Key based Secured Cryptosystems used for Online Data Sharing on the Cloud

Diwakar Bhardwaj, RAKESH KUMAR

Diwakar Bhardwaj, RAKESH KUMAR
Department of Computer Engineering and Application
GLA UNIVERSITY, MATHURA
Department of Computer Engineering and Application
GLA UNIVERSITY, MATHURA

E- Mail: diwakar.bhardwaj@gla.ac.in

ABSTRACT

Since the advent of cloud its has become a efficient platform for sharing data for big enterprises, companies, startups and individuals. Cloud has allowed multiple users to collaborate on data. A certain level of security is desired by the cloud users while sharing data. Data owners would want to store their data on the cloud while preserving confidentiality and would want to allow access or provide decryption details to the users he wants to share data with along with the ability to revoke that access ant any given time. In this paper the system proposed uses broadcast encryption to encrypt the data while using the cloud resources efficiently by reducing the overhead caused in the Key Aggregate Cryptosystem (KAC).

Keywords: Online data sharing, Key-Aggregate Cryptosystem, Security, broadcast encryption, Cloud.

INTRODUCTION

Recent techniques for security while sharing data are majorly of two types - a third party authentication system or using a private key to perform encryption besides maintaining anonymity. KAC is a different point of view of broadcast encryption. While the latter uses a method where a unique cipher text is broad cast to multiple users and the information is decrypted using their personal private key, the former follows a method where an aggregate key is broadcast and can be used to decrypt the information. The cloud is prone to security attacks and can make shared data vulnerable, this is an obstacle in its acceptance as a prime means of sharing the data [11]. The reasons this system cannot be deployed are basically, the number of secret keys would increase with the increase in count of data classes. Furthermore, revocation of access to any user would require re-encryption of the subset of data associated with that user and re-distribution of new keys to other users which causes issues in scalability [12].

This paper attempts to outline a framework for data sharing that is comparatively secure while being efficient to implement. A cost-effective executable sort of the conventional key-aggregate cryptosystem (KAC) has been described in this paper. This method is fully CCA-secure and collusion resistant with lower overhead.

LITERATURE REVIEW

[1] Spice–simple privacy-preserving identity-management for cloud environment, Sherman SM Chow, Yi-Jun He Some of the major security threats in cloud are privacy and identity security. Although there are a few solutions for identity management, not one of these solutions can solve all the occurring issues.

The property of unlink ability in cloud establishes that even during collusion none of the cloud service providers can connect the transactions of a single user. In contrast, entrust able authentication is a signature of the cloud platform where several cloud service providers collaborate to supply a packaged service and only one of those service providers acts as the source while others remain transparent [13]. The source service provider is in charge of interacting with the clients and extending authentication to them. It should be noticed that every service provider has his own method of authenticating clients which depends on a set of different attributes. Also, every service provider is limited to a particular set of attributes [14]. This scheme combines and exploits two group signatures so that it can be randomized to make the same signature look different for different uses and hide parts of messages [2].

Dynamic secure cloud storage with provenance, Sherman SM Chow, Cheng-Kang Chu One of the issues while using cloud storage is that vulnerable data must be kept confidential from the servers that lie outside the trust domain. Moreover, the user would want to remain anonymous while sharing or accessing the data. To utilize the cloud to its full potential a confidential data sharing mechanism that is fine-grained, dynamic, scalable, accountable and secure. A method for constructing a secure cloud storage system is defined which supports dynamic users and data records. The previously existing system does not support the above properties. A system is designed by providing authentication and encryption along with instantiating the design with a revocable signature and broadcast encryption with cipher texts and private keys of constant size.

System Work

A multi-cloud storage system that could leverage some of commodity cloud providers (e.g., Amazon, Google) with the aim of dispensing is given as actual with across distinct administrative domains. This “cloud of clouds” version is receiving growing attention in recent times with cloud storage organizations which encompass EMC, IBM, and Microsoft, presenting merchandise for multi cloud structures. The foe might additionally finish that both by method for leveraging flaws or backdoors in the key-technology programming system or by means of compromising the module that saves those keys (within the cloud or in those person). Since cipher text obstructs are disbursed crosswise over servers facilitated inside exceptional domains.

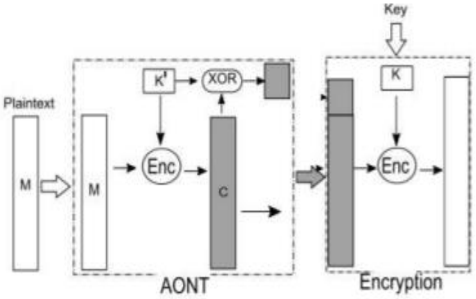


Figure.1. **Preprocessing cipher encryption**

Bastion departs starting with existing AON encryption schemes. Current schemes require an pre-processing round from cipher encryption to the AONT, went with accompany by another round about block cipher encryption figure 1.

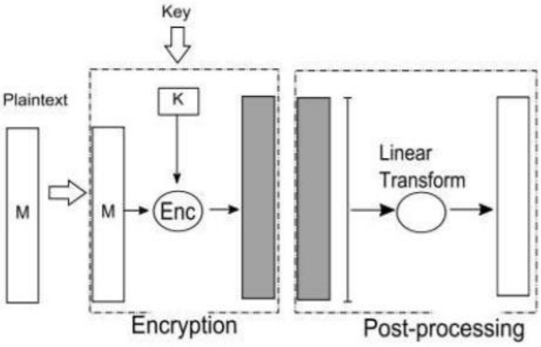


Figure.2. **Post processing Cipher Encryption**

Differently, bastion initially encrypts that detail for one circular about block cipher encryption, following which applies an proficient straight post-processing of the cipher text figure 2. A polynomial-time algorithm a need no unimportant profit on separating those conclusion security from of bastion might be utilized as black-box through another polynomial-time set about decides B should intrude the conclusion security of the underlying encryption mode.

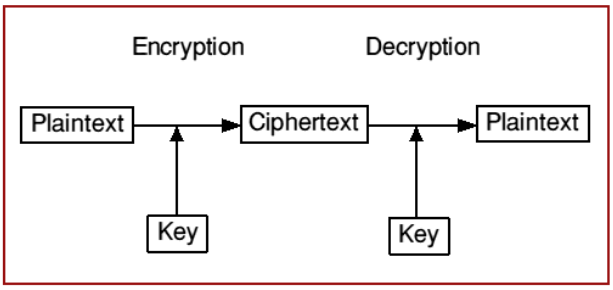


Figure.3. **Cryptographic process**

Those suggested methodology administer absolute cryptographic magic for every information record about client. The figure 2 depicts those generally methodology of the encryption, decryption, and more enter management. Those client substance and CS portal need aid both reliant with one another (to fruition on whole information security furthermore sharing process.

The different modules described in the system are

- File Upload
- File Download
- File Update

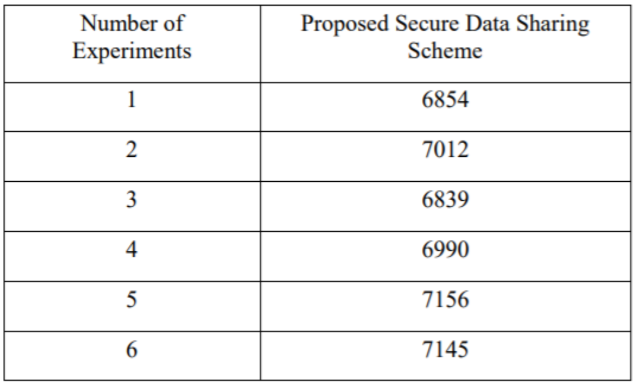
The modules are described below:

File Upload

If the owner wants to share data among a group of users then the first step that he must perform is to send a file encryption request to the cloud. The request also includes attributes like the file *f,* that is to be shared and the list *l,* of users that are to be allowed access to the respective file. The list *l,* also includes the access rights for the users i.e. READ-ONLY or READ-WRITE access. A number of different parameters also set to impose finer control on the data. This list *l,* is used to generate an access control list for the data. A new list is created only when the data has to be shared with a new group or a non-existing group. If the group has been created previously, request for encryption will not contain *l*; instead, the ID of the occuring group will be sent. After the request is received the group of users is created from access control list. the access control list is maintained for every file. It contains information like file ID size, owner details, list of users and their ID’s, etc. If the group previously existed, only the access control list is generated. Next, appropriate cipher like Advanced Encryption Standard (AES) is used to encrypt the file . An encrypted file is generated ad output. A private key is generated for every user for later authentication. In order to protect the file’s integrity, a hash-based message authentication code (HMAC) signature is calculated for every encrypted file.

Table.1. Encryption memory consumption of the proposed Scheme

Upload time = Time of submission of request+ processing of encryption



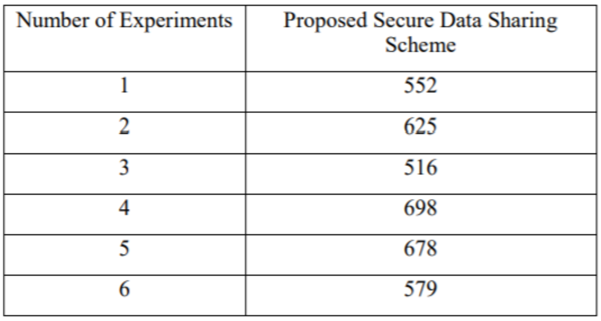
Number of Experiments	Proposed Secure Data Sharing Scheme
1	6854
2	7012
3	6839
4	6990
5	7156
6	7145

Table.1. **Upload Time**

File Download

First the user has to be authenticated or authorized. Only after the user has been authorized he can search for files and request for downloads. The cloud first verifies whether the user is genuine. The download/decrypt request includes a part of the user's key i.e. his private key. This private key provides authentication for the user, Since any two users cannot have the same private key they cannot use another users key as an imitation. After a successful authentication the download/decryption is initiated. If correct private key is received request is accepted else it is denied. Providing that the decryption is successful the file is sect to the user that requests it via. a communication channel. As with the file upload process, file downloading can directly be done by the cloud server instead of the user.

Upload time = Time of submission of request+ processing of decryption



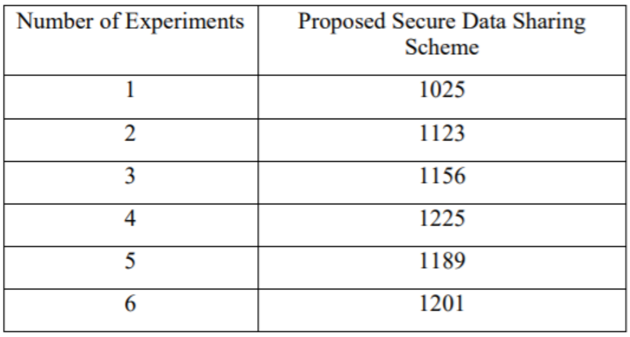
Number of Experiments	Proposed Secure Data Sharing Scheme
1	552
2	625
3	516
4	698
5	678
6	579

Table.2. **Download Time**

File Update

The procedure for updating a file is similar to that of uploading. The only difference being, during updation, the activities for creating the access control list are not performed. The user, after downloading a file performs some changes in it and then requests the CS for an update.

The request contains the details of the user and the files alongside the requested changes. Once it is verified that the user has write access to the file it is encrypted. The latter encrypted file is uploaded to the cloud and the existing file is removed or deleted.



Number of Experiments	Proposed Secure Data Sharing Scheme
1	1025
2	1123
3	1156
4	1225
5	1189
6	1201

Table.3. **Key Generation time**

CONCLUSIONS

This paper attempts to outline a framework for data sharing that is comparatively secure while being efficient to implement. A cost-effective executable sort of the conventional key-aggregate cryptosystem (KAC) has been described in this paper. Our development serves as a productive result to a few information offering requisitions on the cloud, including collaborated oriented information sharing, result permit circulation and more therapeutic information sharing.

An example of how the existing construction has been modified to achieve a secure model is presented. The concept of KAC has been coupled with broadcast encryption to achieve an improved efficiency.

REFERENCES

1. Sherman SM Chow, Yi-Jun He, Lucas CK Hui, and Siu Ming Yiu. Spice–simple privacy-preserving identity-management for cloud environment. In Applied Cryptography and Network Security, pages 526–543. Springer, 2012.
2. Cheng-Kang Chu, Sherman SM Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H Deng. Key-aggregate cryptosystem for scalable data sharing in cloud storage. Parallel and Distributed Systems, IEEE Transactions on, 25(2):468–477, 2014.
3. Selim G Akl and Peter D Taylor. Cryptographic solution to a problem of access control in a hierarchy. ACM Transactions on Computer Systems (TOCS), 1(3):239–248, 1983.
4. Mikhail J Atallah, Marina Blanton, Nelly Fazio, and Keith B Frikken. Dynamic and efficient key management for access hierarchies. ACM Transactions on Information and System Security (TISSEC), 12(3):18, 2009.
5. Fuchun Guo, Yi Mu, and Zhide Chen. Identity-based encryption: how to decrypt multiple ciphertexts using a

single decryption key. In Pairing-Based Cryptography–Pairing 2007, pages 392–406. Springer, 2007.

6. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Advances in Cryptology–EUROCRYPT 2005, pages 440–456. Springer, 2005.
7. Michel Abdalla, C´eline Chevalier, and David Pointcheval. Smooth projective hashing for conditionally extractable commitments. In Advances in Cryptology-CRYPTO 2009, pages 671–689. Springer, 2009.
8. Cong Wang, Sherman S.-M. Chow, Qian Wang, Kui Ren, and Wenjing Lou. Privacy-preserving public auditing for secure cloud storage. Cryptology ePrint Archive, Report 2009/579, 2009.
9. C. Charnes, J. Pieprzyk, and R. Safavi-Naini, "Conditionally secure secret sharing schemes with disenrollment capability," in ACM Conference on Computer and Communications Security (CCS), 1994, pp. 89–95.
10. A. Desai, "The security of all-or-nothing encryption: Protecting against exhaustive key search," in Advances in Cryptology (CRYPTO), 2000, pp. 359–375.
11. D. Bhardwaj and K. Kant, "Qos-Aware Routing Protocol using Adaptive Retransmission of Distorted, Descriptions in MDC for MANETs", International Journal of Ad Hoc and Ubiquitous Computing. Vol. 28, No. 1, pp. 55-67, 9 May 2018 [SCI. Impact Factor: 0.714].
12. Bhardwaj, D., Jain, S.K., Singh, M.P. "Estimation of network reliability for a fully connected network with unreliable nodes and unreliable edges using neuro optimization" International Journal of Engineering, Transactions A: Basics 2(4), pp. 317-332, 2009.
13. Kumar, R., Bhardwaj, D., Mishra, M.K. "Enhance the Lifespan of Underwater Sensor Network through Energy Efficient Hybrid Data Communication Scheme" International Conference on Power Electronics and IoT Applications in Renewable Energy and its Control, PARC 2020 9087026, pp. 355-359, 2020.
14. Kumar, R., Bhardwaj, D. "An improved moth-flame optimization algorithm based clustering algorithm for VANETs" Test Engineering and Management 82(1-2), pp. 27-35, 2020
15. Srivastava, Varun Kar Lal and Asthana, Amit (2019). An Efficient Software Source Code Metrics for Implementing for Software Quality Analysis. International Journal on Emerging Technologies, 10(4): 308–313.
16. B, Madhuravani. (2019). Strong and Secure Mechanism for Data Storage in Cloud Environment. International Journal of Advanced Trends in Computer Science and Engineering. 8. 29-33. 10.30534/ijatcse/2019/0681.32019.