security, *vol. 10, no. 6, pp. 1167-1179, June 2015.*

2. *G. Ateniese, R.D. Pietro, L. V. Mancini, and G. Tsudik, Scalable and Efficient Provable Data Possession," Proc 14th ACM Conf. Computer andComm. Security, pp. 598-609, 2007*

3. *F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte and J. J. Quisquater, "Efficient Remote Data Possession Checking in Critical Information Infrastructures," in* IEEE Transactions on Knowledge and Data Engineering, *vol. 20, no. 8, pp. 1034-1038, Aug. 2008.*

4. *H. Wang, Q. Wu, B. Qin and J. Domingo-Ferrer, "Identity-based remote data possession checking in public*

clouds," in IET Information Security, *vol. 8, no. 2, pp. 114-121, March 2014. doi: 10.1049/iet-ifs.2012.0271*

5. *Q. Wang, C. Wang, K. Ren, W. Lou and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," in IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.*

6. *Y. Zhu, H. Hu, G. J. Ahn and M. Yu, "Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage," in* IEEE Transactions on Parallel and Distributed Systems, *vol. 23, no. 12, pp. 2231-2244, Dec. 2012.*

7. *J. Yu. R. Hao . F.Kong. X. Cheng. J.Fan, and Y.Chen., "Forward –secre Idnetity based signature security Notions and Construction, " Information Sciences , Vol. 181 , Iss 3,p. 648-660, 2011*

8. *C. Wang, S. S. M. Chow, Q. Wang, K. Ren and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," in* IEEE Transactions on Computers, *vol. 62, no. 2, pp. 362-375, Feb. 2013. doi: 10.1109/TC.2011.245*

9. *[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.*

10. *K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel and Distributed Systems, Vol. 24, No. 9, pp. 1717-1726, 2013.*

11. *C. Wang, S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, Vol. 62, No. 2, pp. 362-375, 2013.*

12. *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 and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011. [12] Y. Zhu,*