ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Efficeint, Secure and Flexible Data Sharing in Cloud Storage

N.Tejaswi
M.Tech Student,
Department of CSE,
B V Raju Institute of Technology,
Narsapur, India.

P.BhaskaraRao
Assistant Professor,
Department of CSE,
B V Raju Institute of Technology,
Narsapur, India.

ABSTRACT :

Data sharing is playing vital role inthe cloud storage. Using cloud storage user can store and share their data very securely and efficiently. So data access security becomes the critical section to be focused. Cryptography aids the data owner to stake the data to in harmless approach. Therefore user encodes data and uploads on server. Also dissimilar encryption and decryption keys are produced for dissimilar data. The encryption and decryption keys may be dissimilar for dissimilar set of data. Merely those set of decryption keys are common that the nominated data can be decrypted. At this point a public-key cryptosystems which produce a ciphertext which is of constant size. Thus to handover the decryption rules for number of ciphertext. The variance is one can assemble a set of secret keys and mark them as minor size as a single key with holding the same capability of all the keys that are shaped in a group.

Keywords:

Cloud storage, Attribute baseencryption, Identity base encryption, Cloud storage, data sharing, key aggregate encryption.

I.INTRODUCTION :

In current era Data sharing is a significant functionality in cloud storage. For instance, bloggers can let their associate's opinion a subset of their cloistered pictures; an enterprise may fund her employee's admission to a quota of sensitive data. The thought-provoking problem is in what way we can efficiently share encrypted data. Obviously users can download the encrypted data from the storage, decrypt them, then direct them to others for sharing, but it drops the value of cloud storage. Therefore the users should be capable to give the access rights of the sharing data to others so that they can access these data from the server unswervingly.

Cloud computing is widely increasing technology; customers. As increase in outsourcing of data the cloud computing serves does the management of data [1].Its flexible and cost optimizing characteristic motivates the end user as well as enterprises to store the data on cloud. The insider attack is one of security concern which's needs to be focused. Cloud Service provider need to make sure whether audits are held for users who have physical access to the server. As cloud service provider stores the data of different users on same server it is possible that user's private data is leaked to others. The public auditing system of data storage security in cloud computing provides a privacy-preserving auditing protocol [2].It is necessary to make sure that the data integrity without compromising the anonymity of the data user. To ensure the integrity the user can verify metadata on their data, upload and verify metadata [3].

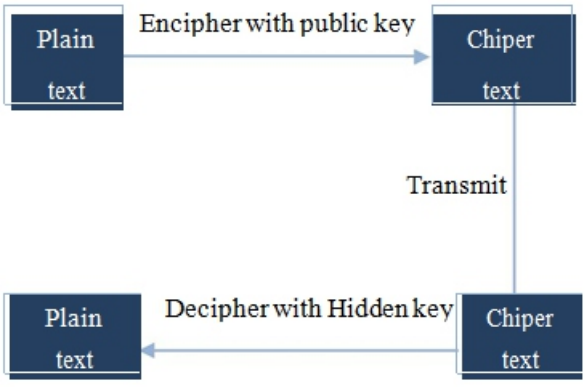


Fig. 1. Cryptosystem

Then there are 2 critical ways data can be saved on cloud remotely and can have access to huge applications with quality services which are shared among

1.Alice encrypt whole picture with one encryption key and give secret key to bob.

2.Encrypt all picture with special key and send

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

organization grant permission for this personal data. But problem is sharing of the encrypted data and effectiveness of that task. Take another example of dropbox for explanation. Alice can collect personal picture on dropbox and she thinks no one can watch her photos. Due data loss possibility Alice does not feel secure and she encrypts all picture using own key before uploading. Another day her friend wants all pictures of the year in which bob appear. Alice use share option of dropbox but problem is that how to delegate decryption rights to bob.

II.LITERATURE SURVEY:

In this section basic KAC scheme is compared with other possible solutions on sharing in secure cloud storage.

a) Cryptographic Keys for a Predefined Hierarchy:

Cryptographic key assignment schemes works on the basis of minimize the expense in storing and managing secret keys for general cryptographic use by using a tree structure [5]. By using ranked tree arrangement, a key for a given division can be used to originate the keys of its child nodes. This can resolve the problem somewhat if one plans to share all files under a certain branch in the pyramid which otherwise means that the number of keys increases with the number of branches. So it is corresponding secret key to bob. Sharing information is main task of cloud. For example, bloggers can want their personal photo, difficult to create a hierarchy that can save the number of total keys to be granted for all individuals concurrently.

b) Compact Key in Identity-Based Encryption (IBE):

In this encryption, there is a trusted party called private key generator in IBE which holds a master-secret key and gives a secret key to each user with respect to the user identity. The encryptor can take the public parameter and a user identity to encrypt a message [7]. The receiver can decrypt this ciphertext by his secret key. Some tried to build IBE with key aggregation. But their key-aggregation comes at the expense of O (n) sizes for both ciphertext and the public parameter, where n is the number of secret keys. This greatly increases the costs to store and transmit ciphertext.

c) Attribute-based encryption (ABE):

This scheme maintains each ciphertext to be associated with an attribute, and the master-secret key holder can extract a secret key for a policy of these attributes so that a ciphertext can be decrypted by this key. But the size of the key often increases linearly with the number of attributes it encompasses, or the ciphertext-size is not constant [8].

III. PROPOSED SYSTEM:

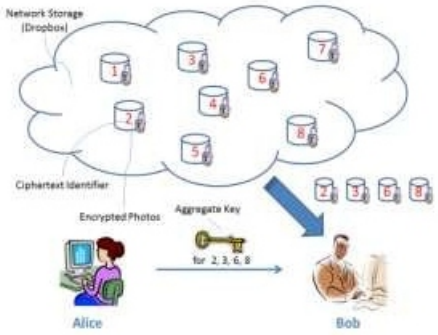


Fig. 2. Alice shares files with identifiers 2, 3, 6 and 8 with Bob by sending him a single aggregate key.

In modern cryptography, a basic problem we often study is about leveraging the secrecy of a small piece of knowledge into the ability to perform cryptographic functions (e.g. encryption, authentication) multiple times. In this paper, we study how to make a decryption key more powerful in the sense that it allows decryption of multiple ciphertexts, without increasing its size. Specifically, our problem statement is: "To design an efficient public-key encryption scheme which supports flexible delegation in the sense that any subset of the ciphertexts (produced by the encryption scheme) is decryptable by a constant-size decryption key (generated by the owner of the master-secret key)."

We elucidate this problem by presenting a singular type of public-key encryption which we call key-aggregate cryptosystem (KAC). Now KAC, users encrypt a message not only below a public-key, furthermore below an identifier of cipher text termed class. That means the ciphertexts are more considered into dissimilar classes. Generally the key owner grips a master-secret called master-secret key, which can be employed to extract secret keys for dissimilar classes. Additional vitally, the extracted key have can be an aggregate key which is as dense as a

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of cipher text classes. By means of our answer, Alice can just send Bob a single aggregate key via a protected e-mail. Bob can download the encrypted photos from Alice's

Dropbox space and then use this cumulative key to decrypt these encrypted photos. The situation is portrayed in Figure 1.

A. KEY-AGGREGATE ENCRYPTION:

A key aggregate encryption has five polynomial-time algorithms as

Setup Phase

The data owner executes the setup phase for an account on server which is not trusted. The setup algorithm only takes implicit security parameter.

Key Gen Phase

This phase is executed by data owner to generate the public or the master key pair (pk, msk).

Encrypt Phase

This phase is executed by anyone who wants to send the encrypted data. Encrypt (pk, m, i), the encryption algorithm takes input as public parameters pk, a message m, and I denoting ciphertext class. The algorithm encrypts message m and produces a ciphertext C such that only a user that has a set of attributes that satisfies the access structure is able to decrypt the message.

Input= public key pk, an index i, and message m Output = ciphertext C.

Extract Phase

This is executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegate. Input = master-secret key mk and a set S of indices corresponding to different classes. Outputs = aggregate key for set S denoted by kS.

Decrypt Phase

This is executed by the candidate who has the decryption authorities. Decrypt (kS, S, i, C), the decryption algorithm takes input as public parameters pk, a ciphertext C, I denoting ciphertext classes for a set S of attributes.Input = kS and the set S, where index i = ciphertextclass.Outputs = m if i element of S.

KAC in meant for the data sharing. The data owner can share the data in desired amount with confidentiality. KCA is easy and secure way to transfer the delegation authority. The aim of KCA is illustrated in Fig. 2.

1)For sharing selected data on the server Alice first performs the Setup.

2)Later the public/master key pair (pk, mk) is generated by executing the KeyGen. The msk master key is kept secret and the public key pk and param are made public.

3)Anyone can encrypt the data m and this data is uploaded on server. With the decrypting authority the other users can access those data.

4)If Alice is wants to share a set S of her data with a friend Bob then she can perform the aggregate key KS for Bob by executing Extract (mk, S).

5)As kS is a constant size key and the key can be shared through secure e-mail. When the aggregate key has got Bob can download the data and access it.

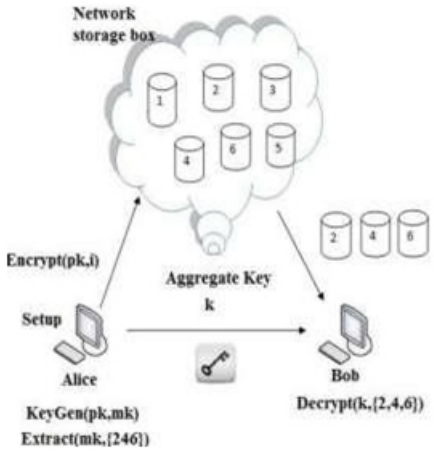
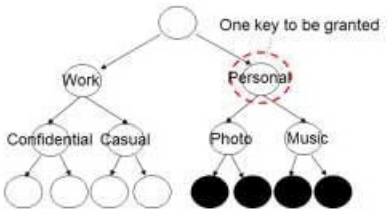


Fig 3 Use of KAC for data sharing

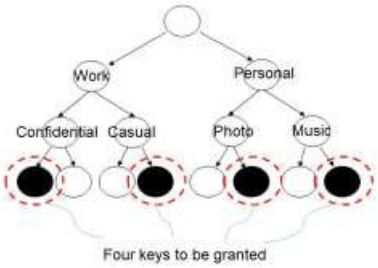
ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

We take the tree structure as an example. Alice can first classify the ciphertext classes according to their subjects like Fig.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.



(a)

In Fig. 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.



(b)

As shown in Fig.4 (b), if Alice shares her demo music at work ("work"→"casual"→"demo" and "work"→"confidential"→"demo") with a colleague who also has the rights to see some of her personal data, what she can do is to give more keys, which leads to an increase in the total key size.

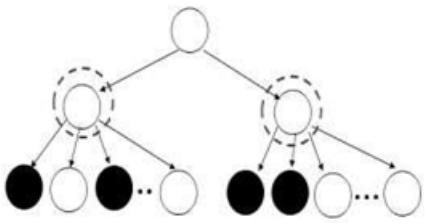


Fig. 5. Key assignment in our approach

Fig.5 shows the flexibility of our approach. We achieve "local aggregation", which means the secret keys under the same branch can always be aggregated. We use a quaternary tree for the last level just for better illustration of our distinctive feature. Our advantage is still preserved when compared with quaternary trees in hierarchical approach, in which the latter either delegates the decryption power for all 4 classes (if the key for their parent class is delegated) or the number of keys will be the same as the number of classes.Patient controlled encryption has been studied in [2]. In the PCE the health record if divided into hierarchical representation depend on the different ontologies and the patients are the parties who create and store secret key. When there is need of accessing record ,a patient will release secret key for the access of record to the healthcare. In the Benaloh et al. [2], proposed three solution.

1.Symmetric key PCE for fixed hierarchy(tree based method)

2.Public key PCE for constant hierarchy(the IBE analogof folklore method).

3.RSA based symmetric key PCE for flexible hierarchy.

Each patient can create her own hierarchy in Fig.6 as per her self need, or follows the set of the catagories recommended by the electronic medical record system such as xray, medications and so on. If patient wants to give access right to her doctor, she choose any subset of different categories and give a single key, from which key total categories computed. Thus, we can basically choose anyhierarchy, useful when the hierarchy can be complex. Finally single healthcare deals with many patient and the data of the patient is possible to stored on the cloude because of his large size, compact size key and easy key management are of the paramount.

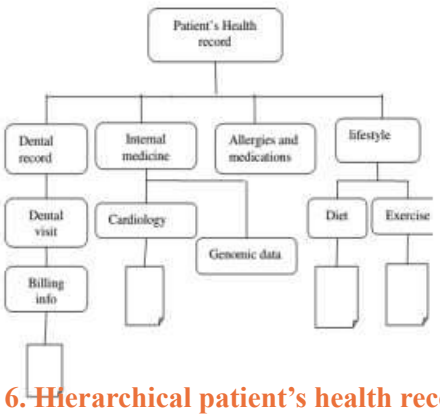


Fig. 6. Hierarchical patient's health record

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

IV. CONCLUSION:

To share data flexibly is vital thing in cloud computing. Users prefer to upload there data on cloud and among different users. Outsourcing of data to server may lead to leak the private data of user to everyone. Encryption is a one solution which provides to share selected data with desired candidate. Sharing of decryption keys in secure way plays important role. Public-key cryptosystems provides delegation of secret keys for different ciphertext classes in cloud storage. The delegatee gets securely an aggregate key of constant size. It is required to keep enough number of cipher texts classes as they increase fast and the ciphertext classes are bounded that is the limitation.

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].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.

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

[4] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," SIAMJournal on Computing (SIAMCOMP), vol. 36, no. 5, pp. 1301–1328, 2007.

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

[6]D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," in Proceedings of Advances in Cryptology - EUROCRYPT '03, ser.LNCS, vol. 2656. Springer, 2003, pp. 416–432.

[7]F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key," Proc. Pairing-Based Cryptography Conf. (Pairing '07), vol. 4575, pp. 392-406, 2007.

[8]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.