Secure Sharing of Data in Cloud through Key Aggregation

LakshmiKantha K.N,
M-Tech Student, Computer Networking Engineering,
Reva Institute of Technology and Management, Bengaluru

Mr. A. Ananda Shankar,
Associate Professor,
Dept. of CSE,
Reva Institute of Technology and Management, Bengaluru.

ABSTRACT-Cloud storage is a model of data storage in which the digital data is stored in logical pools, and the physical storage spans multiple servers. Security thus plays a crucial role in clouds and multiple schemes have been developed for the same. While implementing security, considerable thought must be given to the amount of overhead faced. The proposed model incorporates an aggregate key that ensures security, flexibility and efficiency. It is a new public key cryptosystem producing constant-size encrypt texts such that efficient delegation of decryption rights for any set of cipher-texts are possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a one key, but encompassing power of all keys being aggregated. Secret key holder can release a constant-size aggregate key for flexible choices of cipher-text set in cloud storage, but other encrypted files outside the set remain confidential. This key can then be given to others conveniently.

INTRODUCTION

In modern days cryptography, encryption keys can be obtained in two ways, symmetric and asymmetric (public) key. The asymmetric key encryption tends to be more secured as it involves combination of two different keys, public and private key respectively. Cryptography is the method of storing and sharing the data in a form that allows only authenticated users to access information. It is a paradigm to secure the message by encoding it into an indecipherable format. The basic goal of cryptography is, the ability to send the data stored on cloud, through the internet to the receiver in a way that prevents attackers from accessing it. Most of the encryption algorithms are not efficient and does not provide efficient delegation of decryption rights. The public key encryption algorithm requires a public / private key pair to encrypt/decrypt each of the data file stored on cloud, thus requiring a lot of space to store the keys as the number of data files increases. Also an efficient communication channel is required to transfer the keys.

Over the many years various new schemes and cryptosystems have been proposed. The Attribute Based Encryption technique allows an attribute to be associated with each cipher text which could be decrypted only by a conforming key. Identity Based Encryption allowed an identity string to be associated with the public-key of every user. Homomorphic encryption schemes allowed for operations to be carried out on cipher texts reflecting the result of the operation if carried out on plain texts itself .The Key Aggregate Cryptosystem enabled decryption of a number of cipher texts using a single key that was compact .These schemes have paved the way to a more efficient and secure storage and sharing in cloud. These schemes are more efficient relative to the conventional schemes in the sense that they provide some gain like increased security or decreased costs etc.

I. SCOPE OF THE PROJECT

The main scope of this project is to implement and understand a newly proposed method that strongly, proficiently, and flexibly shares data with other users in cloud. This method produces a constant-size cipher text which enables efficient delegation of decryption rights for any set of cipher texts. The advantage of this method is that the data owner can release a constant-size aggregate key for flexible choices of cipher text set in cloud storage, but his other encrypted files outside the set remain confidential.

II. PROBLEM STATEMENT

"To design an efficient public-key encryption scheme, which supports flexible delegation of decryption rights, in the sense that any subset of the cipher texts produced by the encryption scheme is decrypt-able by a constant-size decryption key, generated by the owner of the master-secret key".

III. EXISTING SYSTEM AND LIMITATIONS

There are two methods to securely share a large number of encrypted on the cloud through public key cryptosystem.

Method 1: All files will be encrypted using a single encryption key and share the secret key with user who is intended to receive the file.

Method 2: All files will be encrypted using separate keys and then send the requestor the corresponding secret keys for the files

Disadvantages:

1) First method is inadequate since all unselected files by the requestor may be leaked.

2) In second method, there are practical concerns on efficiency. The number of such keys is as many as the number of shared files.

3) Also in the second method, transporting these secret keys inherently requires a secure channel, and

storing these keys requires an expensive secure storage. The costs and complexities involved generally increase with the number of the decryption keys to be shared.

IV. PROPOSED SYSTEM

In our proposed system we introduce a unique type of public key encryption which we call key aggregate cryptosystem. In key aggregate cryptosystem, message is encrypted not only under a public-key, but also under an identifier of cipher text called class, which means the cipher texts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More significantly, the extracted key can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys.

Having this solution, the data owner can just send the data user a single aggregate key through a secure e-mail. Data user can then download the encrypted files from data owner's cloud space and then use aggregate key to decrypt the files.

Advantages:

1) Powerful Decryption-Key that aggregates the effect of several secret keys.

2) Efficient and flexible delegation of the aggregate key with easy key management.

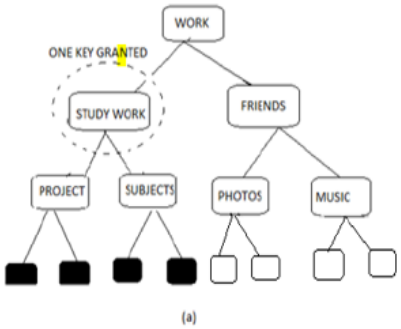
3) Sizes of public key, master secret key, and aggregate key in the KAC schemes are all approximately of constant size.

V. RELATED WORK

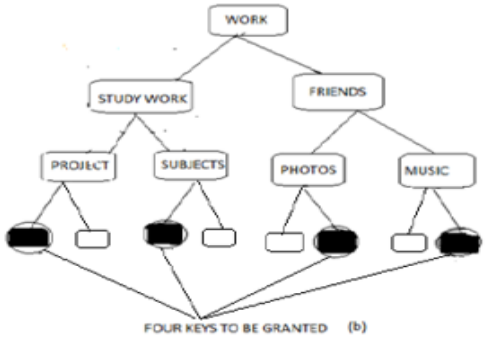
This section provides brief understanding of various cryptographic techniques along with their advantages and disadvantages.

A. Cryptographic Keys for a Predefined Hierarchy

In Predefined Hierarchical Schemes the goal is to minimize cost in storing and managing secret keys for general cryptographic use. Employing a tree structure, a secret key for a given root can be used to procure the secret keys of its leaf nodes. Taking tree structure as an example, first classify the cipher text classes according to their subjects as shown in 1.1 a) One key for hierarchy, b) Four keys for hierarchy. Each node in the tree represents a secret key, while the leaf nodes represent keys for individual cipher text classes. Filled rectangles represent keys for the classes to be delegated and circles circumvented by dotted lines represent keys to be granted. Note that every key of the non-leaf node can derive the keys of its child nodes. In Fig 1.1 a), if the user wants to share all files in the "study work" category, she only needs to grant key for the node "study work", which automatically grants the delegate, keys of all the child nodes. This is the ideal case, where most classes to be shared belong to the same branch and thus a parent key of them is sufficient. However, it is still difficult for general cases. As shown in Fig 1.1 b), if user wants to share files from different branches, the user has to grant as many number of keys as the number of different branches containing these classes else the files from child nodes can also be accessed. One can see that this approach is not flexible when the classification becomes more complex and the user wants to share different sets of files to different people.



a) One key for hierarchy



b) Four keys for hierarchy

B. Attribute Based Encryption (ABE)

ABE, a form of public-key encryption, is a scheme of encryption where both the secret key of a user and the cipher text are dependent on some attributes. Attributes specified can be any key property of that individual such as his native country, type of subscription he possesses etc. In this type of scheme for encryption, decryption of the cipher text is achieved only if set of attributes of user key matches attributes of cipher text. The concept of ABE was first proposed in a landmark work by Amit Sahai and Brent Waters. This scheme possesses good collusion-resistance capabilities. However it possess two challenges non-efficiency and non-existence of attribute revocation mechanism .This scheme has two major sub-groups cipher text-policy ABE (CP-ABE) and Key-policy ABE (KP-ABE). ABE is also the generalization of another encryption scheme known as Identity Based Encryption (IBE). IBE is an important primitive of ID-based cryptography. As such it is a type of public-key encryption in which the public key of a user is some unique information about the identity of the user (e.g. a user's email address).

CP-ABE: In CP-ABE mechanism, access strategy is in control of the encryptor .The complexity of the design of the system public key increases with increasing complexity in the access strategy. This

scheme can be considered as a generalization of IBE but proves to be more flexible. There is a single public key and a master private key that is used to produce private keys. The private keys are associated with sets of attributes or labels, and when encryption is done, we encrypt to an access policy which specifies which keys will be able to decrypt the data allowing complex rules specifying which private keys can decrypt which cipher texts.

KP-ABE: Key-Policy ABE is an important class of Attribute Based Encryption where the cipher texts have sets of attributes associated with them and private keys depend on the access structures controlling the cipher texts a user is able to decrypt. Key-Policy ABE has important applications in data sharing on untrusted cloud storage. However, the cipher text size grows linearly with the number of attributes embedded in cipher text in most existing KP-ABE schemes.

C. Homomorphic Encryption

This type of encryption method permits computations to be carried out on ciphertext. The computations on the ciphertext give an encrypted output which on decryption matches the result of the operations performed on the plain text. This encryption scheme produces a modifiable design, thus providing integrity and confidentiality of the data to be sent. Homomorphic encryption schemes are classified into partially homomorphic and fully homomorphic scheme. Homomorphism is understood taking the RSA algorithm as an example,

If the public key in RSA is modulus 'm' and exponent 'e', then the encryption of a message 'x' is given by 'Є(x) =x e mod m'. The homomorphic property is then

Eq1. Є(x1) .Є(x2) = x1e x2e mod m = (x1 x2) e mod m =

Є (x1. x2 mod m).

Homomorphic encryption schemes are classified into partially homomorphic and fully homomorphic schemes.

Partially-Homomorphic Encryption Schemes: A cryptosystem that shows either additive or multiplicative but not both at the same time is called partially homomorphic. Thus, this subset of homomorphic schemes allows only limited computational capabilities i.e.: operations that can be performed on the cipher text are limited .These schemes allow only specific computations to be carried out onto the cipher text. Processing time and implementation complexity are key advantages of this subset over Fully Homomorphic Encryption Schemes. There exist many cryptosystems that are partially homomorphic like unpadded RSA system and ElGamal system that exhibit multiplicative homomorphism and the Pailler system that exhibits additive homomorphism.

Fully-Homomorphic Encryption schemes: A cryptosystem is considered fully homomorphic if it exhibits both additive and multiplicative homomorphism. Thus, fully homomorphic encryption schemes allow great flexibility in operations that can be carried out onto the cipher text. They possess great computational capabilities and allow creation of generic constructs for computations. Since such a scheme need never decrypt its inputs, it can be run by an untrusted party without revealing its inputs and internal state. The first such construct was proposed by Craig Gentry in 2009. The proposed model was a lattice-based cryptosystem that relies on a complex mesh of ideal lattices for representing keys and cipher text. The next system was proposed in 2010 built with Gentry's construction as a foundation and did not require ideal lattices.

D. Identity-Based Encryption (IBE) with Compact Key

IBE as mentioned is a specific scenario of ABE. It is a public-key encryption scheme where public-key can be set as the identity-string of the user. The Private Key Generator (PKG) holds a master secret key specific to each user with respect to his/her identity. The encryption is done with the help of the public parameter and user identity and the recipient decrypts the message using the secret key. An author Guo made first advances to building an IBE scheme with key aggregation with the constraint that all keys to be aggregated must belong to different "identity divisions". Thus, only polynomial number of keys could be aggregated whereas number of identities and hence secret keys are exponential .An alternative approach to this scheme involved using a hash function to the string denoting the class. In fuzzy IBE, one single compact secret key can decrypt cipher texts encrypted under many identities which are close in a certain metric space, but not for an arbitrary set of identities.

E. Key Aggregate Cryptosystem

The encryption of messages in the Key Aggregate Cryptosystem (KAC) is done not only under a particular public key but also in accordance with an identifier of the cipher-text called class. A cipher text class is an arbitrary integer defining a classification as made by the data owner under which the plain text is to be encrypted. Thus, cipher texts are categorized into different classes. The key owner owns a master-secret key, which can be used to extract secret keys for different classes. The key that is extracted can be an aggregate key which combines the power of many secret keys allowing the decryption of more than one cipher text with the same aggregate key. The sizes of cipher text, public-key, master-secret key and aggregate key in the KAC scheme are all of constant size. This provides great time and security advantages over the one key per request for file and

one key for all files schemes respectively. A limitation of the above scheme is the predefined bound of maximum number of cipher text classes. In cloud storage environment, this parameter grows rapidly and thus, a scheme that is independent of number of cipher text classes is an area for future work.

A general framework for KAC for scalable data sharing in cloud storage proposed by Cheng-Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng is as follows,

(a) Setup Phase: Data owner executes this phase to register his/her account on an un-trusted server. This step also deals with deciding the number of cipher-text classes.

(b) Key Generation Phase: Executed by data owner to randomly generate a public / master–secret key pair.

(c) Encrypt Phase: Executed by any user in order to encrypt the data .This is done using public key, index, and the message and delivers the cipher-text.

(d) Extract Phase: This phase is executed by the data owner for delegating the decrypting power for a certain set of cipher-text classes to a delegate .It uses the master-secret key and set of indices of the cipher-text classes and outputs the aggregate key.

(e) Decrypt Phase: This phase is executed by a delegate who received an aggregate key generated during extraction phase. This phase results in decrypted data to be presented to the delegate.

VI. System Configuration

We have following software and hardware requirements to implement our proposed system.

HARDWARE REQUIREMENTS (minimum):

Processor	:	Pentium IV or upwards.
Hard Disk	**:**	20 GB.
RAM	**:**	1 GB
Key Board	**:**	Standard Windows Key board.
Mouse	**:**	Two or three button mouse.

Any desktop / Laptop system with above configuration or higher level.

SOFTWARE REQUIREMENTS (minimum):

OS	:	Windows XP or upwards
Language used	:	Swings (JDK 1.8)
IDE	:	Eclipse
Database	:	HyperSQL

Cloud Service Provider: Drop Box

VII. Conclusion

The major security issues with respect to cloud include privacy, trust, data portability and conversion, integrity, authentication. But the most important of them is security and how cloud providers are assured of it. Cloud computing has several customers from various backgrounds like ordinary users, academia, and enterprises .If cloud clients are academia ,performance along with security issues are to be taken into account. From an enterprise's point of view security is of higher priority compared to high performance. Security is thus viewed from different point of view by different users based on their requirements. We thus analyze traditional security algorithms in cloud by dividing them into two categories namely symmetric and asymmetric .These algorithms are then compared using various parameters like scalability, security, block size and key length. We then consider modern security schemes and cryptosystems towards effective encryption .This facilitated the study and understanding of the evolution of security algorithms for cloud security from its roots via traditional schemes to its modern day interpretations.

VIII. Acknowledgement

I convey my sincere thanks to my project guide Mr. A. Ananda Shankar who provided me with the opportunity and without whose encouragement this work would not have been
possible. I would like to thank our department and college who have been very supportive.

IX. References:

[1] Md Asif Mushtaque, H, Dhiman, S. Hussain and S.Maheshwari, "Evaluation of DES, TDES, AES, Blowfish and Two fish Encryption Algorithm: Based on Space Complexity", International Journal of Engineering Research & Technology (IJERT), Vol. 3 Issue 4, April – 2014.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-98, 2006.

[3] M. Chase and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009.

[4] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Advances in Cryptology (CRYPTO '01), vol. 2139, pp. 213-229, 2001.

[5] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '05), vol. 3494, pp. 457-473, 2005.

[6] S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," Proc. ACM Conf. Computer and Comm. Security, pp. 152-161, 2010.

[7] 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", IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 2, February 2014.

[8] Xing Zhou, Xiaofei Tang, "Research and Implementation of RSA Algorithm for Encryption and Decryption", Department of Computer Science and Technology Harbin, China, 2013.