Key Aggregate Cryptosystem for Scalable Data Sharing In Cloud Storage

Mr. Akshay Gopal Agrawal, Mr. Shubham Devidas Bhange, Mr. Arun Kumar Mourya

Department of Computer Engineering

Dr.D.Y.PATIL INSTITUTE OF ENGINEERING AND TECHNOLOGY, AMBI, PUNE.

Abstract: ***Cloud Storage is a capacity of information online in the cloud, which is available from different and associated assets. Distributed storage can provide high availability and consistent, quality, reliable, assurance, debacle free restoration, and reduced expense. Distributed storage, imperative, usefulness, i.e., safely, proficiently, adaptability and offering information to others. Data privacy is essential in the cloud to ensure that the user's identity is not leaked to unauthorized persons. Using the cloud, anyone can share and store the data, as much as they want. To share the data in a secure way, cryptography is very useful. By using different encryption techniques, a user can store data in the cloud. Encryption and decryption keys are created for unique data that the user provides. Only a particular set of decryption keys are shared so that the data can be decrypted. A public–key encryption system which is called a Key-Aggregate cryptosystem (KAC) is presented. This system produces constant size cipher texts. Any arrangement of secret keys can be aggregated and make them into a single key, which has the same power of the keys that are being used. This total key can then be sent to the others for decoding of a cipher text set and remaining encoded documents outside the set stays private. The project presented in this paper is an implementation of the proposed system.***

Keywords: KAC, Cloud Computing, Key Aggregation, Cryptography, Data Sharing.

I. INTRODUCTION

Considering data security, an ordinary way to deal with the objective without question is to rely on upon the server to approve the passage way control after affirmation, which suggests any unanticipated advantage increasing speed will reveal all data. In a shared residency disseminated figuring environment, things end up being significantly more severe. Information from different clients can be encouraged on segregated virtual machines (VMs) be that as it may, harp on a single physical device. Because of this, the cloud users may not have a firm belief that the cloud service providers are providing the confidentiality. For this reason, the cloud users may encrypt their data before storing it in the cloud. This can be explained using a simple example.

Using Figure1, let us assume there are two users: Alice and Bob. Alice has an account in Dropbox.com. Alice wants to store her photographs in Dropbox.com. But she doesn't trust the security features provided by Dropbox.com. So, she encrypts all her photos and uploads them to the Drop box. Bob who is a friend of Alice requests her to send the photographs that he is present in. If Alice sends them directly to Bob, then Bob cannot view them because they are encrypted with a key. Here there is a constraint as to how the decryption rights should be provided to Bob. Alice can encrypt all the files with one single key and send the key to Bob or encrypt each file and send Bob the respective keys. The first method is not applicable because all the other data can be sent to Bob. In the second method, the number of keys may be more for the number of photos. This becomes complex and requires key storage and sending of all of these keys may not be secure as it requires a secure channel to be sent. So here we can use the Key Aggregation Concept where all the keys are aggregated into one single key of constant size and can be sent through email or any other means securely. This allows Bob to view only his photos, and not Alice's.

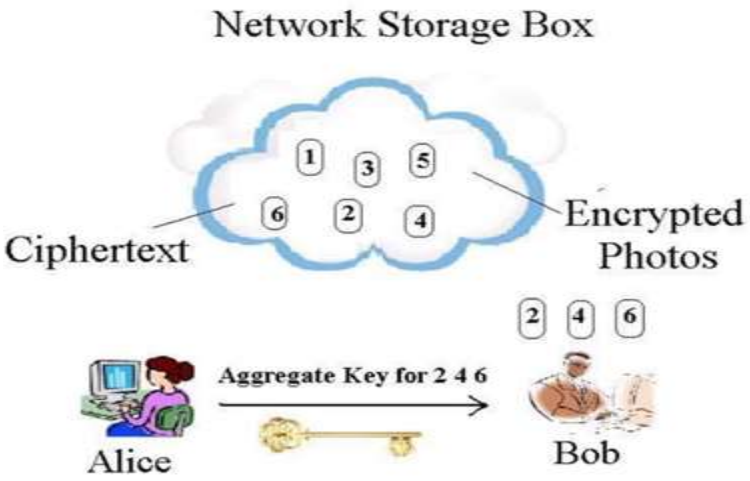

Figure1.Communication between Two Users

Problem Statement

In the present world, sharing the data between the users without any data leakage becomes a big challenge. Data sharing services are one of the essential services provided in cloud computing. Data in the cloud is of higher importance and needs to be protected from unauthorized access. To avoid any data leakage, usually, the data is stored in an encrypted format. However, the problem comes while sharing the encrypted data and providing the decryption rights to the users. An efficient encryption solution is necessary that provides decryption rights to the users for multiple sets of cipher text classes using a single key that comprises the power of several individual keys that are combined.

II. LITERATURE SURVEY

In the present scenario, online data sharing, access, and performance are the preliminary requirements for any organization. The data available is shared across geographical boundaries, and it is available to a multitude of users to perform operations on the data. To do so, the data Owner id really must use different encryption mechanisms before they store the data and provide the decryption power for some of the users while retaining the ability to revoke access for any users at any point in time. The efficient solution would be providing decryption rights to different cipher texts with the use of a single aggregate key, as proposed by (Chu et al. 2014).

According to Cui, Liu, and Wang (2016), the ability to share encoded information via public storage with different clients can significantly reduce the data disclosures in the cloud. To provide that kind of encryption schemas requires excellent management of keys. However, sharing a unique number of selected documents with a different number of users requires different encryption keys to be used for a various number of documents. This implies a large number of decryption keys to be passed securely and to be stored properly. Also, to perform a search over the data, a large number of keyword trapdoors must be passed to the cloud storage to get the data. This scenario seems to be impractical with the current cloud capacity accessed by millions of users.

To address this problem, Cui et al. (2016) proposed the novel concept of Key Aggregate Searchable Encryption (KASE) and instantiated the idea through a concrete KASE scheme. This KASE schema can be applied to any cloud that supports searchable group data sharing. This implies that a user can share a group of files to a group of users and the users can perform the search over the files and retrieve the files they are authorized to access. Efficient key management can be done as follows, the data owner should provide only the aggregate key to the user to share a group of files, and the user needs to provide a single aggregate trapdoor to perform the search over the cloud to get the data.

Recently, Patranabis, Shrivastava, and Mukhopadhyay (2017) proposed Chosen Plain text Attacks (CPA) and Chosen Cipher Attacks (CCA) secure KAC constructions that can be implemented in cloud environments, which uses elliptical curves, and is a first of its kind. This extended broadcast key aggregate cryptosystem is based on the fundamental KAC proposed by (Chu et al, 2014) and the broadcast encryption proposed by Benaloh (2009). In the broad cast scenario, the single cipher text is transmitted to different users, and the respective users can decrypt using their allocated private key. While in KAC a single aggregate key will be distributed to multiple users and they can be used to decode the cipher texts encrypted with reference to different classes.

In the broadcast scenario, the focus is on shorter cipher texts with individual decryption keys and with KAC the focus is on shorter cipher texts with individual lower head aggregation key (Patranabis et al, 2017). They also described how the stand alone KAC scheme could be efficiently combined with broadcast encryption to cater to m data users and M data owners while reducing the secure channel requirement from O(MM).

In the standalone case O(M+M) (Patranabis et al, 2017). They proposed a CCA collusion resistant system and the addition and revocation of new users can be easily implemented. Also, the addition of new users is easily handled in the proposed extended KAC construction by generating a new key for each newly added user in the system and ensuring that all future aggregate key broadcast operations consider the access rights of the new users in addition to the existing ones (Patranabis et al, 2017).

III. PROPOSED SYSTEM

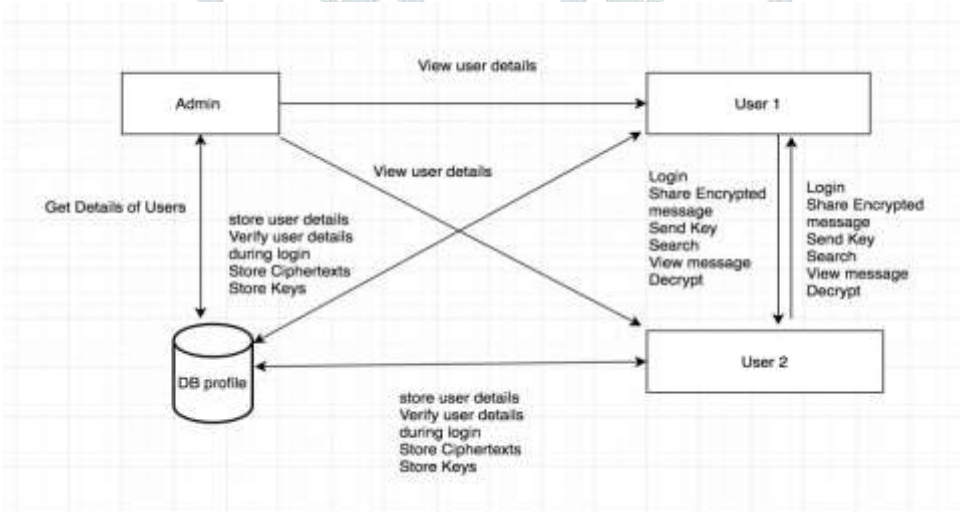

Fig 2. Architecture Diagram

Description and Modules Design:

The architecture diagram shows the various components that are involved in the system and their dependency on each other. The users, admin, and database are the components that are involved, and each component performs a specific operation. The users exchange messages between themselves. The information and keys are stored in the database, and the admin can control users and view user details using the database.

The design phase is to ensure that the project gets developed according to the requirements .The entire design is divided into five modules and each module has a specific functionality, and it is related to other modules. Following are the five modules:

- Registration of user and authentication
- Key generation
- Encryption and storage
- Data sharing using aggregate key
- Decryption and view content

IV. ALGORITHM USED

The ElGamal algorithm is one of the best practices to achieve secure encryption and decryption process of sharing information. There are three components: (a) key generation, (b) encryption, and (c) decryption.

Key generation: The key generator works as follows:

1. Peter generates an efficient description of a multiplicative cyclic group **G** and pick a prime **p** with generator **g**
2. Peter chooses a random **x** from**{0…… p-1}**
3. Peter computes $\mathbf{y= g^x}$
4. Peter publishes **y**, along with the description of **G**, **p**, **g** as his **public key**. Peter retains **x** as his **private key** which must be kept secret

Encryption:
The encryption algorithm works as follows:
To encrypt a message m to Peter under his public key **(G, p, g, y)**

1. David chooses a random **I** from**{1……p-2}** then calculates $\mathbf{c1= g^y}$
2. David calculates the shared secret $\mathbf{s=y^i}$
3. David converts his secret message **m** into an element **m'** of **G**
4. David calculates **c2= m'.s**
5. David sends the cipher ext to Peter $\mathbf{(c1, c2) = (g^i. m'. y^i) = (g^i. m'.(g^x)^i)}$

Decryption: The decryption algorithm works as follows:
To decrypt a cipher text **(c1, c2)** with his private key **x**

1. Peter calculates the shared secret $\mathbf{s = c1^x}$
2. Then he will calculate plaintext message **m**, by converting $\mathbf{c2.s^{-1}}$

Where $\mathbf{c2= m'.y^i}$, $\mathbf{S=y^i}$ and $\mathbf{y = g^x}$ $\mathbf{c2.s^{-1} = m'.y^i.(g^{xi})^{-1}}$

$$\mathbf{=m'.g^{xi}.g^{-xi}}$$

$$\mathbf{=m'.(g^{xi}/g^{xi})}$$

$$=M'$$

V. CONCLUSION

In the present world, the sharing of the data between the users without any data leakage has become a big challenge. Data sharing services are one of the essential functions provided in cloud computing. Data in the cloud is of greater importance and needs to be protected from unauthorized access. In this paper an efficient method is analyzed to avoid any data leakage.
Usually, the data is stored in an encrypted format, which will make the data storage in the cloud more secure. However, the problem comes while sharing the encoded data and providing the decryption rights to the users. An analysis and implementation are provided to share the encrypted data to the end user. In this proposed implementation, encryption processes are done at two places one at the time of storing and another at the time of sharing. The users who share the secret messages should also need to know the decryption process. In the proposed paper, a set of decryption keys are aggregated and sent to the end user. One must decrypt the data with the aggregate key to see the original

text. By this policy of encryption and decryption, it will effectively increase the secure manner of communication and provides successful delegation of decryption rights to the intended user over the cipher texts that they are authorized to view. The implemented application has been tested in various scenarios and executed successfully.

REFERENCES

[1] Bachhav, S., Chaudhari, C., Shinde, N., &Kaloge, P. (2015).Secure multi-cloud data sharing using key aggregate cryptosystem for scalable data sharing. International Journal of Computer Science and Information Technologies, 6(5),4479-4482.

[2] Benaloh,J.(2009).Key compression and its application to digital finger printing. Technical Report, Microsoft Research.

[3] Chu, C. K., Chow, S. S., Tzeng, W. G., Zhou, J., & Deng, R. H. (2014).Key-aggregate cryptosystem for scalable data sharing in cloud storage. IEEE Transactions on Parallel and Distributed Systems,25(2), 468-477.

[4] Cui,B., Liu, Z., &Wang, L.(2016).Key-aggregate searchable encryption(KASE) for group data sharingvia cloud storage. IEEE Transactions on Computers,65(8).

[5] Gan,Q.,Wang,X., &Wu,D.(2017).Revocable key-aggregate crypto system for data sharing in cloud. Security and Communication Networks.

[6] Guo, F., Mu, Y., & Chen, Z. (2007). Identity-based encryption: How to decrypt multiple cipher texts using a single decryption key. In Proceedings of Pairing-Based Cryptography(Pairing'07),4575, 392-406.

[7] Jadhav,R.,&Nargundi,S.(2014).Review on key-aggregate crypto system for scalable data sharing in cloud storage. International Journal of Research in Engineering andTechnology,376-379.

[8] Jansen,W.,&Grance,T. (2011).Guidelines on. NIST Special Publication, 800(144).

[9] Kate, K., &Potdukhe, S. (2014). Data sharing in cloud storage with key-aggregate cryptosystem. International Journal of Engineering Research and General Science,2(6),882-886.

[10] Kumar, S. N. (2015). Cryptography during data sharing and accessing over cloud. International Transaction of Electrical and Computer Engineers System,3(1), 12-18. Retrieved fromhttp://pubs.sciepub.com/iteces/3/1/2

[11] Kumari,G.,&Lakshmi,M.(2014).Keyaggregatecryptosystem&intrusiondetectionfor data sharing in cloud. Multidisciplinary Journal of Research in Engineering andTechnology,3(1),308-317.

[12] Mahalle, R. V., &Pawade, P. (2015).A review of secure data sharing in cloud using key aggregate crypto system and decoy technology. International Journal of Science and Research(IJSR),4(1), 210-213.

[13]Mell,P.,&Grance,T.(2009).The NIST definition of cloud computing. Retrieved fromhttps://www.nist.gov/sites/default/files/documents/itl/cloud/cloud-def-v15.pdf

[14] More, V., & Singh, A. (2015).Key-aggregate crypto system for scalable data sharing in cloud storage. International Journal on EmergingTechnologies,6(2), 182-187.

[15] Patranabis,S.,&Mukhopadhyay,D.(2016).Identity-based key aggregate crypto system from multiline armaps. Retrieved from https://eprint.iacr.org/2016/693

[16] Patranabis, S., Shrivastava, Y., &Mukhopadhyay, D. (2017). Provably secure key-aggregate crypto systems with Broadcast Aggregate Keys for Online Data Sharing on the Cloud. IEEE Transactions on Computers,66(5).

[17] Sahai,A.,&Waters,B.(2005).Fuzzyidentity-basedencryption.InR.Cramer(Ed.),Advancesincryptology-EUROCRYPT, Vol. 3494, pp. 1-15.