A Conceptual Phenomenon for Scalable Data Sharing in Cloud Storage using Key Aggregate Cryptosystem.

Rohit Tate

Dept. of Information Technlogy
Siddhant College of Engineering, sudumbare
Pune, India
E-mail: tate.rohit@gmail.com

Jyoti Pingalkar

Dept. of Information Technology
Siddhant College of Engineering, Sudumbare
Pune, India
E-mail: jyoti_pingalkar@rediffmail.com

***Abstract*—Cloud computing is a popular area of research for inventors. And it is very important in data sharing applications. On cloud the data being shared must be secure. The flexibility and the efficiency of the data is depend upon the security parameter. To achieve this purpose I describe new algorithm which depends upon public key cryptography and produce constant size cipher text. These ciphers can be decrypt by using a secret key. This secret key can generate the constant size key known as aggregate key, for selection of flexible choices of ciphers. The other encrypted files except these cipher remain confidential. I can able to save this aggregate key or can send it to others for further data sharing.**

Keywords-*Cloud storage, data sharing, key-aggregate encryption, patient-controlled encryption*.

INTRODUCTION

Now a day’s internet is most widely used in many applications. In that cloud computing has wide scope of area, so that the data can be upload or download from cloud and can accessed easily. Many number of users can access data and share that data through different virtual machines but present on single physical machine. But the thing is user don’t have physical control over the outsourced data. The need is to share data securely among users. The service provider uses various authentication method to avoid the loss and leakage of data on cloud. Privacy preserving in cloud is done to make sure that user’s identity is not revealed to everyone. Anyone can access large amount of data on cloud as much they want i.e. only selected content can be shared. Cryptography allow the data owner to share the data to in secure way. So user can encrypts data and uploads on server. Different encryption keys as well as decryption keys are generated for each bunch data. The encryption and decryption keys may be different for different set of data. Only those set of decryption keys are shared that the selected data can be decrypted.

This paper propose a public-key cryptosystems which generates constant size ciphertext. So that it transfer the decryption rules for number of ciphertext. The difference is one can collect a set of secret keys and make them as small size as a single key with holding the same ability of all the keys that are formed in a group. Then generated compact key can be send or stored on the cloud in secure manner. The digital data is stored inside the cloud storage as a logical data pool. These cloud storage providers maintains all the data related operations and these are responsible for keeping the data available, protected and running. Other people uses storage capacity from the providers to store end user, they pay for that. Cloud storage services may be accessed through a web service application programming interface (API), such as cloud desktop storage, a cloud storage gateway or Web content management systems.

Cloud storage is based on highly virtualized infrastructure and is like broader cloud. The dada sharing is important application of cloud computing. One can upload or download the data inside cloud. We can store any type of data on cloud. That means data shared may be in the text format or may be in the multimedia format. This sharing of data should be in secure, efficient and flexible manner. Otherwise the data attacker may stole our personal information and may misuse it. So security in the cloud computing plays an important role.

The data storing on cloud is done in flexible and cost optimizing way so it motivates the end user as well as enterprises to store the data on cloud and share it among users. The insider attack is one of security concern which needs to be focused. Cloud Service provider decide whether audits are held for users who have physical access to the server. As CSP 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.

To achieve the integrity along with avoiding anonymity is a major task. For that the user can verify metadata on their own data, upload and verify metadata. The main concern is how to share the data securely the answer is cryptography. The question is how can the encrypted data is to be shared. The user must provide the access rights to the other user as the data is encrypted and the decryption key should be send securely. For an example Alice keeps her private data i.e. photos on drop box and she doesn't want to share it with everyone. As the attacker may access the data so it is not possible to rely on predefine privacy preserving mechanism so she all the photos were encrypted by her on encryption key while uploading it.

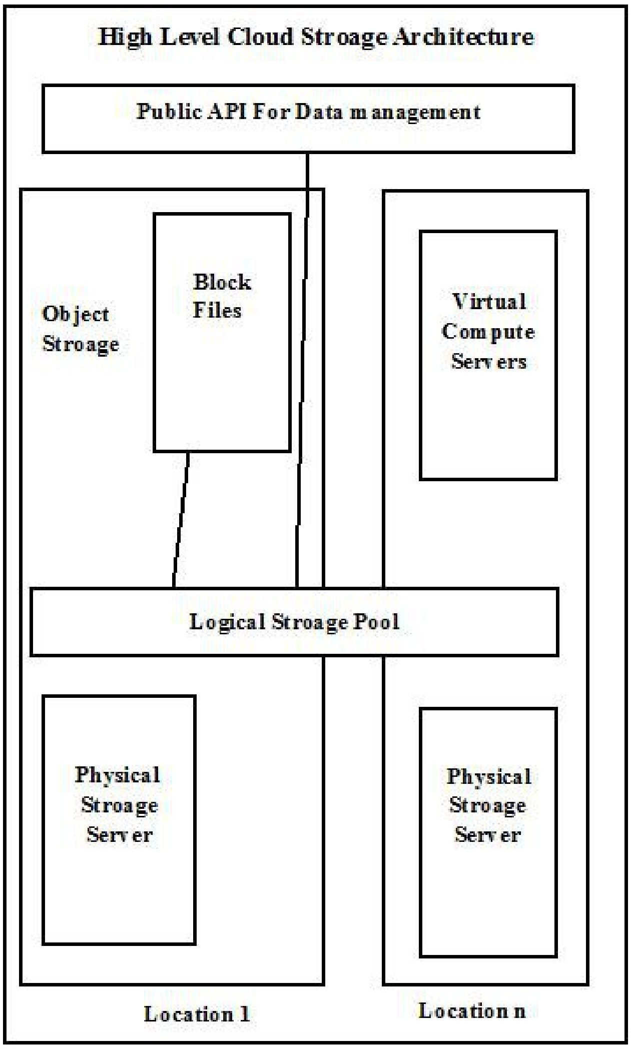

Figure: 1. Architecture of data sharing in cloud storage.

To achieve such type of security inside the cloud we have used the key aggregation technique. In this we are encrypting the data which user want share on the cloud. For this encryption we are using the secrete key. It will generate fixed data size of ciphers. These cyphers can be decrypt by using the aggregate key. This newly generated aggregate key will able to decrypt only bunch of cyphers other remaining ciphers will be confidential

Literature Survey

There were many systems proposed to ensure privacy and security is discussed in a number of existing articles.

M. Chase and S. S. M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption"[8]This paper outlines the requirements for achieving privacy and security in the Cloud and also briefly outlines the requirements for secure data sharing in the Cloud. It provided a survey on privacy and security in the Cloud focusing on how privacy laws should also take into consideration Cloud computing and what work can be done to prevent privacy and security breaches of one's personal data in the Cloud. This explored factors that affect managing information security in Cloud computing. It explains the necessary security needs for enterprises to understand the dynamics of information security in the Cloud.

J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records,"[10]. This paper uses Broadcast encryption which enables a broadcaster to transmit encrypted data or information to a set of users so that only a targeted subset of users can decrypt the data. Other than above characteristics, it also allows the group monitor to include new members by preserving previously computed information, and user decryption secret keys need not be computed again and again, the Aggregation logic and size of cipher texts are remain unchanged and the group encryption key requires no modification.

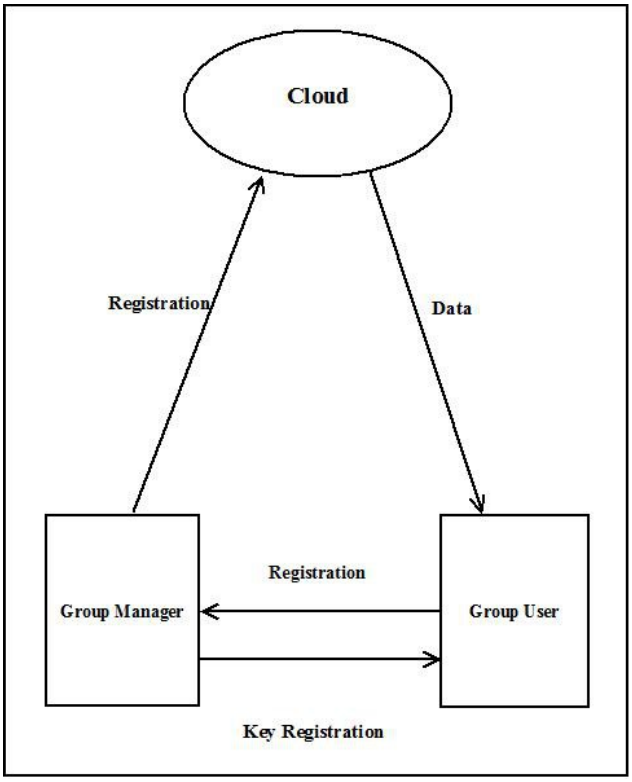

Figure: 2. Dynamic Broadcast Encryption.

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" [4]. This system uses the slice of data cloud to encrypt or decrypt the data. The original data are first divided into a number of slices. When a revocation occurs, the data owner needs only to retrieve one slice, and re-encrypt and re-publish it. The data owner retrieve the signature from secure mediator and then it allows user to upload or download the data over the cloud.

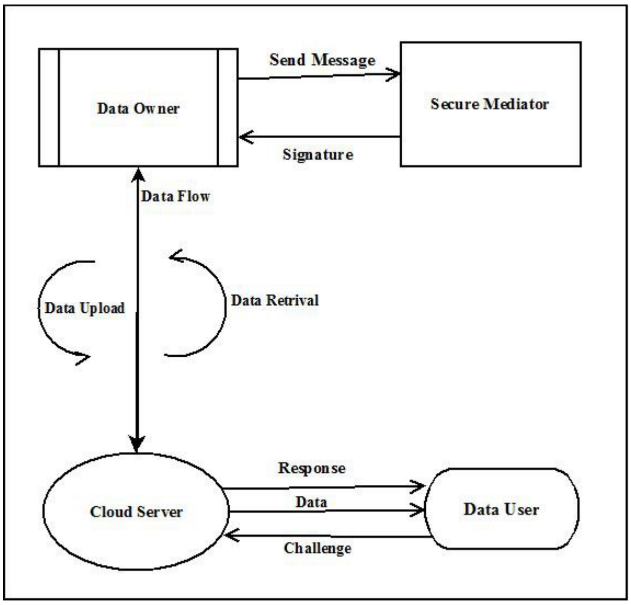

Figure: 3. Data Sharing in Cloud Using Hybrid Cryptosystem

W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," [12]. This system allows sharing of secure file on untrusted servers. It divides files into the group of file and encrypt each group of file with a unique file-key. The data owner can share the file groups with others by delivering the related lockbox key, where the lockbox key is used to encrypt the file-block keys. But it is responsible for heavy key distribution overhead for large-scale file sharing. Additionally, we need to update file key and distributed again for a user revocation.

PROPOSED SYSTEM

In propose system we are using two keys to encrypt and decrypt the data which are secret key and its aggregate key. This system is basically design on the basis of key aggregation encryption. The data owner creates the public system parameter and generates a secrete key which is public key pair. User is responsible for data encryption and he may decides cipher text block associated with the plaintext file which want to be encrypted. The data owner have rights to use the secret key from which he can generate an aggregate key which is use for decryption of a set of cipher text blocks. The both keys can be sent to end user in very secure manner. The authenticated user having an aggregate key can decrypt any block of cipher text.

A key-aggregate encryption scheme consists of five polynomial-time algorithms as follows. The data owner establishes the public system parameter via Setup and generates a public/master-secret3 key pair via KeyGen. Encrypt is use for message encryption by anyone who also decides what ciphertext class is associated with the plaintext message to be encrypted. This project consist of five algorithms which are used to perform the above operations. These algorithms are as follow:

1. Setup: The data owner executes the setup phase for an account on server which is not trusted. The setup algorithm only takes implicit security parameter. The account is created on the untrusted server for sharing of data. This account is generated by data owner.

2. KeyGen: This phase is executed by data owner to generate the public or the master key pair (pk, msk).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 create an aggregate key to access the block of ciphers of limited size.

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

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

4. Extract: This is executed by the data owner for delegating the decrypting power for a certain set of cipher text classes to a delegate. The aggregate key is use for extracting the particular block of the ciphers from the cipher file. But other encrypted data remains secure.

   - Input = master-secret key mk and a set S of indices corresponding to different classes
   - Outputs = aggregate

5. Decrypt: This is executed by the candidate who has the decryption authorities. Decrypt (k_S, 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. The encrypted data is then decrypted by using the same secrete key which is use for encryption

   - Input = k_S and the set S, where index i = ciphertext class
   - Outputs = m if i element of S

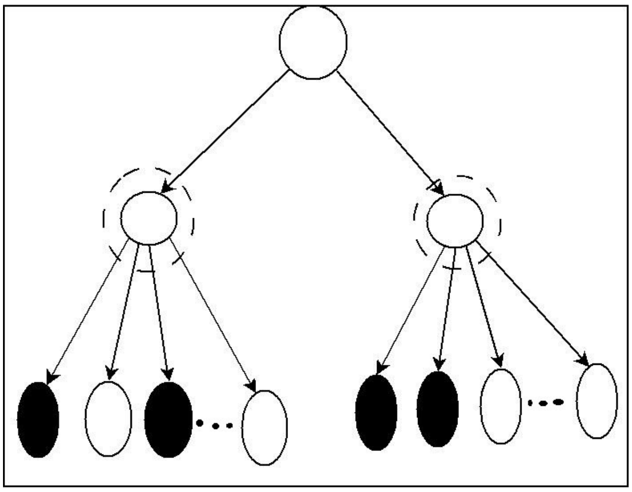

Figure: 4 Key assignment in our system

As the above figure shows, the key assignment is done in dynamic way. The aggregate key is use to decrypt only those cyphers 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 secrete key and the aggregate key, otherwise he will be block forever.

In aggregate cryptosystem authentication is necessary for each user in which user login if user login successfully then proceed for further process. User may be sender or receiver. Permission function of sender it gives the permissions like read, write etc. to data for security and proceeds to encryption function. It encrypt data using aggregate key that key size is fixed for every user but it can be generated dynamically. Split function uploads the data but before uploading it splits the encrypted data into different parts and stored that part on different clouds. Here, Merge is the function of receiver side, it retrieves the data from different clouds like C1,C2,C3…Cn. Decrypt function decrypt the date using the private key and aggregate key and proceed for the further processing.

Extractor checks wheatear that file is accessible to that user or not. In case it accessible then it decrypt from that whole bunch. Figure 4 shows how the key's assigned to the separate users. Each user has separate key as per the aggregation cryptosystem. Basically initially grated key is used to generate separate user key as per their bits status.

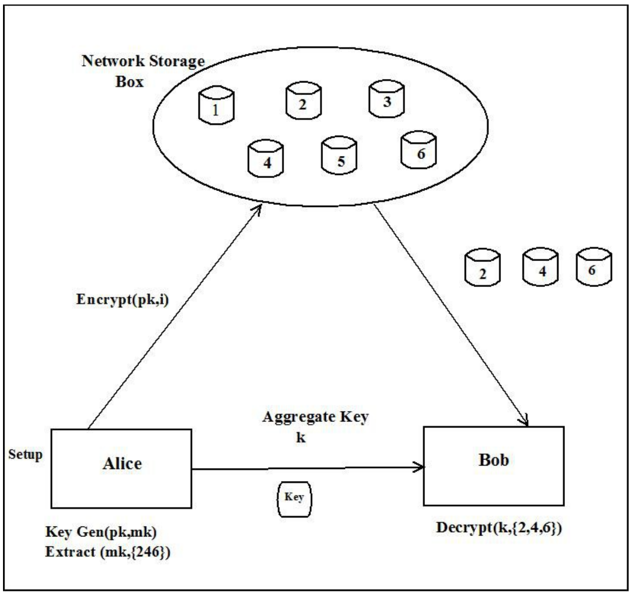

Figure: 5 System Architecture.

Data Sharing

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 Figure 5. For sharing selected data on the server Alice first performs the Setup. 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. Anyone can encrypt the data m and this data is uploaded on cloud. User with the decrypting authority the other users can access those data. 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). 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.

PERFORMANCE ANALISYS

Security: It increases the decryption process performance by using the N2k algorithm it is used to merge the separated file and generate the original form of the data. This algorithm does not require all the parts of the separated file. It only required minimum (n/2)+1 parts of the encrypted file.

Efficiency: For encryption, the value e(g1,gn) can be pre-computed and put in the system parameter. It is fast to compute a pairing.Efficient software implementations exist even for sensor nodes.

Mobility: System can be handled through wireless network or electronic media with any platform.

Comparison Factor: For a concrete comparison, we investigate the space requirements of the tree-based key assignment approach. This is used in the complete sub tree scheme, which is a representative solution to the broadcast encryption problem following the well-known subset-cover framework. It employs a static logical key hierarchy, which is materialized with a full binary key tree of height h, and thus can support up to 2h ciphertext classes, a selected part of which is intended for an authorized delegate.A comparison of the number of granted keys between three methods is depicted. We can see that if we grant the key one by one, the number of granted keys would be equal to the number of the delegated ciphertext classes.

V. EXPERIMENTAL RESULTS.:

1. The various classes of data stored on clouds should not be decryptible by single key.
2. An aggregate master key should be generated for various classes for each user.
3. Upon request of access to data by other users, only exclusive access to requested class of data should be provided.
4. The system provides access to shared data to the users of same group where the data is shared.

VI. CONCLUSION

To share data flexibly is vital thing in cloud computing. Users prefer to upload there data on cloud and among different users. The outsourcing of cloud data to server may causes 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 cipher text classes in cloud storage.

Cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this paper, we consider how to “compress” secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage. My approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. Also provide high level security by storing split file on different cloud means if one cloud data hacked but steel file is secure. The delegate 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 cipher text classes are bounded that is the limitation.

ACKNOWLEDGMENT

Author would like to take this opportunity to express my profound gratitude and deep regard to my Project Guide Mrs. Jyoti Pingalkar, for her exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. Her valuable suggestions were of

immense help throughout my project work. Her perceptive criticism kept me working to make this project in a much better way. Working under her was an extremely knowledgeable experience for me.

References

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.

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.

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

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 Transactions On Parallel And Distributed System, Vol 25, No. 2 February 2014.

V. Goyal, O. Pandey, A. Sahai, 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..

Y. Sun and K. J. R. Liu, “Scalable Hierarchical Access Control in Secure Group Communications,” in Proceedings of the 23th IEEE International Conference on Computer Communications (INFOCOM ’04). IEEE, 2004.

D. Boneh and M. K. Franklin, “Identity-Based Encryption from the Weil Pairing,” in Proceedings of Advances in Cryptology – CRYPTO ’01, ser. LNCS, vol. 2139. Springer, 2001, pp. 213–229.

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, pp. 121–130.

D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

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.

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.

W.-G. Tzeng, “A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy,” IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 14, no. 1, pp. 182–188,2002.