Middle-East Journal of Scientific Research
24 (Special Issue on Innovations in Information, Embedded and Communication Systems): 126-131, 2016
ISSN 1990-9233; © IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.IIECS.23150

Efficient and Secure Data Sharing Using AES and Diffie Hellman Key Exchange Algorithm in cloud

[1]S. Grace Sophia and [2]S. Prabakeran

[1]Computer Science and Engineering, KCG College of Technology, Chennai, India
[2]KCG College of Technology, Chennai, India

Abstract: Sharing data securely in cloud is an important task .This paper explains how data are securely, efficiently and flexibly stored in cloud. The proposed system describes the public-key cryptosystems that gives fixed size cipher texts classes that represents high performance of decryption of keys are available for all the cipher text that are created. The data Owner can release a fixed-size of single key and the remaining files that are encrypted remains confidential. These single key can be sent to others otherwise stored in a card with limited number of storage devices. AES and Diffie Hellman Key Exchange Algorithm are used to improve the security of the method.

Key words: Cloud storage · Key aggregate crypto system · Data sharing · Key exchange

INTRODUCTION

Cloud storage is widely suited because of its real time characteristics. Cloud allows to stores the data in online so that it can be accessed by any user from any place and at any time. It is a core for many online services. At present it is simple to apply for Gmail accounts, sharing of files with size above 25GB. With the use of wireless network, users can access their files and emails through the cellular phones from any place and also in cloud data will be available all time. Two methods can be used to share the keys

- Sender will encrypt all the files from the cloud using a single key and the respective keys are given directly to the receiver.
- Sender will encrypt each file with distinct key and send the secret keys to the receiver.

In First process if sender wants to share only some files, but all the files will be accessed by the data receiver that is present in cloud. In the second method, if sender wants to share 100 files, then the sender share 100 secret key with the receiver, which is not more appropriate. The cost and complexity increases when more number of keys to be shared with the receiver.

Key Aggregate Cryptosystem: Key-aggregate system, user can encrypt a message using a public-key and the ids of the cipher text classes called class the cipher text classes are divided into different divisions. Data owner's has a secret that are called masters-secret key that can be u sed to extract the keys for different set of classes. The extract key is an aggregate key which is compact and similar to the secret key for a single class; it aggregates power of their keys. A key-aggregate encryption cryptosystem includes five algorithmic steps such as Setup, KeyGen, Extract, Encrypt, Decrypt, in which the owner establishes the parameter by using Setup and generates the public /masters key pair by using Key Generation. Files are encrypted using Encrypt. The owner uses their secret keys to provide a Decryption key for a cipher text classes that are produced by the Extract. The provided keys are sending to the data Receivers safely through their mails. User having an aggregate key using that key files are decryption through cipher text using Decrypt.

Encryption can be doing using two methods - symmetric key encryption and asymmetric key encryption.

Symmetric Key Encryption: Symmetric key encryption, the encryption and decryption keys are similar data owner wants to share data to the other party and then they should give their secret keys to the encryptions.

Corresponding Author: S. Grace Sophia, Computer Science and Engineering, KCG College of Technology, Chennai, India.

Asymmetric Key Encryption: Asymmetric encryptions both encrypt and decrypt keys are different. These encryption methods are used for many applications.

For example, in an organization, employees upload the files that are encrypting to the cloud without notice to organization's master secret key. Solution for the method is the senders can encrypt their files using their public key; he will send a single constant-size key to the receiver to decrypt their files. The decryption key must be secure, so he will send through the protected channel. If the key size is smaller it is desirable.

Literature Survey: In 2015 S.Samyuktha *et al.* [1], worked on "Implementation of key Aggregate Cryptosystem with Steganography for Secured Data sharing in cloud Computing". This paper develops the concept of steganography the encrypted files are hidden and changed as image by using steganography and then uploaded in the cloud server. In this module the files, index value are encrypted using the public key. The files are hidden into an image and then stored in the cloud server.

Merits of this paper is Replacing them with information (encrypted mail, for instance). The files can then be exchanged without anyone knowing what really lies inside of them. Flaw of this paper is only a minimum data to be included in image.

In 2015 Suganyadevi *et al.* [2] worked on "Effective Data Sharing in Cloud Using Aggregate Key and Digital Signature". This paper introduces the concept of Advanced Encryption is used for encryption and decryption. Digital Signature is also implemented with Key Aggregate Cryptosystem (KAC) to provide Integrity for data stored in cloud.

Merits of this paper is it provides protection not only to the files stored in memory but also for messages that are broadcast on the network.AES is more secure and it supports larger key sizes up to 128 keys. Flaw of this paper Outsourcing of data to server may lead to leak the private data of user to everyone.

In 2015 Rashm i Khawale *et al.* [3] worked on "Development of improved Aggregated Key Cryptosystem for scalable data sharing". This paper introduces the Blowfish algorithm provides the higher security to the user and faster execution when compared to AES (Advanced Encryption standard) and also DES (Data Encryption Standard). Also the blowfish algorithm blowfish algorithm is unpatented and it does not require any license.

Merits of this paper are Blowfish has not known the weak points of security so it will make excellent encryption algorithm. Flaw of this paper is to improve the data access mechanism in the data sharing.

In 2013 Rushikesh V. Mahalle [4] worked on "A Review of Secure Data Sharing using Key Aggregate Cryptosystem and Decoy Technology". Decoy technology is the technology which is providing the information to the unauthorized user or the attacker. The useless data files on the demand of the system to do attack against the attacker. Using this technique the original information gets changed to other format so that the ex-filtering of the document or information is becomes impossible.

Merits of this paper it stores some of the decoy data files in the database of the customer as the database. Documents are of the Fake types so the original data is gets secured from the malicious insider attack.

In 2014 Baojiang *et al*. [5] worked on" Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage". This paper introduces KASE applies to any cloud storage it supports the search group of data sharing functionality, which means any user can selectively share the group of files with a selected users, while allowing them to perform keyword search over the other document. For each and every data owner, produce the public/master-secret key pair through using the Keygen algorithm. Keywords of each document can be encrypted using the Encrypt with the unique searchable encryption key. Then, the data owner can use the secret key pair to generate an aggregate searchable encryption key for a group of selected documents via the Extract. The aggregate key can be distributed securely (e.g., via secure e-mail or secure devices) to authorized users who need to access those documents. After that, an authorized user can produce a keyword trapdoor via the Trapdoor using this aggregate key and submit the key and store in the cloud. The keys are to search the file and share the details of the file to the others using the cloud.

Merits of this paper securely distributing files to users a large number of keyword for both encryption and also for search and the user stores the received keys securely and submit a more number of keys to the cloud order to perform the search over the shared data using cloud. Flaw of this paper Federated clouds have attracted a lot of attention nowadays, but this method cannot be applied in this case directly.

In 2005 Dan Boneh *et al*. [6] Hierarchical Identity Based Encryption (HIBE) system where the cipher text consists of three group elements and decryption requires two bilinear map computations, for the hierarchy depth. Some applications, such as the time lock encryption, are better served by using this method with short private keys more than cipher texts. To construct a HIBE system private key size grows linearly with hierarchy depth. The idea is to construct a hybrid with the HIBE. In Former system the private key will adjust when the identity depth increases, while in the latter system the private key grows rapidly. The hybrid is based on the similarities between both systems and exploits their opposite behavior with regard to private key size, to ensure that no private key contains more than group elements.

Merits of this paper are used to construct efficient cryptosystems. Forward secure encryption scheme is to guarantee that all messages encrypted before the secret key are remain secret. Cryptosystems designed for the efficient data to a dynamic group of users authorized to receive the data. Flaw of this paper Generic groups do not imply a lower bound at any specific group. Policy-makers need to share the same parameters for their HIBE schemes, which could be difficult to achieve.

In 2006 Matthew *et al.* [7] Identity-Based proxy re encryption, cipher texts are changed from one identity to another. Schemes are compatible with IBE deployments and do not require any extra key from the IBE trusted-party generator. An Identity-Based Proxy Re-encryption (IB-PRE) scheme is an extended Identity Based Encryption scheme. The first extension is an algorithm that generates re-encryption keys that can be given to the proxy. The proxy uses the second algorithm to apply these re-encryption keys to cipher texts and automatically re-encrypt them from one identity to another identity. In a non-interactive scheme, re-encryption generated by the delegator using only their IBE secret key the IBE master secret key is not required.

Merits of this paper it can be used to convert cipher texts from A to B but not vice versa. Proxy re-encryption is to allow B to read A's encrypted emails while she is on vacation. Messages are encrypted and are translated by the proxy into encryptions. The proxy does not learn the content of the messages being translated. Flaw of this paper it does not know the secret keys of A or B and does not learn the plaintext during the conversion.

In 2014 Cheng *et al.* [8] key aggregate cryptosystems the single key are generated for being all keys. The one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of cipher text set in cloud storage, but the other encrypted files outside the set remain confidential.

Merits of this paper are more secure. Flaw of this paper it does not know the details of the data that are uploaded to the cloud. Multi owners cannot access the data at the same time.

Proposed Work: The proposed system designs an efficient public-key encryption scheme it supports flexible allocation of key. In this scheme any set of the cipher texts (produced by the encryption scheme) is decrypt by a decryption key (generated by the data owner and also using the master-secret key).Special method are develop to solve the problem of public-key encryption system called key-aggregate cryptosystem (KAC). In key-aggregate system, user can encrypt a message under a public-key, id of the cipher text called class which means the cipher texts are divided into different. The owner of the key holds a secret key called Master secret key [9].

The master-secret key can be used to extract secret keys for different set of classes. The extracted keys are aggregate key which is compact as a secret key for the cipher text class. By this solution, Alice can send Bob a single aggregate key via a secure channel Eg: Email. Data receiver can download the encrypted photos from the cloud and then use this aggregate key to decrypt these encrypt.

System Architecture: Data owner encrypted and stores the files in cloud storage. According to the Data Requester the files are to be decrypt using aggregate key. The files are to be selected and store in cloud using their id and the password if the user is valid it will allow the user to store and retrieve the file.

The user login into the cloud the user will select the files that are to be uploaded. The files are uploading using the different keys to be encrypted. The master secret key and the symmetric key are to generate the secret keys. Using their keys user will decrypt the file. The single keys are generated for the different file the combination of key are to make the single aggregate Key. According to the user the files that they needed are to be decrypt using the aggregate key. The aggregate has the cipher text and the message and index and the set of indexes are combined together.

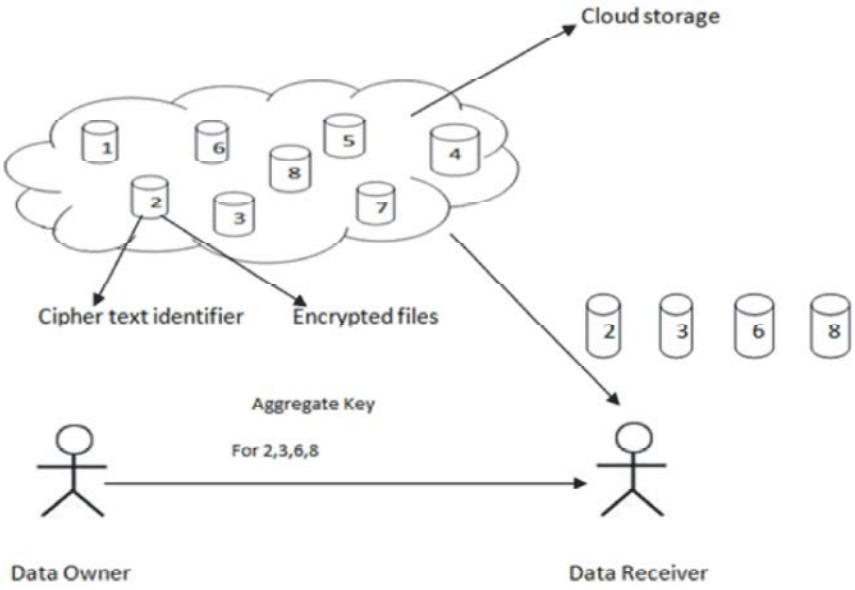

Fig. 1: Data Owner share files with id 2, 3, 6 and 8 with Data Receiver by sending him an aggregate key

Algorithm:

- Setup: Files are encrypted and kept in cloud.
- KeyGen: Generating the keys to the user. It will produce the Output as public and masters-secret key pair.
- Encrypt: For Message and the index i ◎ {1, 2, • • • , n}, the values are randomly selected and the cipher text are generated. AES algorithm is used for encrypting the files.
- Extract: For the set S of indices the aggregate key is computed as KS. Since S does not include 0, can always be retrieved from param.
- Decrypt (KS, S, i, C = c1, c2, c3): If i ◎ S, output. Otherwise, return the message: m for the data owner

AES Algorithm: Advanced Encryption Standard is a symmetric block cipher to encrypt the data that are to be preventing from the malicious Attack. It use same key for Encryption and Decryption. The data Owner store the file in the cloud during the Encryption the key will be generated using the same the respective files are to be decrypted. The key are provided to the users for the files to be received using the mail using that key files are to be decrypt. The 256 bit cipher block keys are used for encrypts and decrypts. Using a powerful 256-bit encryption algorithm, AES safely secure your most sensitive files. Once a file is encrypted, file is completely useless without the password. It simply cannot be read. Using a powerful 256-bit encryption algorithm

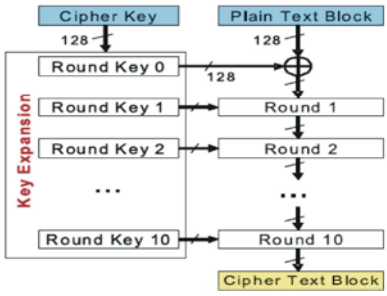

Fig. 2: Data structure of AES Encryption Algorithm.

AES encrypt is the perfect tool for anyone who carries sensitive information.AES encrypt is also the perfect solution for those who wish to backup information and store that data in a cloud-based storage service and any place where sensitive files might be accessible by someone else. AES encrypt is completely free open source software. The AES algorithm has four steps

- Substitute bytes
- Shift rows
- Mix columns
- Add round key

Diffie Hellman Key Exchange Algorithm: A and B, want to use insecure email to agree on a secret "shared key" that they can use to do further encryption for a long message. Diffie-Hellman method provides a way to SSL.A and B, using insecure communication, agrees on a huge prime p and a generator g.

A chooses some large random integer $x_A < p$ and keeps it secret. Likewise B chooses $x_B < p$ and keeps it secret. These are their "private keys".

- A computes her "public key" $y_A \equiv g^{x}{}_{A} \pmod p$ and sends it to B using insecure communication. B computes his public key $y_B \equiv g^{x}{}_{B}$ and sends it to A. Here $0 < y_A < p$, $0 < y_B < p$.
- A computes $z_A \equiv y_B{}^{x}{}_{A} \pmod p$ and B computes $z_B \equiv y_A{}^{x}{}_{B} \pmod p$. Here $z_A < p$, $z_B < p$. But $z_A = z_B$, since $z_A \equiv y_B{}^{x}{}_{A} \equiv (g^{x}{}_{B})^{x}{}_{A} = g^{(x}{}_{A}{}^{x}{}_{B} \pmod p$ and similarly $z_B \equiv (g^{x}{}_{A})^{x}{}_{B} = g^{(x}{}_{A}{}^{x}{}_{B} \pmod p$. So this value is their shared secret key.
- In this calculation, $y_B{}^{x}{}_{A} \equiv (g^{x}{}_{B})^{x}{}_{A}$ involved replacing $g^{x}{}_{B}$ by its remainder y_B, (in the reverse direction.

Implementation: Create an account in the cloud using the user id and the password. After creating the account selects the files to be uploaded in the server .If other user wants any files the secret keys send to the user using the mail ids. Different files are encrypted and stored in cloud using different keys one single key are used to decrypt the files. The AES Algorithm are use to encrypt the files.

Diffie Hellman key exchange Algorithm is to authenticate the user using their id and the password. After validating the user data will allow the user to decrypt the file using the respective aggregate key. The cloud simulators are to create an account in the cloud using the mail id and the password.

Data Owner Repository: The Information owner establishes a parameter for public systems via Setup. On input of a security level parameter 1^{e} and number of cipher text class n, it outputs the public system parameter param. Create an account in the cloud using the id and the name. According to the name and their ids the IP Address are to be generated by the cloud service provider.

Cipher Class Key Generation: It generates the public key and master-secret key pair. It is executed by information owner and it generates randomly a Public Key /Master Secret Key pair. Cipher class consisting of Data owner's id and message and the master/public key of the data owner attributes. Using the master key and public key the secret key are generated. Ciphering algorithms are using Key aggregate cryptosystem.

Aggregate Key: For a message m and an index i ◎ {1, 2… n}, using public key compute the cipher text. It is executed by data owner. Using the public key the message m and index i, it computes the cipher text as C. It is the process of pairing up the attribute information and using master-key and logical derivations of the data-owner attributes. Aggregate key is considered as secret key for data security for the data being outsourced in the cloud. Key aggregate cryptosystem is unique key generation scheme for secure and robust cloud data security mechanism It differs from the normal and cryptography techniques by generating the keys from the various attributes.

Secure Cloud Storage: For the set S of indices an aggregate key is computed as K_{Agg}. It is executed by the information owner for assigning the decrypting power for a particular set of cipher text classes and it outputs the aggregate key for set S denoted by K_{agg}. These files are stored in the cloud servers. In order to do that the cloud server have to configure through the VMware tool. In cloud servers client files are stored as secured files so the crypto processes have applied for crypto process.

Secure Cloud Retrieval: It is executed by a File Requester who received an aggregate key K_{Agg}, created by extract and sent by the data owner. On input K_{Agg}, set S, an index i denoting the cipher text class and cipher text C decrypted output message is produced.Retrieving data consist of retrieval of encrypted files from the cloud server and decrypted using respected secret keys. Data are provided to the users upon the authentication that are provided in Cloud system architecture.

Advantage: Secure key Cryptographic derivation. Privacy preserving Cloud data storage. It provides higher data security. It also supports data integrity process. A decryption is more powerful it allows multiple cipher texts classes during decryption, without increases or expands its size of the keys. The keys size is kept constant size.

Conclusion and Future Work: Key Aggregate Cryptosystem is used to shares the data in the secure manner. The combination of AES and Diffie Hellman key Exchange algorithm are to provide security and Authentication to the user while they uploading the file in the cloud.

A limitation of this process the limited amount of cipher text classes can be used. In cloud, the count of cipher texts are increased rapidly .So we have to allocate maximum cipher classes. Otherwise, we have to increase the public-key values.

REFERENCES

1. Samyuktha, S., C. Vijaya and D. Durai, 2015. "Implementation of key Aggregate Cryptosystem with Steganography for Secured Data sharing in cloud Computing" ISSN: 2277-9655, March, 2015
2. Suganyadevi, G. and S. Punitha Devi, 2015. "Effective Data Sharing in Cloud Using Aggregate Key and Digital Signature" 4(6).
3. Rashmi Khawale, Roshani Ade, 2015. "Development of improved Aggregated Key Cryptosystem for scalable data sharing" Rashmi Khawale *et al.*, / (IJCSIT) International Journal of Computer Science and Information Technologies, 6(2): 1792-1794
4. Rushikesh V. Mahalle, Parnal P. Pawade, "A Review of Secure Data Sharing in Cloud using Key Aggregate Cryptosystem and Decoy Technology" International Journal of Science and Research (IJSR).
5. Baojiang Cui, Zheli Liu and Lingyu Wang, 2014. "Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage" IEEE Transactions on Computers, 6(1).
6. Dan Boneh, Xavier Boyenboyen and Eu-Jin Goh, 2005. "Hierarchical Identity Based Encryption with Constant Size Ciphertext "Advances in Cryptology-EUROCRYPT 2005, volume 3493 of Lecture Notes in Computer Science, pp: 440-456, Springer, 2005.
7. Matthew Green Giuseppe Ateniese, "Identity-Based Proxy Re-Encryption" Information Security Institute Johns Hopkins University 3400 N. Charles St. Baltimore, MD 21218
8. Cheng-Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou and Robert H. Deng, 2014. Senior Member, IEEE Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage, 2(2).
9. Chitraranjan Ngangbamcha and Savitha Shetty, 2015. M. Tech. CSE NMAMIT, Asst. Prof. Dept. of CSE, NMAMIT, Nitte, Karnataka, India Nitte, Karnataka, India "Key Aggregate Cryptosystem Data Sharing in Cloud Storage"International Journal, 5(2).