ISSN 2347 - 3983

**International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.6, Pages : 127- 130 (2015)**
*Special Issue of NCTET 2K15 - Held on June 13, 2015 in SV College of Engineering, Tirupati*
***http://warse.org/IJETER/static/pdf/Issue/NCTET2015sp26.pdf***

# COMPACT KEY TO ACCESS SHARED RESOURCES SECURELY IN CLOUD STORAGE

**M. Keerthi Priya[1], M. Vijayasanthi[2]**
[1] M.Tech CSE Student, S.V.Engineering College for Women, Tirupati, AP ,India
kp.mnp91@gmail.com
[2] Assistant Professor, Dept. of CSE, S.V.Engineering College for Women, Tirupati,AP, India,
vsmaddelasanthi@gmail.com

## ABSTRACT

Cloud expands their IT services in a wide range to increase their potentiality in worldwide, among them data outsourcing is one of the major functionality in cloud storage. Maintaining the security, privacy and effective way of data sharing becomes a critical job for cloud providers. To overcome this problem, we newly presented a symmetric key cryptographic approach which generates a constant size cipher text. An individual user can effectively decrypt the encrypted data from shared cloud. A compact key contains set of secret keys which is generated by the key distribution center on the basis of file and user's privileges who want to access the data from the cloud storages. We can able to store this compact key conveniently in limited stored space.

**Key words:** Cloud storage,compact key encryption, one time password, error message.

## 1. INTRODUCTION

Cloud storage is nowadays very popular storage system. Cloud storage is storing of data off-site to the physical storage which is maintained by third party. Cloud storage is saving of digital data in logical pool and physical storage spans multiple servers which are managed by third party. They are answerable for maintaining data,access and its physical environment should be safe. Instead of storing data to the hard drive or any other local storage, we save data to remote storage which is accessible from anywhere and anytime. It reduces efforts of carrying physical storage to everywhere. By using cloud storage we can access information from any computer through internet which omitted limitation of accessing information from the same computer where it is stored.

While considering data privacy, we cannot rely on traditional techniques of authentication, because unexpected privilege escalation will expose all data. The solution is to encrypt data before uploading to the server with user's own key.user can share data from anywhere and anytime to anyone. For example, an organization may grant permission to access parts of sensitive data to their employees. But challenging task is to share that how the data is encrypted. When the user follows traditional way to download the encrypted data from storage, it decrypts the data and send or share with others, but it loses significance of secured sharing mechanism in cloud storage.

Cryptography technique can be applied in a two major ways- one is a Symmetric key encryption and other is Asymmetric key encryption. In symmetric key encryption, same keys are used for encryption and decryption. By contrast, in asymmetric key encryption different keys are used, the public key for encryption and private key for decryption. Using asymmetric key encryption is more flexible in our approach. This can be illustrated by following example.

Suppose Alice puts all data on Box.com and she does not want to expose her data to everyone. Due to data leakage possibilities she does not trust on privacy mechanism provided by Box.com, so she encrypt all data before uploading to the server. If Bob asks her to share some data, then Alice use share function of Box.com. But the problem now how encrypted data is shared. There are two ways:

1. Alice encrypts the data with single secret key and shares that secret key directly with the Bob.
2. Alice can encrypt data with distinct keys and send Bob corresponding keys to Bob via secure channel to decrypt the data.

In the first approach, unwanted data also get exposed to the Bob, which is inadequate. In the second approach, no. of keys is as many as no. of shared files, which may be a hundred or thousand as well as transferring these keys require secure channel and storage space which can be expensive.

Therefore best solution to above problem is Alice encrypt the data with distinct public keys, but sends single decryption key of constant size to Bob to decrypt the required data. Since the decryption key should be sent via secure channel , kept secret small size is always enviable.

## 2. RELATED WORK:

In latest cryptography area, a fundamental 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) several times. we see how to use decryption key more useful here, it means it allows decryption of multiple cipher texts, without increasing its size. Here we can say it as "To design an efficient public-key encryption scheme which supports flexible delegation 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)."

We now change this problem by using a different type of public-key encryption key-aggregate cryptosystem (KAC). Users can encrypt a message not only under a public-key, but also under an identifier of cipher text called class. Again cipher texts are further divided into various parts.

There is a key called mssaster-secret key, it is used to extract secret keys for various different classes. Mainly 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. The decryption power for any subset of cipher text classes. With our solution, Alice can simply send Bob a single aggregate key via a secure e-mail. Now Bob can download the encrypted photos from Alice's Drop box space and then use the same aggregate key to decrypt these encrypted photos.

## 3. KEY-AGGREGATE CRYPTOSYSTEM

This system basically includes five algorithmic steps as follows-

The data owner founds the public system parameter by using **Setup** and forms a public/master-secret key pair by using **KeyGen**. Messages can be encrypted by using **Encryption method.**

The owner can use the master key to produce an aggregate decryption key for a set of cipher text classes by using **Extract**. The formed keys can be passed to Receivers safely with the help of e-mails. Finally, any user with an aggregate key can **Decryp**t any cipher text..

**Setup(1λ , n):** Data owner executes Setup to create an account on an un-trusted server. With input as security level parameter 1λ and the number of cipher text classes n, it outputs the public system parameter.

**KeyGen:** Data owner executes KeyGen to randomly generate a public/master-secret key pair (pk ,msk)

**Encrypt(pk, i, m):** Anyone can execute this step who wants to encrypt data with input a public-key pk, an index i denoting the ciphertext class, and a message m, which outputs a ciphertext C.

**Extract(msk, S):** Executed by the data owner to handover the decrypting power for a certain set of ciphertext classes to a Receiver . 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 by Ks.

**Decrypt(Ks, S, i, C):** executed by a Receiver who received an aggregate key Ks generated by Extract. On input Ks, the set S, an index i denoting the ciphertext class the ciphertext C belongs to, and C, it outputs the decrypted result m if i є S.

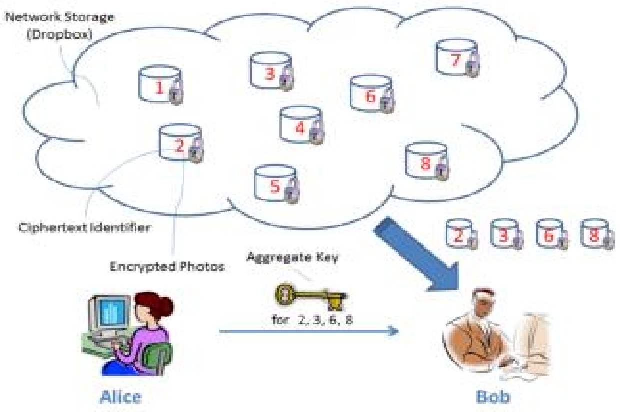


**Figure 1:** Alice shares files with identifiers 2, 3, 6 and 8 with Bob by sending him a single aggregate key.

### 3.1 Other Schemes:

KAC scheme is compared with other possible solutions on sharing in secure cloud storage.

### A) Cryptographic Keys for a Predefined Hierarchy:

.

The schemes which are used in this are to decrease the expense of storage space of secret key and to handle cryptographic key of them by using structure of tree. By using hierarchical tree structure, a key for a given branch can be used to derive the keys of its descendant nodes. This can solve the problem partially if one intends to share all files under a certain branch in the hierarchy which alternatively means that the number of keys increases with the number of branches. So it is difficult to create a hierarchy that can save the number of total keys to be granted for all individuals simultaneously

### B) Compact Key in Symmetric-Key Encryption:

The above method is used to produce a secret value instead of a pair of public/ secret keys [6]. It is made for the symmetric-key setting, then the encryption gets the relative secret keys to encrypt data. Thus it is uncertain how to apply this idea for public key encryption scheme.

### C) Attribute-Based Encryption (ABE):

It maintains each cipher text 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 cipher text can be decrypted by this key. But the size of the key often increases linearly with the number of attributes it encompasses, or the cipher text-size is not constant [8].

| | **Not constant** | **Cipher text size** | **Encryption Type** |
|---|---|---|---|
| **Key assignment schemes for predefined hierarchy** | Non constant | Constant | Symmetric or public key |
| **Symmetric key encryption with compact key** | Constant | Constant | Symmetric key |
| **IBE with compact key** | Constant | Non Constant | Public key |
| **Attribute based Encryption** | Non Constant | Constant | Public key |
| **KAC** | Constant | Constant | Public key |

**TABLE** 1 : comparison of all can be summarized in following table:

## 4. ONE TIME PASWORD:

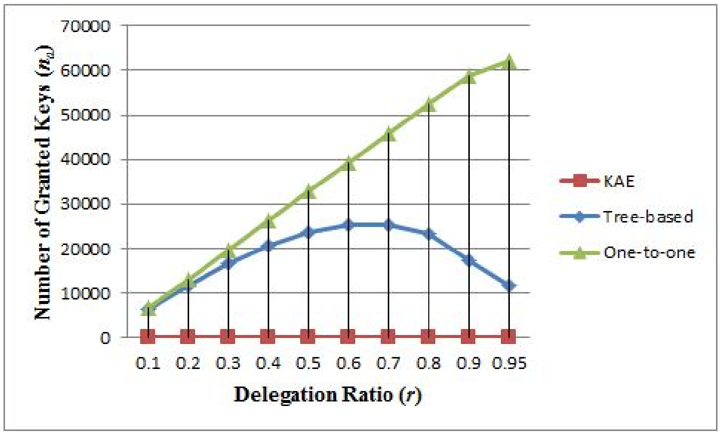
In above schemes we are getting full privacy.so I am trying to implement one new method.In drop box sharing in multiple files (ex: - doc, txt, photos, mp3, videos) without privacy breaches occurring is a major problem that is suppose for example.
"A" is a person uploaded/drag and drop some file to drop box.
**Note:** upload the file are stored in the drop box but it is not in encrypted form. If anybody knows my user id and password of my drop box account they can easily see all the files (which are confidential one also). If we encrypt the file in uploading itself, it will ask a key to view, download that particular file if any person asks.
Consider "A" uploaded 15 photos of his birthday celebration to drop box.
1."A" shares 10 photos to "B" who is his close friend Again "A" shares 5 photos to "C" who is his just friend of him.
2. In this scenario our application will generate individual keys to "B" and "C" to view those photos.
3.So, here we identify a problem that "B" has privilege to see10 photos from shared folder in drop box, here our application will generate a key that will decrypts 10 photos from the shared folder but not the entire 15 photos, Similarly "C" has privilege to see 5 photos from drop box.
If both "B" and "C" exchange the keys what will happen.
To optimize this we are generating ONE TIME PASSWORD (OTP) to access the files share is shared in the drop box.

If customer apply that OTP to view or to download any file from drop box. Our application verifies that applied OTP is intensely generated to the accessing user only or not

If the OTP is generated to intended customer means the customer can access the file.Otherwise application shows an error message to customer that "**this key is intensely not generated for you and my applying this key is expired**".

OTP will expire in 5 minutes. So, user has to apply this key within 5 minutes span of time. Otherwise it will expire.
Generated key has to apply Intended users only but not others.
If a intend user shared the generated key with others. Error Count will increment by applying the key.
Applicant can allow to access up to 2 error count. In case error count is 3, our system automatically blocks the user access.

5. RESULT:



Number of granted keys (na) required for different approaches in the case of 65536 classes of data

## 6. CONCLUSION:

Thus data privacy and security is maintained by designing a public key cryptosystem called as Key Aggregate Cryptosystem (KAC). This KAC helps user to share their data partially over cloud with constant size key pair of public-master keys and also receiver can decrypt this data with single constant size aggregate key. We are applying one time password and error message to get privacy in cloud storage.

## REFERENECE:

[1] 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 Transaction on Parallel and Distributed System, vol. 25, no. 2, February 2014 .

[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, Feb. 2013.

[3] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity- Management for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.

ISSN 2347 - 3983

**International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.6, Pages : 127- 130 (2015)**
*Special Issue of NCTET 2K15 - Held on June 13, 2015 in SV College of Engineering, Tirupati*
***http://warse.org/IJETER/static/pdf/Issue/NCTET2015sp26.pdf***

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

[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] J. Benaloh, "Key Compression and Its Application to Digital Fingerprinting," technical report, Microsoft Research, 2009.

[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.

[9] 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

[10] L. Hardesty, "Secure computers aren't so secure," MIT press, 2009, http://www.physorg.com/news/176107396.html
.

[11] 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.

[12] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng, "Dynamic Secure Cloud Storage with Provenance," in Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442–464.

[13] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamicand Efficient Key Management for Access Hierarchies," ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.

[14] 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.

[15] 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.

[16] V. Goyal, O. Pandey, A. Sashai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data," in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06). ACM, 2006, pp. 89–98.

[17] 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.

[18]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, 1989, pp. 316–322.

[19] C.-K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional Proxy Broadcast Re-Encryption," in Australasian Conference on Information Security and Privacy (ACISP '09), ser. LNCS, vol. 5594. Springer, 2009, pp. 327–342.

[20] 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, pp.316–332.

[21] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Transactions on Information and System Security (TISSEC), vol. 9, no. 1, pp. 1–30, 2006.

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

[23] L. B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, and R. Dahab, "TinyTate: Computing the Tate Pairing in Resource- Constrained Sensor Nodes," in Proceedings of 6th IEEE International Symposium on Network Computing and Applications (NCA '07). IEEE, 2007, pp. 318–323.

.