Comparative Study of Public-key cryptosystems in Cloud Storage

Vaishali B N
PG Student, CSE Dept
Cambridge Institute of Technology
Bangalore, India.
E-mail: vaishalinanaiah@gmail.com

Preethi S
Associate Professor
Cambridge Institute of Technology
Bangalore, India.
E-mail: preethi.srinivas2002@gmail.com

Abstract— Cloud storage is a service model in which data is maintained, managed and backed up remotely and made available to users over a network (typically the Internet). Cloud storage can provide the benefits of greater accessibility and reliability. In cloud storage different members can share that data through different virtual machines but present on single physical machine. But the thing is user don't have physical control over the outsourced data. As a result there is a need of effective method to share data securely among different users. This can be achieved using cryptography, which helps in encrypting the data to be stored in cloud storage to protect against unauthorized access. Here we introduce a public-key cryptosystem which produce ciphertexts of constant size such a way that an systematic assignment of decryption virtue for any number of ciphertexts are possible. The modernity is that one can combine a set of secret keys and make them as mini single key with holding the same ability of all the keys that are formed in each group. This compact aggregate key can be efficiently sent to others or to be stored in a smart card with little secure storage.

Keywords *— cloud storage, key-aggregate encryption, data sharing*

*****

I. INTRODUCTION

Cloud computing means storing and accessing data and programs over the Internet instead of your computer's hard drive. "Cloud computing," allow us to access data or programs over the Internet, without any hardware implementation . Were as Cloud storage means "*the storage of data online in the cloud*," wherein a company's data is stored in and accessible from multiple distributed and connected resources that comprise a cloud. Cloud storage is gaining popularity recently as many enterprise and business setting needs a lots of storage space to store data. It is also used as core technology behind many online services like email where user can store or share lots of data without concerning the space required. With the current wireless technology, data can be saved on cloud remotely and can have access to huge applications with quality services which are shared among customers. As a result any user can access the files or data that is stored in cloud storage from any corner of the world.

Considering a data privacy the traditional way is that server has a right to control access for authentication[1]. If any unauthorized person get access then whole data in cloud get expose because in cloud data from different clients are stored in single place which can be stolen easily[2].so the main concern is how securely the data can be shared in cloud. The one way is to store the data in an encrypted form, so cryptography plays a main role in cloud storage, cryptography allows the user in storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. Cryptography is the best solution in terms of confidentiality [4]. Regarding availability of data there are number of cryptographic methods that support a variety of computations on encrypted data [3]. Whenever the user is not happy with the security of cloud these users are motivated to encrypt their data with their own keys before uploading them in cloud and then these users must provide the access rights to the intended person by sending the decryption key in a secured manner example through mail.

Suppose for example if any one user puts all his data say for example pictures on cloud and as he doesn't want to expose his pictures to everyone he will encrypt it and store it in cloud. Suppose after some days if an another person request for some of the pictures in which he is present then the user either encrypt all the pictures with the single key and send that key to intended person or user should encrypt each pictures with a distinct keys and send these keys to intended person. But both the ways has disadvantage that is in the first way even the un-chosen data will get leaked and in the second way the number of decryption keys will be more suppose if there are hundred pictures then there will be hundred keys to send which required more space and more bandwidth which is inefficient.

There are two basic encryption methods: symmetric encryption and asymmetric encryption. Symmetric encryption also called private-key cryptography, in Symmetric encryption a sender encodes a message into ciphertext using a single key, and the receiver uses the same key to decode it since a single key is used for encryption and decryption this method is less secure. Were asymmetric is also called as public key cryptography. This type of cryptography uses two keys, a "private" key and a "public key," to perform encryption and decryption. The use of two

keys overcomes a major weakness in symmetric key cryptography, since a single key does not need to be securely managed among multiple users. Hence the use of public-key encryption gives more flexibility for our applications.

Therefore, the best solution for the above problem is that the user encrypts files with different public-keys, but only sends a single key to the one who request data which is stored in cloud this single is made up by adding all the decryption key into single key, the receiver will decrypt the data using this single key. By using this small single key the resource-constraint problem can be solved this consumes less amount of bandwidth.

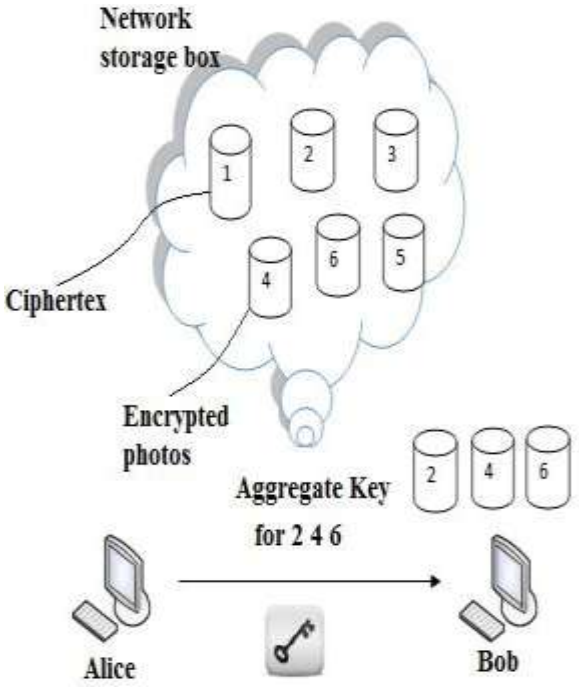

Fig 1 File sharing between sender(Alice) and receiver(Bob) were sender shares files with identifiers 2,4,6 with receiver by sending him a single aggregate key.

We introduce a special type of public-key cryptosystem called key-aggregate cryptosystem (KAC) users encrypt under an identifier of ciphertext called class i.e. ciphertexts are further classified into different classes. The key owner has the master secret key which is helpful for extracting secret key in each classes. The extracted key have can be an aggregate key for a single class, but aggregate the power of any subset of ciphertext classes. So in above scenario now the sender (Alice) can send an aggregate key to receiver (Bob) through a email and the encrypted data is downloaded from cloud storage through the aggregate key. This is shown in figure1. The sizes of ciphertext, public-key, master-secret key and aggregate kay in KAC are constant size.

In this paper, we study how to make a decryption key more powerful by allowing decryption of multiple ciphertexts, our problem statement is – "To design an efficient public-key cryptosystem scheme which support an effective decryption using a constant-size decryption key on the any subset of the ciphertexts (produced by the encryption scheme)." Previous results may achieve a similar property featuring a constant-size decryption key, but the classes need to conform to some pre-defined hierarchical relationship[5]. Our work is flexible in the sense that this constraint is eliminated, that is, no special relation is required between the classes.

II. KEY-AGGREGATE ENCRYPTION

A. FRAMEWORK

A key-aggregate encryption scheme consists of five polynomial-time algorithms as follows:

- **Setup(1^{λ}, n) :** this is executed by the data owner this setup algorithm takes no input other than the implicit security parameter 1^{λ} represents the security level parameter and n represents the number of classes, the data owner establishes the public system parameter .
- **KeyGen :** this phase is executed by data owner to generate the public or the master key pair (pk, msk). The data owner after establishing the public system parameter via setup next he generates a pk or msk via this phase.
- **Encrypt(pk, i,m) :** this phase is used to encrypt the data, the encryption algorithm takes as input the public parameters pk, an **index** i denoting the the ciphertext class, and a message m. The algorithm will encrypt message m and produce a ciphertext C such that only an intended user who satisfy access structure is able to decrypt the message.
  - Input= public key pk, index i, message m.
  - Output= ciphertext C
- **Extract (msk, S) :** this step is executed by the data owner in order to generate an aggregate single key for the number of secrete keys generated during each classes .On input the master-secret key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted ks.
  - Input= master-secrete key and set S of indices of different classes.
  - Output=aggregate key for set S denoted ks.
- **Decrypt (ks, S, i, C) :** this phase is executed by the one who has right for decryption. The decryption algorithm takes an input ks an aggregate key, the set S, an index i denoting the ciphertext class the C belongs

to and an ciphertext C , with all these inputs it is possible to generate the original message m.

- Input=aggregate-key ks, the set S, an index i ,ciphertext C
- Output= message m if i € S.

In the above framework the data owner will establish the public system via the setup phase and randomly generates the public or master-secrete key via the KeyGen phase then in Encrypt phase the message is encrypted into different classes. Then the data owner will produce an aggregate key by adding all the keys into an single key via an extract phase .The generated aggregate key is sent to the intended user via an secured email finally any user can extract an encrypted data with the decryption key provided ciphertexts's class is present in the aggregate key via decrypt phase.

III. DATA SHARING

A important application of KAC is an data sharing , it gives an efficient and flexible method to share data in cloud this scheme allows the user to share data in a confidential and selective way, with classes of chiphertexts by sending only a single and small aggregate key. The idea of data sharing is illustrated in figure 2

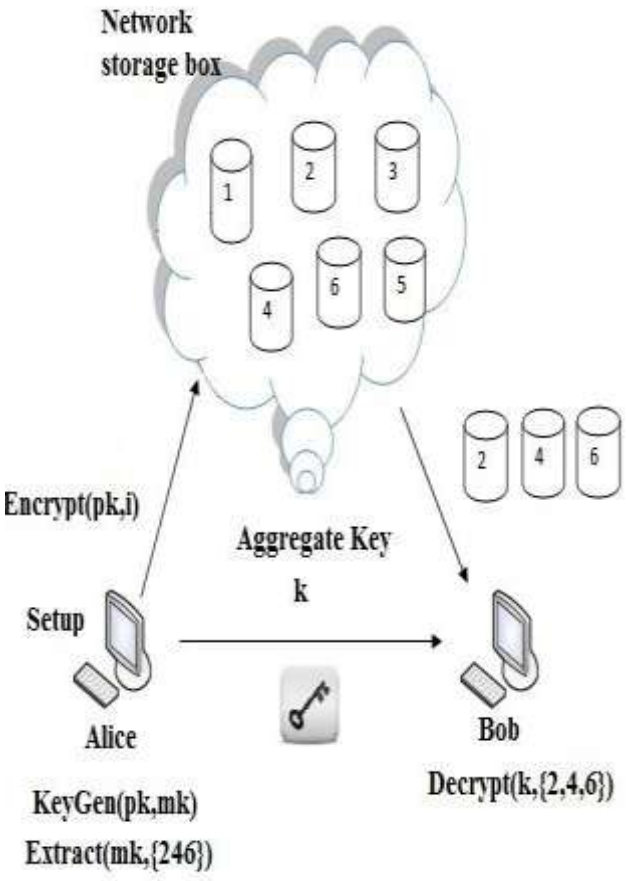

Fig 2 usage KAC for data sharing in cloud storage

In order to share the data to the cloud the Alice (sender) will first perform the setup in order to have an public system environment. Later public key (pk) or master key (mk) is generated randomly by using KeyGen phase.the public key pk is made public and master-secret key msk is kept secret only the sender will know that. Using encryption algorithm the data is encrypted and then the encrypted data is uploaded in cloud in figure 2 Alice will encrypt the data using the public-key pk and i denoting the chiphertext class, suppose if the Bob (receiver) request for file say {2,4,6} as shown in figure 2 then Alice will compute an aggregate key k for file {2,4,6} by performing Extract (msk, S) and send that aggregate key k for Bob since k is constant size key it is easy to send through secure email. After obtaining the aggregate key the Bob can download the requested encrypted data and can be easily decrypt via decrypt(ks, S , i, C) . i.e in figure 2 the Bob can decrypt the file {2,4,6} by using an aggregate key k.

IV. ARCHITECTURE

General architecture

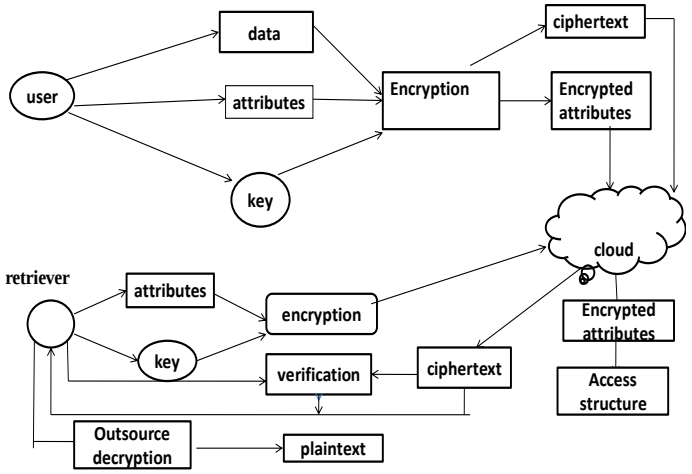

Fig 3 the general architecture of usage of cryptography in cloud

The fig 3 tells that general use of public key cryptosystem in cloud storage. The user will have a data which he likes to store it in the cloud in order to prevent the data from the unauthorized access the user will encrypt the data using public key by making use of an encryption algorithm and then this encrypted data is stored in the cloud then the authorized person access the encrypted data from cloud and then perform decryption by using his private key and obtain the original message. In this way the one can Make use of cryptography in cloud for storing the data and to prevent the data from the data leakage. Public-key cryptography are fundamental security ingredients in cryptosystem , application and protocols.

The design of our basic scheme is inspired from the collision-resistant broadcast encryption scheme [6] . Although this scheme supports constant-size secret keys, every key only has the power for decrypting ciphrertexts associated to a particular index. Thus a new Extract algorithm and the corresponding Decrypt algorithm is introduced in this paper. In order to get constant-size aggregate key and constant-size ciphertext simultaneously comes from the linear-size system parameter. Our motivation is to reduce the secure storage and this is a trade-off between two kinds of storage. The parameter can be placed in non-confidential local storage or in a cache provided by the service company. They can also be fetched on demand, as not all of them are required in all occasions.

V. RELATED WORK

This section we compare our basic KAC with other possible solutions on sharing in secure cloud storage

CRYPTOGRAPHIC KEYS IN HIERARCHY BASED CRYPTOSYSTEM

This scheme [5] aims to minimize the expense in storing and managing the secrete keys, this scheme make use of tree structure were the key of a given node can be used to derive the key for its descendant nodes hence by granting a single key that is a parent key one can able to get the key for all remaining nodes. We take the tree structure as an example Alice can first classify the ciphertext classes according to their subjects like Figure 4. Each node in the tree represents a secret key, while the leaf nodes represents the keys for individual ciphertext classes. Filled circles represent the keys for the classes to be delegated and circles circumvented by dotted lines represent the keys to be granted. Note that every key of the non-leaf node can derive the keys of its descendant nodes. In Figure 4(a), if Alice wants to share all the files in the "personal" category, she only needs to grant the key for the node "personal", which automatically grants the delegatee the keys of all the descendant nodes ("photo", "music"). 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.

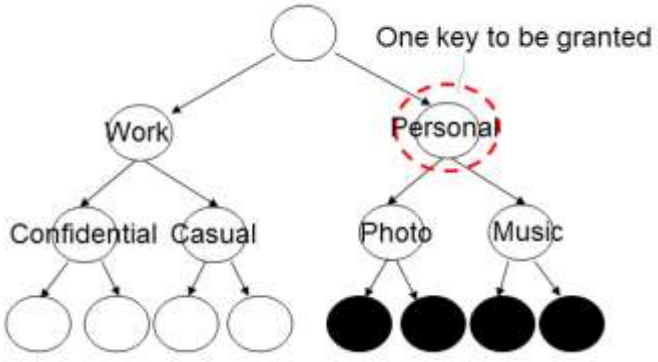

Fig 4 (a)

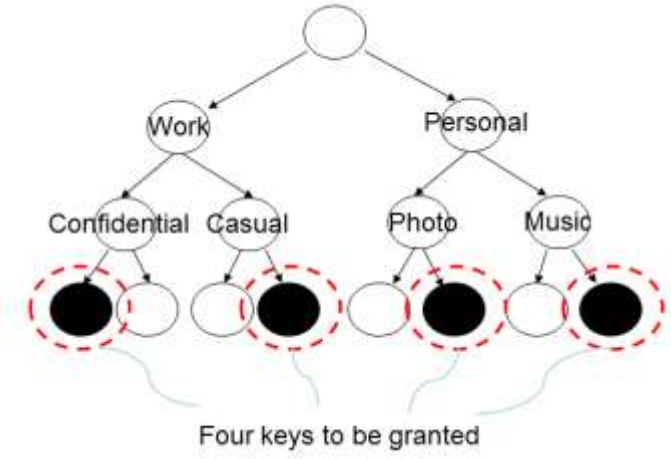

Fig 4 (b)

However this method has an disadvantage that number of keys increase with the number of branches that is as the tree structure increases the number of key also increases suppose if we want to access the data which is in last branch or in leaf node then its requires all other keys from parent node to that leaf node so as a result it requires more space to store all the keys and also it requires more bandwidth which is an efficient way. In general, hierarchical approaches can solve the problem partially if one intends to share all files under a certain branch in the hierarchy. On average, the number of keys increases with the number of branches. It is unlikely to come up with a hierarchy that can save the number of total keys to be granted for all individuals (which can access a different set of leaf-nodes) simultaneously.

VI CONCLUSION

The data privacy is one the important feature in cloud storage, as a result there are many cryptographic schemes exist which results in multiple keys for a single application. In this paper, we consider how to compress all secret keys in public key cryptosystem into an single small key called aggregate key which consist of powers of all the other key. Even though the cryptosystem produce an set of keys our approach reduce them into single key and sends only one small aggregate key to the authorized receiver, as a result the memory required for storage is very less and also it consume less bandwidth. This approach also plays an important role in sharing an decryption key in an secured manner. Our approach is more efficient than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges.

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to Prof. Preethi S. I am so deeply grateful for her help, professionalism, and valuable guidance throughout this paper. I would also like to thank to my friends and classmates who always stood by me in difficult situations

also helped me in some technical aspects and last but not the least I wish to express deepest sense of gratitude to my parents who were a constant source of encouragement and stood by me as pillar of strength for completing this work .This accomplishment would not have been possible without them. Thank you.

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.

[2] L. Hardesty, "Secure computers aren't so secure," MIT press, 2009, http://www.physorg.com/news176107396.html

[3] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy- Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362–375, 2013.

[4] B. Wang, S. S. M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," in International Conference on Distributed Computing Systems - ICDCS 2013. IEEE, 2013.

[5] S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Transactions on Computer Systems (TOCS), vol. 1, no. 3, pp. 239–248, 1983.

[6] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," in Proceedings of Advances in Cryptology - CRYPTO '05, ser. LNCS, vol. 3621. Springer, 2005, pp. 258–275.