ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 6, June 2017

Secure Sharing of Group Data in Public Clouds

[1] Anusha C H, [2] Dr.R.Aparna
[1] Siddaganga Institute of Tchnology ,[2] Siddaganga Institute of Tchnology
[1]anushaharish18@gmail.com,[2] raparna27@gmail.com

***Abstract*— Cloud computing is a rapidly emerging technology that provides storage and computing services to the customers on demand. Due to loss of control over the data and computation, many security issues will arise in cloud computing environment. In case of data shared among group members, there exist certain additional security issues that need to be addressed. The main purpose of this paper is to provide data security and efficient key management in group shared data using multi-clouds, for storing the encrypted data in blocks. Secure Sharing of Group Data in Public Clouds counters security issues in group shared data by providing data confidentiality, integrity, backward and forward access control, using AES 256 bit encryption technique and efficient key management. The main key, which is used for encrypting the data, is not stored anywhere. But the key is divided into two parts, one part of the key is stored in the trusted server and the other part is sent to the user. Encrypted data is divided into blocks and stored in different clouds. Whenever the user requests for the data, he sends his key to the server. The server retrieves the data blocks stored in different clouds and combines it to get the encrypted data. The encrypted data is finally decrypted using the key generated by combining the key sent by the user and the key stored in the server for the corresponding user.**

***Index Terms*—Backward and forward access control, confidentiality, integrity, multi-clouds.**

I. INTRODUCTION

One of the applications of cloud is Cloud Storage that makes organizations free from hosting in-house data storage systems. Low budget organizations can use high computing and storage services without investing on maintenance and infrastructure. However, in cloud storage systems, the loss of control over the data and computation gives rise to security issues for organizations and this prevents organizations using public clouds for storage. Due to loss of control over the data and computation, the cloud users are motivated to establish access control over the data. However, the confidentiality and privacy of the data should also be taken care by the cloud users. The confidentiality of data should be managed by the cloud users to ensure that the cloud provider does not know about the data stored on the cloud by the user. A typical tool used to preserve confidentiality and privacy of the data is Cryptography. The cloud users should ensure that their data is encrypted before uploading to the cloud. To ensure data security in cloud, the processes such as encryption, decryption and key management are all managed by the cloud user.

The cryptographic algorithms should be flexible enough to manage different users, maintain the access control and to handle the keys in an efficient way to ensure data security when the data is shared among a group. Some additional characteristics need to be addressed, when data is shared among a group, as opposed to the data handling that involves single user or two-party communication. A group consists of existing, departing and newly joining group members. They can act as an insider threat, violating confidentiality and privacy of the data. As insider threats are launched by trusted entities, they can be proved to be more damaging than other attacks. Due to different types of users in a group, there may arise many security issues.

The proposed methodology, Secure Sharing of Group Data in Public Clouds, deals with the security issues of sharing the data among group members in cloud. This scheme consists of three main entities. They are 1) Data owner and data users (Group), 2) a Trusted Server, and 3) Public Clouds. A group consists of data owner and data users in order to share the data among each other. Whenever the data owner wants to share the data with the users, he sends the data to be shared, the list of users with whom he wants to share the data and access permission to the trusted server. The server then encrypts the data and divides the data into blocks of equal size and uploads the encrypted data block by block among different public clouds.

Here the main key that is used to encrypt the data is not stored anywhere. Instead for each user, a random key is generated and this random key is stored in the server for each user. This random key is XORed with the main key that is used to encrypt the data for each user and corresponding XORed key is sent to each user. The data user, who wants to decrypt the data, sends his key to the server. The server then obtains the blocks of the data from different clouds and combines the blocks of data to get the entire encrypted data. The encrypted data is then decrypted using the key sent by the user.

The advantage of this scheme is, there is no need to re-encrypt the data for frequent changes in group membership. There is no computational overhead on the user as the operations such as encryption, decryption, key management, splitting and merging of data blocks, are taken care by the trusted server. As this scheme divides the data into blocks after encryption, it consumes less time than other schemes

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 6, June 2017

where the data is divided into blocks and then each block is encrypted. The key management is also easy and efficient.

II. LITERATURE SURVEY

In case of group-shared data, the group members might generate security issues such as backward access control i.e. accessing of past data by the newly joining member of the group and forward access control i.e. accessing of future data by the departing member of the group.

An efficient and provably-secure Group Key Management (GKM) was proposed in [1] based on key derivation method. The key derivation method uses keyed hash and XOR operation, this makes this scheme efficient. This scheme also used rekeying mechanism for the group member who is offline. The simple solution to backward and forward access control is to use rekeying mechanism. It involves generating a new key, decrypting all the encrypted data and again encrypting all the data with the new generated key. This scheme was proposed in [2], but this solution is not efficient for dynamic groups and frequent changes in group members.

For each member in the group a separate key is generated, but it is a cumbersome solution. In this case, the data is encrypted separately for each member with his/her key and if there are any changes in the data, then all the copies of data must be decrypted and again encrypting with changed contents [2]. If the whole symmetric key is given to the group member, he might turn into malicious user and he can change the data illegally being legitimate member of the group. He can also manipulate, decrypt and re-encrypt the data maliciously within a group and he can become insider threat [3].

A certificate-less proxy re-encryption (CL-PRE) scheme was proposed in [4], for secure data sharing in a group using the public cloud. In this scheme, the data to be shared is encrypted by the data owner using symmetric key encryption. Then the public key of the data owner is used to encrypt the symmetric key. The encrypted data and the encrypted key are uploaded to the public cloud. Here the cloud acts as proxy re-encryption [4] entity and it again encrypts the encrypted key. This re-encrypted key is only decrypted by the private key of the legitimate data recipient. And the certificates are not used to generate user's public and private key pairs. The public and private key pairs are generated by using the identity of the user. In this scheme, Bilinear pairing and BDH are used in proxy re-encryption. This makes CLE-PRE scheme computationally intensive. As compared to standard operations in finite fields, the computation cost of bilinear pairing is very high.

An efficient certificate-less encryption scheme was proposed in [5], for securely sharing data in public cloud, to overcome the computational complexity of bilinear paring. The key escrow problem in identity based encryption and the certificate revocation problem in public key cryptography were solved by Mediated Certificateless Public Key Encryption (mCL-PKE). Here the cloud is responsible for both the secure storage and the key generation. The cloud generates public and private key pair for each user and sends the public key to all the users. The data owner encrypts the data to be shared with the cloud generated public keys based on the access control policies and the encrypted data is uploaded to the cloud. Partial decryption of encrypted data is done at the cloud, after successful authorization, for the users. Then the users using their private key fully decrypt the partially decrypted data. The data is not fully decrypted by the cloud, so preserves confidentiality of the data and the keys.

A group key management scheme was proposed in [6], it was based on meta Proxy re-encryption scheme and it is RSA-based re-encryption scheme with special features. It is the first RSA-based PRE scheme proposed for group key management and has desired features of uni-directionality and multi-hop. No matter how many times the update of group key occur, the size of rekey history is a constant O (N) and no matter how many times the update of group key missed, the computation time of generating newest group key from rekey history is always O (log N). This property gives a practical solution for group key update when the users go off-line from time to time. However, this scheme is vulnerable against collusion attack of other members.

Secure cloud storage using AES encryption was proposed in [7], for secure data storage in cloud. This paper discusses various data security and privacy issues in cloud. AES 128 bits is used in this paper for providing data confidentiality and integrity. Here the data is divided into blocks and each block of the data is encrypted separately. These blocks are then uploaded to the cloud. But the disadvantage of this scheme is, it consumes more time as it encrypts each block of the data separately and also key management is difficult.

Secure Data Sharing in Clouds was proposed in [8], for sharing sensitive data in clouds. They have used efficient key management but the entire encrypted file is stored on the public cloud. This lead to insider threats if the user gets the entire data which is stored on the single cloud.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 6, June 2017

III. SYSTEM DESIGN

The proposed system, Secure Sharing of Group Data in public clouds, consists of three main entities as shown in Fig 1

- Data owner and data users (Group)
- A Trusted Server, and
- Public Clouds.

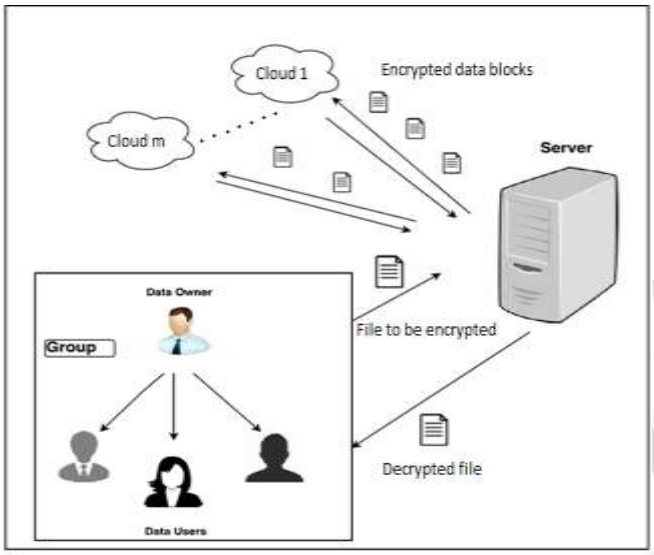

Fig 1: Architecture of the proposed system

Data owner and data users (Group): A group consists of data owner and data users. Data owner is the one who owns the data to be shared among group members or data users. Whenever he wants to share any data with group members, he sends encryption request to the server along with the data to be encrypted, group Id and list of data users.

Trusted Server: It is responsible for processes such as encryption and decryption of the data, key management, splitting and merging of encrypted data blocks. So there is no overhead on the user as well as cloud, which is an untrusted entity.

Public cloud: Different public clouds are used for storing encrypted data blocks. In this paper, we have used Dropbox and Amazon Web Services, public clouds. As we are using the cloud for only storing the data and it consists of basic operations such as uploading and downloading of data, it does not require any changes in the implantation on the cloud.

IV. IMPLEMENTATION

In this section, we present how Secure Sharing of Group Data in Public Clouds is implemented. We have implemented two algorithms for encryption and decryption of data to achieve security goals.

Algorithm 1: Algorithm for key generation and encryption of data

Input: File to be encrypted F,List of n users $u_1,u_2,...,u_n$,
m no. of clouds $cloud_0$ to $cloud_{m-1}$,
AES-256 algorithm,SHA-256 algorithm H
Output: Encypted blocks
Compute:
Generate a random number R_k of size 256 bits
Main key,$M_k = H(R_k)$
C=AES(F,M_k)
for *each user* u_i,*i=1 to n* **do**
Generate a random number K_{s_i} of size 256 bits
$u_1 \rightarrow K_{s_1}, u_2 \rightarrow K_{s_2}, ..., u_n \rightarrow K_{s_n}$
$K_{u_i} \leftarrow K_{s_i}$ XOR M_k , i=1 to n
save K_{s_i} in server database for user u_i , i=1 to n
send K_{u_i} to user u_i , i=1 to n
endfor
Divide c into b no of blocks $c_0,c_1,...,c_{b-1}$
Upload block c_i to cloud i (mod m), i= 0 to b-1

Fig 2: Algorithm for Key generation and encryption of data

Fig 2 shows algorithm for key generation and encryption of data. The data owner who wants to share the data with other group members sends request to the trusted server. The request to the server consists of file to be encrypted F, group Id and the list of users with whom the data owner wants to share the file F. The server, after receiving the request to encrypt the data, it will generate a random key R_k of length 256 bit. The random key R_k is given to SHA 256 bit hash function to obtain a hash value of length 256 bit. This hash values obtained is used as a main key M_k to encrypt the File F using AES 256 bit encryption technique. After the encryption of the file F, the server generates a random key of length 256 bit K_{s_i} for each user u_i, this key K_{s_i} will be stored in the server database for the corresponding user, u_i.

For each user, the corresponding K_{s_i} is XORed with the main key M_k, K_{s_i} is obtained which is sent to the corresponding user. The encrypted File C is then divided

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 6, June 2017

into blocks of equal size. The blocks with their filename consists of odd number will be uploaded to one public cloud (Dropbox) and the blocks with even number filename will be uploaded to another public cloud (Amazon Web Services).

Fig 3 shows an algorithm for decryption. Whenever the data user wants to access the File F stored as encrypted data blocks in different clouds, sends the download request to the server along with the corresponding key K_{u_i} which was sent to the user. After the server receives the request to decrypt the data, it fetches the corresponding K_{s_i} which is stored in the

Algorithm 2: Algorithm for decryption

Input: Blocks $c_0,c_1,...,c_{b-1}$ from cloud 0 to m-1,AES-256 algorithm
Output: Decrypted data
Compute: collect the blocks $c_0,c_1,...,c_{b-1}$ from cloud 0 to m-1
Merge the blocks $c_0,c_1,...,c_{b-1}$ to get c
Get the user key K_{u_i} from the requesting user u_i, i= 1 to n
Retrieve K_{s_i} from the server database for the corresponding user u_i, i
if K_{s_i} *does not exist in the database* **then**
  return access denied message
else
  $M_k = K_{s_i} XOR K_{u_i}$, i=1 to n
  F= AES(C, M_k)
  end

Fig 3: Algorithm for decryption

server database for that user. The K_{u_i} is then XORed with the K_{s_i} to obtain the main key M_k to decrypt the data. The encrypted data blocks from both the clouds are obtained and merged them to get the encrypted file C. The encrypted file C is then decrypted using main key M_k and sent to the user.

V. CONCLUSION

Secure Sharing of Group Data in Public Clouds achieved security goals such as data confidentiality, integrity, backward and forward access control when the data is shared among group members. As the compute intensive operations are performed at the server, there is no overhead on users and the cloud is used only for storing the encrypted data. This scheme used efficient key management. The encrypted data is divided into blocks and stored in different clouds increased security of the data and it takes less time when compared to schemes that used a technique where the data is divided into blocks and then each block is encrypted.

REFERENCES

[1] Y. Chen and W. Tzeng, "Efficient and provably-secure group key management scheme using key derivation," in *Proc. IEEE 11th Int. Conf. TrustCom*, 2012, pp. 295–302.
[2] A. N. Khan, M. M. Kiah, S. A. Madani, M. Ali, and S. Shamshir-band, "Incremental proxy re-encryption scheme for mobile cloud computing environment," *J. Supercomput.*, vol. 68, no. 2, pp. 624–651, May 2014.
[3] A. N. Khan, M. L. M. Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," *Future Gen. Comput. Syst.*, vol. 29, no. 5, pp. 1278–1299, Jul. 2013.
[4] L. Xu, X. Wu, and X. Zhang, "CL-PRE: A certificateless proxy reencryption scheme for secure data sharing with public cloud," in *Proc. 7th ACM Symp. Inf. , Comput. Commun. Security*, 2012, pp. 87–88.
[5] S. Seo, M. Nabeel, X. Ding, and E. Bertino,"An Efficient Certificateless Encryption for Secure Data Sharing in Public Clouds," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2107–2119, Sep. 2013.
[6] Y. Chen, J. D. Tygar, andW. Tzeng,"Secure group key management using uni-directional proxy re-encryption schemes," in *Proc. IEEE INFOCOM*, pp. 1952–1960.
[7] Babitha.M.P, K.R. Remesh Babu, "Secure Cloud Storage Using AES Encryption," *2016* International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT) International Institute of Information Technology (I²IT), Pune.
[8] Mazhar Ali, Revathi Dhamotharan, Eraj Khan, Samee U.Khan, Athanasios V. Vasilakos, Albert Y. Zomaya, "SeDaSC: Secure Data Sharing in Clouds," 2015 IEEE SYSTEM JOUNAL.