KAC SCHEME FOR SCALABLE DATA SHARING IN CLOUD STORAGE

[1]S. R. Jankar, [2]S. R. Khandelwal
[1]ME Student, [2]Professor
[1]ME Dept. of Computer Science & Engineering,
[1]M.S. Bidve Engineering College Latur, Maharashtra, India

Abstract : Cloud computing is information technology paradigm or model in which network of remote servers are hosted on internet to store maintain and process data rather than a local server (PC). Cloud computing provide different services, facility such as application, server, cloud storage. Nowadays we can see rise of cloud storage as it allows to store large documents, it also allows consumers to use applications without installation and access their files as any computer with internet access. That means it reduces cost of installation and storage. Cloud storage plays an important role that is data sharing. But when it comes to privacy and security, cloud storage is not secure for data sharing. So users prefer to encrypt data before uploading on cloud. Key aggregate cryptosystem (KAC) is public key encryption scheme which supports flexible delegation. Secret key holder produces aggregate key for flexible choices of cipher texts. This aggregate key holds the power of all aggregated key. It is compact but powerful key. This paper demonstrates cryptography techniques for scalable data sharing in cloud storage using this constant size aggregate key.

IndexTerms **- Cloud storage, data sharing, data security, key aggregate cryptosystem (KAC), cryptography.**

I. INTRODUCTION

Cloud computing has abundant of benefits to the cloud user so cloud system can be used to empower data sharing capabilities. Nowadays enter prizes are engaged in this effort how to scalable the data sharing. We can see there is rise in demand of data outsourcing especially in business quarter, which assists in the strategic management of corporate data. It is also based on core technology behind many online services for personal applications. Because of today's wireless technology user can access almost all data or emails on their mobile or PC at anywhere at any time.

Cloud storage is online data storage where the digital data stored in logical pools, and physical storage spans physical multiple server (and often locations) and physical environment is typically owned and managed by hosting company. This is the cloud storage provider's responsibility to keep the data available and accessible and also keep the physical environment protected and running. People buy the storage capacity from cloud storage provider to store their data. According to cloud user it may be owned by individual or enterprise cloud storage categorized into personal and public cloud storage. There are also another two types private and hybrid cloud storage. Difference between public and private is in public, cloud storage provider fully manages enterprise's cloud while in private only the infrastructure is managed by storage provider. Hybrid is the combination on private and public cloud storage. Cloud storage services may accessed through co-located cloud compute service, a web service application programming interface (API) or by applications that utilizes the API such as cloud desktop storage, a cloud storage gateway or web based content management system.

If we consider the data privacy and security a traditional way of giving an access control after the authentication. It means for security and privacy user is totally relying on cloud server. It means any unexpected privilege escalation will expose all data. Because of shared tenancy of cloud computing environment situation becomes worse. It means data on target virtual machine can be stolen by co-related virtual machine to the target virtual machine as they are situated on same physical machine. Cloud user encrypts their data with their own keys. At the time of data sharing he or she share key so key can be shared in two way-

1. Sender will encrypt all files with a single key and gives matching secret key to receiver, but this method is not adequate if sender wants to share only few files but receiver can access all files.
2. Sender will encrypt each data with distinct keys and sends the matching secret keys to receiver which he wants, but this is not appropriate if sender wants to share 1000 files she or he has to give 1000 matching secret keys which increases time and storage complexity and also cost.

Encryption comes in two flavors-

1. Symmetric encryption: In this if sender wants data to be originated from third party then he or she has to give her encryption key to encryptor which is not desirable.

2. Asymmetric encryption: In this encryption and decryption keys are different so it is suitable for our application.

In the cloud storage efficient public key encryption scheme which support flexible delegation in the sense that any subset of the cipher texts is decryptable by a constant-size decryption key. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of ciphertext set in cloud storage [1].

In KAC user can encrypt message not only under a public key but also under an identifier of cipher texts called class. The ciphertexts are further categorized into different classes. The key owner holds a master-secret key called master secret key. The extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregate the power of many such keys, i.e., the decryption power of any subset of cipher text classes.

Cryptography helps the data owner to share the data to in safe way. Cryptography is the practice and study of hiding information. It is the Art or Science of converting a plain intelligible data into an unintelligible data (i.e. encryption) and again retransforming that message into its original form (i.e. decryption). It provides Confidentiality, Integrity, and Accuracy [7].

A cryptographic solution, with proven security relied on number-theoretic assumptions is more desirable Data sharing is important functionality in cloud storage [8]. For example bloggers can let their friends view private data or an enterprise may grant their employee access to important data. But the problem is how to effectively share encrypted data. Of course users can download the encrypted data from the storage, decrypt them and then send them to others for sharing, but it loses the value of cloud storage. So user should be able to give access rights of sharing data to others so that they can access these data from server directly. Cloud system is shown in figure 1, which depicted data sharing in cloud storage.

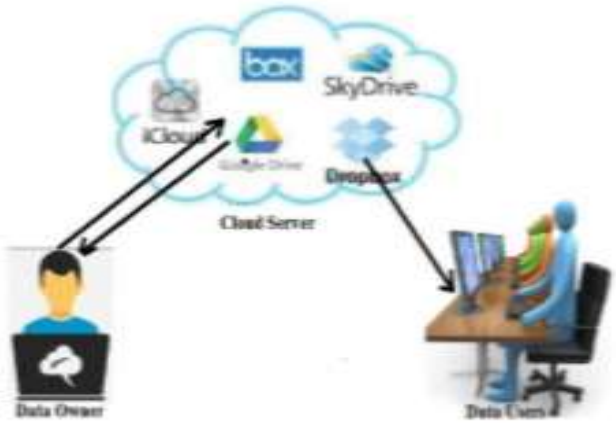

Figure 1. Cloud System

II. RELATED WORK

Key assignment scheme aim to minimize the expense in storing and managing secret keys for general cryptographic use. Only hash functions are used for a node to derive a descendant's key from its own key. The space complexity of the public information is the same as that of storing hierarchy and is asymptotically optimal; the private information at a node consists of a single key associated with that node and updates are handled locally in the hierarchy [2].

Presented an encryption scheme which is originally proposed for concisely transmitting large number of keys in broadcast scenario. In this paper build an efficient system that allows patients both to share partial access rights with others, and to perform searches over their records. They formalize the requirements of a Patient Controlled Encryption scheme, and give several instances, based on existing cryptographic primitives and protocols, each achieving a different set of properties. The encryptor needs to get the secret keys to encrypt data which is not suitable for many applications. It is unclear how to apply this method for public key encryption scheme [3].

Identity-based encryption (IBE) is a type of public-key encryption in which the public-key of a user can be set as an identity-string of the user (e.g., an email address). There is a trusted party called private key generator (PKG) in IBE which holds a master-secret key and issues 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. The recipient can decrypt this ciphertext by his secret key [4].

Attribute-based encryption (ABE) allows 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 if its associated attribute conforms to the policy. However, the major concern in ABE is collusion-resistance but not the compactness of secret keys. Indeed, the size of the key often increases linearly with the number of attributes it encompasses or the ciphertext-size is not constant [5].

III. PROPOSED SYSTEM

This project consists of five algorithms which are used to perform the above operations. These algorithms are as follow:

Setup: The account is created on the untrusted server for sharing of data. This account is generated by data owner. **KeyGen**: This algorithm is use for the generation of public key. The data owner generates a public secrete key to encrypt the data over cloud. He also creates an aggregate key to access the block of ciphers of limited size.

Encrypt: This algorithm encrypts the data provided by the data owner by using the secrete key. This encrypted data is then share among the cloud.

Extract: The aggregate key is use for extracting the particular block of the ciphers from the cipher file. But other encrypted data remains secure.

Decrypt: The encrypted data is then decrypted by using the same secrete key which is use for encryption. As the above figure 2 shows, the key assignment is done in dynamic way. The aggregate key is use to decrypt only those ciphers which user wants. This key will not decrypt the other remaining ciphers. The main encryption and decryption is done by the secrete key. If any user enters the wrong secrete key or wrong aggregate key then the user contains will be blocked by the data owner. And the information which that user tries to retrieve is then added into non confidential storage. Only data owner can unblock that user contents and he may transfer the information from non-confidential storage to confidential storage. The user can only access the data on cloud if he has secreted key and the aggregate key, otherwise he will be block forever.

MD5: MD5 algorithm can be used as a digital signature mechanism. It takes as input a message of arbitrary length and produces as output a 128 bit fingerprint or message digest of the input. It intended where a large file must be compressed in a secure manner before being encrypted with a private key under a public-key cryptosystem such as PGP. The main steps of MD5 algorithm to generate the hash value are given as below:

1. Append padding bits so message becomes 448 modules 512.
2. Append length to the input message so that it becomes exact 64-bit in length.
3. Initialize the 32 bit MD buffer A, B, C, D.
4. Process the message in 16-word block,

F (X, Y, Z) = XY or not (X) Z
G (X, Y, Z) = XZ or Y not (Z)
H (X, Y, Z) = X xor Y xor Z
I (X, Y, Z) = Y xor (X or not (Z))

5. The final digest message will be stored in buffer.

IV. SYSTEM ARCHITECTURE

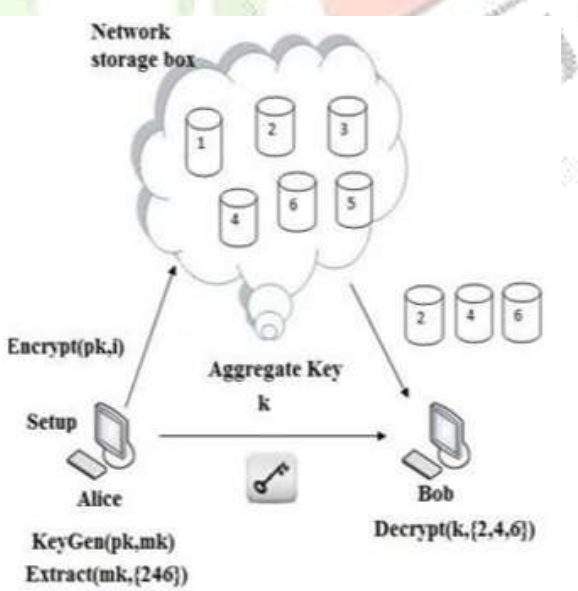

Figure 2: System Architecture.

KAC is developed for the secure data sharing. Data owner can send his data with secure and confidently. KAC is very secure and reliable method for sharing data in cloud computing. The aim of KCA is illustrated in Figure 2. For sharing the selected file with user cloud service provider first checks the rights of particular user. If he having rights for that file then only user can perform particular

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

1. User Module

Set (C) = $\{C_0, C_1, C_2, C_3, C_4\}$

C_0= User Registration

C_1 = Upload file.

C_2 =Generate secret key.

C_3= Encrypt or decrypt files.

C_4= Download file.

2. Cryptographic modules

Set (G) = $\{g_0, g_1, c_2, c_3, c_4\}$

G_0=Secrete key generation.

G_1= Encrypt and share file.

G_2=Decrypt file using secrete key.

3. Extraction Modules

Set (E) = $\{e_0, e_1, e_2, c_3, g_3\}$

E_0=Receive key.

E_1=Extract key data.

E_2=Decrypt Data.

4. Union and Intersection

Set (G) = $\{g_1, g_2, g_3, c_2, c_3, c_4\}$

Set (E) = $\{e_0, e_1, e_2, c_3, g_3\}$

CIntersection$G = c_2, c_3, c_4$

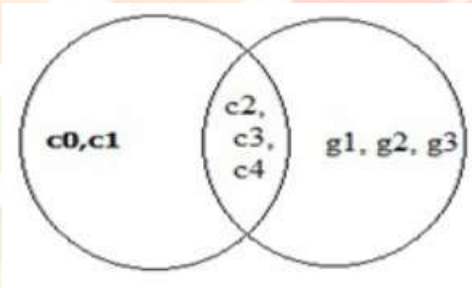

Figure 3: Set Representation

V. RESULT ANALYSIS

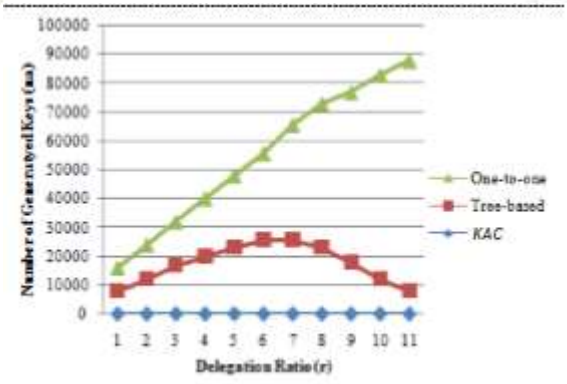

Figure 4: Number of granted keys (na) required for different approaches

Looking at the performance analysis, a comparison of the number of keys granted between three methods is shown in the Figure 4. Here we can see, in one by one key granting, the number of granted keys will be same as the number of ciphertext delegate classes.

With the tree based structure, the number of keys granted can be saved depending on the delegation ratio. Whereas in KAC scheme, it is efficiently implemented with the fixed size aggregate key. The constant-size aggregate key and constant-size ciphertext is the greatest advantage of this scheme. The Key Aggregate Cryptosystem (KAC) is the most efficient scheme when compared to the tree based structure and one by one granting of the keys.

VI. CONCLUSION

To share data flexibly and securely in cloud computing is vital thing. Users always prefer to upload there data on cloud and share the uploaded data among different users. The main drawback of cloud computing is the security issue. Cryptography is a one of best solution which provides security to share selected data with desired cloud data users. Sharing of decryption keys in secure way plays important role. The proposed Public-key cryptosystems provides delegation or leader key of secret keys for different cipher text classes in cloud storage. The proposed system creates user group and can share file to all group members simultaneously. Cryptographic schemes are getting more versatile and trustable, it involve multiple keys for a single application. In this paper, we consider how we can ”compress” secret keys by combining the multiple keys which support delegation or aggregation of secret keys for different cipher text classes in cloud storage system.

REFERENCES

[1] J. Benaloh et.al, “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.

[2] S. S. M. Chow et.al, “Practical Leakage-Resilient IdentityBased Encryption from Simple Assumptions,” in ACM Conference on Computer and Communications Security, 2010, pp. 152–161.

[3] M. Chase and S. S. M. Chow, “Improving Privacy and Security in Multi-Authority Attribute-Based Encryption,” in ACM Conference on Computer and Communications Security, 2009, pp. 121–130.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] L. Hardesty, “Secure computers aren’t so secure,” MIT press, 2009, http://www.physorg.com/news176107396.html

[9] D. Boneh, R. Canetti, S. Halevi, and J. Katz, “ChosenCiphertext Security from Identity-Based Encryption,” SIAM Journal on Computing (SIAMCOMP), vol. 36, no. 5, pp. 1301–1328, 2007.

[10] F. Guo, Y. Mu, and Z. Chen, “Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key,” in Proceedings of Pairing-Based Cryptography (Pairing ’07), ser. LNCS, vol. 4575. Springer, 2007, pp. 392–406.

[11] 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.