Key-Aggregate Searchable Encryption (KASE) Framework Using Single Aggregate Key

Aishwarya Chavan[1], G.T.Chavan[2]

[1]Student, Department of Computer Engineering, SCOE, Vadgaon, Pune, Maharashtra
[2]Professor, Department of Computer Engineering, SCOE, Vadgaon, Pune, Maharashtra

---

Abstract- *Large amount of data can be stored on cloud but it's an untrusted. There is need for secure storage of confidential data.*

In this paper we defined a system that can manages multiple users and their access specifications. User will get access to documents Instead of generating multiple keys for multiple users, a single key having multiple authorization codes will be used for encryption/decryption purpose. When user search the document using keywords (words) then user request for the searching document. That time he/she sends search index according to the cloud server. Once the data is available in particular document then that document will be decrypt using AES algorithm. The AES Key can also be encrypt and decrypted by the user using his public and private key with RSA algorithm.

Keywords: **Aggregate Key, Trapdoor Key, Cloud Storage, AES, RSA.**

1. INTRODUCTION

Cloud computing assures to elastically store and process large amount of data. The storage and dealing with data on Cloud facilitate the aggregation and analyze the data from dissimilar data sources. Alternatively, data accumulated on cloud frequently contains confidential and sensitive information. In present Cloud platforms, the data owner is unable to find control on the data once it goes through the Cloud. So the designing of cloud is such that owner can trust to handle her confidential data securely. Some of the security architecture [11] gives end-to-end data access control by the data owner. Security architecture is a promising extension of today's Cloud offers. In recent years number of clients store number of records on distributed storage. This sharing of encrypted data is preferred in public cloud storage; it also straightforward practice of file distribution in cloud. When a file is to be shared with group of people, lot of keys need to be generated and shared with users. A key challenge to designing such encryption schemes lies in the efficient management of encryption keys. The flexibility of sharing any important document with many user by providing different encryption keys for different documents. i.e for each document separate encryption key is provided. still, it gives firmly share large number of keys for both encryption and search, and those users will have to store the received keys, securely and user have to put forward same number of trapdoors key to the cloud in order to carry out search operation on shared data. This paper gives (KASE) key aggregate searchable encryption as an enhanced way to allocate a single aggregate key.

2. RELATED WORK

Searchable symmetric encryption (SSE) [1][3] The more literature is available on SSE. It gives permission to the user for outsourcing the stored data to another party in a privately. If another random group of parties except the owner submit search query then it expand the ability of searching to that parties.

The paper [4] gives concept of public key encryption with keyword search (PEKS) schemes in which the use of key (public key) used by anyone to encrypt messages through the keyword search is to generate a PEKS that corresponds to Identity Based Encryption (IBE). It enables the server to identify all messages containing some specific keyword. Without learning anything else about the document. A lot of work is available on PEKS. [8][10].

Chosen ciphertexts attacks (CCA) [5] is a fundamental notion of security for any encryption and broadcast encryption scheme. It gives cipher texts and fix-size secret keys. CCA are safe proposals with a transmit encryption with fix-size cipher texts and they prove to be secure under assumptions that are realistic generalizations of earlier assumptions under failed CCA security.

Cloud computing offer an reasonable and proficient resolution for distribution group resource among cloud users. Because of the common change in sharing data [6] while storing the data and identify privacy from untrusted cloud is still a difficult issue.

Multi key searchable encryption (MKSE) [7] gives a single keyword trapdoor to the server, and let the server to look for that trapdoor's keyword in the data encrypted with different keys. KASE allows distributing the aggregate key to user in a group data sharing system while the aim of MKSE is to ensure the cloud server can perform keyword search with one trapdoor over different documents due to a user.

3. PROPOSED SYSTEM

This given KASE system helps for group data sharing functionality, such that any user can contribute a group of selected files with a group of selected users, while consent to execute keyword search over the previous information. The data owner only have to share a single key to a user for allocation of a large number of documents, and the user simply required to suggest a single trapdoor to the cloud for querying the shared documents.

Users of multiple departments will be storing data on public cloud. Each document will be stored on cloud server in encrypted format. Each document is encrypted using a private key of the user which can only be decrypted when user gives permission. Tokens are generated for each document and stored in database for later searching. Tokens are encrypted using server's symmetric AES key. Instead of generating multiple keys for multiple users, a single key having multiple authorization codes will be used for decryption purpose. Each user will have to sign in to our system; its role and department will be extracted from login information. Department and role will be used for extracting user's trapdoor key from the aggregate key. Once the trapdoor key is extracted, it will used for searching and locating the user document.

A. Aggregate key generation process

Aggregate key generation is a 7 step process

Parameter Setup –The cloud service provider setup the system by using this step.

Key Generation – the algorithm is run by the data owner to generate a random key pair

Encryption – Data owner generate an aggregate key by owner's public key and file index i

Key Extraction –owner's master secret key MSK is takes as a input, and a set S which contains the indices of documents, then outputs the aggregate key Kagg.

5. Trapdoor Generation- this is used by the end user with whom the document is shard. It takes input as aggregate key and document index, and generates a trapdoor key which is used for document searching.

Trapdoor adjustment- this is performed by the cloud service provided to generate the right key for document decryption using the end user's trapdoor key.

Trapdoor Testing – this algorithm checks if the given word exists in the file and revert true or false. In our proposed system we will be using it to search and download the file.

Elliptic curve cryptography is used to generate public key and private key pairs for each user. It's the most powerful key exchange algorithm available till today. Multiple user's and their access specifications. Employee will get access to documents by their department.

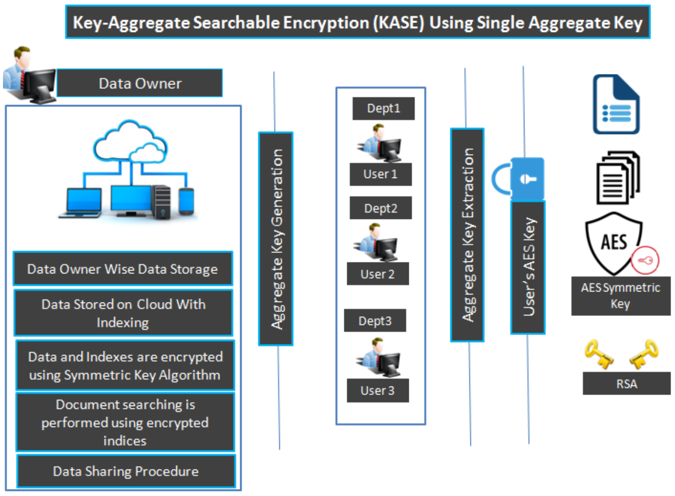
Key-Aggregate Searchable Encryption (KASE) Using Single Aggregate Key
Data Owner
Data Owner Wise Data Storage
Data Stored on Cloud With Indexing
Data and Indexes are encrypted using Symmetric Key Algorithm
Document searching is performed using encrypted indices
Data Sharing Procedure
Aggregate Key Generation
Dept1
User 1
Dept2
User 2
Dept3
User 3
Aggregate Key Extraction
User's AES Key
AES
AES Symmetric Key
RSA

Fig-1: Diagram for Proposed System

4. ALGORITHM USED

A. AES Algorithm Generate single keys

a. Encryption

You take the following AES steps of encryption for a 128-bit block:

1. Derive the set of round keys from the cipher key.
2. Initialize the state array with the block data (plaintext).
3. include the primary round key to the early state array.
4. Perform nine rounds of state manipulation.
5. Perform the tenth and final round of state manipulation.
6. Copy the final state array out as the encrypted data (cipher text).

Each round of the encryption process requires a series of steps to alter the state array.

These steps occupy four kind of action called:

1. Sub-Bytes
2. Shift-Rows
3. Mix-Columns
4. Xor-Round Key

b. Decryption

As one might be expecting, decryption engross turn around all the in used steps in encryption using inverse functions:

1. InvSub-Bytes
2. InvShift-Rows
3. InvMix-Columns

Operation in decryption is:

1. Perform initial decryption round:
    - Xor-Round Key
    - InvShift-Rows
    - InvSub-Bytes
2. Perform nine full decryption rounds:

    Xor-Round Key
    InvMix-Columns
    InvShift-Rows
    InvSub-Bytes
3. Perform final Xor-Round Key

c. RSA Algorithm

The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers p and q.
    - For security purposes, the integers p and q should be chosen at random, and should be similar in magnitude but 'differ in length by a few digits to make factoring harder. Prime integers can be efficiently found using a primality test.
2. Compute n = pq.
    - n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
3. Compute φ(n) = φ(p)φ(q) = (p − 1)(q − 1) = n − (p + q − 1), where φ is Euler's totient function. This value is kept private.
4. Choose an integer e such that 1 < e < φ(n) and gcd(e, φ(n)) = 1; i.e., e and φ(n) are coprime.
5. Determine d as d ≡ e−1 (mod φ(n)); i.e., d is the modular multiplicative inverse of e (modulo φ(n)).

- This is more clearly stated as: solve for d given d⋅e ≡ 1 (mod φ(n)).

- e having a short bit-length and small Hamming weight results in more efficient encryption – most commonly 216 + 1 = 65,537. However, much smaller values of e (such as 3) have been shown to be less secure in some settings.

- e is released as the public key exponent.

- d is kept as the private key exponent.

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and φ(n) must also be kept secret because they can be used to calculate d.

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and φ(n) must also be kept secret because they can be used to calculate d.

5. CONCLUSIONS

The proposed system provide robust security of personal data and providing single key Instead of generating multiple keys for multiple users, a single key having multiple authorization code will be used for decryption purpose. Users have to store the received keys firmly, and give a uniformly large number of keyword trapdoors to execute search operation over the shared data on cloud.

The document stored on cloud is encrypted using a private key of the user which can only be decrypted when user gives permission. Tokens are generated for each document and stored in a system for later searching. Tokens are encrypted using a servers symmetric AES key.

REFERENCES

[1] Searchable symmetric encryption: Improved definitions and efficient constructions",Reza Curtmola,Journal of Computer Security 19 (2011) 895–934 895DOI 10.3233/JCS-2011-0426 IOS Press.

[2] S. Kamara, C. Papamanthou, T. Roeder. “Dynamic searchable symmetric encryption”, Proceedings of the 2012 ACM conference on Computer and communications security (CCS), ACM, pp. 965-976, 2012.

[3] P. Van,S. Sedghi, JM. Doumen. “Computationally efficient searchable symmetric encryption”, Secure Data Management, pp.87-100, 2010.

[4] Public Key Encryption with keyword Search",Dan Boneh,Giovanni Di Crescenzo.

[5] Duong-Hieu Phan,David Pointcheval,”Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts”’Int. J. Inf. Secure.(2013) 12:251265,DOI 10.1007/s10207-013-0190-0.

[6] X. Liu, Y. Zhang, B. Wang, and J. Yan. “Mona: secure multiowner data sharing for dynamic groups in the

cloud", IEEE Transactions on Parallel and Distributed Systems, 2013.

[7] R. A. Popa ,N. Zeldovich. "Multi-key searchable encryption".Cryptology ePrint Archive, Report 2013/508, 2013., 24(6): 1182-1191.

[8] C. Bosch, R. Brinkma, P. Hartel. "Conjunctive wildcard search over encrypted data", Secure Data Management. LNCS, pp. 114-127, 2011.

[9] J. W. Li, J. Li, X. F. Chen, et al. "Efficient Keyword Search over Encrypted Data with Fine -Grained Access Control in Hybrid Cloud", In: Network and System Security 2012, LNCS, pp. 490-502, 2012

[10] J. Li, Q. Wang, C. Wang. "Fuzzy keyword search over encrypted data in cloud computing", Proc. IEEE INFOCOM, pp. 1-5,2010

[11] René Hummen, Martin "A Cloud design for user-controlled storage and processing of sensor data", , Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference .