Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024

PROTECTING YOUR MOBILE CLOUD DATA CHAOS-BASED ENCRYPTION

Mrs. Kanukuntla Rama Devi Assistant professor CSE, Vaagdevi College of Engineering(Autonomous),India
NEERUDU RAGAL BANDE, UG Student,CSE,Vaagdevi College of Engineering(Autonomous),India
MAMINDLA VAMSHI ,UG Student,CSE,Vaagdevi College of Engineering(Autonomous),India
MIRZA RAMYAN BAIG ,UG Student,CSE,Vaagdevi College of Engineering(Autonomous),India
NIDIGONDA SAGARIKA, UG Student,CSE,Vaagdevi College of Engineering(Autonomous),India

Abstract:
This paper considers the security problem of outsourcing storage from user devices to the cloud. A secure searchable encryption scheme is presented to enable searching of encrypted user data in the cloud. The scheme simultaneously supports fuzzy keyword searching and matched results ranking, which are two important factors in facilitating practical searchable encryption. A chaotic fuzzy transformation method is proposed to support secure fuzzy keyword indexing, storage and query. A secure posting list is also created to rank the matched results while maintaining the privacy and confidentiality of the user data, and saving the resources of the user mobile devices. Comprehensive tests have been performed and the experimental results show that the proposed scheme is efficient and suitable for a secure searchable cloud storage system

1. INTRODUCTION

CLOUD computing is a model to enable convenient, ondemand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) [1]. In the current Internet, people can easily access their data stored in the cloud with their mobile devices from anywhere e.g., check emails, read the history of online chatting applications, view previously saved photos, videos or other kind of documents. To provide security in all such scenarios, it is essential to store and access the outsourced data in a secure and efficient manner. For the protection of data privacy and control, data is usually encrypted before outsourcing, which makes its effective utilization a challenge. In particular, indexing and searching the outsourced encrypted data becomes problematic. Searchable encryption (SE) allows searching over encrypted data in the cloud and returns to the user the data that correspond to the given keywords, without having to reveal the keywords. It is thus a critical enabler for securing outsourced data. Traditional searchable encryption [2]-[7] schemes allow a user to securely search over encrypted data through keywords but only support 1) exact keyword matching, which is not a practical requirement for current mobile phone input methods and 2) boolean search without capturing the relevance of data files. The system usability can be greatly enhanced by the use of fuzzy keyword search [1], [8]- [10] instead of traditional searchable encryption. Fuzzy, or error tolerant, searchable encryption returns to the user the files that match not only the exact predefined keywords but also the closest possible matched files based on keyword similarity semantics. Similarly, system usability is greatly enhanced by ranked search [11], [12] which returns the matched files in a ranked order determined by appropriate relevance criteria. This paper investigates the problem of supporting both ranked and fuzzy keyword search in a single scheme to achieve effective utilization of remotely stored encrypted data in mobile cloud computing applications. Many approaches are proposed to enable fuzzy search. Researchers in [8] consider the use of wildcards to enlarge the range of possible similar keywords searched, but this technique only covers part of the possible close keywords. A wildcard only permits capturing of errors provided we know where they are located in the keyword [1]. In [9], the authors proposed a new cryptographic primitive called Public Key Error Tolerant Searchable Encryption (PKETS) which is based on public key encryption with keyword search proposed in [2]. This algorithm was applied to the biometric data in [13]. Acceptable

erroneous keywords did not have to be specified in advance in their algorithm. However, this approach was designed for a special type of data i.e. iris code. This technology is useful at airports as a replacement for passports but it is not designed for text documents. The authors in [14], proposed to embed edit distance (Levenshtein distance) into Hamming distance to obtain a fuzzy keyword search suitable for strings and then text files. This method uses existing locality sensitive hashing (LSH) to enable the fuzziness in the search method and has a very low distortion. However, this method is mainly theoretical and the proposed embedding technique introduces a lot of redundancy, which increases the dimension of the stored data, and is not suitable for the case of mobile usage because of the small amount of memory available. Another method, proposed in [15], uses bloom filters and Jaccard similarity to perform the translation and the LSH. It also introduces ranking of the retrieved encrypted data. However, the ranking has to be performed by the user himself and not automatically by the server which can add unwanted burden for a mobile user's device. Chaotic Searchable Encryption for Mobile Cloud Storage Abir Awad, Adrian Matthews, Yuansong Qiao, Brian Lee Actually, very few searchable encryption schemes support the ranking of matched items though this problem has recently attracted the attention of some researchers [11], [12], [16]. Fuzziness and ranking are currently two different research axes and very few researchers have considered combining them [15], [17]. However, these methods are either not practical for mobile usage as is the case in [15] or they suffer from security problems as is the case in [17]. In this paper, we propose a new fuzzy transformation by introducing chaos and enhance the fuzziness through amplification of the LSH, which significantly improves both the security and the efficiency of the fuzzy searching process compared to the existing solutions. Furthermore, comprehensive tests on different LSH methods are performed in order to select the best one to be used in our algorithm. Chaotic systems are widely used in the cryptography domain and have attracted the attention of many researchers [21]-[23] due to the interesting characteristics of chaos. However, to the best of our knowledge, this is the first paper proposing to use chaos in the searchable encryption schemes. Our proposed system is, in addition, designed to support fuzzy and ranking mechanisms and is proven to be practical for mobile usage.

2. PROBLEM STATEMENT

In the existing system, Bringer et al. proposed a newscheme permitting search over encrypted data with anapproximation of a keyword. An application in the biometricdomain is also proposed. A biometric identification schemearises from this construction; it permits identification of aperson using his biometrics in an encrypted way. A specificdifficulty concerning biometrics is their fuzziness. It is nearlyimpossible for a sensor to obtain the same image frombiometric data twice. The classical way to solve this problemis to use a matching function, which basically tells if twomeasures represent the same biometric data or not, but thesemethods do not meet the privacy requirements that someonecan expect from an such identification scheme. The Bringer et al. algorithm resolves this issue and provides the privacymissing in the existing algorithms. This method uses acombination of LSH method specific for an iris code (beaconindexes) to enable the fuzziness and a Bloom filter withstorage to accelerate the search on the encrypted data. In [14], the authors modified the above mentionedalgorithm to allow its usage for text messages. The changesentail on applying embedding and sketching methods on themessage which enables the application of the above mentionedalgorithm in [9], [13] that was previously used for thebiometric information. However, the algorithm is stilltheoretical and no implementation or test is provided.

3. PROPOSED SYSTEM

The system proposes a new fuzzy transformation by introducing chaos and enhances the fuzziness through amplification of the LSH, which significantly improves both the security and the efficiency of the fuzzy searching process compared to the existing solutions. Furthermore, comprehensive tests on different LSH methods are performed in order to select the best one to be used in our algorithm. Chaotic systems are widely used in the cryptography domain and have attracted the attention of many

researchers due to the interesting characteristics of chaos. However, to the best of our knowledge, this is the first paper proposing to use chaos in the searchable encryption schemes. Our proposed system is, in addition, designed to support fuzzy and ranking mechanisms and is proven to be practical for mobile usage.

4. IMPLEMENTATION

4.1 Cloud

In this module, the Cloud has to login by using valid user name and password. After login successful he can perform some operations such as View All Users, View All Documents, View Top 'K' Keywords, View Keywords and Links, View Time Delay of Files, View User Transactions, View File Rank Results ,View Time Delay Comparison Results

4.2 Friend Request & Response

In this module, the admin can view all the friend requests and responses.Here all the requests and responses will be displayed with their tags such as Id, requested user photo, requested user name, user name request to, status and time & date. If the user accepts the request then the status will be changed to accepted or else the status will remains as waiting.

4.3 Client

In this module, there are n numbers of clients are present. Client should register before performing any operations. Once Client registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Verify finger print and LoginOnce Login is successful Client can perform some operations likeView Profile, Upload Document, Edit / Delete Document, Search Cloud Data, View Document Search Comparison, View Keyword and Fetched Files, View Same Data Files

4.4 Searching Users to make friends

In this module, the user searches for users in Same Network and in the Networks and sends friend requests to them. The user can search for users in other Networks to make friends only if they have permission

Steps of Implementation

Using image-encryption algorithms to encrypt textual data is conceptually quite simple. The steps of this procedure are as follows:

1. Convert a text message into an image
2. Apply the selected image-encryption algorithm to the received image

The first step above is to convert text into an image. This allows text to be input into the encryption algorithm. In the second step, the received image is encrypted according to the selected image-encryption algorithm.

The following subsections describe the above steps in detail.

4.6 Text-to-Image Conversion

The authors in [35] present a simple algorithm for changing the message format from text to image. This procedure was not related to one of the encryption steps. It was mainly intended to save disk space occupied by data files. So using it in the context of encryption is a new idea. This algorithm requires that the text message must be encoded with UTF-8. A simplified version of this algorithm is presented below to better illustrate the procedure:

1. The text message is read into the buffer.
2. Text characters ti are converted to numbers in the range 0–65535 (UTF-8 encoding).
3. Each of the values ti is stored in the positional system with base 256 according to the equation:

$$ti=ai\cdot 256+bi \tag{1}$$

where ai and bi are the coefficients of ti in the positional numeral system with base 256. Values ai and bi are stored in one-dimensional table T.

4. The image size (height and width) is calculated using the formula

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024

$$size = \left\lceil \sqrt{\frac{length_T}{3}} \right\rceil \quad (2)$$

where $0\cdot$ stands for the ceiling function, and $lengthT$ is the length of the table T.

5. If

$$length_T < 3 \cdot size^2, \quad (3)$$

length is exactly then T is first padded with the value 3 ("End-of-Text" character) and then with random values from set $\{0,1,\cdots,255\}\{0,1,\cdots,255\}$ so that its

6. T is saved as a three-dimensional array $TPsize \times size \times 3$, i.e., every three consecutive elements of the array T are stored as an element of a matrix with dimensions of $size \times size$

7. Array $TPsize \times size \times 3$ is treated as an array of pixels and saved as a png file.

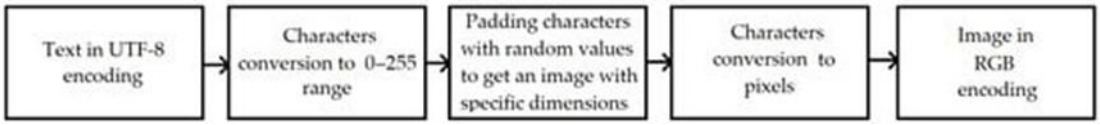


Figure 2 *shows a general scheme of the proposed procedure for converting text into an image.*

Figure 2. Text to image conversion scheme.

Any text message in UTF-8 format can be converted with this procedure. This is because Formula (1) reduces the characters from this encoding to the pixel value range, i.e., the range 0–255. For the next step, which is image encryption, the proposed *png* image file extension is the best choice. This is due to the fact that this graphic file format has an RGB structure, which allows the use of a very wide range of encryption algorithms known from the literature. Moreover, thanks to the compression algorithms built into this format, simply storing the text in the form of a *png* file usually saves disk space. However, when choosing a format other than *png*, graphic formats with built-in lossy compression (such as *jpg*) must be avoided. In this case, it will be impossible to recover the plaintext, as the compression algorithm will modify its content.

The above algorithm allows the conversion of a text message into exactly one image. There may be times when the resulting image is too large in dimensions. In this case, the text can be divided into several parts, and the image conversion algorithm can be applied to each of them separately.

The proposed procedure of text-to-image conversion is fully reversible. In addition, the added random values in step 5 of the text-to-image algorithm are ignored in the image-to-text conversion. They only play the role of padding to achieve the fixed file size.

To receive a text message from a *png* file, follow these steps:

1. Successive image pixels are loaded in the buffer until the value of one of the components is equal to 3.
2. Loaded RGB pixel components values are stored in one- dimensional table T.
3. Every two values from T are retrieved and marked as ai and bi. These values are the coefficients of the number ti written in the number system with base 256, i.e.,

$$t_i = a_i \cdot 256 + b_i. \quad (4)$$

4. The ti values are stored as consecutive characters in UTF-8 encoding in the text message.

Figure 3 shows a general scheme of the proposed procedure for converting images into text.

Figure 3. Image to text conversion scheme.

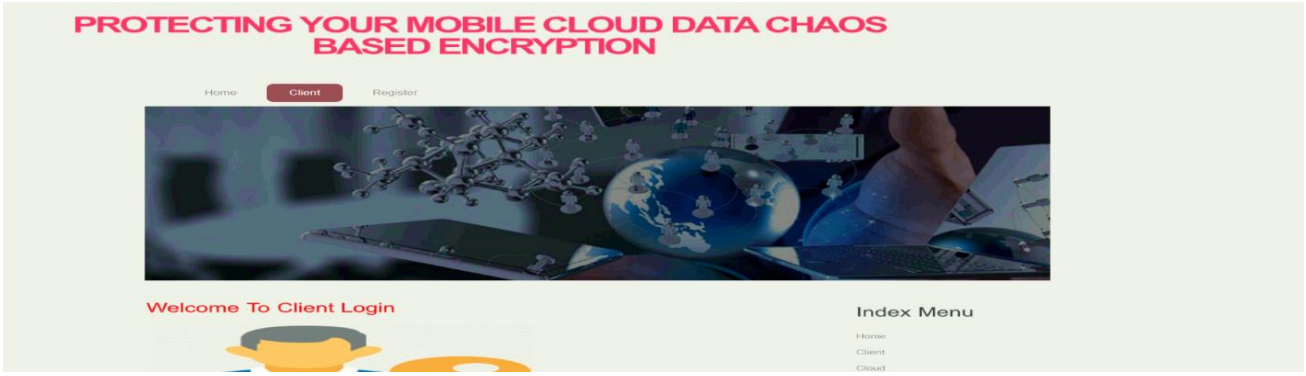
Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024

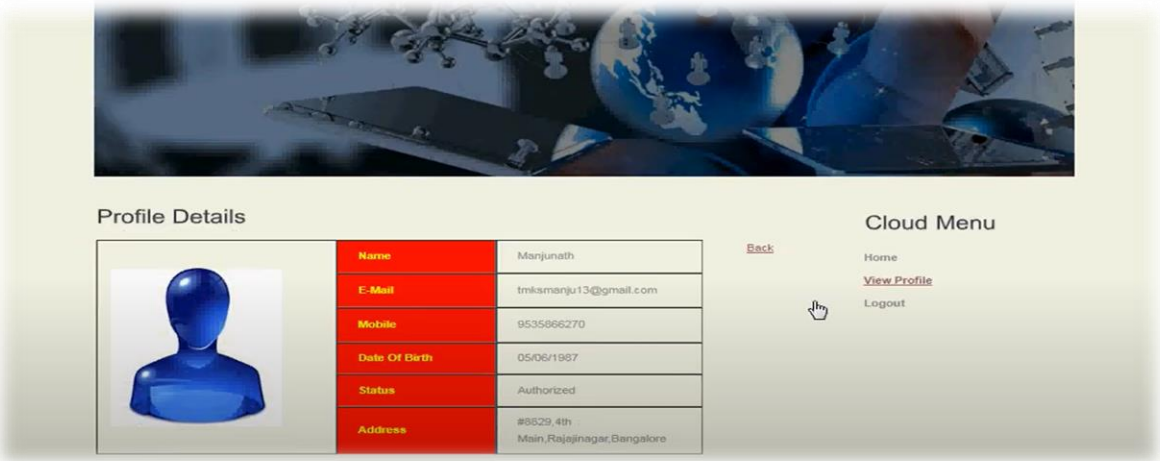
The above method of decoding an image into text is correct and allows the conversion of any RGB image to text in UTF-8 encoding. This is because the pixel range, i.e., 0–255, is extended to the UTF-8 encoding range, i.e., the value 0–65,535, by the Formula (4).

4.7. Image-Encryption Algorithm

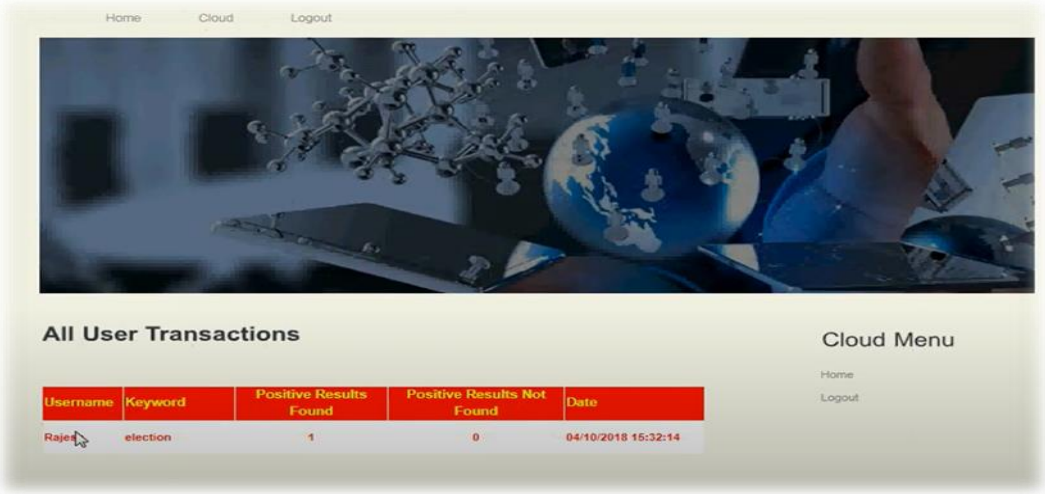
This step requires selecting an image-encryption algorithm known from the literature. It is not important which algorithm is chosen because the text is in the form of an image and can be encrypted with any algorithm. However, it should be noted that the algorithm has no image size limitations. If they are, the text should be divided into enough parts so that the size of each of the obtained images does not exceed the assumed sizes in the algorithm.

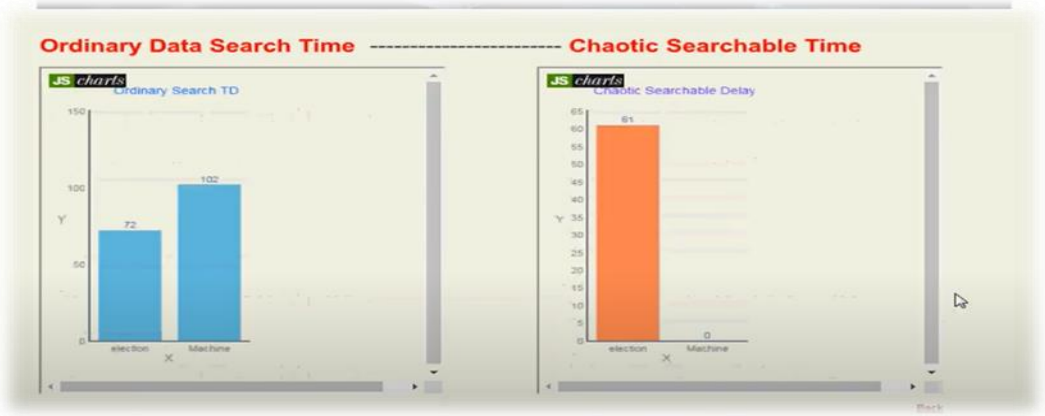
5. RESULT

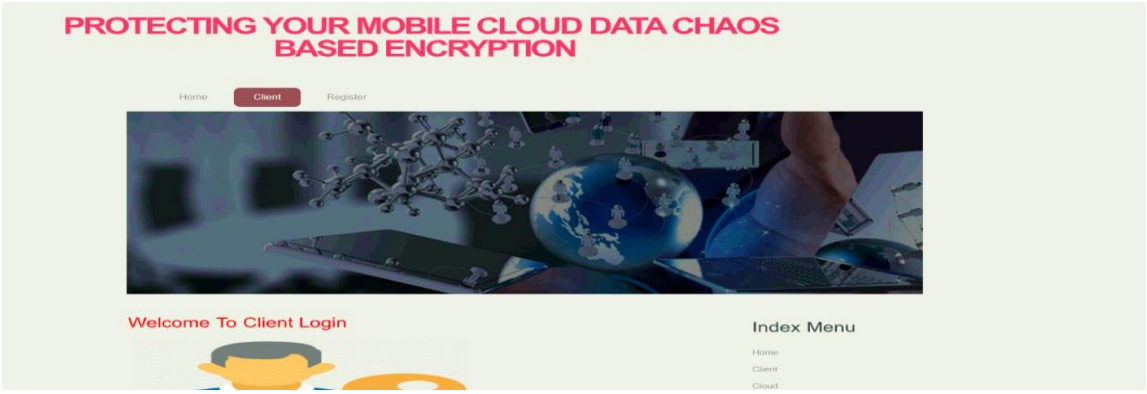




Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024







6. CONCLUSION

In this paper, we proposed the first chaos based searchable encryption approach which also allows both ranked and fuzzy keyword searches on the encrypted data stored in the cloud. Our approach guarantees the privacy and confidentiality of the user even vis-à-vis the cloud provider who is semi-trusted in our case. The proposed method is designed to achieve effective retrieval of remotely stored encrypted data for mobile cloud computing scenarios. This scheme is implemented and evaluated using two databases: RFCs and the Enron database. Comprehensive tests have been performed to prove the efficiency of our proposition. First, the chaotic locality sensitive hashing method with 0‰ failure is selected. Then, effects of different parameters of the amplification method (AND-OR construction) and the chaos, on the efficiency of the algorithm, are shown when different numbers of files are requested. The algorithm is also tested when different kind of errors (deletions, insertions, permutations and substitutions) occur

in the query and similar precision, recall and retrieved ratio curves are obtained. Our proposed algorithm supports the search with only one keyword and an extension of the proposed algorithm to enable conjunctive and disjunctive multi-keywords search, will be considered in the future work.

7. REFERENCES

[1] B.Yang, X. Pang, Q. Du, and Dan Xie, "Effective Error-Tolerant Keyword Search for Secure Cloud Computing," Journal of computer science and technology, vol. 29, no.1, pp. 81-89, Jan. 2014.
[2] D. Boneh, G. D. Crescenzo, "Public key encryption with keyword search," in C. Cachin and J. Camenisch, editors, Advances in Cryptology, Eurocrypt, vol. 3027 of LNCS, pp. 506–522, Springer, 2004.
[3] S. Kamara, K. Lauter, "Cryptographic cloud storage, " in Financial Cryptography and Data Security, pp. 136-149, Springer Berlin Heidelberg, 2010.
[4] S. Kamara, C. Papamanthou, T. Roeder, "CS2: A searchable cryptographic cloud storage system," Microsoft Research, Tech. Report MSR-TR, 2011.
[5] Y. Earn, R. Alsaqour, M. Abdelhaq, T. Abdullah, "Searchable symmetric encryption: review and evaluation," Journal of Theoretical and Applied Information Technology, vol. 30, 2011.
[6] R. Koletka, A. Hutchison, "An architecture for secure searchable cloud storage," IEEE, Information Security South Africa (ISSA), pp. 15-17, Aug., 2011.
[7] E. Stefanov, C. Papamanthou, E. Shi, "Practical Dynamic Searchable Encryption with Small Leakage," IACR Cryptology ePrint Archive, 2013. [8] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," INFOCOM, 2010 Proceedings IEEE, Dept. of ECE, Illinois Inst. of Technol., Chicago, IL, USA , Mar. 2010.
[9] J. Bringer, H. Chabanne, B. Kindarji, "Error-tolerant searchable encryption," Communication and Information Systems Security Symposium, International Conference on Communications (ICC), Dresden, Germany, pp. 14-18, Jun. 2009.
[10] J. Yu, J. Li, X. Wang, W. Gao, "Conjunctive Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," TELKOMNIKA Indonesian Journal of Electrical Engineering, vol.12, no.3, pp. 2104-2109, Mar. 2014.