ISSN2515-8260 Volume03, Issue06, 2016

KEY AGGREGATE CRYPTOSYSTEM FOR SCALABLE DATA SHARING BY CIPHER TEXT DATA IN THE CLOUD

M.ERAMMA,T.ABDUL RAHEEM,
Assistant professor, Assistant professor,.
Department of CSE
St.Johns College of Engineering and Technology, Yemmiganur, Kurnool (Dist).

Abstract:
Data sharing is a significant usefulness in distributed storage. In this article, we tell the best way to safely, proficiently, and deftly share information with others in distributed storage. We depict new open key cryptosystems which produce steady size ciphertexts with the end goal that effective appointment of decoding rights for any arrangement of ciphertexts are conceivable. The curiosity is that one can total any arrangement of mystery keys and make them as minimized as a solitary key, however including the intensity of the considerable number of keys being totaled. At the end of the day, the mystery key holder can discharge a consistent size total key for adaptable decisions of ciphertext set in distributed storage, however the other scrambled documents outside the set remain confidential. This minimized total key can be helpfully sent to others or be put away in a brilliant card with constrained secure stockpiling. We give formal security examination of our plans in the standard model. We additionally portray other utilization of our plans. Specifically, our plans give the main open key patient-controlled encryption for adaptable chain of importance, which was at this point to be known. In our cryptosystem, ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption.
Keywords -- Cloud storage, Cloud storage, data sharing, key-aggregate encryption, Attribute-based encryption

I. INTRODUCTION

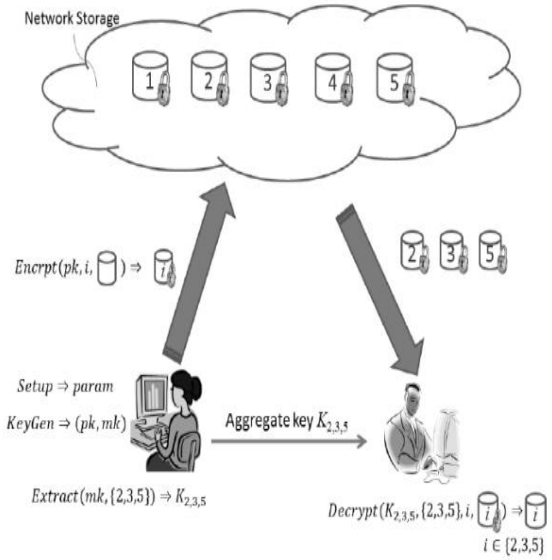
Cloud computing has created tremendous momentum in the IT industry that can be used to understand the kinds of computing, storage, and applications. Several IT companies dump data to cloud storage. Different users can access or send information stored in the cloud, regardless of their location. Distributed storage is picking up ubiquity as of late. In big business settings, we see the ascent popular for information redistributing, which aids the key administration of corporate information. It is likewise utilized as a center innovation behind numerous online administrations for individual applications. These days, it is anything but difficult to apply with the expectation of complimentary records for email, photograph collection, record sharing or potentially remote access, with capacity size more than 25GB (or a couple of dollars for additional than 1TB)[1]. Together with the present remote innovation, clients can get to practically the entirety of their documents and messages by a cell phone in any side of the world.

Thinking about information security, a customary method to guarantee it is to depend on the server to implement the entrance control after validation, which implies any unforeseen benefit heightening will uncover all information[2]. In a common occupancy distributed computing condition, things become much more terrible. Information from various customers can be facilitated on isolated virtual machines (VMs) however dwell on a solitary physical machine. As to of records, there are a progression of cryptographic plans which go similarly as permitting an outsider examiner to check the accessibility of documents for the information proprietor without spilling anything about the information [3], or without settling the information proprietors namelessness [4]. Moreover, cloud clients likely won't hold the solid conviction that the cloud server is working admirably as far as secrecy.

If the storage is compromised the amount of information loss will be limited. One disadvantage of encrypting data is that it severely limits the ability of users to selectively share their encrypted data at a fine-grained level. Suppose a particular user wants to grant decryption access to a party to all of its Internet traffic logs for all entries on a particular range of dates that had a source

IP address from a particular subnet. The user either needs to act as an intermediary and decrypt all relevant entries for the party or must give the party its private decryption key, and thus let it have access to *all* entries. Neither one of these options is particularly appealing.

SYSTEM ARCHITECTURE:



Here we portray the primary thought of information partaking in distributed storage utilizing KAC, represented in following figure. Assume Alice needs to share her information m1,m2,… .mn on the server. She initially performs Setup (1λ ; n) to get param and execute KeyGen to get general society/ace mystery key pair (pk, msk). The framework parameter param and open key pk can be made open and ace mystery key msk ought to be stayed discreet by Alice. Anybody (counting Alice herself) would then be able to scramble every mi by Ci = Encrypt (pk, I, mi). The scrambled information are transferred to the server. With param and pk, individuals who help out Alice can refresh Alice‟s information on the server. When Alice is happy to share a set S of her information with a companion Bob, she can figure the total key KS for Bob by performing Extract (msk, S). Since KS is only a steady size key, it is anything but difficult to be sent to Bob by means of a protected email. In the wake of getting the total key, Bob can download the information he is approved to get.

PROPOSED SYSTEM

With the development of various stages for sharing the data's by means of cloud server, cloudlets and so on, the keywords search is necessary to search the files from the cloud storage. To maintain the information safety the security level should be enhanced. The proposed system is designed using three modules namely. Data Owners, Data Users, Cloud Servers as illustrated in the figure 1.

Figure 1 : System architecture of ciphertext data

RELATED WORK:

a) Cipher Cloud :

Cipher Cloud provides a unified cloud encryption gateway with award-winning technology to encrypt sensitive data in real time before it„s sent to the cloud. It also protects enterprise data by using operations-preserving encryption and tokenization in both private and public cloud communication without affecting functionality, usability, or performance. Cipher cloud provides ability to create a unified data protection policy across all clouds that users probably used to store data, such as Google, Amazon, Azure and others. By applying encryption in a cloud security gateway, Cipher Cloud eliminates the inherent security, privacy, and regulatory compliance risks of cloud computing.

b) Cryptographic Cloud Storage:

In this, proposed a virtual private storage services that would satisfy the standard demands (Confidentiality, integrity, Authentication .etc.). Most of the demands are done by encrypting the documents stored in the cloud. However, such encryption leads to hardness in both the search processes through documents and the collaboration process in real time editing. the architecture of the cryptographic storage service that are used in solving the security problems of "back-ups, archival, health record systems, secure data exchange and e-discovery". It contains three main components: Data Processor (DP) that processes data before sending it to the cloud, Data Verifier (DV) which verifies data"s integrity and finally, Token Generator (TG) that generates tokens allowing the service provider to retrieve documents. Before uploading data to the cloud, Aliceuses the data processor to encrypt and encode the documents along with their metadata (tags, time, size, etc.), then she sends them into the cloud. When she wants to download some documents, Alice uses the TG to generate a token and a decryption key. The token is sent to the storage provider to select the encrypted files to be downloaded. After that, the DV is invoked to verify the integrity of the data using a master key. The document is decrypted using the decryption key.

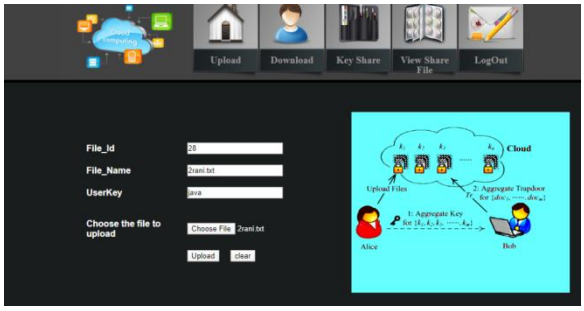
System Implementation:

a) Key Aggregate framework: The proposed system is basically design on the basis of key aggregation encryption. Here we are using two keys to encrypt and decrypt the data which are secret key and its aggregate key. The data owner creates the public system parameter and generates a secrete key which is public key. Data can be encrypted by any user and he may decides ciphertext block associated with the plaintext file which want to be encrypted.

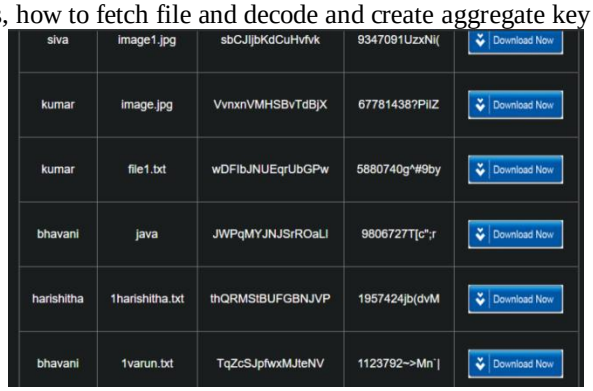
b) Aggregation of Secret Keys: Introducing a special type of public-key encryption which we call key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public key, but also under an identifier of ciphertext called class. The key owner holds a master-secret called master secret key, which can be used to extract secret keys for different classes. More importantly, the extract key can be an aggregate key which is 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 ciphertext classes.

EXPERIMENTAL RESULTS AND DISCUSSION

Data Security in Cloud Using Key Aggregate Cryptosystem using the developed technique generates much less load on system about secret key as cyphertext classes increases. In order to show the effectiveness of the developed method over the conventional and state-of-the-art Key Aggregate Cryptosystem techniques, several example of different area with different features are used. We have checked our system on several input documents which are belongs to different category like Cryptographic Keys for a Predefined Hierarchy, Attribute-based encryption, Cloud Encryption Models, Compact Key in Identity-Based Encryption, Compact Key in Symmetric Key Encryption, etc. And every time our developed system has proved superiority among all existing systems.

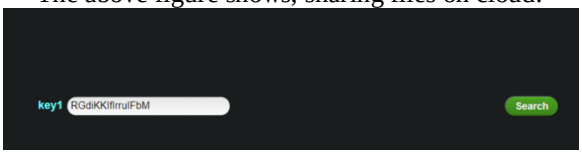


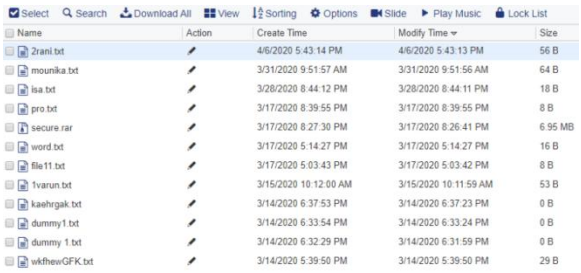
The above figure shows, how to fetch file and decode and create aggregate key for sharing it on cloud.



siva	image1.jpg	sbCJIjbKdCuHvfvk	9347091UzxNi(	Download Now
kumar	image.jpg	VvnxnVMHSBvTdBjX	67781438?PIIZ	Download Now
kumar	file1.txt	wDFIbJNUEqrUbGPw	5880740g^#9by	Download Now
bhavani	java	JWPqMYJNJSrROaLI	9806727T[c";r	Download Now
harishitha	1harishitha.txt	thQRMStBUFGBNJVP	1957424jb(dvM	Download Now
bhavani	1varun.txt	TqZcSJpfwxMJteNV	1123792~>Mn`|	Download Now

The above figure shows, sharing files on cloud.





If we grant the key one by one, the number of granted keys would be equal to the number of the delegated ciphertext classes. With the tree-based structure, we can save a number of granted keys according to the delegation ratio. On the contrary, in our developed approach, the delegation of decryption can be efficiently implemented with the aggregate key, which is only of fixed size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges.

CONCLUSION AND FUTURE WORK

Step by step instructions to secure clients' information protection is a focal inquiry of distributed storage. With progressively numerical devices, cryptographic plans are getting increasingly flexible and regularly include various keys for a solitary application. In this article, we consider how to "pack" mystery keys out in the open key cryptosystems which bolster appointment of mystery keys for various ciphertext classes in cloud capacity. Regardless of which one among the force set of classes, the delegatee can generally get a total key of steady size. Our methodology is more adaptable than progressive key task which can just spare spaces.

In the event that every single key-holder share a comparative arrangement of benefits. A limitation in our work is the predefined bound of the number of maximum ciphertext classes. In cloud storage, the number of ciphertexts usually grows rapidly. So we have to reserve enough ciphertext classes for the future extension. In this paper, we addressed an important issue of secure data sharing on untrusted storage. We investigated the challenges pertained to this problem and proposed data security in cloud using key aggregate cryptosystem. In this paper, proposed system is found to be very efficient for sharing the data on cloud. For this we have used Key aggregate encryption algorithm which support delegation of secret keys for different ciphertext classes in cloud storage. It also produces constant-size ciphertexts such that efficient delegation of decryption rights for any set of ciphertexts which is here possible. Since in traditional methods unexpected privilege escalation will expose all data. And that we are able to avoid and provide more security by using key aggregate algorithm.

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 – ACNS 2012*, ser. LNCS, vol. 7341. Springer, 2012, pp. 526–543.

[2] L. Hardesty, "Secure computers aren't so secure," MIT press, 2009, 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 Data on the Cloud via Security-Mediator," in *International Conference on Distributed Computing Systems - ICDCS 2013*. IEEE, 2013.

[5] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng, "Dynamic Secure Cloud Storage with Provenance," in *Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday*, ser. LNCS, vol. 6805. Springer, 2012, pp. 442–464.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," in *Proceedings of Advances in Cryptology - EUROCRYPT '03*, ser. LNCS, vol. 2656. Springer, 2003, pp. 416–432.

[7] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," *ACM Transactions on Information and System Security (TISSEC)*, vol. 12, no. 3, 2009.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] G. C. Chick and S. E. Tavares, "Flexible Access Control with Master Keys," in *Proceedings of Advances in Cryptology – CRYPTO '89*, ser. LNCS, vol. 435. Springer, 1989, pp. 316–322.

[13] 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.

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

[15] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," *Information Processing Letters*, vol. 27, no. 2,.
[17] Q. Zhang and Y. Wang, "A Centralized Key Management Scheme for Hierarchical Access Control," in *Proceedings of IEEE Global Telecommun*