Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# A Literature Survey on Key Aggregate Cryptosystem for Multi File Data Sharing in Cloud

**Archana Sharma C.N, Dr. K Thippeswamy**
Department of PG studies, VTU,
Mysore, Karnataka, India

***Abstract:*** ***In cloud, data sharing plays an important role. This survey explains how to securely, efficiently and flexibly distribute the data with others in cloud. This survey depicts a key-aggregate or public key cryptosystem, which produces a constant size ciphertext such that efficient delegacy of ciphertext is possible. This new scheme can aggregate several set of secret keys into a solitary key and using this solitary key, multiple files can be decrypted and the files that are outside the ciphertext remain confidential. This aggregate key is stored in a smart card and can be shared with others in a safe channel.***

***Keyword: Data sharing, key aggregate cryptosystem, encryption, cloud, decryption***

## I. INTRODUCTION

Cloud storage is being accepted and is adopted widely because of its commercial characteristics. Cloud allows to stores the data online so that it can be accessed by the cloud user from any place and at any time. Cloud storage is a core for many online services. At present it is simple to apply for gratis mail accounts, sharing of files with size above 25GB. With the utilization of wireless technology, the users can access all their files, emails through their mobile from any place and in cloud data will be available all time (24*7).

When it comes to privacy of data in cloud, a conventional way is to relay on server to impose the access control after authentication i.e. if some unpredicted privilege may expose all the data. The data of different users will be hosted on different virtual machines, but all these reside in one physical machine. By making utilize of a new virtual machine with the intention one, the data can be stolen from the targeted virtual machine. There are several cryptographic schemes are proposed about availability and security of data in cloud. This will make use of an auditor which will check whether the files are there in the cloud on support of the user. The auditor will not reveal any information.

Cloud users can't rely on cloud server in terms of privacy. User will not rely on protection provided by the virtual machine. So the users will encrypt their files with their own keys and then will upload the files. The most important functionality in cloud is sharing the data. Say for example, a blogger will allow his friends to vision the part of his private pictures, and then an organization will allow their employees to view a part of private data. Here confront is how efficiently the users can share the encrypted data. Users from the storage can download the encrypted files and then they can decrypt and can share with others, by doing this it may lose the importance of cloud. Users should be capable to give access to the others from the server for sharing the data. But finding the way to share data securely and efficiently is not trivial.

Keys can be shared with two methods.

1. Sender will encrypt all the files in the cloud with a single key and gives the matching secret key directly to the receiver.
2. Sender will encrypt each file with distinct key and provides the matching secret key to the receiver.

The first method is not adequate as if sender wants to share only few files, but all the files will be accessed by the receiver that is present in cloud. In the second method, if sender is sharing say some 100files, then the sender needs to share 100 secret key with the receiver, which is not appropriate. The cost and complexity increases with increase in number of keys to be shared with the receiver.

Encryption comes in two flavors- symmetric key encryption and asymmetric or public key encryption.

1. symmetric key encryption

In symmetric key encryption, if the sender wants the data to be originated from the third party, then she should give her secret key to the encryptor which is not desirable.

2. Asymmetric key encryption

In asymmetric key encryption both the encryption and decryption keys are different. The use of this asymmetric or public key encryption is suitable for our application.

For example, in an organization, employee uploads the files that are encrypted to the cloud without notice to the organization's master secret key. So the solution for the problem is that the sender encrypts the files with distinct public-keys, but he will send a single constant-size key to the receiver to decrypt the files. This decryption key must me secure, so he will send via a protected channel. If the key size is small it is desirable.

 

## II. EXISTING SYSTEM

### *A. Identity Bases Encryption (IBE)*

IBE is a type of a public-key encryption. Identity-string is set for encryption which is nothing but user's public key. In IBE, master secret keys are generated by the private key generator and here the secret key is provided based on user's identity. Sender wants to share files. So sender will encrypt the files by making use of user identity and public parameter and sends the files. Receiver will decrypt these files by making use of his secret key. Guo et al. [6][3] tried to develop IBE with key aggregation. But out of key-aggregation and IBE, only one assumes random oracles. Key aggregation is inhibited as keys to be aggregated will come from various "identity".

Advantages

- Encryption type is public-key encryption.
- This scheme has a reliable party which will hold secret key.
- Based on the identity, secret key will be provided.
- The size of decryption key is constant.

Disadvantage

- Ciphertext size is non-constant.
- Cost of storing ciphertext and transmitting it expensive.

### *B. Symmetric Key Encryption*

Benaloh et al. [4] proposed an encryption scheme, where a huge number of keys can be sent rapidly in a broadcast scenario. The key origin is as follows. Initially choose two prime numbers p and q for a composite module. At random, master secret key will be chosen. Dissimilar prime numbers will be allied with each class. A public system parameter is considered for which all the prime numbers will be put. The outcome of this is a constant size key. This method is designed for symmetric-key setting. So here the sender should encrypt files with corresponding secret keys which will not be feasible.

Advantages

- Ciphertext size is constant.
- Decryption key size is constant
- Requires less space to store ciphertext and keys.
- Construction is simple.

Disadvantages

- Both encryption and decryption is done by same key.
- Encryptor should get corresponding key to encrypt files.

### *C. Attribute Based Encryption (ABE)*

In Attribute Based Encryption method an attribute will be linked with ciphertext. From master secret key, the secret key will be derived. This secret key is used to decrypt the files merely if all its connected attributes go after the rules. Before Attribute Based Encryption method was introduced, the user who wanted secret key must go to third party and proving he is real by providing his identity and then he was capable to decrypt the file. Later in ABE scheme the secret key of user was not allowed to a single centre. Instead it was authorized by independent authorities. But still this scheme has drawback i.e. no solidity on secret key. Here in this scheme there is linear increase in key size, with the increase in attributes.

Advantages

- Encryption type is public key encryption.
- Ciphertext size is constant.

Disadvantages

- Decryption key size is non-constant.
- Requires more space to store keys.
- Decryption key size increases linearly.
- Managing keys is expensive\

## III. PROPOSED SYSTEM

The proposed system is designed with an efficient public-key encryption. In this any number of subset of the ciphertext can be decrypted by the decryption key. The problem is solved by the introduction of key aggregate cryptosystem. In key aggregate cryptosystem user will encrypt message not merely in a public key but also beneath an identifier. These ciphertexts are more characterized into classes. The owner will have the master secret key. The secret keys are extracted from the master secret key, these secrets keys are used to encrypt the files. The extracted key can be aggregate key which is as compact as a single key. By this solution, sender shares the single aggregate key by means of a safe channel say email. The receiver downloads the encrypted files from sender's drop box and then decrypts those files with single aggregate key. This scenario is shown in Figure 1.

 

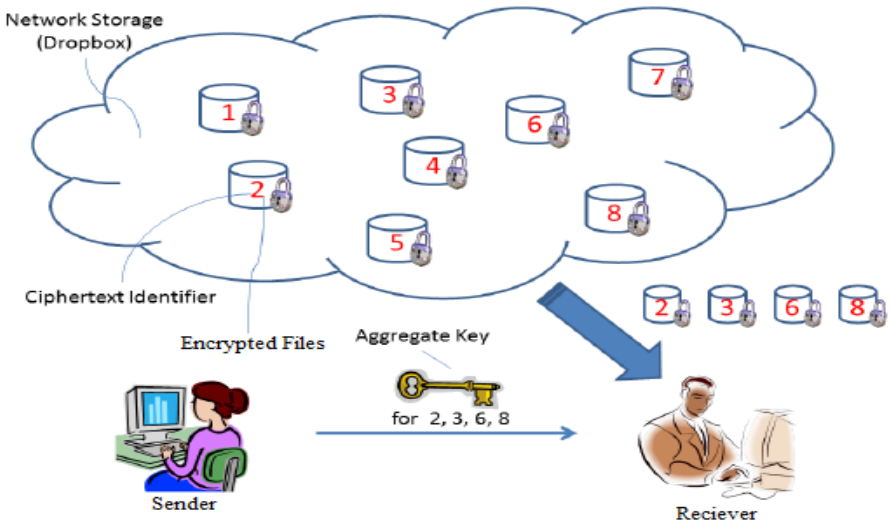


Fig. 1. Sender shares 4 files with identifiers 2,3,6,8 with the reciever by sending him a single aggregate key

Advantages

- With a single decryption key, multiple files can be decrypted.
- The size of the aggregate key, master key, ciphertext and public key are constant in KAC scheme.
- KAC provides efficient data sharing.
- The user can share the data in a private and selective way.
- Encryption type is public-key cryptosystem.
- In KAC scheme, it requires less space to store the aggregate key.
- Key management is easy in our KAC scheme.
- Only selective members can view the message.
- Management of keys is not expensive.

## IV. CONCLUSION

In this survey, we study how to compact secret keys in key-aggregate cryptosystem. This approach is flexible as by using single key able to decrypt multiple files. To store the aggregate key, less space is required. Through this KAC scheme, key management is easy. The space can be saved in compressed key, if all the key-holders share the set of privileges which are alike.

## REFERENCES

[1] C. K. Chu, Sherman S. M. Chow, W. G. Tzeng, J. Zhou, and R H. Deng, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed systems, vol. 25, no. 2, Feb 2014.

[2] R.S. Sandhu "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2,pp. 95–98.

[3] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," in Proceedings of Information Security and Cryptology (Inscrypt '07), ser. LNCS, vol. 4990. Springer, 2007, pp. 384–398.

[4] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records, " in Proceedings of ACM Workshop on Cloud Computing Security (CCSW '09). ACM, 2009, pp. 103–114.

[5] D. Boneh and M. K. Franklin, "Identity-Based Encryption From The Weil Pairing," in Proceedings of Advance in Cryptology – CRYPTO '01, ser. LNCS, vol. 2139. Springer, 2009 pp. 213–229.

[6] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in Proceedings of Advances in Cryptology - EUROCRYPT '05, ser. LNCS,vol. 3494. Springer, 2005, pp. 457–473.

[7] S. S. M. Chow, J. Weng, Y. Yang, and R. H. Deng, "Efficient Unidirectional Proxy Re-Encryption," in Progress in Cryptology AFRICACRYPT 2010, ser. LNCS, vol. 6055. Springer, 2010.

[8] G. C. Chick and S. E. Tavares, "Flexible Access Control with Master Keys," in Proceedings of Advances in Cryptology CRYPTO'89, ser. LNCS, vol. 435. Springer, pp. 316–322.

[9] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," IEEE Transactions On Knowledge and Data Engineering (TKDE), vol. 14, no. 1, pp. 182–188.

 