Cloud Provider using Authentication Privacy Cloud Cryptography Encrypt Ciphertext in Cloud Computing

Mr.Tushar T. Hedaoo[1] Prof.Anil Chhangani[2]

[1,2]Department of Computer Engineering

[1,2]Lokmanya Tilak College of Engineering, Mumbai University, NaviMumbai, India.

Abstract— Data sharing is an important functionality in cloud storage over the internet. The computing resource that is delivered as a service over a network are used in cloud computing. Data sharing in the cloud depend on performance of the network of the data center. In this, how to securely, efficiently, and flexibly share data with others in cloud storage. Cloud storage is a storage of data online in cloud which is accessible from multiple and connected resources. Cloud storage can provide good accessibility and reliability, strong protection, disaster recovery, and lowest cost. exploitation the cloud storage, users store their information on the cloud while not the burden of information storage and maintenance and services and high-quality applications from a shared pool of configurable computing resources Cryptography is may be the foremost necessary side of communications security and is turning into progressively necessary as a basic building block for security. Using Key-Aggregate cryptosystem produce constant size ciphertexts such that efficient delegations of decryption rights for any set of cipherttext are possible. In particular schemes give the first public key patient controlled encryption for flexible hierarchy and the remaining encrypted files outside the set are remains confidential.

Key words: Cloud Computing, Data Sharing, Cryptosystem, Ciphertext, Encryption

I. Introduction

Cloud computing environment all services being delivered in internet cloud. Cloud computing is term used to refers to almost any services. Information technology services totally depend on the Internet cloud. Cloud storage is storing of data off-site to the physical storage which is maintained by third party.

Data privacy is a traditional way to ensure is it to rely on the server to enforce the access control after authentication. Instead of storing data to the hard drive or any other local storage, 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 it can access information from any computer through internet which omitted limitation of accessing information from same computer where it is stored.

The challenging problem is how to effectively share encrypted data. Of course users can download the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly. However, finding an efficient and secure way to share partial data in cloud storage is not trivial data privacy, it cannot rely on traditional technique of authentication, because unexpected privilege escalation will expose all data. Solution is to encrypt data before uploading to the server with user's own key. Data sharing is again important functionality of cloud storage, because user can share data from anywhere and anytime to anyone. Cryptography technique can applies i.e asymmetric key encryption different key are used and in symmetric key encryption same keys are used so public for encryption. Using asymmetric key encryption is more flexible approach [1],[2].

A. Data Center

Data center is the foremost ingredient of cloud computing. It is the collection of the software and hardware resources in data center. The cloud user and cloud service provider suffer the problem of managing the occupant and non-occupant resources. Cloud vendors provide same or different resources share by some cloud server. Data center placed in different area but data shared by cloud user from different cloud vendor, such types of task managing by the IT organization. Utility computing play a very important role in cloud computing. Utility computing is useful for managing effectively when the resources are outsourced [14].

B. Data Sharing

In cloud computing ,data sharing is a very important aspect, it play a pivot role in cloud computing by providing it with an abundant benefits to the user .According to a survey by IT organization it is been seen that most of the organization share the data with customers yielding high productivity ,redundancy, mutual sharing of resources and low cost. Cloud is suspect able to much privacy and security attacks thus hindering the progress and adoption of cloud hence, sharing resources mix the cloud is more vulnerable to attack and the data stored is highly preserved and used[4],[14].

II. Related work

A. Cryptographic Keys

The most relevant study in the literature of cryptography/security. Cryptographic key assignment scheme aim to minimize the expense in storing and managing secret keys for general cryptographic use. Utilizing a tree structure, a key for a given branch can be used to derive the keys of its descendant nodes. Just granting the parent key implicitly grants all the keys of its descendant nodes. Sandhu proposed a method to generate a tree hierarchy of symmetric keys by using repeated evaluations of pseudorandom function/block-cipher on a fixed secret. The concept can be generalized from a tree to a graph. More advanced cryptographic key assignment schemes support access policy that can be model by an acyclic graph or a cyclic graph. Most of these schemes produce keys for symmetric-key cryptosystems, even though the key derivations may require modular arithmetic as used in public-key cryptosystems, which are generally more expensive than “symmetric-key operations” such as pseudorandom function[1],[7].

B. Symmetric-Key Encryption

The same problem of supporting flexible hierarchy in decryption power delegation, Benaloh et al. present an encryption scheme which is originally proposed for concisely transmitting large number of keys in broadcast scenario. The construction is simple and it briefly review its key derivation process here for a concrete description of what are the desirable properties want to achieve[1],[2].

C. Identity-Based Encryption

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. Guo et al. [6] [9] tried to build IBE with key aggregation. One of their schemes assumes random oracles but another does not. In their schemes, key aggregation is constrained in the sense that all keys to be aggregated must come from different “identity divisions”. While there are an exponential number of identities and thus secret keys, only a polynomial number of them can be aggregated. Most importantly, their key-aggregation comes at the expense of O(n) sizes for both ciphertext and the public parameter, where n is the number of secret keys which can be aggregated into a constant size one. This greatly increases the costs of storing and transmitting ciphertext, which is impractical in many situations such as shared cloud storage. As mentioned, schemes feature constant ciphertext size, and their security holds in the standard model. One single compact secret key can decrypt ciphertext encrypted under many identities which are close in a certain metric space, but not for an arbitrary set of identities and therefore it does not match with our idea of key aggregation.

Previous results may achieve a similar property featuring a constant-size decryption key, but the classes need to conform to some pre-defined hierarchical relationship. Work is flexible in the sense that this constraint is eliminated, that is, no special relation is required between the classes[11],[12].

D. Objective

- The Objective of the project is to provide best solution for the Existing problem is that user1 encrypts files with distinct public-keys, but only sends Bob a single (constant-size) decryption key.
- Since the decryption key should be sent via a secure channel and kept secret, small key size is always desirable. Using the Public-key Cryptosystem (public key Encryption algorithm).
- The challenging problem is how to effectively share encrypted data. Of course users can download the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage.
- Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly.

III. System Architecture

A. Architecture of Proposed System

Introduce a special type of public-key encryption which call key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public-key, but also under an identifier of cipher text called class. That means the cipher texts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have 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, i.e., the decryption power for any subset of cipher text classes.

With a solution, USER 1 can simply send USER 2 a single aggregate key via a secure e-mail. USER2 can download the encrypted files from USER 1 Drop box space and then use this aggregate key to decrypt these encrypted files [13]. USER1 encrypts all files with a single encryption key and gives USER 2 the corresponding secret key directly. USER1 encrypts files with distinct keys and sends USER2 the corresponding secret keys.

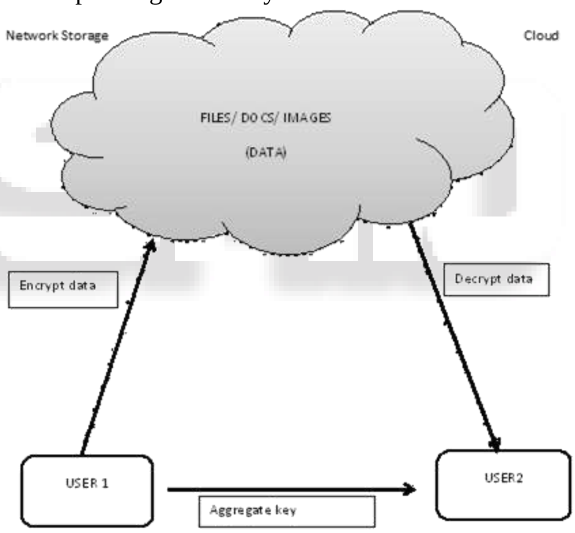

Fig. 1: Cryptosystem Architecture

“To design an efficient public-key encryption scheme which supports flexible delegation in the sense that any subset of the ciphertext (produced by the encryption scheme) is decryptable by a constant-size decryption key (generated by the owner of the master-secret key).Solve this problem by introducing a special type of public-key encryption which we call key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertext are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, 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 for any subset of ciphertext classes. With solution, USER1 can simply send USER2 a single aggregate key via a secure e-mail. USER2 can download the encrypted data from USER1 Dropbox space and then use this aggregate key to decrypt these encrypted photos. The sizes of ciphertext, public-key, master-secret key and aggregate key in our KAC schemes are all of constant size. The public system parameter has size linear in the number of ciphertext classes, but only a small part of it is needed each time and it can be fetched on demand from large (but non-confidential) cloud storage. Previous results may achieve a similar property featuring a constant-size decryption key, but the classes need to conform to some pre-defined hierarchical relationship. Work is flexible in the sense that this constraint is eliminated, that is, no special relation is required between the classes. Propose several concrete KAC schemes with different security levels and extensions in this article.

Therefore, the best solution for the above problem is that USER1 encrypts files with distinct public-keys, but only sends USER2 a single (constant-size) decryption key. Since the decryption key should be sent via a secure channel and kept secret, small key size is always desirable [13].

B. *Properties*

- Increased admin security: The PC should be highly secured and accessible only by the administrator to avoid the misuse of the application.
- Portability: The GUIs of this application is user-friendly so it is very easy for the user to understand and respond to the same.
- Reliability: This system has high probability to deliver us the required queries and the functionalities available in the application.
- Response time: The time taken by the system to complete a task given by the user is found to be very less.
- Scalability: The system can be extended to integrate the modifications done in the present application to improve the quality of the product.
- Robustness: The application is fault tolerant with respect to illegal user/receiver inputs. Error checking has been built in the system to prevent system failure.

IV. ALGORITHM AND DESIGN

A. *AES_Encrypt*

Public-key Cryptosystem (public key Encryption algorithm it to be proposed)

Public-key cryptosystems, which are generally more expensive than “symmetric-key operations” such as pseudorandom function .It take the tree structure as an example. USER1 can first classify the ciphertext classes according to their subjects. Each node in the tree represents a secret key, while the leaf nodes represent the keys for individual ciphertext classes. Note that every key of the non-leaf node can derive the keys of its descendant nodes.

The Advanced Encryption Standard or AES is a symmetric block cipher used to protect classified information and is implemented in software and hardware throughout to encrypt and decrypt sensitive data.

- Setup(1_; n): Executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1_ and the number of ciphertext classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter param, which is omitted from the input of the other algorithms for brevity.
- KeyGen: Executed by the data owner to randomly generate a public/master-secret key pair (pk; msk).
- Encrypt(pk; i;m): Executed by anyone who wants to encrypt data. On input a public-key pk, an index I denoting the ciphertext class, and a message m, it outputs a ciphertext C.

1) *Extend version*

- Extract(msk; S): Executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegate. On input the mastersecret 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 delegatee who received an aggregate key KS generated by Extract. On input KS, the set S, an index i and c denotes ciphertext.

B. *Design*

Functional requirements

- Sign Up: User will Sign up in to and Make Account in the System.
- File Upload: User Can Upload File in to the System.
- Generate Key: System Generates the Key for uploaded File.
- Encrypt Data: Data in the File will be encrypted By Algorithm based by system.
- Sharing: Multiple Files Sharing between Different user using A Single Key.

V. FUTURE SCOPE

In this project how to protect user data privacy in cloud storage. With cryptographic schemes are getting more versatile and often involve multiple keys for a single application. Use public-key cryptosystems which support delegation of secret keys for different ciphertext classes in cloud storage. A limitation in work is the predefined bound of the number of maximum ciphertext classes. In cloud storage, the number of ciphertexts usually grows rapidly. So reserve enough ciphertext classes for the future extension. Otherwise, need to expand the public-key. Although the parameter can be downloaded with ciphertexts, it would be better if its size is independent.

REFERENCES

[1] 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 - ACNS2012, ser. LNCS, vol. 7341. Springer, 2012.

[2] L. Hardesty, “Secure computers aren’t so secure,” MIT2009, http://www.physorg.com/news176107396.html.

[3] 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.

[4] B. Wang, S. S. M. Chow, M. Li, and H. Li, “Storing Shared Dataon the Cloud via Security-Mediator,” in International Conference.

[5] 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.

[6] 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.

[7] 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.

[8] G. C. Chick and S. E. Tavares, “Flexible Access Control with Master Keys,” in Proceedings of Advances in Cryptology

[9] G. Ateniese, A. D. Santis, A. L. Ferrara, and B. Masucci, “Provably-Secure Time-Bound Hierarchical Key Assignmen Schemes,” J. Cryptology, vol. 25, no. 2, pp. 243–270, 2012.

[10] R. S. Sandhu, “Cryptographic Implementation of a Tree Hierarchy for Access Control,” Information Processing Letters, vol. 27, no. 2

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

[12] Q. Zhang and Y. Wang, “A Centralized Key Management Scheme for Hierarchical Access Control,” in Proceedings of IEEE GlobalTelecommunications Conference (GLOBECOM ’04). IEEE, 2004, pp.

[13] Cheng-Kang Chu ,Chow, S.S.M, Wen-GueyTzeng, Jianying Zhou, and Robert H. Deng , ―Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage‖, IEEE Transactions on Parallel and Distributed Systems. Volume: 25, Issue: 2. Year 2014.

[14] International Journal of Research in Advent Technology, Vol.2, No.12, December2014 E-ISSN: 2321-9637 73 Bandwidth Allocation Dynamically to the Suspicious user in Cloud Computing