Mutable Encryption for Oblivious Data Access in Cloud Storage

Mahmood Ahmad, Shujjat Hussain, Zeeshan Pervez, Sungyoung Lee and Tae Choong Chung
Dept. of Computer Engineering, Kyung Hee University, Korea

e-mail : {rayemahmood, shjaat.hussain, zeeshan,sylee}@oslab.khu.ac.kr, tcchung@khu.ac.kr

Abstract

Data privacy and access control policies in computer clouds are a prime concerns while talking about the sensitive data. Authorized access is ensured with the help of secret keys given to a range of valid users. Granting the role access is a trivial matter but revoking user access is tricky and compute intensive. To revoke a user and making his data access ineffective the data owner has to compute new set of keys for the rest of effective users. This situation is inappropriate where user revocation is a frequent phenomenon. Time based revocation is another way to deal this issue where key for data access expires automatically. This solution rests in a very strong assumption of time determination in advance. In this paper we have proposed a mutable encryption for oblivious data access in cloud storage where the access key becomes ineffective after defined number of threshold by the data owner. The proposed solution adds to its novelty by introducing mutable encryption while accessing the data obliviously.

1. Introduction

Cloud computing is a major drift in information technology that took its trend from limited pool of resources to boundless provision of computing powers and storage facility. Enormous volume of databases and resource hungry applications are potential candidates for this infrastructure. The contents that are deposited on this cloud infrastructure are not disclosed publically always; secret or private information demands security followed by authorized access. Security is achieved through encryption and authorized access is made possible with the help of access keys. Various models [1,2,3] of secure access have been formulated which guarantee data confidentiality and privacy. Trapdoor [6] is one methodology that can facilitate the secure access model to some extent but has limitations with its predefined limited sets of entries. Private matching [7] is one way to obliviously evaluate common entries between two communicating parties. To perform basic arithmetic operations Homomorphic encryption [5] is there to serve. Proxy re-encryption is used when availability of communicating party is not 24/7, where this technique transforms one cypher text into another cypher text without knowing the inner contents and without giving the donor key to the recipient. After achieving the secure model of data access which is known as oblivious access models, data availability is ensured only for those recipients who have a valid key to decrypt the information

This model has one major concern which is the user revocation. After the user is given with valid keys to access the encrypted data, its validity is effective until the key pair is not regenerated by the data owner. For any system having 'N' numbers of users need to assign 'N' number of keys to all its users. In case, if for 'k' users (where1 $\leq k \leq$ N) are required to be ceased with their access at a particular time, the data owner has to generate all the keys for 'N-K' users. This is a workable solution but very tedious and compute intensive where number of users is very large and frequency of withdrawal is much higher. To solve this issue a technique has been found out which is the predefined time limit to access the data along with the access key

The moment a particular user access the system, its key is validated along with the time stamp when the request is received. In the setup phase, the data owner share few keys with the cloud service provider to check the validity of time stamp against any request to access the information. The solution appears to be good in few cases but not all. Determining the time stamp is not viable in all situations. For example in medical domain, a patient may share her critical information with the physician. If we follow the data model depending on time, the doctor has to follow the strict time limits otherwise he would be unable to access the patient information. To further augment these limitations consider another situation where a user is given with access on Data set 'A' for time bracket (T1 $\leq$ valid time $\leq$ T2). Prior to accessing system 'A' this user needs to access system 'B' which is open but takes undetermined period of time. After receiving response from system 'B' the user expire his time making his access key ineffective beyond the given period of time. Keeping these two scenarios we have coupled the time based access system with round based which is or can be made independent from time

2. Mutable Encryption: Workflow Architecture

The idea of mutable encryption has been derived from the DES encryption standard where it uses predefined number of rounds to calculate key for subsequent rounds [4]. Using the same methodology at abstract level, we have proposed a

mutable encryption which expires on predefined number of rounds (threshold defined by data owner). The overall setup of complete system is shown in fig-1.

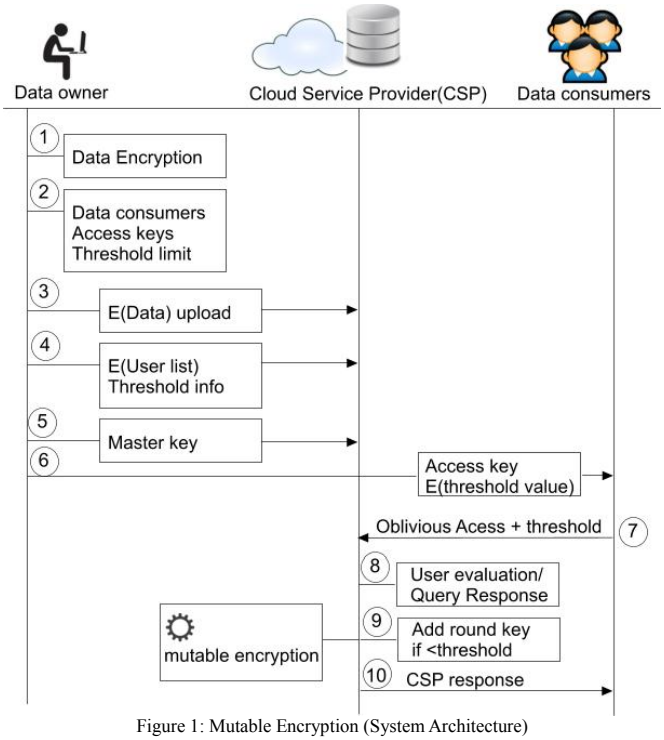

Figure 1: Mutable Encryption (System Architecture)

Data owner starts from data encryption and ends up in 6 steps by handing over the encrypted data and master key to cloud service provider (CSP) and access key to its users. It is assumed that transfer of this information has been carried out in a secure environment. The master key given to CSP will be used for knowing the threshold value along with the user request. With each request this threshold value is evaluated obliviously by the CSP and is entered into the next round, this information is updated and maintained at the CSP. The data consumer will always come with the same access keys thus utilizing the maximum hardware resources at the CSP. After the user request is exhausted or meets the threshold value all results that are given in step 10 of figure:1, ends up in invalid and random response. The activity that takes place at step 9 (which is the core of this architecture) takes input key from user request and its matching round is transformed into the next round if it is less than the threshold.

The algorithm that will work at step 9 will work as shown in the figure:2 . All the operations performed in the algorithm are encrypted and are evaluated using the Homomorphic encryption. Employing Homomorphic encryption ensures that all the calculations are unknown or oblivious to CSP yet effective in their nature. The result given to the end user will either be a meaningful reply for the enduser or a random one. In both cases the CSP will learn nothing and will be unable to infer any knowledge about the result.

```
1 #This algorithm resembles with DES encryption standards
2 #and evaluates user key with its given threshold
3# START
4    if user key is valid
5         reterive its threshold_count
6         reterive its current_round
7              if current_round<threshold_count
8                   current_round++
9                   update CSP_log
10                  update user_log
11             else
12             exit
13   else
14   exit
15 #END
```

Figure 2: Algorithm for Mutable Encryption

3. Mutable Encryption Realization

The realization of mutable encryption is desired in situations where the user access is indefinite while accessing the cloud storage. Applications in health care domain or where the calculations are done in a distributed environment, making time estimation is really hard to define in advance. The proposed idea of mutable encryption is most suitable option in these scenarios.

4. Conclusion

The volume of information that is being deposited on the cloud is enormous and sensitive too. Ensuring privacy of information is not a hard task but making it equally useful at the same time is something which is not trivial. The proposed solution that is presented in this paper will not only exploit these resources effectively but will add another dimension in handling users effectively in the cloud environment.

Acknowledgement

This work (Grants No. 00048272) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2011.

References

[1] Qin Liu,Guojun Wang, Jie Wu. "Time Based Proxy Re-Encryption Scheme for Secure Data Sharing in a Cloud Enviornment."Elsevier Information Sciences, September 18, 2012.

[2] Zeeshan Pervez, Ammar Ahmad Awan,Asad Masood Khattak, Sungyoung Lee and Eui-Nam Huh , "Privacy-aware Searching with Oblivious Term Matching for Cloud Storage", The Journal of Supercomputing (SCI, IF:0.578), 2012

[3] Zeeshan Pervez, Asad Masood Khattak, Sungyoung Lee, Young-Koo Lee and Eui-Nam Huh, "**Oblivious Access Control Policies for Cloud Based Data Sharing Systems**", Springer, (SCI, IF: 1.08), ISSN: 0010-485X, August 21, 2012

[4] Man Young Rhee, Internet Security Cryptographic Principles, Algorithms and Protocols,2003, South Korea, Wiley, ISBN 0-470-85285-2.

[5] Gentry, Craig. A fully homomorphic encryption scheme. Diss. Stanford University, 2009. [6] Cimino JJ. Use, usability, usefulness, and impact of an infobutton manager. Proc AMIA Annu Fall Symp 2006:151–155.

[6] Yao, Andrew C. "Theory and application of trapdoor functions." Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on. IEEE, 1982.

[7] Freedman, Michael J., Kobbi Nissim, and Benny Pinkas. "Efficient private matching and set intersection." Advances in Cryptology-EUROCRYPT 2004. Springer Berlin Heidelberg, 2004.