Searching & Sharing Of Data over Cloud Using Key Aggregate Cryptosystem

Miss. Swapnali V Arote, Prof. Rahul L Paikrao

ME Student in computer department , AVCOE Sangamner, Maharashtra ,India
HOD of computer department , AVCOE Sangamner, Maharashtra ,India

ABSTRACT

Now a day, Data stored on cloud can be shared with the multiple users. In our project we are using cloud for data storage. We are focusing on securely, efficiently, and flexibly share data among multiple users and data searching over encrypted cloud data. Privacy of user's data is a critical question of cloud storage. In our project client outsourced his/her encrypted data to a cloud server and authorize the latter to search on his/her behalf by using symmetric searchable encryption scheme. We are using public key cryptography for storing and sharing data. In our system we are creating a fixed size cipher texts that is impartial of data size. Therefore, it is possible to share data and to give authorized right for decryption of any set of cipher text. The data owner can share subset of files with other users by authorizing them an access rights and constant size of aggregate key. As per access rights search can be applied over restricted data set. This newly generated Aggregate Key can be send via email, mobile OTP or be stored in a smart card with very limited secure storage. Proposed approach aims to reduce security risk and is more flexible than existing schemes.

Keyword : - *Aggregate key encryption , cloud storage, data privacy, multiparty searchable encryption scheme.*

---

1. INTRODUCTION

Security of data is challenging in today's world because of sharing of data over internet. Data owner make services available to users for searching/accessing require data. Burt providing such services may attract more attacks. Cloud computing is most popular application which provide the data sharing and searching. In cloud computing the main issue is provide security to the end user to protect files or data from unauthorized user. Cloud services have most risky issues of data integrity and privacy protection because data does not store on his own servers. The security can be achieve by encrypting data. The clients can get information from anywhere in world client can store their information in the remote storage servers. But storing data at remote server is not secure. Hence the information's are scrambled before putting into server. But data encryption doesn't provide high security and prevent data from being modified. An unauthorized user may get access to the data while transferring data from data owner to cloud storage. An attacker may modify/alter data and store modified copy of data to storage or he may decrypt the data directly from the cloud server by getting cryptographic keys. When user get access to that data he can't recognize difference between original data and modified data. The challenge is how to securely, efficiently, and flexibly share data with others in cloud storage. Cryptography can be achieved in a two ways - one is symmetric key cryptography where same key will be used for encryption and decryption and other is asymmetric key cryptography where different keys are used i.e. public key for encryption and private key for decryption. Asymmetric key cryptography is more flexible and secured way for proposed approach [4].

Suppose user A want to upload his/her data on server and at the same time he want to share his/ her data with limited user. For security purpose he encrypt his data before uploading to the server. If another user B ask him to share his data. User A do not want to share all the data,he wants to share the data which is useful to user B.

There are two ways:

1. Symmetric cryptography in that user A can encrypt data with single secret key and share that secret key directly with the user B.

2. Asymmetric cryptography in that user A can encrypt data with distinct keys and send user B corresponding keys via secure channel. Both schemes are not adequate as transferring these keys require secure channel and if distinct keys are used for encryption then storage space can be expensive.

First method i.e. single key encryption is inadequate because it will allow the user B to get the access of all data. But user A do not want to allow for all data access. In second method encryption key is different and the decryption key is different so for multiple user we require multiple decryption keys, if the number of user increase then difficulty may arise in key management process. Both process having their own disadvantages.

The best solution for this problem is aggregate key encryption scheme that applies to any cloud storage that supports the searchable group data sharing functionality, a single aggregate key is used and it will be send to another user via a secure e-mail or mobile OTP which means any user may selectively share a group of selected files with a group of selected users, while allowing the latter to perform keyword search over the former. To support searchable group data sharing the main requirements for efficient key management
are twofold.

2. RELETED WORK

In this section, an overview of existing techniques are provided. The objective of this survey is clearly understand the limitations of existing schemes.

S. Yu, C. Wang, K. Ren, and Goyal et al[2]. uses **Attribute Based Encryption.** Attribute based encryption allows each cipher text to be linked with an attribute. User who want a secret key must go to the third party and get key by providing his identity. Then only he will decrypt the file. The secret key of user is authorized by independent authorities. Limitations of such a scheme are it requires more space to store keys. The size of Decryption key rises linearly and key management is expensive

R. Lu, X. Lin, X. Liang, and X. Shen [3], proposed secure provenance SP scheme based on the bilinear pairings in cloud computing model. This scheme is used provide security and trusted confirmation for data forensics in cloud computing.

X. Song, D.Wagner, A. Perrig [4], provides the proofs of security with the help proposed cryptographic scheme. It supports searching functionality without losing the confidentiality of the data. This technique is secure for encryption as it provides control searching over the data. This system also supports random-access decryption. In this length of every word require to be saved with particular word.

R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky [5], proposed stronger security technique that is **Searchable Symmetric Encryption (SSE).** In this technique user can store data on remote server and can access it privately. In this system user least the data from large dataset, Single-database PIR used to fetch data from a server that involved an unencrypted data.

D. Boneh et al. [6] uses **Identity Based Encryption.** Identity Bases Encryption is a type of a public key encryption. User's public key i.e. set of Identity-String is used for encryption. (e.g. email address, mobile number). The sender will encrypt the data by using identity string and public parameter and sends the data to receiver. By using his secret key receiver will decrypt data. In this approach the size of decryption key is Constant and cipher text size is non-constant. The cost of storing cipher text and transmitting it is expensive. This technique also uses **Chosen-Cipher text Secure Proxy Re-Encryption. Instead of transferring** secret key to the receiver, a useful primitive is proxy re-encryption used. Proxy encryption converts cipher text to original text.

F. Zhao, T. Nishide, K. Sakurai [7], proposed data sharing scheme based on attribute-based cryptosystems is. It is finegrained as well as flexible for cloud storage. This scheme decreases the data leakage from keyword search process also user revocation and key updating can be easily achieved
users to enable the access to their data.

Table 1 shows different types of related work.

Table 1: Related Work

Papers	Related work
Nothing is for Free: Security in Searching Shared and Encrypted Data	Use the bilinear property of Type-3 pairings and its security is based on the bilinear Diffie–Hellman variant
Key Aggregate Cryptosystem For Scalable Data Sharing In Cloud System	Produce constant size cipher text with private key to decrypt.
A Public Key Cryptosystem Based On Number Theory.	Based on numeric data and exploits the features of computationally hard problems.
Storing Shared Data on the Cloud via Security Middleman	Security Middleman generates verification signatures for data owners.
Attribute Union in CP-ABE Algorithm for Large Universal Cryptographic Access Control.	Integrate certain number of attributes into attribute union.

3. PROPOSED SYSTEM

The proposed system with an efficient public-key encryption is shown in Figure 1 and Figure 2. There are two modules, data owner module and data follower module. In this work, any number of subsection of the cipher text can be decrypted by using the decryption key. The problem is solved by the key aggregate cryptosystem.

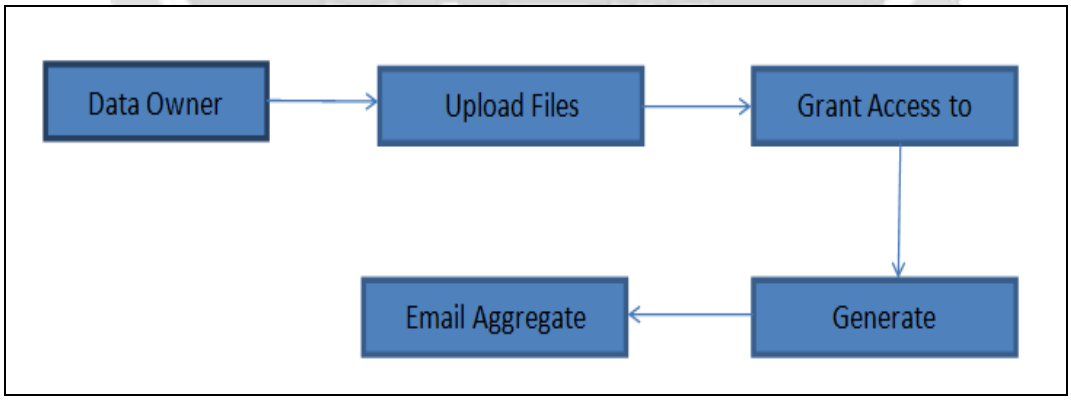

Fig 1 : Data Owner Module

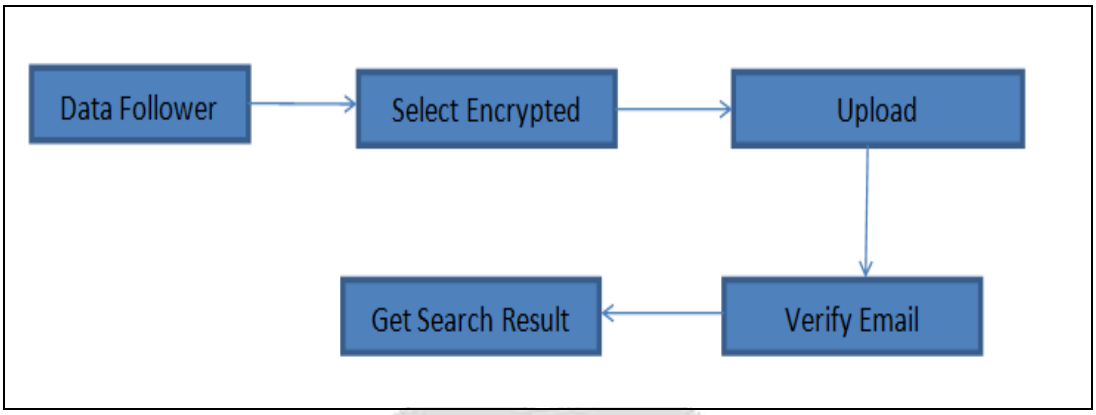

Fig 2: Data Follower Module

Proposed System includes following steps.

3.1 Set up:

The setup algorithm does not accept any input other than the implicit security parameter then it result into public parameters PK and a master key MK.

3.2. Encrypt:

Encrypt (PK, M, A). The encryption algorithm accepts input namely, public parameters PK, a message M, and an access structure A. An algorithm then encrypts message M and produces a cipher text CT such that only a user that have a set of attributes will satisfies the access structure and able to decrypt the message. We will assume that the cipher text implicitly contains A. Important keywords are extracted from document and index in generated.

3.3 . KeyGen:

Key Generation (MK, S) key generation algorithm accepts the master key MK and a set of attributes S as input. After processing it results SK i.e. a private key.

Aggregate Key Generation Algorithm:

a: Data Owner Selects n Files to encrypt using n keys as follows:

File 1 -> key 1
File 2 -> key 2
File 3 -> key 3
File 4 -> key 4 ……
File n ->key n

b: Data Owner allocates who can access which File.

For E.g. User 1 has allocated to access file 1,3,5 then

c: User 1's aggregate key will be generated as follows:

key1 + separator + key3 + separator + key5

convert this string to byte array and this byte array passed to Base64Encoder will give u encoded String. This encoded string converted to number using string to number converter api. This Number is the Aggregate Key to the User 1. Email or send mobile OTP of this aggregate key to User 1 or data follower 1.

3.4. Share Key:

As per permission rights and user mentioned in share list keys are shared to the respective users via secure sharing medium.

3.5. Decrypt:

Decrypt (PK, CT, and SK). The decryption algorithm accepts as input the public parameters PK, a cipher text CT, which contains an access policy A, and a private key S. If S is the set of attributes satises the access structure A then the algorithm will decrypt the cipher text and return a message M.

keys are generated runtime by decryptingsed.

3.6. Email Verification

Before allowing search in allocated files, the system will verify data follower's email by sending random number string in mail content to registered mail id. This random number string will be required before decryption.

3.7. Grant Access and Search Result

After passing all security checks like aggregate key verification and email verification, user has to enter search token to make search in allocated files. Search results in highlighted and graphical format will be displayed to the data follower.

5. IMPROVEMENTS OVER EXISTING SYSTEM

a. Existing system searches data at cloud-side which is insecure but proposed system user can search in allocated data at client-side. This improvement makes proposed system more secure than existing system.
b. No need to store 'n' number of distinct keys for 'n' number of files because of Aggregate key generation.
c. Only allocated bundle will be downloaded at data follower side. Hence it will reduces amount of data downloaded to data follower.
d. Amount of data reduced hence search and download will be more faster.

5. CONCLUSIONS

This system will be secure as encryption technique is involved. Also it is efficient as aggregate key for multiple documents are shared with group of user. Which is not case in existing system Decryption key should be sent via a secure channel and kept secret e.g. email hence data will be secure. This system will be efficient public-key encryption scheme which supports flexible delegation for searching also. Searching over encrypted data is performed efficiently since important public information is retrieved and mapped with the document in encryption format.

6. ACKNOWLEDGEMENT

A very firstly I gladly thanks to my project guide Prof. R.L. Paikrao for their valuable guidance for implementation of proposed system. I will forever remain a thankful for their excellent as well as polite guidance for preparation of this report. Also I would sincerely like to thank to HOD Prof. R.L. Paikrao and other staff for their helpful coordination and support in project work.

7. REFERENCES

[1] Qiang Tang, "Nothing is for Free: Security in Searching Shared and Encrypted Data", IEEE Transaction on Information Forensics and Security, Vol. 9, No. 11,November 2014
[2] S. Yu, C. Wang, K. Ren, and W. Lou, Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing,Proc. IEEE INFOCOM, pp. 534-542, 2010.
[3] R. Lu, X. Lin, X. Liang, and X. Shen, Secure Provenance:The Essential of Bread and Butter of Data Forensics in Cloud Computing, Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010
[4] X. Song, D.Wagner, A. Perrig. Practical techniques for searches on encrypted data, IEEE Symposium on Security and Privacy,IEEE Press, pp. 44C55, 2000.

[5] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions, In: Proceedings of the 13th ACM conference on Computer and Communications Security, ACM Press, pp. 79-88, 2006.
[6] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," SIAM J. Computing, vol. 36, no. 5, pp. 1301-1328, 2007.
[7] F. Zhao, T. Nishide, K. Sakurai. Multi-User Keyword Search Scheme for Secure Data Sharing with Fine-Grained Access Control. Information Security and Cryptology, LNCS, pp. 406-418,2012
[8] Q. Tang, "Search in encrypted data: Theoretical models and practical applications," in Theory and Practice of Cryptography Solutions for
[9] Secure Information Systems. Hershey, PA, USA: IGI, 2013, pp. 84–108. R. A. Popa and N. Zeldovich. (2013). Multi-Key Searchable Encryption. [Online]. Available: http://eprint.iacr.org/2013/508
[10] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. 3rd Int. Conf. Appl. Cryptography Netw. Security, vol. 3531. 2005, pp. 442–455.
[11] M. Chase and S. Kamara, "Structured encryption and controlled disclosure," in Advances in Cryptology—ASIACRYPT (Lecture Notes in Computer Science), vol. 6477, M. Abe, Ed. Berlin, Germany: Springer-Verlag, 2010, pp. 577–594.
[12] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation," in Proc. Netw. Distrib. Syst. Security Symp. (NDSS), 2012, pp. 1–15.
[13] C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," in Proc. 22nd Annu. IFIP WG 11.3 Work. Conf. Data Appl. Security XXII, vol. 5094. 2008, pp. 127–143.
[14] C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," J. Computer Security, vol. 19, no. 3, pp. 367–397, 2011.
[15] F. Kerschbaum and A. Sorniotti, "Searchable encryption for outsourced data analytics," in Proc. 7th Eur. Workshop Public Key Infrastructure., Services Appl., vol. 6711. 2011, pp. 61–76.
[16] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in Proc. IEEE 28th Int. Conf. Data Eng., Apr. 2012, pp. 1156–1167.