International Journal of
Scientific Engineering and Technology Research

www.ijsetr.com

ISSN 2319-8885
Vol.06,Issue.29
September-2017,
Pages:5656-5658

Updating the Outsourcing of Key for Cloud Storage Auditing Based On Time Period

AMOS R[1], VINOD A M[2]

[1]Assistant Professor, Dept. of MCA, Maharaja Institute of Technology, Mysore, Karnataka, India.
[2]Assistant Professor, Dept. of MCA, Malnad College of Engineering, Hassan, Karnataka, India.

Abstract: Cloud computing has to be designed to maintain the large amounts of data which offers the pay as you go service. It offers more resources such as applications, platforms, and infrastructures. Cloud services are available throughout the world. In cloud computing, users can outsource storage and infrastructure to servers using Internet. Cloud storage can able to store large volume of data, but one major issue here is to maintain the privacy and security of the cloud storage. In one part, the user must authenticate itself before making any access to data, and on the other hand, it must be ensured that the cloud does not make any changes or alteration with the data that is outsourced. The key exposure became a 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 is exposed to cloud, the cloud is able to easily hide the incident of data loss for maintaining its reputation, even some times discard the client's data which are not used regularly for saving the storage space. So, to overcome this problem, encrypt the key before exposing the key to authorized party. The authorized party updates the encrypted key based on time period. Whenever user want to change the data, first verify the validation of the key and decrypt the key and perform the updation of data.

Keywords: Authorization, Encryption, Decryption, Cloud Storage, Auditing.

I. INTRODUCTION

Cloud computing is a technology which offers more services through the internet. Cloud computing put attention towards both academic and industrial worlds. Cloud services are available throughout the world. In cloud computing, users can outsource storage and infrastructure to servers using Internet [1]. Clouds can provide different types of services like infrastructures(e.g,Amazon's EC2, Eucalyptus), applications (e.g., Microsoft online, pdf, google sheets), and platforms to help developers to perform the task. Data stored in clouds is highly sensitive, for example, medical records and social networks. So, providing security and privacy are, thus, very important issues in cloud computing. In one part, the user must authenticate itself before making any access to data, and on the other hand, it must be ensured that the cloud does not make any changes or alteration with the data that is outsourced [2]. Maintaining privacy is also important so that the cloud or other users do not know the identity of the user. The cloud can hold the user accountable for the data it outsources, also the services it provides. In recent years, outsourcing computation creates a wide impact on researchers. It has been considered in many applications including scientific computations [2], linear algebraic computations [1], linear programming computations [3] and modular exponentiation computations [4], 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. The key exposure became a problem in cloud storage auditing, has been considered [5] recently. The problem itself is non-trivial by nature. Once the client's secret key for storage auditing is exposed to cloud, the cloud is able to easily hide the incident of data loss for maintaining its reputation, even some times discard the client's data which are not used regularly for saving the storage space. So, to overcome this issue,Yu et al. [5] developed a protocol for cloud storage auditing with key-exposure resilience by periodically updating the user's secret keys. In this way, we can minimize the damage of key exposure in cloud storage auditing [12]. But it also creates a complexity for the client because the client has to execute the key update algorithm periodically to make his secret key move forward. Users who are having limited resources they do not have intension to update the key regularly. In regular key updations it would be better to maintain the transparency to the user. In this paper, the main objective is to maintain transparency by outsourcing key updates. To achieve this there are so many considerations [9].

1. If the genuine user shares a secret key with the authorized party who performs the key updations then

there s a chance of security threat, so the key should be shared in an encrypted form for cloud storage auditing.

2. The authorizing party performs the outsourcing computations on the encrypted secret key only [8].
3. It should be very efficient for user to retrieve the real secret key.
4. User has to verify the validation of the key after getting from the authorized party.

II. EXISTING SYSTEM

Cloud storage has became the most important services of cloud computing. Even though the cloud storage provides great resources to users, it brings new security threats. The main security issue is to maintain the integrity (The trustworthiness of data in the cloud) of the stored data in cloud. Recently, to address this issue so many auditing protocols has been proposed. The key exposure problem is main important problem in cloud storage auditing. The key exposure became a problem in cloud storage auditing, has been considered [5] recently. The problem itself is non-trivial by nature. Once the client's secret key for storage auditing is exposed to cloud, the cloud is able to easily hide the incident of data loss for maintaining its reputation, even some times discard the client's data which are not used regularly for saving the storage space [7]. User thinks they are the only customer of private inputs, but they will not be able to answer corruption without customer identification. Using numerical and scientific calculations, we want to know what needs to be counts, but computing resources (computing power, proper software, or programming skills) make them locally to create a customer who counts for performing Wants to use an external agent, does not want to outsource the structure of the review [6].

Limitations:

1. Only maintaining integrity is not efficient.
2. If the real user exposes the key to the authorizing party there may be chance of hiding the data loss incidents.

III. PROPOSED SYSTEM

We propose a new method known as updating the outsourcing of key for cloud storage auditing based on time periods. In this new strategy, updation of keys are not done by the user, this will be carried by an authorized party. But here the problem is if the user exposes his key to the authorizing party it creates a threat for security so the user exposes his key in encrypted form [10]. The authorized party keeps on updating the key based on time periods. Whenever the client requires storing or accessing any data from the cloud then first he get the key from the authorized party and decrypts the original key and based on that he performs the required operation. Rather than decryption first the user verifies the validity of the key. So in this paper first we design a protocol for outsourcing key updates foe cloud storage auditing. In this protocol, the third party auditor (TPA) plays the role of the authorized party who is going to take care of key updates based on time periods. TPA does not know the real secret key of the client. Traditional encryption algorithms are not suitable for encryption because it makes the key update difficult to complete also difficult to verify the genuine user. To resolve these challenges, we propose to implement the blinding technique with homomorphic property to efficiently "encrypt" the secret keys. It allows key updates to be smoothly performed under the blinded version, and further makes verifying the validity of the encrypted secret keys possible. It makes the key updates are as transparent as possible for the client. In the designed SysSetup algorithm, the TPA is having an initial encrypted secret key and the client is having a decryption key which is used to decrypt the encrypted secret key. In the designed KeyUpdate algorithm, homomorphic property makes the updation of secret key under encrypted state aregoing smoother, and makes verifying the encrypted secret key possible. The VerESK algorithm s used to check the validity of encrypted keys.

A. Modules

Here we are having 3 modules

1. Client Module
2. Cloud Module
3. Third Party Auditor (TPA)

1. Client: 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.

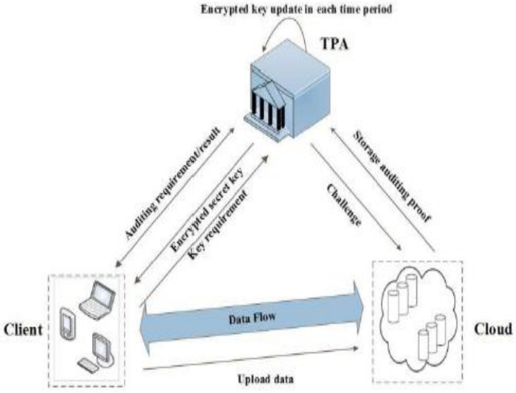

Fig 1. System model for cloud storage auditing

2. Cloud: The cloud stores the client's files and provides download service for the client.

3. TPA: The TPA plays two important roles: the first is to audit the data files stored in cloud for the client; the second is to update the encrypted secret keys of the client in each time period.

IV. CONCLUSION

In this paper we focused on the best way to provide key updations in an encrypted form which goes in smoother manner. TPA update the encrypted key continuously based on time period. If the user want to make the changes to given data then first the user has to validate the key and then decrypt the key. 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.

V. REFERENCES

[1]. 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.

[2] 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.

[3] 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.

[4] 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.

[5] Yu, Jia, Kui Ren, and Cong Wang. "Enabling cloud storage auditing with verifiable outsourcing of key updates." IEEE Transactions on Information Forensics and Security 11.6 (2016): 1362-1375.

[6] Aparna, D., and K. Jaya Shree. "Enabling Cloud Storage Auditing With Verifiable Outsourcing of Key Updates."

[7] 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.

[8] F. Zhao, T. Nishide, and K. Sakurai, “Realizing Fine-Grained andFlexible Access Control to Outsourced Data with Attribute- BasedCryptosystems,” Proc. Seventh Int’l Conf. Information SecurityPractice and Experience (ISPEC), pp. 83-97, 2011.

[9] 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.

[10] 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.

[11] G. Atenieseet al. , “Provable data possession at un-trusted stores,” in Proc. 14th ACM Conf. Comput. Commun. Secur.,2007, pp. 598 – 609.

[12] A. Juels and B. S. Kaliski, Jr., “PORs: Proofs of irretrievability for large files,” in Proc. 14th ACM Conf. Comput. Commun. Secur. , 2007, pp. 584 – 597.