Innovative Broadcast Key Scheme for Cryptosystem in Online Cloud Data Sharing

Rigam Sowmiya.R, Ramkumar.M.O

Abstract— **In a practical data sharing environment with millions of users who tend to store and share their data, the major issue faced is Security. The data sharing environment which this paper focuses on is Cloud environment. Every data owner wants his data to be stored in an encrypted manner and share his data only with the user having proper decryption rights for the particular file. In this paper, we propose a Multiple KAC (Key-aggregate cryptosystem) using the ECC (Elliptic curve cryptography) algorithm and multiple key protocol. This system is combined with broadcast encryption scheme for distributing multiple aggregate-keys to arbitrary number of users in a cloud environment.**

Index Terms— **Cloud environment, data sharing, Broadcast encryption, Multiple Key-Aggregate Cryptosystem, Elliptic Curve Cryptography (ECC).**

I. INTRODUCTION

Nowadays Internet is being widely used all over the world. It has many applications. One of the most popular application is cloud computing, which has been invented by Joseph Carl Robnett Licklider around 1960's to connect people and data at anytime from anywhere. The users tend to store their data in cloud and share it to other privileged users. This is said as online data sharing. Cloud storage also provides natural disaster proof backup, as there are 2 or 3 different servers around the globe. The users need to pay only for the operating expenses (storage they actually use) and not the capital expenses.

Though there are many advantages in using the cloud storage, they are susceptible to privacy and security attacks. This is because the online data always resides in shared environments (i.e., multiple virtual machines sharing on the same physical device). Regarding the data privacy, a traditional method is to rely on the server for enforcing the access control mechanism [1], in which any unexpected privileged escalation could expose all the data. So, the data must be made encrypted before uploading or sharing data with others. Some primary requirements that are to be provided in online cloud data sharing are data confidentiality, user revocation, scalability and efficiency, collusion between entities [2].

A special type of public-key encryption called Key-aggregate Cryptosystem (KAC) [3] concept is used here. Multiple aggregate-keys are used at the side of decryption.

II. LITERATURE SURVEY AND RELATED WORK

This section provides a brief overview of existing public and private key cryptographic schemes that are used for secured online data sharing. The special focus is on providing constant size keys for decrypting an arbitrary number of encrypted entities.

A. *Predefined Hierarchical Encryption*

One of the most popular and efficient method for access control in online data storage is using a predefined hierarchy of secret keys [4], [5], [6] which is done to minimize the cost of storing and managing secret keys for cryptographic use. It was defined with a tree-like structure. Here, the keys of the descendent nodes can be derived from a key that is assigned to a particular parent node i.e. access to the key corresponding to any node implicitly grants access to all the keys in the sub tree (all the descendent nodes) rooted at that node. Sandhu [7] proposed a method to generate a tree hierarchy of symmetric keys using repeated evaluations of pseudo-random function. Some advanced schemes extend access to cyclic and acyclic graphs.

Generally, the hierarchical approaches seem to be efficient only in case of sharing all files under certain branch of hierarchy. A main disadvantage is the number of secret keys increases with the number of branches. This in turn enlarges the size of keys shared.

B. *Symmetric-key Encryption using Compact Key*

This encryption scheme presented by Benaloh et al [8] is actually for transmitting large number of keys in broadcast scenario. It is used for symmetric key setting. The encryptor needs to get the corresponding secret key to encrypt the data using secure channel.

The use of secured channel for transmission of keys may be expensive and may not be always suitable for many applications in cloud. Moreover, this method only generates a secret value rather than a pair of secret keys. This makes it unclear to apply in public-key encryption scheme.

C. *Attribute Based Encryption (ABE)*

In Attribute based encryption [9], a set of attributes is used to identify each user and each ciphertext is related to particular

***Manuscript received March, 2017*.**

Rigam Sowmiya.R, *Computer Science and Engineering Department, IFET College of Engineering, Villupuram, Tamil Nadu, India, Mobile No: +91-733950537.*

Ramkumar.M.O, *Assistant Professor, Computer Science and Engineering Department, IFET College of Engineering, Villupuram, Tamil Nadu, India, Mobile No: +91-9944026683.*

set of attributes. Decryption of encrypted data can be done only by the user who has the corresponding secret key. The transmission of secret key is done to the user who satisfies the access control policies set by data owner.

Major drawbacks of this scheme are the size of keys increases linearly with the number of attributes and ciphertext size is not constant. It is collusion resistant but not suitable for key-size compression. Each time the access right to the user is revoked, the entire ciphertext needs to be re-encrypted in the cloud environment.

D. Identity Based Encryption (IBE) using Compact Key

Identity based encryption is one of the public-key encryption in which the public key of the user can be set using some unique information about the user as his/her identity (e.g., mobile number, an email address). A trusted party called Private Key Generator (PKG) holds a master secret key and issues a secret key to each user with respect to their user identity. Public parameters and user identity are considered by the encryptor to encrypt the message. Using the secret key, the recipient can decrypt the message.

A single compact secret key [10] is used to decrypt the cipher texts those are encrypted based on many identities in Fuzzy IBE. The main disadvantage is the cost of managing and transmitting ciphertexts increases and it becomes not suitable for the cloud storage.

Table 1: Comparative study of various data sharing schemes

Schemes	**Decryption key size**	**Secure channels**
Predefined Hierarchical encryption	Non-constant	$O(mm^0)$
Symmetric encryption	Constant	$O(mm^0)$
Attribute Based Encryption	Constant	O(m)
Identity Based Encryption	Non-constant	O(m)
Basic KAC	Constant	$O(mm^0)$
Generalized KAC	Constant	$O(m+m^0)$

In the above table, various data sharing schemes are compared for m data users and m^0 data owners.

III. Existing System

The existing system focuses on combining the standalone KAC scheme with broadcast encryption [11] to serve m data users and m^0 data owners reducing the channel requirement from $O(mm^0)$ to $O(m+m^0)$ [2]. The generalized KAC constructions are made to be CPA and CCA secure. This type of construction is implemented using Elliptic Curve Cryptography (ECC) [12] efficiently and is suitable for cloud based data sharing environments.

A CCA-secure fully collusion resistant construction for the basic KAC scheme with low overhead ciphertexts and aggregate keys is implemented.

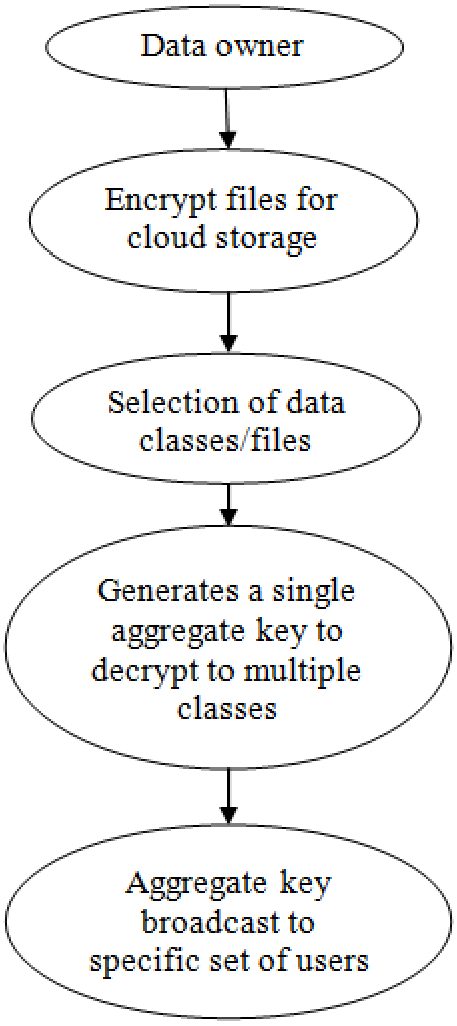
Data owner
Encrypt files for cloud storage
Selection of data classes/files
Generates a single aggregate key to decrypt to multiple classes
Aggregate key broadcast to specific set of users

Fig 1: Procedure- KAC online data sharing scheme

The above Fig.1 shows the procedure for a generalised KAC scheme for sharing data. In this scheme, the data owner on storing a set of files into the cloud encrypts the data before uploading. A single constant size decryption key that combines the decryption rights of the data classes. It uses a public key framework to broadcast this key to the target set of users in the form of a low overhead broadcast aggregate key i.e., when a sender sends a same file to a specific set of receivers, a single aggregate-key is being broadcasted to all the privileged receivers to decrypt the data files.

This scheme is efficient and scalable to any arbitrary number of data classes and data users. It avoids the use of secret channels which are costly.

The extended KAC framework [2] with aggregate-key broadcast includes: Setup, OwnerKeyGen, OwnerEncrypt, SystemEncrypt, UserKeyGen, Extract, Broadcast and Decrypt.

The disadvantage is a single aggregate-key is used for decrypting the file by multiple users, it may be insecure. This is when the single aggregate-key that is to be sent is exposed (hacked), then any user within the set who may be privileged will also be able to decrypt the shared file using the exposed single aggregate-key and the hacker could access the file.

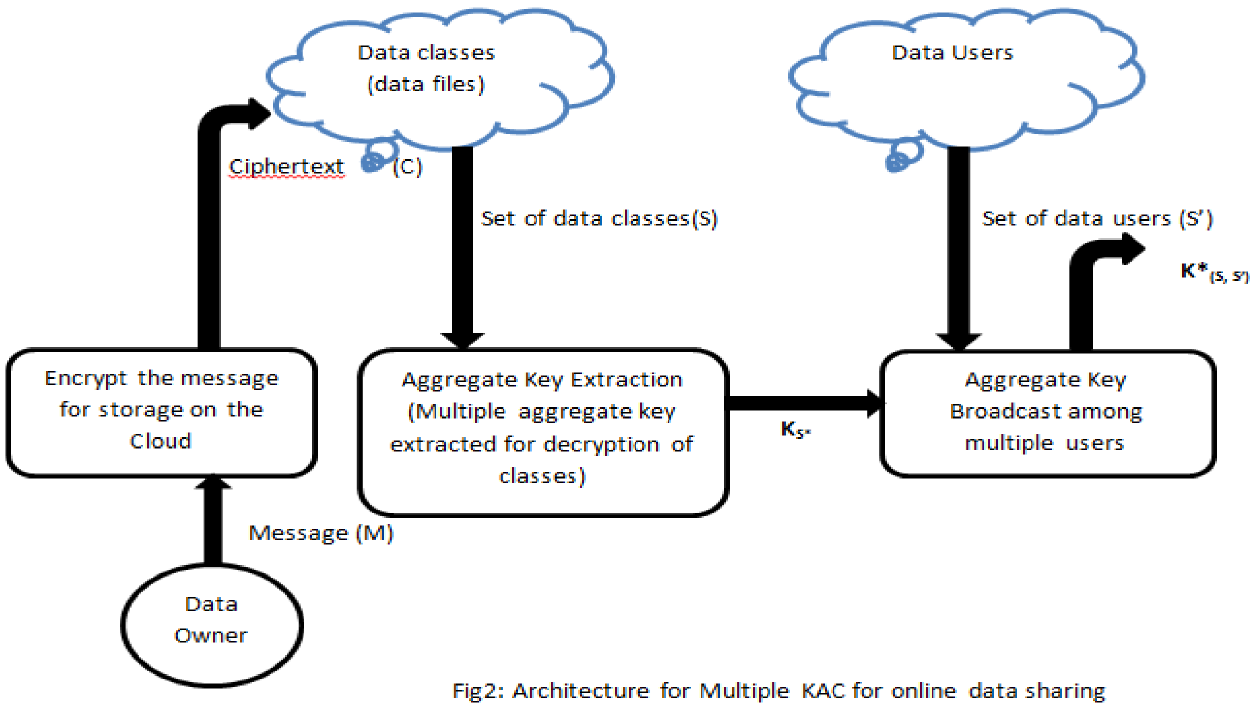
Data classes
(data files)
Ciphertext
(C)
Set of data classes(S)
Data Users
Set of data users (S')
K*(S, S')
Encrypt the message for storage on the Cloud
Aggregate Key Extraction (Multiple aggregate key extracted for decryption of classes)
KS'
Aggregate Key Broadcast among multiple users
Message (M)
Data Owner

Fig2: Architecture for Multiple KAC for online data sharing

IV. Proposed System

As the broadcasting of a single aggregate-key for all set of users is insecure, we step into the following new system.

This system focuses on the construction of a Multiple Key-aggregate cryptosystem (Multiple KAC) using the Multiple-key protocol and Elliptic Curve Cryptography (ECC). This type of Cryptosystem is then combined with the broadcast encryption key for distributing different aggregate keys to arbitrary number of users in cloud environment.

By implementing this system, multiple aggregate-keys could be generated and broadcasted to multiple users for their decryption of data instead of broadcasting a common aggregate-key (single key) to multiple users. The major advantage is the increase in efficiency and security. None other than the privileged user having their own decryption key can access the file.

The Fig.2 shows the data owner uploading message M and encrypting the data for storage in cloud. The encrypted data is called ciphertext C. S and S' are the set of data classes and data users. K^*_S refers to the multiple aggregate key generated for decryption of data classes. The Multiple aggregate keys broadcasted to a set of users to decrypt set of data classes is represented as $K^*_{(S, S')}$.

This system includes four modules: User/Client module, public Cloud Server (PCS), Proxy and the Key Generation Centre (KGC).

A. User/Client Module

This module describes the client entity. A new client creates his own account in cloud by filling some of his details like username, mail-id, phone number, etc. He sets his own password for later access of the account. Existing client could login to his account with the help of username and password. The client uploads massive data into the cloud server for storage. It can perform the remote data integrity checking.

B. Public Cloud Server (PCS)

Public Cloud Server (PCS) is managed by a cloud service provider. Is holds the significant storage space and computation resource to maintain the client's data. If some blocks are modified /deleted, then the malicious PCS cannot generate a valid remote data integrity proof. Also, a practical Multi key protocol needs to convince the client that all of his outsourced data is kept integrated with a high probability. This server stores all the data uploaded by the client.

C. Proxy

Proxy entity acts as an intermediate in this online data sharing. The proxy holds all the keys for transmission. It acts as intermediate between the endpoint device and another server from which the user or client is requesting a service.

D. Key Generation Centre

Key Generation Centre (KGC) is an entity which on receiving an identity generates the Multi private key which corresponds to the received identity. It enjoys very efficient multi generation. This module focuses on generating keys for the cryptosystem.

V. System Requirements

This section describes about the platform and components required for implementation of the proposed scheme. It involves a public cloud based setup consisting of three VMs -the data owner VM that performs Encryption operation, the data user client that performs decryption operation, and trusted third party VM for other operations.

The software requirements include Windows 8/10 operating system and Asp.net as coding language. The Integrated Development Environment (IDE) used is Android Studio 2010 and MySQL database.

VI. Conclusion

Providing Security is most important issue in online cloud data sharing. Users upload their data and the data privacy is maintained by initially storing it in encrypted form. Here the study and comparison of different techniques for data sharing is made and we found that the Multiple-key aggregate system more efficient and secure than others. This system is developed for generating a unique key for accessing file by multiple users in the aggregate server. The delegation of decryption is made to be efficiently implemented. This secure ensures constant size ciphertexts and aggregate key. The Elliptic Curve Cryptography (ECC) is used as it well suits in cloud environment, providing shorter key length with faster computations. ECC provides a level of security with 164-bit key which equals to 1024 bit-key in other systems. ECC is widely used in mobile applications due to its high security with low computing power and low resource usage. The Multiple-key protocol is used to generate and broadcast multiple keys for sharing of same file among multiple users.

REFERENCES

[1] S. S. M. Chow, Y. J. He, L. C. K. Hui, and S.-M. Yiu, "SPICE -Simple Privacy-Preserving Identity-Management for Cloud Environment", in *Applied Cryptography and Network Security – ACNS 2012*, ser. LNCS, vol. 7341. Springer, 2012, pp. 526–543.J.

[2] Sikhar Patranabis, Yash Shrivastava and Debdeep Mukhopadhyay, "Provably Secure Key-Aggregate Cryptosystems with Broadcast Aggregate Keys for Online Data Sharing on the Cloud", Citation information: DOI 10.1109/TC.2016.2629510, IEEE.

[3] Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H, Deng, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", Parallel and Distributed Systems, IEEE Transactions on, 25(2):468–477, 2014.

[4] Selim G Akl and Peter D Taylor, "Cryptographic solution to a problem of access control in a hierarchy", ACM Transactions on Computer Systems (TOCS), 1(3):239–248, 1983.

[5] Gerald C Chick and Stafford E Tavares, "Flexible access control with master keys", In Advances in CryptologyCRYPTO89 Proceedings, pages 316–322. Springer, 1990.

[6] Wen-Guey Tzeng, "A time-bound cryptographic key assignment scheme for access control in a hierarchy", Knowledge and Data Engineering, IEEE Transactions on, 14(1):182–188, 2002.

[7] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control", Information Processing Letters, vol. 27, no. 2, pp. 95–98, 1988.

[8] 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.

[9] John Bethencourt, Amit Sahai, and BrentWaters, "Ciphertext-policy attribute-based encryption", In Security and Privacy, 2007. SP'07. IEEE Symposium on, pages 321–334. IEEE, 2007.

[10] Fuchun Guo, Yi Mu, and Zhide Chen, "Identity-based encryption: how to decrypt multiple ciphertexts using a single decryption key", In Pairing-Based Cryptography–Pairing 2007, pages 392–406.Springer, 2007.

[11] Dan Boneh, Craig Gentry, and Brent Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys", In Advances in Cryptology–CRYPTO 2005, pages 258–275. Springer, 2005.

[12] Kulkarni Mayuri A, V. R. Chirchi, "Key-Aggregate Cryptosystem based on Elliptic Curve Cryptography for Data Sharing in Cloud Storage", International Journal of Computer Applications (0975 – 8887) Volume 150 – No.2, September 2016.