RESEARCH ARTICLE OPEN ACCESS

Owner Secured Data Storage on Cloud and Sharing With Key Aggregate System

Miss.Snehal Lande, Miss. Varsha Mhaske, Mr.Mahesh Lagad, Prof. Raskar R.B
Department of Computer Science and Engineering
Shree Chhatrapati Shivajiraje College of Engineering, Ahmednagar
Maharashtra - India

ABSTRACT
In this project, we study how to securely and efficiently, also flexibly share data with others over cloud storage. Nowadays data sharing is in very large scale. Also outsourcing of data is big data.We introduce new public-key crypto-systems which produce constant size cipher texts that efficiently representative for decryption rights for any set of cipher texts are possible. The aggregate key is combination of any set of secret keys and single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can create a constant-size aggregate key for flexible choices of cipher text set in cloud storage, but the other encrypted files outside the set remain confidential and secure that is not decrypted. The compact aggregate key can be conveniently sent to others through secure channel like mail then user can access data efficiently.
Keywords:- Cloud storage, Attribute based encryption ,Identity based encryption, Fuzzy Identity based Encryption, Key aggregate Cryptosystem ,Data sharing.

I. INTRODUCTION

Cloud storage is a service where data is remotely maintained, managed, and backup. As increase in outsourcing of data the cloud computing serves does the management of data. Nowadays, outsourcing of information is very high. Information from various user can collected from separate virtual machine but reside on single physical machine. Cloud users do not have guaranteed that cloud server can keep their information secure. Cloud storage is saving of digital data which are managed by third party or cloud manager. Third party keeps data available and can access anytime, anywhere and physical environment should be protected at all time. Store data in hard disk or any other storage, we save data to remote storage means cloud or internet which is access anywhere & anytime. It minimizes efforts of physical storage to everywhere. We can easily access information from any computer or any smart device through internet by using cloud which avoid limitation of accessing information from same computer or smart device where it is stored.

Data sharing is main functionality in cloud storage, but need of sharing data very securely and maintaining privacy. Solution is cryptosystem owner of data encrypt data before uploading to the cloud with its own key.
Cryptography technique can be applied in following ways-
1. Symmetric key encryption
2. Asymmetric key encryption

In the symmetric key encryption, one public key are used for encryption as well as decryption of all photos.
In the asymmetric key encryption, two different key is used; a public key used for encryption and private key used for decryption.

Suppose Seema put all data on Cloud and she does not want to share her own data to everyone (fig 1). Because of data leakage possibilities she does not trust on privacy mechanism provided by third party or cloud manager, so she encrypt all plain text data before uploading to the server. If Sham asks her to share some data then Seema use share function of Cloud. But problem is how to share encrypted cipher text data.
There are two ways:

1. Seema can encrypt all data with one secret key and send that single key to sham through secure channel.
2. Seema can encrypt data with different keys and send Sham corresponding keys to Sham through secure channel.

In first way, personal data also get share with the Sham, which is inefficient way. In second way, number of keys is as many as number of share files, which may be hundred as well as sharing of keys require secure channel and storage space which is expensive.

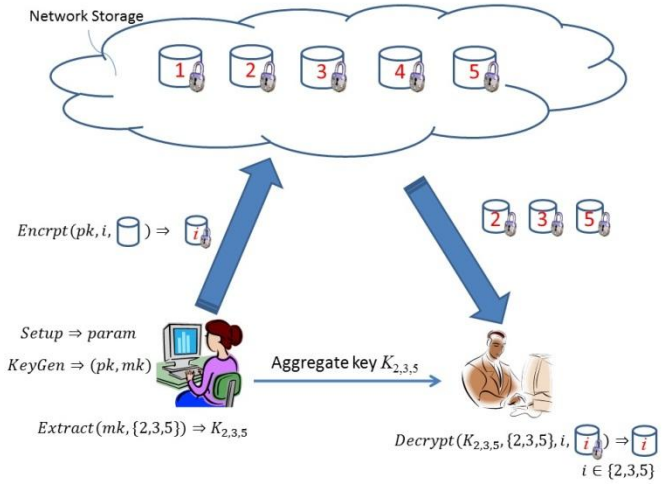

Fig1: Seema shares file with identifiers 2,3 ,6 and 8 with Sham by sending him a single aggregate key.

Therefore, best solution is that encrypt data with distinct public key but only sends one decryption key. Since send Single Compact size key via secure channel and kept private, limited size key is always desirable.

A new way for public-key encryption is used called as key-aggregate cryptosystem (KAC).The encryption is done through an identifier of Cipher text known as class, with public key. The classes are formed by classifying the cipher text. The key owner has the master secret key which is helpful for extracting secret key. So in above scenario now the Seema can send a aggregate key to Sham through a email and the encrypted data is downloaded from cloud storage through the aggregate key.Ther is shown in figure1.

II. LITERATURESURVEY

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

CRYPTOGRAPHIC KEYS FOR A PREDEFINED HIERARCHY

S.G. Akl and P.D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," [1], presented 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 . 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 the number of secrete keys increases with the number of branches of tree. So it is difficult to create a hierarchy of tree can save the number of total keys to be granted for all individuals concurrently

SYMMETRIC-KEY ENCRYPTION WITH COMPACT KEY

"SPICE"(Simple Privacy-Preserving Identity-Management for Cloud Environment). Presented as encryption scheme which is originally introduced for transmission of large number of keys in broadcast scenario. The implementation is simple and we also review key derivation process. The key derivation for a set of classes for decrypting is as follows. A composite modulus is developed where x and y are two large primes numbers are selected randomly. A master secret key is also selected randomly. Each class is associated with a different prime X and Y.Prime numbers can be put in the public system parameter. A constant size key for set can be generated using public parameters. These keys are used for decrypting. The owner (who wants to upload data) needs to get the secret keys to encrypt data which is not suitable for many applications. becouse this method is used to generate a secret value instead of a pair of public or secret keys.

ATTRIBUTE-BASED ENCRYPTION

V. Goyal, O. Pandey, and B. Waters, "Attribute Based encryption for fine Grained Access Control of Encrypted Data," [5], introduce a technique called Key-Policy Attribute Based Encryption (KP-ABE). In this crypto-system, cipher texts are labelled with set of attributes and secret keys are related with access the data which control with cipher texts a user is able to decrypt that cipher text. While this primitive was shown to be useful for fault-tolerant encryption with biometrics, the need of impressibility seems to limit its applicability to larger system. The KP-ABE constructions do not hide the set of attributes under which information is encrypted. This is the drawback of KP-ABE cryptosystem. In 2007 new encryption technique introduced named as cipher text policy attribute based encryption (CP-ABE). Data owner only believes the key provider as CP-ABE technique addresses the difficulty of KP-ABE.

IDENTITY BASED ENCRYPTION WITH COMPACT KEY

D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," [6] is a public-key encryption in which the public key of user contains distinct information of user's identity like Gmail id or mobile number. The key can be textual value or domain name, any distinct value etc. Integrated development environment is used to deploy the public key infrastructure. The identity of the user is used as identity string for public key encryption. A Trusted party

called private key generator (PKG) in IBE which has the Master secret key and gives secret key to users according to the user

Identity. The data owner uses the public key and the identity of user to encrypt the data. The cipher text is decrypted using secretKey For denoting the class we apply hash function to the string, and repeatedly keep hashing until output of the function prime is obtained. We mentioned, our schemes feature constant cipher text size and their security is in the standard model. In fuzzy IBE, for decrypt cipher text one single compact secret key can used, which is encrypted under many identities, close in a certain metric space, but not for an arbitrary set of identities and therefore it do not match with our idea of key aggregation.

FUZZY IDENTITY-BASED ENCRYPTION

Sahai and B. Waters, "Fuzzy Identity-Based Encryption," [8], presented a new type of identity-based encryption is called fuzzy IBE. In fuzzy IBE, it observes an identity as set of descriptive attributes. In this technique allows for a secret key for an identity, p, to decrypt a cipher text encrypted with an identity, q, if and only if the identities p and q are close to each other. Therefore, this scheme allows a certain amount of fault tolerance in the identities. Fuzzy-IBE [8] produces to the two new applications. The first is an IBE system that uses biometric identities. That is it can show a user's identity such as iris scan, fingerprint. Since biometric measurements are inaccurate, we cannot use already present IBE systems. However, the fault-tolerant property of fuzzy-IBE allows for a secret key to decrypt a cipher text encrypted with a small change in measurement of the same biometric. Another application in fuzzy-IBE is Attribute Based Encryption. This technique is already explained in previous paragraph.

MULTIAUTHORITY ATTRIBUTE BASED ENCRYPTION

M. Chase and S. S. M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," [9] ,presented as a multi attribute-authorities numbers of attributes are analysed regarding the decryption key and the user must get a particular key related to the attribute while decrypting a message. The decryption keys are allocated independently to users those who have attribute identity without interaction between each other. Multi-authority attribute-based encryption allows real time deployment of attribute based privileges as different attributes are issued by different authorities. The attribute authorities ensure the honesty of the user privilege so the confidentiality is maintained by central authority .

IV. PROPOSED SYSTEM

1.Setup Phase

In this application set up is done by deploying rar files on cloud server and extracting it. The data owner executes the setup phase for an account on server which is not trusted. The setup algorithm only takes implicit security parameter.

2. KeyGen Phase

This phase is executed by data owner to generate the public or master key Pair.

3. Encrypt Phase

This phase is executed by any user who wants to send the encrypted data. Encrypt, the encryption algorithm takes input as public parameters pk, a message m, and i denoting cipher text class. The algorithm encrypts message m and produces a cipher text C such that only a user that has access or permission to decrypt the message.

4. Extract Phase

This phase is executed by the data owner for delegating the decrypting power for a certain set of cipher text classes to a delegate.

5. 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 cipher text C, i denoting cipher text classes for a set S of attributes.

6. Data Sharing:

KAC in meant for the data sharing. The data owner can share the data in desired amount with confidentiality. KAC is easy and secure way to transfer the Delegation authority.

IV. ARCHITECTURE

1. Requirement gathering: All the necessary requirement of the project is gathered.

2. Cryptosystem selection: Appropriate crypto system is selected.

3. Designing of encryption decryption: Appropriate algorithms are used for encryption and decryption.

4.Testing: Whether everything is running correctly. Everything is tested for proper working of project.

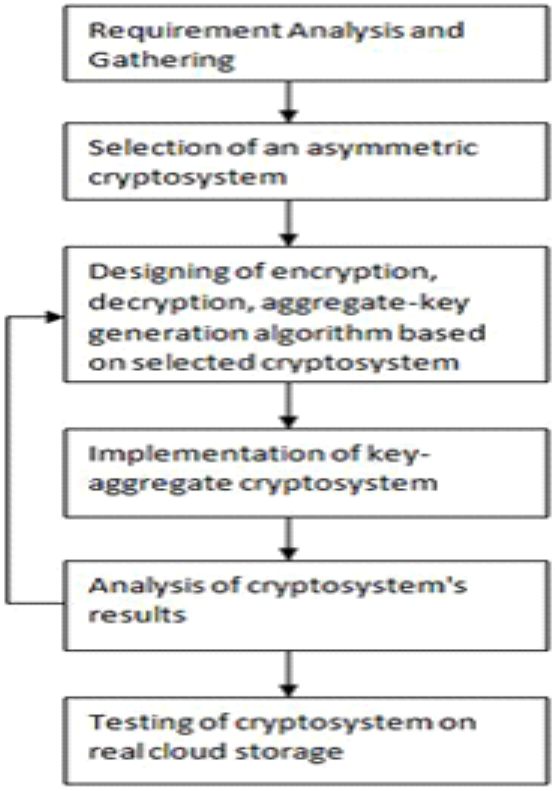

Figure 3.3: Frame Work for Our Proposed Approach

V. ALGORITHMS

Algorithm for public/private-key pair

- Randomly choose p and q large prime numbers.
- Compute the value of n as n = p*q.
- Compute the value of Euler totient function for n as phi (n) = (p-1) *(q-1).
- Choose randomly the value of e such that 1<e<n and gcd(phi(n)) = 1.
- Compute the value of d such that (e*d) mod phi(n)= 1.

Now e, n will be publically announced now d must be kept secret.

Algorithm for encryption

Now here comes an addition in RSA.

In RSA, cipher text C = power (M,e) mod n.

But in our approach, cipher text C = power (M,e*i) mod n, where i the cipher text index of message M.

To encrypt a message M having i as cipher text index

C = power (M,e*i) mod n

Algorithm to generate Aggregate Key

Say S is the set of cipher text indices of those files whose aggregate-key is to be

Generated. Following is the Pseudo-code to generate aggregate-key.

Extract_ Aggregate_ Key (d,S)

aggr_key = d

s <- S. size ()

i <- 1

While i<=s

aggr_ key <- aggr_ key * S[i]

Return aggr_ key

Algorithm for decryption

Here comes another modification and addition in RSA.

In RSA, cipher text C can be decrypted as M = power(C,d) mod n

In our approach, to decrypt a set of files whose cipher text indices are kept in set S, following is the pseudo code of our approach.

Decryption(C,aggr_ key,S)

S <- S.size()

i <- 1

while S!=empty

temp = temp * S[i]

20

"Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage"

dd = aggr_ key/temp

i <- 1

while S!=empty

Mi = power(Ci,dd/i) mod n

SYSTEM MATHEMATICAL MODELING

The proposed system S is defined as follows:

S { I, O, F, U }

Where,

I: Input

O: Output F: Functions

U: User

I= {U, IF, AU, ISM, DH}

Where ,

U = User which require Data Security in Cloud Using Key Aggregate Cryptosystem.

IF = Input files on cloud for sharing with others.

AU = Authenticated user login in to cloud by proving proper user id and password.

ISM = Input secret messages which system explicitly wants to share with other instead of sharing all files.

DH: RSA algorithm input value for Key Exchange on cloud to share files on cloud like prime number etc…
O = {AK, DHSK, SF, CT, RFM}
Where,
below are the output generated from system processing;
AK = Key Aggregate Cryptosystem generates Aggregate Key for sharing files and Data Security in Cloud.
DHSK = RSA secret key for exchanging aggregate key on cloud.
SF = Sharing of files on cloud with help of class identification function.
CT = System generate cipher text.
RFM = Finally, receiver will receive files and display Received File Message.
U = {SD, FS, A, CA}
Where ,
SV = System developer
FS = Files Sender
FR = Files Receiver
A= Administrator
CA = Cloud Administrator
F= {F1, F2, F3, F4, F5}
Where,

Function F1: To store the files on cloud for exchanging or sharing with other by providing data security.

Function F2: Sender sends files toward receiver by encrypting with the help of key aggregate cryptosystem(KAC) .

Function F3: It is use to generate aggregate key of any set of public keys and create them as compact as a single key.

Function F4: It uses RSA algorithm for exchanging secret key to ensures the confidentiality of the data on the cloud by using symmetric encryption.

Function F5: Receiver receives data files (cipher text) and decrypts it only those having class identification tags.

VI. CONCLUSION

We learned that to protect users data privacy on cloud storage. With more mathematical tools, cryptographic schemes we protect data of user. In this project, we use single compact size key called aggregate key for decrypting number of files. This aggregate key is combination of multiple secret keys. Aggregate key is used to decrypt data which is encrypted by that corresponding secret key .we also learned the other technique of encryption but that have some drawback which are overcome in key aggregate cryptosystem. Using key aggregate cryptosystem we protect data and maintain privacy.

REFERENCES

[1]. S.G. Akl and P.D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Trans. Computer Systems, 1983

[2]. 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 IEEE, 2013.

[3]. 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 2006.

[4]. D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," in Proceedings of Advances in Cryptology '2001.

[5]. F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Cipher texts

[6]. Using a Single Decryption Key," Proc. Pairing-Based Cryptography Conf. 2007.

[7]. [A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques 2005.

[8]. M. Chase and S. S. M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," in CM Conference on Computer and Communications Security, 2009.

[9]. 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 – ACNS 2012.