Fuzzy Keyword Search over Encrypted Data in Cloud Computing Using Advanced Encryption and Pattern Matching Algorithm

R. Vigneshwar[1], A. Tejhaskar[2]
[1]*UG Scholar, Sri Sai Ram Institute of Technology*
[2]*UG Scholar, Sri Sai Ram Institute of Technology*

***Abstract*- With the evolution in Cloud Computing more and more sensitive data is being incorporated into the cloud. To ensure security and privacy these data are first encrypted before being uploaded onto the cloud servers thus making search a complicated task. Although in traditional cloud computing encryption searching schemes allows user to search encrypted data through keywords securely. These techniques employed exact keyword search and will fail if there are any morphological variants or spelling errors. This leads to low in efficiency and also affects system usability very badly. Fuzzy keyword search increases the system usability by allowing matching the exact or closet match text to the stored keywords and retrieving the approximate closest results. We shall be using edit distance to quantify keywords. We ensure the privacy of the data against unauthenticated users by encrypting the data using Honey encryption before uploading to the cloud servers. We tend to resolve this problem by using a cloud server and employing fuzzy keyword search based on N grams. Thus efficiency of our proposed system would be demonstrated through experimental results.**

***Index Terms*- HoneyEncryption, Fuzzy Keyword, Wildcard mechanism, Cloud Computing**

I. INTRODUCTION

With the advancement in cloud computing, cloud servers are widely being used for storing data centrally. This includes various social accounts, game data, website login and more type of data. The cloud services provides relief to user as it reduces storage overheads and risk of losing the data due to hardware failures i.e. it might happen the hard disk of our system or due to malicious activity and we would end up losing all the important data. The other problem may be poor maintenance and low configuration service as compared to cloud configuration services. On the other hand cloud also has some drawbacks because cloud servers cannot be trusted by the data owners so it is the user's responsibility to encrypt the data before upload. By implementing data encryption, there's overhead of data utilization in more efficient manner as the data is secured and cannot be accessed by unauthenticated users. Also, in cloud computing, data owners share their outsourced data with large number of users due to which privacy of the data is not ensured. Thus it is required that every individual should retrieve specific data files which they are looking for within a session. To apply this type of system we need to deal with keyword search that retrieve the required files instead of retrieving all the encrypted files.

In plaintext search scenarios such as Google search, the keyword search technique is used which allows users to selectively retrieve the required files. Unfortunately, encrypted data restricts user's ability to use the keyword search technique and thus makes the plaintext search methods no use for Cloud Computing. Apart from this, encrypted data files which consist of file name needs to be protected as it may also describe the quality and sensitivity of information related to the data files. But by encrypting file name the traditional plain text methodology get totally useless as it is only able to search over plain text. In this paper, we are implementing fuzzy keyword search over cloud without compromising the privacy of our data. By employing fuzzy keyword search the usability of our system is enhanced. Users can search their text with possible values and get the desired result when exact keyword match fails. This failure of exact keyword could be because of some spelling or morphological

error. Thus fuzzy keyword search helps to overcome this and give desired results to the user. In our proposed system, edit distance technique to quantify keywords similarity by implementing the advanced algorithm technique for storing, matching and searching fuzzy keyword sets. These algorithms eliminate the need for storing all fuzzy keywords to improve efficiency in terms of privacy as well as overhead of storing large number of keywords by reducing the number of keywords which helps us to retrieve fast data and overhead of matching to all fuzzy keyword is reduced. We shall be implementing Honey encryption algorithm before uploading our documents over the cloud servers. This is done to ensure secure and privacy of our data against unauthenticated users. Fuzzy keyword search would be then implemented using N-grams and wildcard-based technique.

2. RELATED WORK

Plaintext Fuzzy Keyword Search

Currently much of the importance is given to fuzzy search for plain text with the help of fuzzy keyword search by many communities [2]. This problem was solved by rejecting the idea of using of try and see approach for searching related work and instead using using string matching algorithms. But again this method

Honey Encryption:

Ari Juels together with Thomas Ristenpart of the University of Wisconsin has developed a new encryption system known as Honey Encryption. [10]This Encryption system proves to be highly resilient against Brute force attack. With the help of this Encryption system, if cipher text is decrypted with the incorrect key, it produces a plausible looking yet incorrect plaintext. The incorrect key will generate a fake plaintext when used while decrypting the data. The attackers think of the fake plaintext as a legal message as it looks like a plausible plaintext.

Complete Search

In complete search user types the keyword letter by letter and system retrieves all the records that contain the keyword.

3. PROBLEM FORMULATION:

3.1 System Model

Here we assume a cloud data system which consist of data owner, data user and cloud server. Cloud Server is responsible for mapping searching request for the authorized users over the encrypted data C. This encrypted data C consist of n encrypted data files and a predefined set of distinct keywords W = {w1, w2,…

3.2 Design Goals

In this paper, we address the problem of fuzzy keyword search services over the encrypted cloud data [11]. Specifically, we have the following goals: (1) to explore new mechanisms which can construct storage efficient fuzzy keyword sets; (2) to design an effective and efficient fuzzy search scheme which is based on the constructed fuzzy keyword sets; (3) to validate the security of the proposed scheme [12].

3.3 Preliminaries

3.3.1 Edit Distance

There are many methods to quantitavely measure the similarity of the strings [13]. In this paper, for our purpose we use the well-studied edit distance [16]. The edit distance ed(w 1, w2) between two words w1 and w2 is defined as the number of operations required to transform one word into another. The three operations are: (1) Substitution: changing one character to another in a word;Deletion: deleting one character from a word; (3) Insertion: inserting one character into a word. For a certain integer d, Sw,d denotes the set of words w' which satisfy the edit distance ed (w,w') < d.

3.3.2 Fuzzy Keyword Search

On the basis of edit distance, we define fuzzy keyword as a set of distinct keywords W={w1, w2,….,wn} with edit distance d, n . Encrypted data files C=(F1,F2,….FN) which are stored on a cloudserver and a searching input (w,k) with edit distance k where k <d, the execution of fuzzy keyword search returns a set of File Ids whose set of distict keywords matches the input word w, provided by the user, otherwise if the input word does not matches the set of distinct keywords of the encrypted data files stored on the cloud server then it returns those File Ids whose edit distance ed(w,wi) < k.

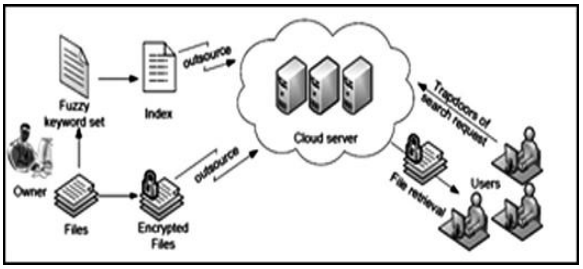


Fig 1. Architecture of fuzzy keyword search

4. PROPOSED SYSTEM:

Honey Encryption:

We will describe how the original Honey Encryption works, which Juels and Ristenpart developed. [11]For implementing Honey Encryption, we need a message space M that contains all possible messages. By using Distribution Transforming Encoder (DTE), we map each message with a seed space S. DTE acts as a function F mapping seeds, which are just bit strings of a length n (n being a predetermined value), from a seed space into the message space. For example, in Figure 4, "JFK" from message space is being mapped to Seed Space "000". A good Distribution Transforming Encoder is one that models the message distribution well, in the sense that if you pick a seed uniformly at random and then apply the DTE to it, you will get back the message distribution. Moreover, the function F of DTE is invertible which means that if you pick a message, you can find the corresponding seed. Honey encryption also makes use of a mapping from Key (Password) Distribution to Seed Space using a function G. G maps keys (passwords) randomly onto the seed space like a Hash function. For example, we are assuming that password "ABC" is mapped to 000, 001 and 010.Therefore, to encrypt a message M under a Password K, we first compute a seed corresponding to the message of the function F inverse. After the seed is computed for the message, we compute a seed corresponding to the password. Once both the seeds are computed, encryption can be achieved by XOR both the seeds. For example, consider the encryption of Airport codes in Figure 4. Our message space M consists of different Airport codes, M = {JFK, LGA, EWR, ATL, SFO, LAX, LAS, MIA}. We have taken the seed space of 3 bits in this example. Now, consider the process of encrypting an airport code, say "ATL" under the password "XYZ". Using the DTE, we find the seed corresponding to our airport code "ATL" to get the seed value as 011. At the same time, DTE finds the seed corresponding to our password "XYZ" to get the seed value as 110. After retrieving the seed value for both the message and the password, DTE XOR them together, i.e. 011 XOR 110 to get 101 as the cipher text.To decrypt now, DTE finds the seed value for the password "XYZ" to retrieve 110. After retrieving the seed value of the password, DTE XOR the seed value 110 with the cipher text 101 to recover the original seed as 011. DTE takes the seed after decryption, maps it to the message distribution and recover the correct airport code as "ATL". Now, if you try to decrypt the message with the wrong password, it will generate a plausible looking yet incorrect message. For example, let us try to decrypt the same cipher text 101 using another password "DEF". DTE will find the seed value for the password "DEF" to retrieve 011. After retrieving the seed value of the password, DTE XOR the seed value 011 with the cipher text 101 to get the seed value as 110.DTE takes the seed to get the airport code as "LAS". In this case, we get a perfectly valid looking airport code but the airport code was incorrect. We get the airport code as "LAS" instead of the airport code as "ATL". The decryption under any password yields a valid message.The above-explained method of Honey Encryption can be used for encrypting the messages in the Cloud Architecture. After the messages are encrypted using Honey Encryption, we introduce a method of Secure Repository Manager (SRM) specifically for Cloud computing. [12] SRM can be used for securely storing the data on the Cloud Architecture. The job of SRM is to divide the data into small chunks and upload the data randomly onto the cloud server. The size of the chunk was directly related to the level of the security of the data. Division of the data into smaller chunks means highly sensitive and critical data. Division of the data into medium chunks means the data is less sensitive and critical. Division of the data into bigger chunks means the data was not at all sensitive or critical.After the encryption process, the data chunks were stored at random locations. The information about the location and the server was stored in the Secure Repository Manager (SRM). For the data retrieval process, necessary information in the form of a password, chunk size and location is gathered by SRM at the client side. After the

information gathering is done, decryption process takes place and the data is kept in order for maintaining the original form of the message. For avoiding the damage of data while transferring from cloud to the user, secure connection in the form of HTTPS and SSH is used. If any means still damage the data, a request is resent to the cloud server, for the user to have the data in the original form.

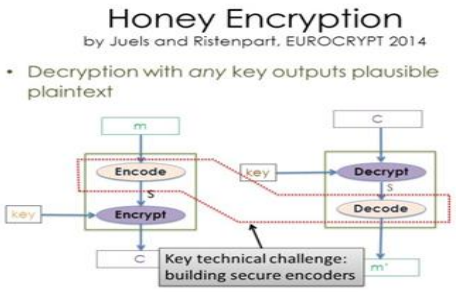


Fig. 2: architecture of honey encryption

Wild-card based Technique for Constructing Fuzzy Keyword Set:

The wildcard-based fuzzy set of wi with edit distance d is denotedas Swi,d={S'wi,0, S'wi,1, …….S'wi,d}, where S'wi,t denotes the set ofwords wi. For example, fuzzy sets for the keyword CAMERA using wild-card based technique and with edit distance 1 can beconstructing as, SCAMERA,1={CAMERA, *CAMERA, *AMERA, C*AMERA, C*MERA, …….. CAMER*, CAMERA*}. So wecan see that total number of variants in the fuzzy set of keyword CAMERA is 13+1. Using straightforward approach it would be 13 × 26 + 1. So we can see the proposed approach reduce the storage required. Using this approach 30GB storage required can be reduce to 40MB, approximately.

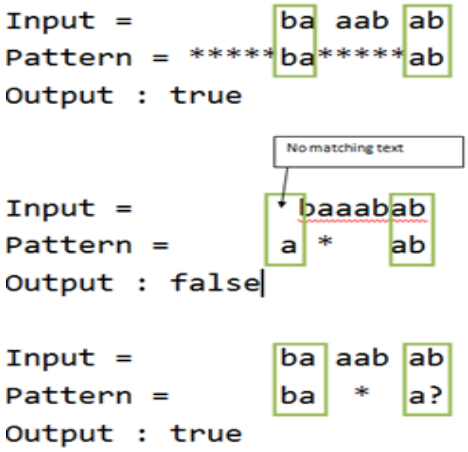
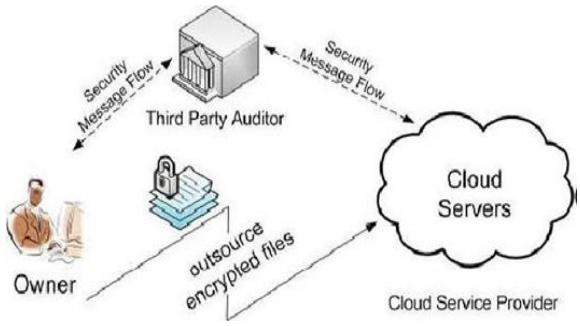


Fig. 3: wildcard pattern matching



5. CONCLUSIONS:

In this paper, we have solved the problem secure and privacy preserving fuzzy search for effective utilization of encrypted data uploaded onto the cloud servers. We have created fuzzy sets using gram and edit distance technique. After thorough security analysis we have found that our proposed system is secure and privacy preserving while realizing the goal of Fuzzy keyword search.

REFERENCES

[1] A REVIEW PAPER ON FUZZY SEARCH OVER ENCRYPTED DATA IN CLOUD COMPUTING by Neel Gala ISSN:2393-9842

[2] Fuzzy keyword search over encrypted data in cloud computing" by T. Balamuralikrishna.

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

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

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

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

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

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

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

[10] A Behm, S. Ji, C. Li,, and J. Lu, " Space-constrained gram-based indexing for efficient approximate stringsearch," in Proc. of ICDE'09 .

[11] Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, " Public key encryption with keyword search," in Proc. ofEUROCRYP'04 , 2004.
[12] Implementation of Fuzzy Keyword Search Over Encrypted Data in Cloud Computing by ChandniChandawalla.
[13] Security in Cloud Computing using Honey encryption by Fancy Arora
[14] Effective and Efficient Fuzzy Keyword Search over Encrypted data in cloud computing by AreebaTauseef , Manas Gupta , AshiVarshney , Ankit Agarwal , Manish Gupta.