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

Aishwarya Chavan[1] G. T. Chavan[2]
[1]Student [2]Professor
[1,2]Department of Computer Engineering
[1,2]SCOE Vadgaon, Pune, Maharashtra, India

Abstract— With the advancement of information technologies particularly cloud storage is used for outsourcing data. Now a day's users store a large amount of information on the cloud but it's untrusted and we store secure data on the cloud. To conquer this downside we suggest the idea of searchable encryption that gives promising way in resolving the privacy problem at the time of storing data to the cloud. This method consent to store the information in encryption format on an untrusted server, and then hand over the server to search on their behalf by the private key and encrypted search indices. When a file is to be shared with a group of people, lot of keys need to be generated and shared with users. A main confront to scheming such encryption method lies in the well-organized encryption key management. Our System is an advancement of key aggregate searchable encryption (KASE), where a data holder only wants to share out a single key to multiple users for allocation of the huge documents, and the user only has to offer a single trapdoor to the cloud for getting the shared documents. The safety analysis and performance estimation confirm that the given method is probably safe and practically capable.

Key words: Key Aggregate Searchable Encryption (KASE), Untrusted Server, File Sharing, Security Analysis, Encryption Keys, Trapdoor Generation

I. Introduction

Cloud computing is extremely admired in organizations and institutions because it provides storage and computing services. However, it also introduces new challenges for ensuring the privacy, reliability and access control of the data. Many methods exist to make sure these safety requirements but they are not satisfactory in some ways for example violation of data privacy because of involvement attack and serious computation.

Security architecture is a promising extension of today's Cloud offers. When a file is to be shared with the group of people, the lot of keys need to be generated and shared with users.

Chu et al. [5] give the reduction in large spread data encryption keys. To contribute numerous credentials using various encryption keys by the same user, the data holder will be required to distribute the different such keys to all the users. This conventional approach is generally unrealistic.

There is a rich literature available on searchable encryption, including SSE schemes [6] and PEKS system [7] [8] in which the use of key (public key) used by anyone to encrypt messages through the keyword search is to create a PEKS that corresponds to Identity-Based Encryption (IBE)[9]. It enables the server to recognize the entire messages having some definite keyword.

A general method of keyword search is performed by different users and that name as the multi-user setting. In this Multi-User Searchable Encryption (MUSE) [2][3] the data holder share a file with different authorized users and every user having the right of accessing the data can make searching over the document with trapdoor method.

The attribute-based encryption (ABE) [12] is used to attain fine-grained access control aware keyword search.

Multi-key searchable encryption (MKSE) [11] 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 the sharing of the aggregate key to the user in a group data sharing system while the aim of MKSE is to make sure that the cloud server can carry out the keyword search with one trapdoor over different documents due to a user.

Scalable multiparty searchable encryption (MPSE)[10] scheme that takes different server-user involvement probability by engaging the bilinear property of Type-3 pairings and demonstrate its sanctuary relayed on the bilinear and symmetric exterior Diffie Hellman inference in the arbitrary oracle model.

II. Proposed System

A. *Store Documents on Cloud*

Users of multiple departments will be storing data on the public cloud. Each document will be stored on the cloud server in encrypted format. Each document is encrypted by a private key of the user which is only decrypted when the user gives permission. Tokens are generated for each document and stored in the database for later searching. Tokens are encrypted using server's symmetric AES key.

B. *Key Sharing between User Groups*

Instead of generating multiple keys for several 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 the 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 be used for searching and be locating the user document.

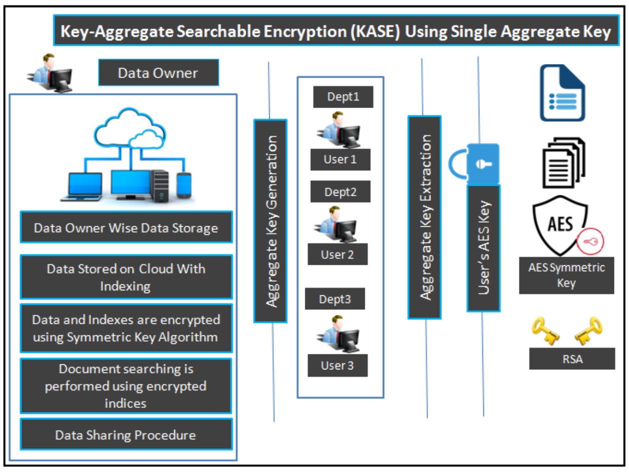
Key-Aggregate Searchable Encryption (KASE) Using Single Aggregate Key
Data Owner
Data Owner Wise Data Storage
Data Stored on Cloud With Indexing
Data and Indexes are encrypted using Symmetric Key Algorithm
Document searching is performed using encrypted indices
Data Sharing Procedure
Aggregate Key Generation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

Fig. 1: System Architecture

C. *Aggregate Key Generation Process*

Aggregate key generation is a 7 step process

Aggregate key generation is a 7 step process

1) Parameter Setup
   The algorithm is executed by cloud service provider to set up the format.
2) Key Generation
   The algorithm is run by the data holder to create a random key pair.
3) Encryption
   It carries out by the data holder to create an aggregate key using owner's public key and file index i
4) Key Extraction
   It gets an input the holder's master secret key MSK, and a set S which include 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 the shard. The input as aggregate key and document index, and creates a trapdoor key which is used for document searching.
6) 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.
7) Trapdoor Testing
   It checks if the certain word is present in the file and returns true or false. In our proposed system we will be using it to search and download the file.

III. Proposed Method

A. *AES Encryption*

To generate single key:

The key size of AES is in general 128 bits. Whereas 256 bits and 512 bits keys are also possible to use.

1) Encryption

1) The AES steps of encryption for a 128-bit block is as follows:
2) Take the list of round keys from the cipher key.
3) Initialize the status array with the plaintext.
4) Include the primary round key to the early state array.
5) Carry out nine rounds of state management.
6) Carry out the tenth and last round of state handling.
7) Copy the ending status array out as the encrypted data.

Every cycle of the encryption procedure need a series of steps to change the state array.

These steps occupy four kind of action called:

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

2) Decryption

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

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

Process in decryption is:

1) Carry out 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

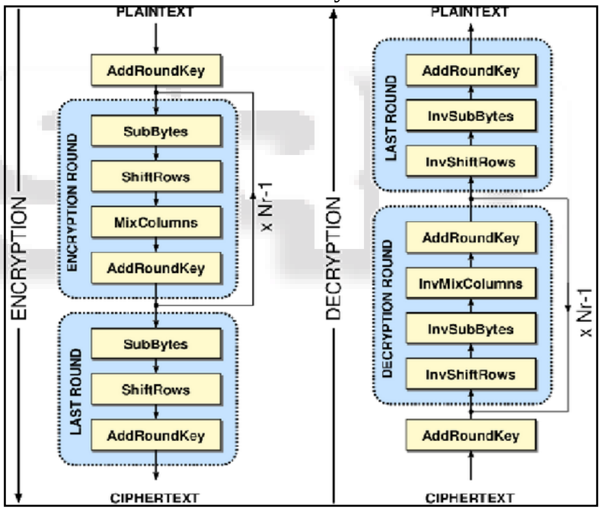

Fig. 2: AES Algorithm [8]

3) RSA Algorithm

The RSA algorithm are created by allowing the below steps:

1) Select two distinct prime numbers p and q.
   – For safety the integer's p and q should be unsystematically selected at, and must be identical in extent but 'differ in length by a few digits to make factoring harder.
2) Compute n = pq.
   – n is used as the modulus for the public and private keys. The length, generally given in bits, is the key length.
3) Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1)$, where φ is Euler's totient function its value and is kept confidential.
4) Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and φ(n) are coprime.
5) Determine d as $d \equiv e^{-1} \pmod{\varphi(n)}$; i.e., d is the modular multiplicative inverse of e (modulo φ(n)).

- This is obviously given as: solve for d given d·e ≡ 1 (mod φ (n)).
- e having a short bit-length and small Hamming weight results in more proficient encryption – most commonly 216 + 1 = 65,537. conversely, lesser values of e have to be less protected in some cases.
- e is released as the public key model.
- d is set as the private key exponent.
- The public key consists of the modulus n and the public exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, that have to 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 exponent d that have to be secret. p, q, and φ(n) kept private as they can be used to calculate d.

IV. Experimental Results

The application uses the Ever data Cloud Server for public hosting. The system's GUI was designed in HTML. Core Technologies used were Ajax, JQUERY, JSP, etc. The overall development was done in the Eclipse IDE and for DB we used MY SQL GUI browser. The database basically used for user storing user details like Registration, encrypted documents.

The time required for key generation of RSA algorithm with different key sizes is shown in below table.

Algorithm Name Key Size (bits)	Generation	Encryption	Decryption
RSA 512	36.363823	0.089972	0.980457
RSA 1024	156.672739	0.26239	5.282019
RSA 1024	128.24768	0.263075	5.23173
RSA 2048	822.579944	0.922985	33.650631
RSA 2048	1364.251272	0.927433	33.914047
RSA 2048	1241.695787	0.92709	33.765234

Table 1: RSA Key Generation with Different Key Sizes

The key generation process is inured to create the keys that are utilized by the signing method and the validation procedure. Every time it creates a key pair containing Private Key and the equivalent Public/confirmation key. Below is the plot of time for the key generation with different sizes.

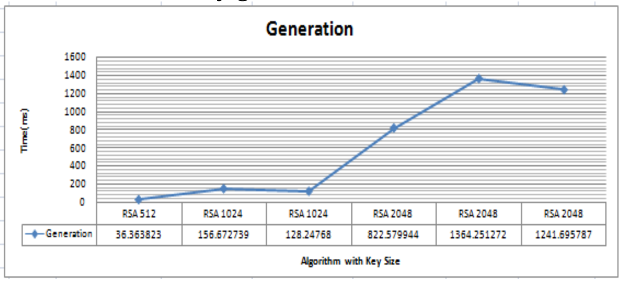

Fig. 3: Key Generation

The operation time required for RSA algorithm to encryption and decryption is like given in figure 4.

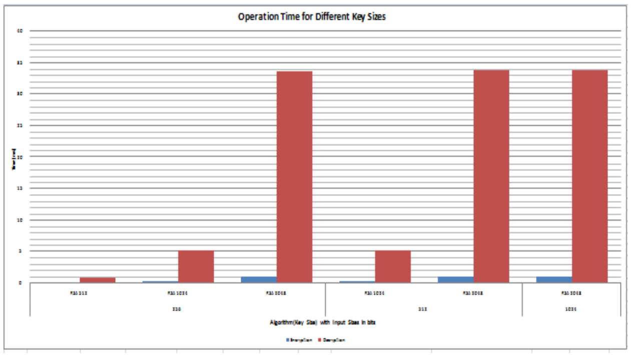

Fig. 4: Key Generation

V. Conclusion

The system will manage multiple users and their access specification by providing the single key instead of multiple keys for individual sharing. Multiple authorization codes will be used for decryption purpose. This system provides the vigorous security to the private data and it gives only access to the authenticated users.

References

[1] Baojiang Cui,Zheli Liu,Lingyu Wang, "Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage",IEEE Transactions on Computers (IEEE T COMPUT).
[2] 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.Transactions on Knowledge and Data Engineering, vol. 27, no. 8, pp. 2107–2119, Aug 2015.
[3] Z. Liu, Z. Wang, X. Cheng, et al. “Multi-user Searchable Encryption with Coarser-Grained Access Control in Hybrid Cloud”, Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, pp. 249-255, 2013.
[4] 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..
[5] C. Chu, S. Chow, W. Tzeng, et al. “Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage”, IEEE Transactions on Parallel and Distributed
[6] 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.
[7] 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 .
[8] Public Key Encryption with keyword Search",Dan Boneh,Giovanni Di Crescenzo..
[9] 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.

[10] Qiang Tang,"Nothing is for Free: Security in Searching Shared and Encrypted Data",IEEE TRANSACTIONS ON INFORMATION FORENSIC AND SECURITY, VOL. 9, NO. 11, NOVEMBER 2014.

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

[12] C. Dong, G. Russello, N. Dulay. "Shared and searchable encrypted data for untrusted servers", Journal of Computer Security, pp. 367-397, 2011.