Cloud Computing's Fuzzy Search Over Encrypted Data

CHETAN N. RATHOD[1], BHUMIKA K. CHARNANAND[2]
[1]*Vivekanand College for BCA, Surat*
[2]*Smt. Z.S. Patel College, Surat*

Abstract— Since all data is stored on the cloud, cloud computing is known to be centralised. Data may or may not be sensitive, but it is unacceptable to compromise privacy. Encrypting data is a prerequisite for outsourcing. Although encryption makes it difficult to use data, traditional software can still retrieve data even when encryption is in place. However, in order to obtain data, the exact keyword must match during the search in order to download the necessary file; otherwise, you will receive blank results. Even a small typo is unacceptable, and we regularly encounter errors of this kind. This results in low efficiency and negatively impacts system usability. Thus, by achieving them and maintaining data encryption against unauthorised users and cloud owners, we are able to formalise the issue and find solutions through a variety of methods. By matching the exact or closest match text to the stored keywords and retrieving the approximate closest results, fuzzy keyword search improves the system's usability. Here, we're quantifying keywords using edit distance and building the keyword set using sophisticated techniques that help cut down on overhead and storage. With the aid of file mapping based on usage, this also aids in search ranking, which places the most frequently viewed documents first, followed by others. In order to keep it safe even from cloud users, we are also putting one of the more sophisticated encryption algorithms into practice before uploading it over the cloud. The more sophisticated function outlined here involves using a selective encryption algorithm to work with images and videos in addition to documents.

Indexed Terms- Cloud computing, encryption, fuzzy search, edit distance, ranking search

I. INTRODUCTION

A variety of data types, including social media accounts, game data, website logins, and many more, are stored centrally in the cloud. The cloud is used because it relieves data owners of the burden of keeping their data on their own, which can occasionally be fatal due to hard drive failure or other related issues. The other issue might be data maintenance, which includes data availability and dependability, and the inability to receive high-quality service because the configuration is inferior to that of cloud servers. However, cloud computing also has some disadvantages. For example, cloud servers are not in the same domain as data owners, so it is the user's responsibility to encrypt data before uploading it. Implementing data encryption results in an overhead of efficiently using data. Additionally, data owners share their outsourced data with a large number of users in cloud computing. Within a session, each person will only retrieve the particular data files they are searching for. In order to implement this kind of system, we must deal with keyword searches that only return the necessary files rather than all of the encrypted ones.

This keyword search method has been widely used in plaintext search scenarios, like Google search, and enables users to selectively retrieve files of interest [1]. Sadly, users cannot use the keyword search method on encrypted data, which renders plaintext search methods useless for cloud computing.

In addition, encrypted data files that contain the file name must be safeguarded since they may also contain sensitive and high-quality information. However, the conventional plain text method becomes completely ineffective when the file name is encrypted because it can only search plain text.

In recent years, searchable encryption techniques have been developed to search over encrypted files [2]–[10]. Techniques for searchable encryption generate a keyword index and associate it with the relevant file [3]–[10]. This method is efficient and helps us search safely, but it is not suitable for cloud computing because all searchable encryption methods rely on exact keywords; even a small typo will prevent the file from being retrieved. It is also common for user input to not match the pre-defined keywords, such as beautiful instead of beautiful, or to have variations in representation, such as R.I.P. and RIP, which may also

be the result of a lack of detailed knowledge about the data.

The conventionally used algorithm might not be able to fully solve the problem if another valid keyword is accidentally typed. For instance, if someone searches for "hate" and accidentally types "late," the algorithm won't work because it would be difficult to distinguish between the two valid words. As a result, the disadvantage forces us to use a different algorithm that supports searching in terms of typos and flexibility.

In this paper, we apply various encryption algorithms to implement fuzzy keyword search over cloud while protecting document privacy from unauthorised users. By comparing the input search text with the pre-defined keywords, fuzzy keyword search helps us improve the usability of our system while searching for files. When an exact search fails to yield the desired result, it also looks for the closest match by modifying the user's search text with potential values. By using the sophisticated algorithm technique for storing, matching, and searching fuzzy keyword sets, we are using the edit distance technique to quantify the similarity of keywords. [J. Li 2009, Xin Zhou 2006]. By decreasing the number of keywords, this algorithm helps us retrieve data quickly and reduces the overhead of matching to all fuzzy keywords. It also eliminates the need to store all fuzzy keywords, improving efficiency in terms of privacy and the overhead of storing a large number of keywords. In order to secure our documents, even from cloud owners, we will use the AES encryption algorithm. We will also use a selective algorithm to encrypt images and videos. After analysis, we can conclude that the following solution is as secure as it was previously, while also enhancing the system's flexibility in terms of cloud usage.

II. SURVEY OF LITERATURE

Fuzzy keyword search for plain text: Many communities are currently placing a lot of emphasis on fuzzy keyword search for plain text [1]. They were able to resolve this issue by utilising a similar string matching algorithm instead of a try-and-see strategy to find relevant information. Eventually, if we put this string matching algorithm into practice, it would also be useless in terms of privacy because hackers might use dictionary or statistics attacks, which would allow unauthorised individuals to access the files.

Searchable encryption: In the context of cryptography, traditional searchable encryption [2]–[8], [10] has been extensively researched. The majority of algorithms search for ways to increase security and efficiency. Song et al. [3] proposed the first searchable encryption construction, in which a unique two-layered encryption construction encrypts each word in the document separately. According to Goh [4], the indexes for the data files should be constructed using Bloom filters. Chang et al. [7] and Curtmola et al. [8] both proposed similar "index" strategies to increase search efficiency, where an encrypted hash table index is made for the entire collection. Each keyword entry in the index files used by this algorithm is connected to an encrypted collection of files containing related keywords. With a scenario similar to [3], Boneh et al. [10] offered a public-key based searchable encryption scheme as a supplementary method. All of these schemes are ineffective for cloud computing because they only function with precise searches.

Complete Search: Bast et al. suggested methods to facilitate "Complete Search," where a user enters keywords one letter at a time, and the system locates records that contain these terms (perhaps in various locations) [10].

Selective Encryption: When encrypting an image, the entire image is first compressed, and then the entire image is encrypted using a standard cypher technique like RLE, AES, etc. All of these encryption techniques are not as effective as they should be because encrypted data typically requires higher transmission and not all of them have that higher bandwidth. compressed the entire picture and then encrypted it, which is referred to as fully layered. Another issue with it is that different users might not have the same codec functionalities, which restricts how the algorithm can be used online. One emerging technique is selecting encryption, which encrypts only the chosen portion of the image rather than the entire image.

III. METHODS OF SOLUTION

System Model:

The administrator, user, and cloud server are linked to the cloud system in this paper. Using a predetermined set of distinct keywords over the encrypted data, the user searches the uploaded data files in the cloud system—a feature that the cloud only allows to authorised users.

By identifying the file and a specific link to a group of keywords, the cloud server's job is to retrieve the appropriate file based on the user's interests [1]. When a fuzzy keyword scheme is implemented in specific protocols, the following outcomes are produced:

1. The server is in charge of providing the correct file when the user enters the exact file name.
2. The server should be able to send the closest result if the user fails to enter the exact keyword.

Threat Model:

In light of this, it is possible that sensitive information could be compromised by a user's request via the cloud if the data files are kept on a less secure server. The procedure has been carried out in a secure manner to remove this risk factor, and sensitive data is unaffected even though the cloud server obtains the data from the user-inputted keyword [2].

Design Goals:

The main challenge in this paper [5] is to provide effective fuzzy keyword search over cloud-encrypted data while maintaining confidentiality. The following objectives are primarily taken into account: I) Fuzzy keyword storage should be efficient; ii) Fuzzy keyword design should be efficient; and iii) The implied scheme's security should be maintained.

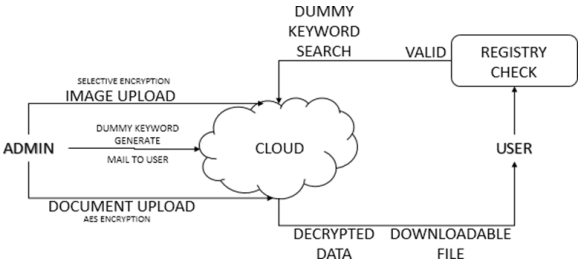

Figure 1: Architecture of Proposed System

Preliminaries:

Modify Distance We have selected the edit distance as the quantitative measure for matching strings after conducting a thorough analysis. To change one word to another, different operations—substitution, deletion, and insertion—are needed between the corresponding words (k1, k2). Assuming a keyword k, let (Wk,c) represent the collection of words k_ that satisfy $c(k,k\_) \leq c$ for any given integer c.

Advanced Methods for Building Sets of Fuzzy Keywords:

We now move on to its advanced techniques in order to improve the simple approach and the effective practical use of fuzzy keywords with storage and search capabilities.[7] While suppressing the fuzzy keyword, the search matching scheme will remain a concern. By concentrating on the edit distance state where d = 1, the search's generality over the encrypted data will also be preserved. The same logic will apply to higher values of d.

Wildcard-based Fuzzy Set Construct:

As we saw in the previous section of the simple method, if the list is even, all of the keywords operate out to the same position. Our proposal was to use a fuzzy set based on wildcards to indicate the edit operations at the same position.

Gram-based Fuzzy Search:

A substring that can serve as a signature for effective search is called a gramme of a string. The following observation is used by these algorithms to respond to a fuzzy query on a set of strings: There should be a specific number of common grammes between the query string and a string r in the collection if they are similar. To facilitate effective search, gram-inverted lists for string ids can be created using this count filter.

The Symbol-based Trie-Traverse Search Scheme:

We now suggest a symbol-based trie-traverse search scheme to improve search efficiency. This scheme uses a multiway tree to store the fuzzy keyword set {S wi, d } i wi∈W over a finite symbol set.[1]–[3] This construction's main premise is that all trapdoors with a common prefix might share nodes. The symbols in a trapdoor can be recovered by searching from the root to the leaf that closes the trapdoor, and the root is

linked to an empty set. A depth-first search will find every fuzzy word in the trie.

Summary:

This paper allows us to design fuzzy search over cloud, eliminate the issue of exact match search, and increase search possibilities by partially allowing typos. This was made possible by the application of two sophisticated techniques (the gram-based and wildcard-based techniques), which also offer other crucial advantages like storage efficiency. In order to accomplish multi way tree structure using symbols over fuzzy keyword sets, we also employed another effective search method, namely the symbol-based trie traverse scheme. By analysing our system, we can conclude that the suggested system is effective without sacrificing security.

REFERENCES

[1] Fuzzy keyword search over encrypted data in cloud computing" by C. Anuradha.

[2] Implementation of Fuzzy keyword search over encrypted data in cloud computing" by D. VASUMATHI.

[3] Fuzzy keyword search over encrypted data in cloud computing", Illinois Institute of Technology, ISSN: 2321-8134.

[4] Practical techniques for searches on encrypted data" by D. Song, A. Perrig. In IEEE, 2000.

[5] Privacy preserving keyword searches on remote encrypted data" by Y. C. Chang in ACNS, 2005.

[6] Overview on selective encryption of image and video" by A Massoudi in EURASIP, 2008.

[7] Efficient interactive fuzzy keyword search "by J. Feng, G. Li in WWW, 2009.

[8] International Journal of Advanced Research in Computer Science and Software Engineering" Research Paper by P.Kalidas, R.Chandrasekaran.

[9] A. Behm, S. Ji, C. Li,, and J. Lu, "Space-constrained gram-based indexing for efficient approximate string search," in

[10] Proc. of ICDE'09. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. of EUROCRYP'04, 2004.