STM JOURNALS

International Journal of
Data Structure Studies

Volume 2, Issue 1, 2024
January—June
DOI (Journal): 10.37591/IJDSS

https://journals.stmjournals.com/ijdss

Review IJETA

Searchable Encryption Based on Key Aggregation

Nesha Kavya Urs[1], Lekha Achuth[2,*]

Abstract

The ability to selectively distribute data with individuals in public cloud services may alleviate safety concerns about inadvertent privacy violations in cloud storage. The need for flexibility in sharing specific sets of documents with various user groups requires the use of distinct encryption keys for each document. This undertaking tackles practical difficulties by presenting the notion of Key-Aggregate Searchable Encryption (KASE). Natural language processing was used to extract essential phrases from a file that would be uploaded by the data owner. It removes the punctuation, special character, and numeric values from the file and finds the term frequency of the remaining keywords. With help of the term frequency, the keyword rank of all the words is derived. This helps in retrieving the file at a faster rate. Hash code is also derived from all these keywords obtained to maintain the security of the file.

Keywords: Key searchable encryption, data sharing, cloud computing, data privacy, searchable encryption

INTRODUCTION

Cloud storage has emerged as a promising solution, offering convenient and on-demand access to vast amounts of data. It significantly streamlines collaboration and facilitates the sharing of files. Users may simply share files and folders with others, giving them access to specific works or allowing collaborative editing. Multiple users can work on the same files at the same time, avoiding version control difficulties, which promotes efficient teamwork and increases productivity. Users are worried about unintended cloud privacy violations, but they also appreciate the convenience of sharing data via public cloud storage. The use of cloud storage offers convenience and accessibility but also introduces risks. Encrypting files within the cloud can enhance security, but efficient management of encryption keys is crucial, particularly in large-scale applications [1]. In Cloud storage, a malevolent enemy or a negligent cloud provider may be to blame for data leaks, which can seriously compromise customer or business information. While cloud storage allows users to share data, it also exposes them to the risk of inadvertent data breaches. To address user worries about data breaches in cloud storage, a cryptographic cloud is used, where all files uploaded by the data owner are encrypted in the cloud. Looking for data and retrieving only the information containing the supplied keywords in such big cloud storage has proven to be a significant problem for the user [2]. While integrating a searchable encryption scheme with cryptographic cloud storage can fulfil the fundamental security needs of a cloud storage system, deploying such a system for large-scale applications with millions of users and billions of files may face practical challenges related to the efficient administration of encryption keys. Notably, existing literature has largely overlooked these key management issues. User is directed to the keyword search engine page, where he or she can make a keyword search and, if the keyword matches, the file is shown, once the user clicks on download, the file is downloaded. Searching for particular files using keywords is possible for users, but this procedure might be time-consuming.

***Author for Correspondence**
Lekha Achuth
E-mail: lekha@pes.edu

[1]Student, Department of Computer Applications, People's Education Society University, Bengaluru, Karnataka, India
[2]Associate Professor, Department of Computer Applications, People's Education Society University, Bengaluru, Karnataka, India

Received Date: February 02, 2024
Accepted Date: February 29, 2024
Published Date: March 11, 2024

Citation: Nesha Kavya Urs, Lekha Achuth. Searchable Encryption Based on Key Aggregation. International Journal of Data Structure Studies. 2024; 2(1): 21–25p.

Balancing privacy concerns and the benefits of cloud storage remains an ongoing challenge, as both user convenience and data security are of utmost importance in the cloud storage landscape [3]. A malevolent enemy or a negligent cloud provider may be to blame for data leaks, which can seriously compromise customer or business information. To share many files in the cloud, the data owner must provide the user with a set of aggregate keys. A user will send many execute keyword searches over the shared data and use an index-based file retrieval mechanism, which will take a long time to find the file in the cloud.

LITERATURE SURVEY

Many encryption methods are not designed efficiently to overcome the problems faced in cloud computing [4]. While merging a searchable encryption scheme with cryptographic cloud storage can satisfy the fundamental security needs of cloud storage, creating such a system for extensive applications with millions of users and billions of files may encounter practical challenges related to the effective administration of encryption keys. Notably, existing literature, to the best of our knowledge, has largely overlooked these key management issues [4]. To begin with, the need to selectively share encrypted data with different people typically necessitates the use of separate encryption keys for distinct files. However, this implies that the quantity of keys required to be provided to users for them to search and decrypt encrypted files will be directly proportional to the number of these files. Given all of the requirements, the two existing systems for cryptographic cloud would be:

Multi-user Searchable Encryption (MUSE)

The most common cloud storage setting is a multiuser environment in which users conduct keyword searches. Searches are conducted concurrently. A scenario known as "Multiuser Searchable Encryption" (MUSE) happens when a data owner distributes a collection of documents with a number of approved users, and any individual with the requisite authorization rights may use a cloud gateway to search for keywords throughout the shared data.

Multi-key Searchable Encryption (MKSE)

The ratio of trapdoors to searched documents is the same in multi-user systems. This algorithm is described as requiring the data used to provide the cloud server with a single trapdoor, which is made up of a single term. However, the cloud server also offers the option to search via the keyword phrase using various keys.

PROPOSED SYSTEM

Searchable encryption based on cloud storage has attracted much attention recently [5]. The proposed system is applicable to any cloud based platform which has comprehensive collective information-sharing capabilities, allowing any cloud provider to exchange a bunch of documents with a collection of individuals while allowing the latter to do search queries over the former. The hybrid cloud storage with group data sharing capabilities is covered. There are two primary criteria for effective key management in order to facilitate searchable group data sharing [2]. First, A data owner just must provide a user with a single composite key to share any bunch of documents. Second, to do a web search across any number of uploaded files, the individual simply must enter a single consolidated search to the cloud. Only by performing a single trapdoor, the user will be able to retrieve the file at a faster rate [2]. The admin will produce two keys in the admin module for the encryption as well as decryption processes. The Asymmetric algorithm will be used by the administrator to produce the main confidential key and public key. Admin will clean up the file by removing special characters and punctuation and searching for keywords [6, 7]. Using the MD5 technique, transform each keyword into hash code and save the resultant hash code in the Index Array. The data user can use their login and password to sign in to the User section and do a search based on keywords across any number of supplied files. When the aggregate key is matched, the user is directed to the keyword search engine page, where he or she can make a keyword search and, if the keyword matches, the file is shown; once the user clicks on download, the file is downloaded. Following the correct credentials, users will also be able to alter their profile and change their passwords.

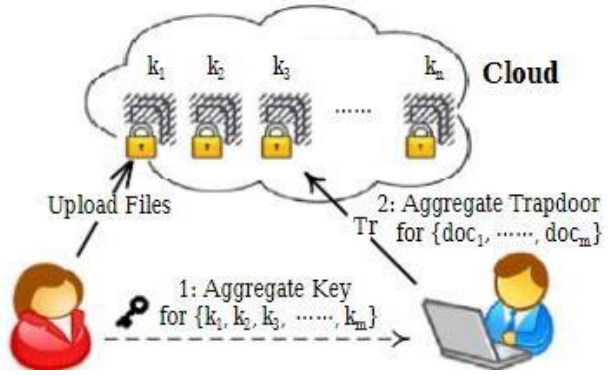

Figure 1. Process of Key-aggregate encryption.

METHODOLOGY

To set up the KASE scheme, the cloud server would generate public system parameters using the Setup process, and these public parameters could be reused by other data owners to share their files [8]. Each data owner is required to generate a public/master-secret key pair using the Keygen algorithm. Subsequently, the keywords associated with each document can be encrypted using the Encrypt algorithm and a unique searchable encryption key. Following this, the data owner can utilize the master-secret key to generate an aggregate searchable encryption key for a specified group of documents. This aggregate key can be securely distributed to authorized users who require access to those documents [8]. An authorized user has the capability to create a keyword trapdoor using the Trapdoor algorithm with this aggregate key and submit the trapdoor to the cloud.

Key Generation

In the field of cryptography, key generation refers to the procedure of producing keys. A key, employed for the encryption and decryption of data, is created during this process. A key generator, commonly referred to as a keygen, is a tool or software that generates keys. The data owner executes this algorithm to generate a random key pair.

Access Control Mechanism

The admin will give access control to the files that are uploaded; only authorised users will be able to read the files that the admin uploads, and while uploading, the admin will encrypt the file with the use of a master secret key for cloud security [8].

Generating Aggregate key

The data owner executes this algorithm to create a collective searchable encryption key, enabling the delegation of keyword search privileges for a specific set of documents to other users (Figure 1). The algorithm takes the owner's master-secret key (MSK) and a set containing document indices as input, and produces the aggregate key as output.

Hashing Technique

It aids in hashing the data into an unintelligible form. It is similar to encryption, except that hashing is one-way only, whereas encryption is two-way. A word is broken down into numerous words and alphabets [9]. There will be different hashing for small and large letters, making decoding the hash word very hard. Hashing operates in a manner such that if even a single character of the message is altered, it will produce a distinct hash.

Trapdoor for Performing the Keyword Search

The user with the aggregate key executes this algorithm to conduct a search [6]. The algorithm takes the aggregate searchable encryption key (KAAG) and a keyword (w) as input, and produces a single trapdoor as output.

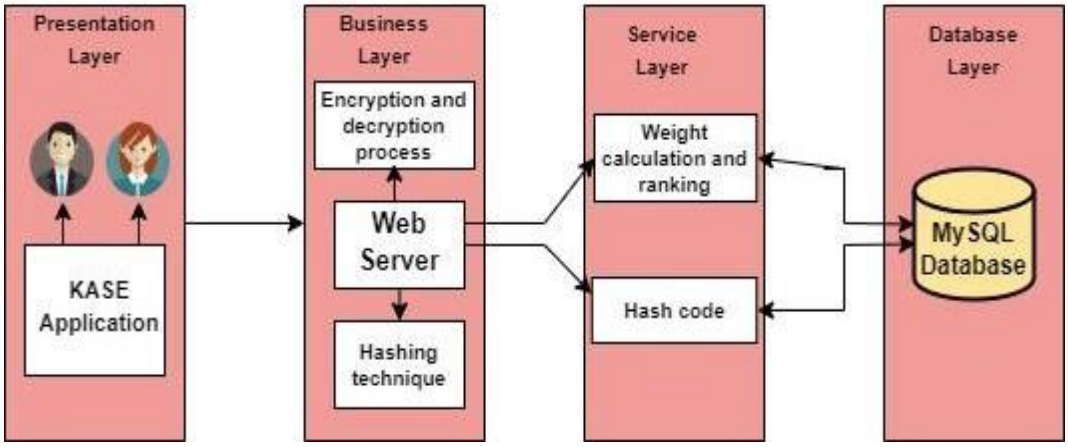

Figure 2. Architecture diagram.

Keyword Search

Initially, the user needs to select the aggregate key before entering the search query. Transform the keyword into a hash code. Encrypt the whole Key, Separate and obtain hash keys, as well as separate and obtain the public Key. Generate hash codes with Hash Key and keyword (Trapdoor) (Figure 1). Send the hash codes to the server; the server must verify the keyword index and, if any matching files are found, list all of the file names to the user [8]. View the server's shortlisted files, download the files, and lastly decrypt the file with the owner's public key.

Cryptography

It is an encryption method that is mostly used for safety precautions to protect sensitive data. It makes use of public key encryption, which has been determined to be the most secure type of encryption [4]. It is a type of public-key encryption method, also known as Asymmetric cryptography, in which the public key is shared with anybody who wants to access the data while the private key is kept private and not shared with anyone.

SYSTEM ARCHITECTURE

Users in the presentation layer will utilise the KASE interface. In the business layer, there is a web server that basically performs two operations: encryption, decryption, and hashing [2]. Following that, it retains the weight calculation, ranking, and hash code (Figure 2). The administrator will gain entry to the website by providing the admin ID and password. Admin will create a user profile based on the user's information, upload the content to the cloud, and do keyword ranking on that file [2]. The administrator will provide the user access to the programme, and the user will enter the aggregate key to do the keyword search and obtain the encrypted file [8, 10].

CONCLUSION AND FUTURE WORK

The Key-aggregate searchable encryption was designed to overcome all the issues of cloud data management systems. To transfer any collection of files in the cloud, the data owner will only be required to transmit a single shared key to the user. The user will just need to input the public key that was supplied to them and do a keyword search on the shared file. The file can be retrieved more quickly and efficiently with the help of a ranking system. After uploading the file to the cloud, a hash key will be generated. This aids in data security of the event on unintentional data leaks in the cloud. In the future, we hope this system helps users easily access the cloud application without any hustle. Key-aggregate searchable encryption will eventually be available as an Android application, making it more practical and convenient to use. By including a time frame for the public key shared during key generation, it is possible to further increase security and prevent unauthorized access. In the future, as soon as the file is submitted by the data owner, notifications will also be sent to the user via SMS.

REFERENCES

1. Boneh D, Gentry C, Waters B. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Annual international cryptology conference. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005 Aug 14; 258–275.
2. Li J, Chen X, Li M, Li J, Lee PP, Lou W. Secure deduplication with efficient and reliable convergent key management. IEEE Trans Parallel Distrib Syst. 2013 Nov 8; 25(6): 1615–25.
3. Li J, Kim K. Hidden attribute-based signatures without anonymity revocation. Inf Sci. 2010 May 1; 180(9): 1681–9.
4. Yu S, Wang C, Ren K, Lou W. Achieving secure, scalable, and fine-grained data access control in cloud computing. In 2010 Proceedings IEEE INFOCOM. 2010 Mar 14; 1–9.
5. Lu R, Lin X, Liang X, Shen X. Secure provenance: the essential of bread and butter of data forensics in cloud computing. In Proceedings of the 5th ACM symposium on information, computer and communications security. 2010 Apr 13; 282–292.
6. Oliveira LB, Aranha DF, Morais E, Daguano F, López J, Dahab R. Tinytate: Computing the tate pairing in resource-constrained sensor nodes. In 6th IEEE International Symposium on Network Computing and Applications (NCA 2007). 2007 Jul 12; 318–323.
7. Phan DH, Pointcheval D, Shahandashti SF, Strefler M. Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts. Int J Inf Secur. 2013 Aug; 12: 251–65.
8. Chu CK, Chow SS, Tzeng WG, Zhou J, Deng RH. Key-aggregate cryptosystem for scalable data sharing in cloud storage. IEEE Trans Parallel Distrib Syst. 2013 Apr 11; 25(2): 468–77.
9. Li M, Lou W, Ren K. Data security and privacy in wireless body area networks. IEEE Wirel Commun. 2010 Feb 18; 17(1): 51–8.
10. Song DX, Wagner D, Perrig A. Practical techniques for searches on encrypted data. In Proceeding 2000 IEEE symposium on security and privacy; S&P 2000. 2000 May 14; 44–55.

