International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 07 | July-2016 www.irjet.net p-ISSN: 2395-0072

ideal Aggregated Key Cryptosystem for Maintaining Security of Cloud Using AES Algorithm

Faizan Ahmad
PG Scholar, Department of computer technology
KITS, Ramtek,india.

S.V Hemant
Assistant Prof., Department of computer technology
KITS, Ramtek,india.

***ABSTRACT**: The popularity of the cloud has been increase day by day for sharing data or to access different cloud services. We know that cloud provides various kinds of services like storage, networking, business application, etc. but the reason behind the success of cloud is the storage facilities that are provided by CSP. As we outsource our data on to the cloud server we cannot ensures the security of data. In order to secure data on to the cloud server we propose a hybrid approach of AES algorithm and aggregated key cryptosystem. Here different classes of cipher text are emerges by using splitting algorithm so rather giving actual decryption key for each class of cipher text we are giving only master secret key which is an aggregated key that posses the power of all keys. Finally data are decrypt and merge using merging algorithm at the requested user system. In these cryptosystem we make decryption key more powerful.*

Index term: Cloud storage, AES algorithm, , splitting and merging algorithms and Key aggregated encryption.

I.INTRODUCTION

Cloud computing has been emerges as the central source of services that providing different functionality over the period of time. Different functionalities of cloud are using in business, entertainment and other social activity that turnout to be 80% of assets of 1000 companies are turning to other CSP for their business need mainly in I.T sector.

What is cloud computing? U.S national institute of standards and technology gives definition: The cloud enables convenient ,on demand network access to a shared pool of computing resources-networks, servers, storage applications, and services among others several key features are crucial to the cloud offering and to today's dynamic business environment. Cloud computing are lacks in some security aspects the common security issues of cloud is data theft attacks. Sharing resources is one of the important functionalities of cloud storage. In this paper we show how to ensure security of the outsource data on to the cloud server which is going to share.

We are proposing new public key cryptosystem which is an hybrid approach of Key aggregated cryptosystem and AES algorithm that forms same size of cipher texts such that it validate and give the decryption rights for any set of cipher text to the requested user. Here we are using Microsoft azure as cloud storage platform and AES algorithm for the encryption of data along with splitting and merging algorithm at the server side. Data which is going to be share is divided into four parts by using splitting algorithm after that there is key generation using auto key generation. By using this key encryption is performed and the data is going to be store in different containers of cloud storage which consists of several blobs. Now if valid user wants this data than data owner give Master secret key (Msk) that posses the power of all decryption keys to the valid user now this Master secret key (Msk) will be enter by the valid user than automatically data is merge by using merging algorithm from different cloud storage containers and it is going to be downloaded by the valid user. In this way resources are going to share in secure manner without losing confidentiality of data which is the main objective of paper.

II. AES ALGORITHM

The principal drawback of triple DES (which was recommended in 1999) . Federal Information Processing Standard FIPS PUB 46-3 as new standard with 168-bit key) is that the algorithm is relatively Sluggish in software. A secondary drawback is the use of 64-bit block size, for these reasons or both efficiency and security. a larger block size is desirable. In 1997 National Institute of Standards and Technology (NIST) issued a call for proposals for a new Advanced Encryption Standard (AES), which should have security strength equal to or better than 3DES and significant improved efficiency. In addition, NIST also specified that AES must be a symmetric block cipher with. Block length of 128 bits and support for key length of 128,192 and 256 bits. In first round of evaluation, 15 proposed algorithms were accepted. A second round narrowed to 5 algorithms. NIST completed its evaluation process publish final standard (FIPS PUB 197) in November 2001. NIST selected Rijindel as the proposed AES algorithm. The 2 researchers of AES are Dr. Joan Daemon end Dr. Vincent Rijmen from Belgium.

The Advance Encryption Standard (AES) was announced by the National institute of Standards and Technology (NIST) in Novemher2001.It is the successor of Data Encryption Standard (DES) which cannot be considered as safe any longer. Because or its short key with a length of only 56 bits, to determine which algorithm would follow DES. NIST called for different algorithm proposals in a sort of competition. The best of all suggestion would become the new AES. In the final round of this competition the algorithm Rijndael, named after its Belgian inventors Joan Daemen and Vincent Rijmen won because of its security, ease of implementation and low memory requirements. There are three different versions of AES. All of them have a block length of 128 bits. Whereas the key length is allowed to be 128,192 or 256 bits. In this application report, only key length of 128 bits is discussed. The AES algorithm consists of ten rounds of encryption as can be seen in figure 1.

First the 128-bit key is expanded into eleven so-called round keys, each of them 128 bits in size. Each round includes a transformation using the corresponding cipher key to ensure the security of the encryption.

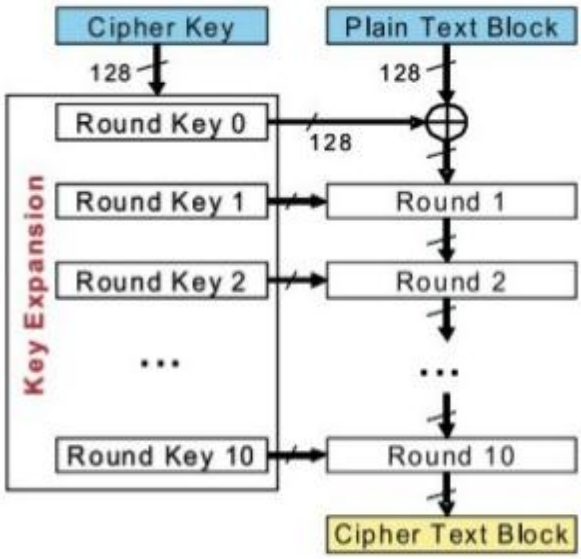


Figure1. AES algorithm Structure

III SPLITTING AND MERGING ALGORITHMS

When the user get register than user come into the actual web page for uploading the data on to the cloud server. When the data is success fully uploaded on to the cloud server then it is going to be split in the four parts using Splitting algorithm. When decryption is performed than at the server data automatically merges using merging algorithm.

Splitting Algorithm:-

(i) Open a large file in read mode by Binary reader stream.

(ii) Execute step (iii) and step (v) until file read reach at the end of file.

(iii) Read from that stream and set these bytes in byte array.
(iv) Make a new file name from original file name with splitting number.
(v) Save these bytes form the array to a new file with 'File class' 'Write All Bytes' method.
(vi) Close the .net binary stream.

Merging Algorithm

(i) Create a binary writer stream and open a binary file in append mode.
(ii) Execute from step iii to v
(iii) Generate file name in runtime depends on pervious file name.
(iv) Check for last file slice, last file slice name ends with last character 'E'.
(v) Read all bytes from file and set in a byte array then write these byte data by binary writer.
(vi) Close the binary writer.

IV. KEY AGGREGATE ENCRYPTION

- **Important Components of KAC:-**

KAC scheme will have five algorithms that run in polynomial time. Data owner will maintain the public system parameter through Setup which generates pair of public key/Master secret key through KeyGen. Encryption of messages will be done through Encrypt by anyone or data owner and also decides which cipher text class is associated with the plaintext message which is to be encrypted. Owner of the data will use master secret key to generate an aggregate decryption key for a particular set of cipher text classes through Extract. Now the keys are passed to delegates in secure manner at last any user with an aggregate key can decrypt any cipher text through Decrypt.

Setup phase: The setup algorithm takes no input other than implicit security parameter. It outputs the public parameters PK and a master key MK.

Encrypt Phase: Encrypt (PK, M, A). The encryption algorithm takes as input the public parameters PK, a message M and an access structure A over the universe of attributes. The algorithm will encrypt M and produce a cipher text CT such that only a user that possesses a set of attributes that satisfies of the access structure will be able to decrypt the message. We will assume that cipher text implicitly contains A.

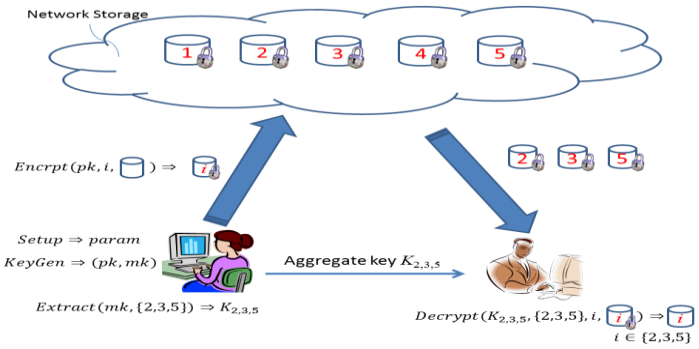
Key Gen Phase: Key Generation (MK, S). The key generation algorithm takes as input the master key MK and a set of attributes S that describes the key; it outputs a private key SK.

Decrypt Phase: Decrypt (PK, CT, SK). The decryption algorithm takes as input the public parameters PK, a cipher text CT, which contains an access policy A and private key SK, which is a private key for a set S of attributes satisfies the access structure A then the algorithm returns a message M.

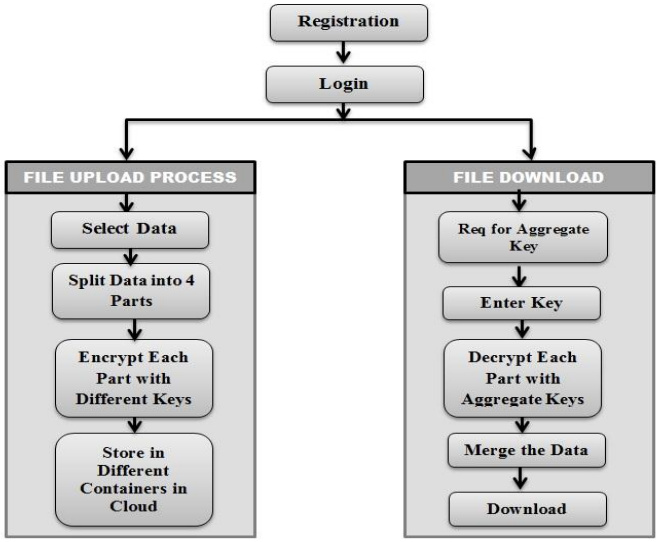
User Management : This part of the module mainly deals with the users who are associated with proposed system. In this part of the module that the username and password are checked before they can log into the system. The user management module is also used for the management of the registered user by the administrator. The user management is the place where the administrator manages the GUI that helps the users to easily use the system without difficulty. Here we implement the side that is directly in contact with the user. A user is allowed to enter the system after authentication of that particular user. The users of the system have to provide username and password, if particular user is not in the login table, then he can't access the system.

Data management: The storage cloud is maintained by a third party cloud provider (e.g. Amazon S3) and keeps the data on behalf of data owner. We emphasize that we do not require any protocol and implementation changes on the storage cloud to support our system even a naive

storage service that merely provider file upload/download operations will be suitable.



Flowchart:



V. ACKNOWLEDGMENT

I would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

VI. CONCLUSION

Data security on to the cloud server becomes one of the most important measure of the cloud computing. There are several cryptographic techniques that involve only management of key user's class hierarchy. In this paper is on how to aggregate secret key in a public key cryptographic environment that support delegation of secret keys for the different classes of cipher text in cloud server also the person that want to access to the particular data will get same size master secret key.

VII. REFRENCE

[1] Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng Senior Member, *IEEE* key –Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage IEEE Transactions on Parallel and Distributed Systems. , 25(2) , 468.2014

[2] 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

[3] 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

[4] 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.

[5] 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.

[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