Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Secure Search in Encrypted Data Using Aggregate Key Mechanism

Swapnali Vilas Arote[1] And Prof. Rahul L.Paikrao[2]
[1,2]PG Student, Department of Computer Engineering,Assistant Professor, Department of Computer Engineering AVCOE, Sangamner, India

Abstract— Privacy of user's data is a critical question of cloud storage. A 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. These solutions are not secure and scalable for the multiclients. It is a challenge to have a secure and scalable multiparty searchable encryption scheme because of the likelihood that the cloud server might connive with a few malignant clients. Proposed scheme describe public key encryption and efficient delegation. Proposed technique aggregates the set of secrete keys and make a compact single key. 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.

Keywords—Aggregate Key Encryption; Cloud Storage; Data Privacy; Multiparty Searchable Encryption Scheme.

I. INTRODUCTION

Security of data is great concern in today's world because of sharing of data over internet. Data providers make services available to users for searching/accessing purposes. These services may attract more attacks. One of the most important service provided by internet to users is cloud computing. In cloud computing the main issue is the 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 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 s/he can't recognize difference between original data and modified data. The challenge is how to effectively share encrypted data over cloud server.

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 s/he does not want to share his/ her data with anyone else. For security purpose s/he encrypt his/her data before uploading to the server. If another user B ask him/her to share his/her data. User A share his/her data. There are two ways: 1. User A can encrypt data with single secret key and share that secret key directly with the user B. 2. 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.

The best possible solution to above problem is public-key encryption. Instead of using distinct keys for encryption, a single aggregate key is used and it will be send to another user via a

secure e-mail or mobile OTP. User B can now get access to data and again use this aggregate key to decrypt these encrypted data.

II. RELATED WORK

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

The searchable encryption schemes in the **Symmetric Key Encryption** was first introduced by Song et al. [2], Q. Tang,et al.[8], Y.-C. Chang et al. [11], M. Chase[12], C. Dong et al.[14],[15]. These techniques assumes one user and one server, where the user can produces searchable data and stores them at the server, and later on authorize the server to search on her behalf. Data owner needs to produce different key to protect different document and then shares the key with the one who want to search this document. As a result, the number of keys will be equal to the number of documents. Hence it required more storage space and hence can be expensive.

D. Boneh et al. [3] 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.

Goyal et al. 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.

Q. Tang [1] proposed a technique that uses bilinear property of Type 3 pairing and security is based on bilinear Diffie-Hellman variant. This technique searches data at cloud-side which is insecure and also store 'n' number of distinct keys for 'n' number of files which increase space on disk.

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.

III. PROPOSED SYSTEM

The proposed system with an efficient public-key encryption is shown in Figure 1(a) and Figure 1(b). 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.

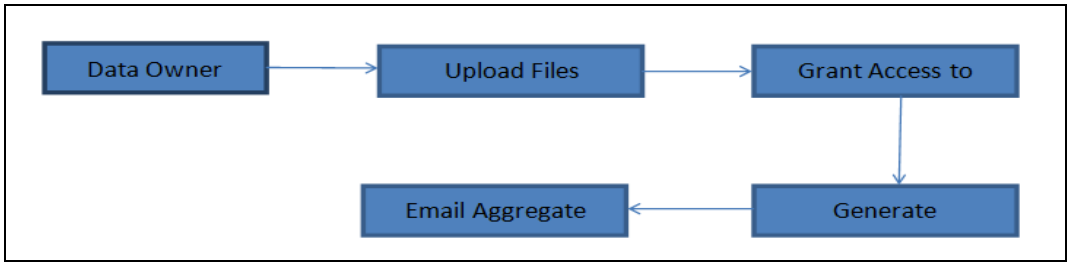

Figure 1(a) Data owner module

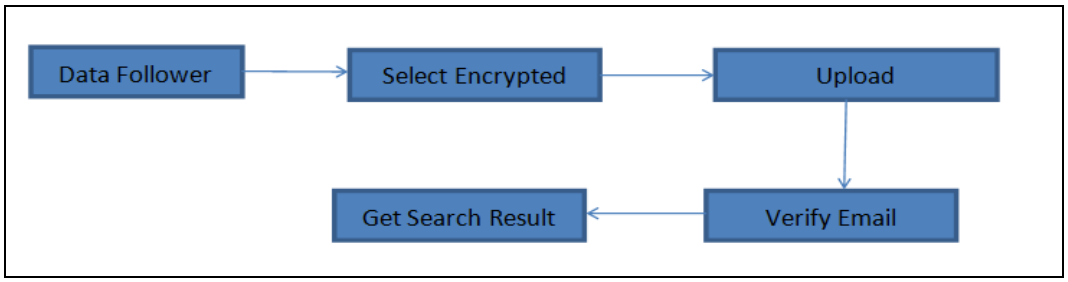

Figure 1(b) Data follower module

Proposed System includes following steps.

3.1. Login and Authenticate

There are two types of login.

a. Data Owner :
   Data owner uploads data and encrypts data. He also authorizes other users to access data.

b. Data Follower
   Data follower receives aggregate key from data owner and by using aggregate key and search keyword, he tries to access/search data.

3.2. User Group Creation

User create groups by sending and accepting friend requests. Like face book or Google plus.

3.3 Upload, Encrypt and Authorize

Data owner login allows user to upload files on server. Then user encrypts each and every file using separate key and then authorize users who want to access that files. Different aggregate keys will be generated using aggregate key generation algorithm for different users who want to access certain files. These aggregate keys will be emailed to data followers or mobile OTP will be sent to data followers.

3.4. Aggregate Key Generation and Email

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.

3.5. Aggregate Key Validation:

Using Data Followers Login user can access allocated file and make search in these files using following steps :

a. Download aggregate key from mail.
b. Select encrypted file bundle which contains allocated files.
c. Type aggregate key.
d. Aggregate key validation contains following steps :

keys are generated runtime by decrypting aggregate key and removing separator used.

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.

IV. 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.

V. CONCLUSION AND FUTURE WORK

In this paper, we reviewed different techniques on searching in encrypted data. Existing system based on Attribute Based Encryption and Multi Identity Single Key Decryption. Proposed system work by using any number of subsection of the cipher text that can be decrypted using the decryption key. Proposed scheme describe public key encryption and efficient delegation. Proposed technique aggregates the set of secrete keys and make a compact single key. 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. Public-key cryptography offers allocation of secret keys for dissimilar cipher text classes in cloud storage.

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] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Symp. Security Privacy, May 2000, pp. 44–55.

[3] 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.

[4] http://www.careerride.com/Networking-symmetric-cryptography-key.aspx

[5] B. Hore, S. Mehrotra, M. Canim, and M. Kantarcioglu, "Secure multidimensional range queries over outsourced data," VLDB J., vol. 21, no. 3, pp. 333–358, 2012.

[6] E. J. Goh, "Secure indexes," IACR, Tech. Rep. 216, 2003, [Online]. Available: http://www.iacr.org.

[7] C. Bosch, Q. Tang, P. Hartel, and W. Jonker, "Selective document retrieval from encrypted database," in Proc. 15th Inf. Security Conf. (ISC), vol. 7483. 2012, pp. 224–241.

[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.