Impact Factor (SJIF): 5.302

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 5, Issue 3, March-2018

Key Aggregate Tagged File Searching(KATFS)

Trupti Tanpure[1], Shamal Tavandkar[2], Monika Tope[3], Dipali Wadekar[4]

Prof: V.N.Dhage[5]

[1]Computer Department, P K Technical Campus, Chakan

[2]Computer Department, P K Technical Campus, Chakan

[3]Computer Department, P K Technical Campus, Chakan

[4]Computer Department, P K Technical Campus, Chakan

[5] Assistant Professor, Computer P K Technical Campus, Chakan

Abstract :

The important functionality in cloud is a sharing .To address user concerns over potential data leaks in cloud storage a common approach is for the data owner to encrypt all data before uploading them to the cloud, such that later the encrypted data may be retrieved and decrypted those who have the decryption keys. A key challenge to designing such encryption schemes lies in the sufficient management of encryption keys. This also implies the necessity of securely distributing to users a large number of keys for both encryption in search and user will have to securely store the received key and submit an same large number of keywords trapdoors to the cloud in order to perform search over the shared data. The practical problem of privacy preserving data sharing system based on public cloud storage which requires a data owner to distribute a large number of keys to users to enable them to access his/her documents. By addressing this practical problem which is hugely neglected in literature, we propose the novel concept of key aggregate Tagged File Searching (KATFS) in which data owner only need to distribute a single to user for sharing large number of data and user only needs to submit a single trapdoor to the cloud for querying the shared a large number of documents.The ability of selectively distributing unreadable data with many users via public cloud storage may greatly ease security concerns over in advertent data leaks in the cloud. To designing the encryption schemes lies in the efficient management of encryption keys.The desired flexibility of sharing any group of selected documents with any group of users demands different encryption keys to be used for different data .The necessity of safely distributing to users a large number of keys for both encryption and search, and those users will have to securely store the received keys, and submit an equally huge number of keyword trapdoors to the cloud in order to perform search over the shared data. The required for secure communication, storage, and complexity clearly renders the approach impractical. The concept of Key Aggregate Tagged File Searching (KATFS) and instantiating the concept through a concrete KATFS scheme, in which a data owner only needs to distribute a single key to a user for sharing a large number of documents, and the user only needs to submit a single trapdoor to the cloud for querying the shared documents. The security analysis and performance evaluation both confirm that our proposed schemes are provably secure and practically efficient.

Keywords-

Encryption, big-data, data sharing, cloud storage, Data Privacy, Tagging

I.INTRODUCTION

Cloud systems can be used to enable document distributing abilities and this can provide an abundant of benefits to the user. There is currently a push for IT organizations to increase their data sharing efforts. In enterprise settings, demand for data outsourcing is increased today. Data outsourcing should be assists in the strategic management of corporate data. This scheme is also used as a core technology behind many online services. These online services used for online application. Currently this scheme was easy to apply for free accounts for mail, photograph album, sharing of file with storage size more than 25GB. Together by using the present wireless technology, cloud users can access almost all of their files, directories and emails by a mobile phone in any corner of the world. Some of major requirements of secure data sharing in the Cloud are as follows. Firstly the data owner should be able to specify a group of users that are allowed to view his or her data.
Any member within the group should be able to gain access to the data anytime, anywhere without the data owner's intervention. No-one, other than the data owner and the members of the group, should gain access to the data, including the Cloud Service Provider. The data owner should be able to add new users to the group. The data owner should also be able to revoke access rights against any member of the group over his or her shared data No member of the group should be allowed to revoke rights or join new users to the group. One trivial solution to achieving secure data sharing in the Cloud is for the data owner to encrypt his data before storing into the Cloud, and hence the data remain information-theoretically secure against the Cloud provider and other malicious users. When the data owner wants to share his data to a group, he sends the key used for data encryption to each member of the group. Any member of the group can then get the encrypted data from the Cloud and decrypt the data using the key and hence does not require the intervention of the data owner. However, the problem with this technique is that it is computationally inefficient and places too much burden on the data owner when considering factors such as user revocation.

II. RELATED WORK

Considering the practical problem of security preserving information system based on public cloud storage which requires a data owner to distribute a large number of keys to users to enable them to access his/her documents, we for the first time propose the concept of Key Aggregated Tagged Files Searching (KATFS) and construct a concrete KATFS scheme. Both analysis and evaluation conclusion confirm that our work can give an effective solution to building practical data distributed system based on public cloud storage. In a KATFS scheme, the owner only needs to distribute a single key to a user when sharing lots of documents with the user, and the user only needs to submit a only one trapdoor when he queries overall file shared by the equal owner. However, if a user wants to query over documents shared by multiple owners, he must generate multiple trapdoors to the cloud. How to reduce the number of trapdoors under multi owners setting is a future work. Moreover, federated clouds have attracted a lot of attention now a days, but our KATFS cannot be applied in this case directly.

III. PROPOSED WORK

1. Primary user:

This is the person or user who have all permission to uploads information on the cloud. Data having images, files or documents etc. This user upload files on cloud by Tagging File. The tag includes the user that can access the file. Cause of this tagging, only those data can be downloaded by other user which he/she is permitted to download (tagged in that document). This user generate trapdoor which having information about those persons whose data is present on cloud. So the person can easily download their data by requesting to cloud. Trapdoor having information about secondary user i.e. users id or name. Primary user save information on cloud by using RSA algorithm.

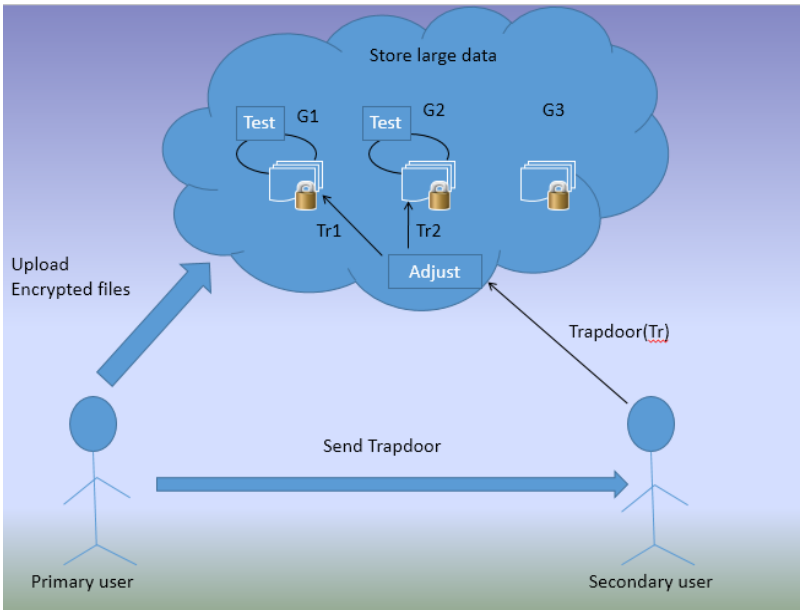

Fig: System Architecture

2. Cloud:

Cloud is used to store large amount of data. This data can be anything like images, files, documents etc. Data is store on cloud by primary user. Cloud provides privacy, so it is not dangerous for uploading big amount of data on it. By using trapdoor cloud can easily search out that data in which that secondary user is present. When cloud get request from secondary user for accessing or downloading the data, that time cloud checks trapdoor, which having secondary users information.

3. Secondary user:

Here secondary user is that person who wants information present on cloud. Here secondary user is that person who receive data present on cloud. This user get trapdoor from primary user. Secondary user send trapdoor to cloud. Cloud firstly adjusting that trapdoor then check it. Because of this trapdoor cloud receive information that who is this user and what he wants. So after testing trapdoor, cloud provide that user only those data in which he/she is tagged. A scenario where two worker of a company would like to distributed some other confidential.

IV. METHODOLOGY

1. Login & Registration Module:
Login & Registration Module is divided into two parts
A) Sender side login & Registration
B) Receiver Side login & Registration

2. Authentication Module:
Authentication is done at sender side by public key and at the receiver side Authentication is done by private key of each receiver.

3. File Uploading:
File Uploading is done by Sender to the cloud and encrypting it while Uploading.

4. Request Download file:
At receiver side the trapdoor is accessed and the request for list of file is generated from receiver to cloud. After receiving the request cloud responses to the user with the list of file.

5. File Download:
After receiving the list of file user can download the files .

Algorithms used in this system :
1 .RSA Algorithm
2 .SHA Algorithm

ALGORITHAM: RIVEST SHAMIR ADALEMAN (RSA)

Steps used for Key Generation:

1. Select to prime large numbers. Let's call them p and q such that p!= q(at least 512 bits each)

2. Compute their system modulus N= p*q Where f(N) = (p - 1)(q - 1)

3. Compute our n to n = p * q.

4. Select a small, odd integer (e) that is relatively prime to f(N) and not 1 Where 1 < e < f(N), gcd(e,f(N))1

5. Compute d to be the multiplicative inverse of emodulo f(N) Where e.d = 1 mod f(N) and 0 = d = N

6. The order pair (e,n) is our RSA public - key(Encryption key). Publishth is key.

7. The order pair(d,n)is our RSA private - key(Decryption key).Keep secrete this key.

8. Make sure that 'p'and 'q' are completely annihilated. Otherwise the security of our cryptosystem may be comprised.

A .Steps used for RSA Encryption:

For encrypting a message M with RSA , the sender first obtain the Public key of the recipient i.e. KU e,N. once the sender has the public key of the recipient, he computes the encrypted message as follows
C=M mod N

Where 0= M = N

Thus C is the Encrypted message which is sent to the recipient over public network. No one other person except the original recipient can decrypt the encrypted message C to get the original message M.

B. Steps used for RSA Decryption :

when the recipient receives the encrypted message C, he/she can decrypts the message with he/she Private Key . He/she can decrypt it only if the message was encrypted with he/she Public Key. Thus the recipient uses his/her private key KR d,N to get the original message M in the following way.

M =C d mod N

Where 0= M = N

The message M must be smaller than the modulus N

Explanations of Keyword used in RSA Algorithm:

p and q= two prime numbers used for key generation.

n = Modules for both public and private key.

f(N)= Euler's totient function. this value is kept Private.

d = Modular Multiplicative Inverse. Kept as the public key Exponent.

e = Released as public key Exponent.

M = Encrypting a message.

C = Encrypted message.

ALGORITHAM : SECURE HASHING ALGORITHM (SHA 2.0)

1. Append Padding Bits

Message is padded with a 1 and as many 0s as necessary to bring the message length to 64 bits fewer than an even multiple of 512.

2. Append Length

64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating the length of the original message.

3. Prepare Processing Functions

4. Prepare Processing Constants

5. Initialize Buffers
SHA1 requires 160 bits or 5 buffers of words (32 bits)

6. Processing Message in 512-bit blocks (L blocks in total message)
This is the main task of SHA algorithm which loops through the padded and appended message in 512-bit blocks.

V. FUTURE WORK

With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this article, we consider how to compress secret keys in public-key cryptosystems which support delegation of secret keys for different cipher. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. In cloud storage, the number of cipher texts usually grows rapidly. So we have to reserve enough cipher text classes for the future extension. Otherwise, we need to expand the public-key. Although the parameter can be downloaded with cipher texts it would be better if its size is independent of the maximum number of cipher text classes. On the other hand , when one carries the delegated keys around in a mobile device without using special hardware, the key is prompt to leakage, designing a leakage cryptosystem yet allows efficient and flexible key delegation is also an
interesting direction.

VI. CONCLUSION

Considering the practical problem of privacy- preserving data sharing system based on public cloud storage which requires a data owner to distribute a large number of keys to users to enable them to access his/her documents, we for the first time propose the concept of key-aggregate searchable encryption (KASE) and construct a concrete KASE scheme. Both analysis and evaluation results confirm that our work can provide an effective solution to building practical data sharing system based on public cloud storage. In a KASE scheme, the owner only needs to distribute a single key to a user when sharing lots of documents with the user and the user only needs to submit a single trapdoor when he queries over all documents shared by the same owner.

VII. REFERENCES

1. Kallahalla, E. Riedel, R. Swaminathan, Q.Wang, and K. Fu, Plutus: Scalable Secure document sharing on Unauthorized Storage, Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

2. E. Goh, H. Shacham, N. Modadugu, and D. Boneh, Sirius: Securing Remote Untrusted Storage, Proc. Network and Distributed Systems Secu- ritySymp.(NDSS), pp.131-145, 2003.

3. D. Naor, M. Naor, and J.B. Lotspiech, Revocation and Tracing Schemes for Stateless Receivers, Proc. Ann. Intl Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-62, 2001

4. G. Ateniese, K. Fu, M. Green, and S. Hohenberger, Improved Proxy Re- Encryption Schemes with App lications to Secure Distributed Storage, Proc. Network and Distributed Systems Security Symp. (NDSS), pp.29-43, 2005

5. S. Yu, C. Wang, K. Ren, and W. Lou, acquring Secure, Scalable, and Fine- Grained file Access maintain in Cloud Computing, Proc. IEEE IN- FOCOM, pp. 534-542, 2010.

6. In detail. [6] proposed a secure provenance scheme, which is built upon group signatures and cipher text-policy attribute- based encryption techniques. Particularly, the system in their scheme is set with a single attribute. Each user obtains two keys after the registration: a group signature key and an attribute key.Thus, any user is able to encrypt a data file using attribute- based encryption and others in the group can decrypt the encrypted data using their attribute keys. Meanwhile, the user signs encrypted data with her group signature key for privacy preserving and traceability. However, user revocation is not supported in their scheme Key aggregated Tagged File Searching.

7. In Xuefeng Liu, Yuqing Zhang [7] Mona: Secure Multi Ownership Data Sharing for Dynamic Groups in the Cloud In this paper, they propose a secure multi owner document sharing scheme, named Mona, for dynamic groups in the cloud. By leveraging group signature and dynamic broadcast encryption techniques, any cloud user can anonymously share data with others. Meanwhile, the storage overhead and encryption computation price of our scheme are does not dependent with the number of revoked users.

8. In Cong Wang KuiRen[8] Privacy Preserving Public Auditing for Secure Cloud Storage enabling public audit ability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced file and be worry without cost. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user. In this paper, a secure cloud storage system supporting security preserving public auditing. They further extend result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient