Secure and Dynamic Multi-Keyword Ranked Search

M. Ambika, SSN College of Engineering, India
Mangayarkarasi N., SSN College of Engineering, India
Raghuraman Gopalsamy, SSN College of Engineering, India
L. Sai Ramesh, Anna University, India
Kamalanathan Selvakumar, NIT Trichy, India

ABSTRACT

Information storing in third party storage is increased. Outsourcing the data to other storage devices or servers may bring question to the secure environment. However, sensitive data like medical information should need privacy when it is stored in cloud storage. In this paper, a secure keyword search that provides the resultant data in an encrypted form where the end user can decrypt using the key given to them is shown. It uses blowfish to encrypt the data, and it also supports the data owner to delete or modify the content of their document. It also ensures accurate relevance score calculation between encrypted index and query vectors.

KEYWORDS
Blowfish, Decryption, Encryption, K-Gram, Searching, Secure

INTRODUCTION

Encrypting File System (EFS) is a feature of Windows that can be used to store information on hard disk in an encrypted format. Encryption is the strongest protection that Windows provides to help the use to keep the information secure. Encryption is the process of translating plain text data (plaintext) into something that appears to be random and meaningless (cipher text). To encrypt more than a small amount of data, symmetric encryption is used. A symmetric key is used during both the encryption and decryption processes. The goal of every encryption algorithm is to make it as difficult as possible to decrypt the generated cipher text without using the key. For a good encryption algorithm, the longer the key, the more difficult it is to decrypt a piece of cipher text without possessing the key. It is difficult to determine the quality of an encryption algorithm. Algorithms that look promising sometimes turn out to be very easy to break, given the proper attack. When selecting an encryption algorithm, it is a good idea to choose one that has been in use for several years and has successfully resisted all attacks. Decryption is the process of converting cipher text back to plain text. A symmetric key is used for both the encryption and decryption processes. To decrypt a particular piece of cipher text, the key that was used to encrypt the data must be used.

Recently, there has been interest in the problem of searching on encrypted data. Consider a user with limited resources storing the data on remote and untrusted storage servers. To preserve confidentiality, it is desirable to store the data in encrypted form. However, encryption makes it hard
to retrieve data selectively from the server. In other words, when a user wishes to only retrieve specific content (e.g., containing certain words), it is hard to let the server perform the search for desired data without any loss of data confidentiality. For providing secure searchable encryption, most of these schemes encrypt the content of data and attach related keywords to it in encrypted form.

Large number of data owners have moved their data into servers for their convenience of multiprocessing, accessing the data from anywhere, reducing their work. The main objective of this project is to secure the sensitive data by encrypting it before storing into the server. Currently, the user search for the data used as the keyword-based search. Here, the proposed method is secured multi keyword ranked search and avoiding the duplicate data from the encrypted data. When uploading the files into the server data owner create the keyword sets for the uploaded file. While searching that file using a keyword sets, user construct a special index structure and retrieve maximum files based on the generated keywords.

RELATED WORKS

P. Xu et al. (2013) proposed a novel concept called the Public-key Encryption with Fuzzy Keyword Search (PEFKS). It allows a third party knowing the search trapdoor of a keyword to search encrypted documents containing that keyword without decrypting the documents or knowing the keyword. M. Kuzu et al. (2012) proposed an efficient scheme for the similarity search over encrypted data. Encrypted storage protects the data against illegal access. To mitigate the concerns, sensitive data is usually outsourced in encrypted form which prevents unauthorized access.

U. Draisbach et al. (2012) proposed a system which focused on Duplicate Count Strategy (DCS) a variation of SNM that uses a varying window size. Ramesh et al. (2015) provides a secure searching of data for personalized recommendation systems for user search. D. Wagner et al. (2013) proposed the cryptographic schemes that enable searching on encrypted data without leaking any information to the untrusted server. The techniques also support query isolation, meaning that the untrusted server learns nothing more than the search result about the plaintext.

T. Papenbrock et al. (2015) proposed two novel progressive duplicate detection algorithms that significantly increase the efficiency of finding duplicates if the execution time is limited. Duplication removal is the main strategy of this work which partitioned the data into chunking and delete the duplicate files. K. Deng et al. (2015) proposed a system that, the objects in a spatial database are associated with keyword(s) to indicate their services. The great advantage of this work is to investigate the search based on the closest keywords.

M. Wallace et al. (2004) proposed the incremental update of fuzzy binary relations, when focusing on both storage and computational complexity issues. K. Elmagarmid et al. (2007) proposed a thorough analysis of the literature on duplicate record detection. The existing tools are concluded with the coverage of a brief discussion of the big open problems in the area.

Z. Bao et al. (2010) proposed an IR-style approach which basically utilizes the statistics of underlying XML data. They first proposed specific guidelines that a search engine should meet in both search intention identification and relevance oriented ranking for search results.

This proposed system provides the efficient search system for searching the files from the server using multi-keyword. It has a server to generate the fuzzy keyword set from the file name. Here, the fuzzy keyword set is used to create all the possible misspell keywords. Search keyword will be encrypted and it will be checked with the collection of original encrypted file name in the server. If the keyword gets matched then the fuzzy keyword set is connected with that particular keyword. So that any search for the file is based on the list of fuzzy keywords rather than fetching the files directly from the server and here the searching performance is also considered. Like how much time it will take to complete the task and how many files it will retrieve.
PROPOSED SYSTEM

Figure 1 shows the architecture of the system. In user interface, the data owner will upload the files. The uploaded file has been stored in database. The uploaded files is encrypted using the blowfish algorithm. Then the file name and keyword will be extracted. After extraction, the fuzzy search will be processed to retrieve the files. For that, the key set will be generated. Then the three types of searching process will be held by the user. The user has to search the file by giving the keyword. Then the user can use anyone of the three types of searching process to retrieve the file like kgram, semantic and linear searching process.

The proposed system consists of four modules namely user interface, admin upload encrypt data, keyword set generation and file search. User interface is the front end which will have the basic login for the data owners and the data users with their respective access like File upload, File search and Report generation.

Key management is the management of cryptographic keys in a cryptosystem. This includes dealing with the generation, exchange, storage, use, and replacement of keys. It includes cryptographic protocol design, key servers, user procedures, and other relevant protocols. Generating keys manually or enabling or disabling the generation of keys are tasks that require you to recycle the node agents and application servers to accept the new keys. If any of the node agents are down, run a manual file synchronization utility from the node agent machine to synchronize the security configuration from the deployment manager.

Data Owner

Figure 2 shows the data owner process. The data owner is responsible for updating the files. The update operation will be like inserting and deleting the files. While updating, the data owner generates the update information locally and sends it to the server. Data owner has a collection of documents which the user wants to outsource to the server in encrypted format, while still keeping the capability to search on them for effective utilization. Data owner will first upload the file. The uploaded file will be stored in database in encrypted format. Afterwards, the data owner outsources the encrypted
collection and the secure index to the server, and securely distributes the key information of trapdoor generation and document decryption to the authorized data users.

**File Search**

This section has the logic to create an encrypted keyword set based on the kgram, linear and semantic algorithms. It has the file search and file retrieval logics based on the different keyword set generated. The report generation logics in this section to understand the performance based on various conditions during file upload, keyword set creation, file search for the correct keyword search and the typing errors/representation or inconsistent keyword search. Figure 3 shows the different searching process of the user.

**Figure 2. Data Owner Process**

[Diagram of Data Owner Process]

**Figure 3. File Search**

[Diagram of File Search]
K-Gram Search

This section has the kgram1 searching process. kgram1 process of searching a file by using the keyword is very essential searching process for retrieving a file from the server. In this kgram1 searching, the user have to search the file by using the keyword, which can be given by leaving single character. The authorized user will enter the keyword to search a file, only for the valid keyword the file will be displayed. Depending upon the number of visit count the security will be increased. To download the particular file from the server the user has to give the secret key, which has been send to the user’s registered e-mail ID from the admin. Figure 4 shows the kgram1 searching process. Likewise, kgram2 process also done until validation occur.

Figure 4. K-Gram search Process

Semantic search seeks to improve search accuracy by understanding the searcher’s intent and the contextual meaning of terms as they appear in the searchable dataspace, whether on the web or within a closed system, to generate more relevant results. Semantic search systems consider various points including context of search, location, intent, variation of words, synonyms, generalized and specialized queries, concept matching and natural language queries to provide relevant search results.

IMPLEMENTATION

In algorithm design, there are two basic ways to ensure that the key is long enough to ensure a particular security level. One is to carefully design the algorithm so that the entire entropy of the key is preserved, so there is no better way to crypt analyze the algorithm other than brute force. The other is to design the algorithm with so many key bits that attacks that reduce the effective key length by several bits are irrelevant. Blowfish is the algorithm used for encryption which undergoes 522 iterations to test the key which is effective among the cyber attacks.

Results

The experiments are carries out with cloud set up using Cloudsim. Data owner uploaded the file in the data storage in decrypted form and user search the data by using the semantic search. The uploaded file will be stored in the database. The key set will be generated for the uploaded file, and then it will be stored in database in a separate table.

Figure 5 shows the encrypted files stored in the database. In this process of encrypting a file, first create a key from a given byte array for a given algorithm. Then get an instance of Cipher class for a given algorithm transformation. Initialize the Cipher with an appropriate mode (encrypt or decrypt) and the given Key. Invoke doFinal(input bytes) method of the Cipher class to perform encryption or decryption on the input bytes, which returns an encrypted or decrypted byte array. Read an input file to a byte array and write the encrypted and decrypted byte array to an output file accordingly.
In kgram1 searching process the user can search the file by leaving a single character in the keyword. In Figure 6, it shows the keyword given to search the file, by leaving a single character. It also shows the particular file, which has been searched by the user from the server. It shows the count of files which is available in the server and along with that it shows the time taken to search the file. The user will enter the keyword, only for the valid keyword the file will be displayed. Depending upon the number of visit count the security will be increased for the file.

The kgram2 searching process is the process of searching the file by leaving the double character in the keyword. In Figure 7, it shows the keyword given to search the file, by leaving the double character. It also shows the particular file which has been searched by the user from the server. It shows the count of files which is available in the server and along with that it shows the time taken to search the file. The user will enter the keyword, only for the valid keyword the file will be displayed. Depending upon the number of visit count the security will be increased for the file.

The secret key which is send by the admin to the authorized user. The user has to enter the secret key to download the file. The secret key which is send to the user's registered e-mail ID from the admin. The authorized user only can use the secret key to retrieve the file. Whenever the visiting count is increased in kgram1 and kgram2 search, it will ask for a secret key. The linear searching process generally needs the secret key to retrieve the file. The semantic searching process also needs the secret key for retrieving the file. In the semantic search the searching process will be processed by using the
meaning of the file name to search and retrieve the file. In Figure 8, it shows the keyword given by
the user. It contains the number of files and along with that it shows the time taken to search the file.

Finally the accuracy of the K-Gram search results is compared and analyzed with the existing
works. Table 1 shows the response time and memory consumed for search based on the number of
keyword given for search. The accuracy is calculated and compared with existing work of Deng et
al (2015).

The accuracy of the proposed system increasing while the number of keywords are increased.
So, it can also consumes less memory when we go for more keyword search.

| No. of Query keyword | Response Time (ms) | Memory Consumed (MB) | Accuracy (%) |
|----------------------|--------------------|----------------------|--------------|
|                      | Deng et al. | Proposed | Deng et al. | Proposed | Deng et al. | Proposed |
| 1                    | 25         | 15        | 0.40        | 0.42     | 98.0        | 94.0       |
| 2                    | 75         | 22        | 0.45        | 0.43     | 96.5        | 93.8       |
| 3                    | 185        | 35        | 0.48        | 0.44     | 91.7        | 92.6       |
| 4                    | 280        | 48        | 0.50        | 0.46     | 87.5        | 91.2       |
| 5                    | 350        | 54        | 0.52        | 0.48     | 83.5        | 90.7       |
CONCLUSION AND FUTURE WORK

This project proposed the keyword search over encrypted data, and constructs the security requirement. The secure, efficient and dynamic search scheme is proposed, which supports not only the accurate keyword search but also the dynamic deletion and insertion of documents. It also performs the duplicate file detection process in this project. The efficient Blowfish algorithm is used to encrypt the data. The data owner is responsible for uploading the file and sending them to the server. The authorized user will perform different searching process to search and retrieve the file. The user will search the file by using the keyword. The owner will send the secret key to the user’s registered e-mail ID, and by giving the appropriate secret key the user can retrieve the file. Finally, the secure keyword searching process has performed well for the authorized user’s. In future, the secured keyword searching over encrypted data can be performed in the cloud environment. So that the process of keyword search over encrypted cloud data can be implemented with a variety of security requirements. In this proposed scheme, the data owner is responsible for updating information and sending them to the cloud server.
REFERENCES

Bao, Z., Lu, J., Ling, T. W., & Chen, B. (2010). Towards an effective XML keyword search. *IEEE Transactions on Knowledge and Data Engineering, 22*(8), 1077–1092. doi:10.1109/TKDE.2010.63

Christen, P. (2012). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Transactions on Knowledge and Data Engineering, 24*(9), 1537–1555. doi:10.1109/TKDE.2011.127

Deng, K., Li, X., Lu, J., & Zhou, X. (2015). Best keyword cover search. *IEEE Transactions on Knowledge and Data Engineering, 27*(1), 61–73. doi:10.1109/TKDE.2014.2324897

Draisbach, U., Naumann, F., Szott, S., & Wonneberg, O. (2012, April). Adaptive windows for duplicate detection. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on* (pp. 1073-1083). IEEE. doi:10.1109/ICDE.2012.20

Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2007). Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering, 19*(1), 1–16. doi:10.1109/TKDE.2007.250581

Kuzu, M., Islam, M. S., & Kantarcioglu, M. (2012, April). Efficient similarity search over encrypted data. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on* (pp. 1156-1167). IEEE. doi:10.1109/ICDE.2012.23

Papenbrock, T., Heise, A., & Naumann, F. (2015). Progressive duplicate detection. *IEEE Transactions on Knowledge and Data Engineering, 27*(5), 1316–1329. doi:10.1109/TKDE.2014.2359666

Ramesh, L. S., Ganapathy, S., Bhuvaneshwari, R., Kulothungan, K., Pandiyaraju, V., & Kannan, A. (2015). Prediction of user interests for providing relevant information using relevance feedback and re-ranking. *International Journal of Intelligent Information Technologies, 11*(4), 55–71. doi:10.4018/IJIIIT.2015100104

Wallace, M., & Kollias, S. (2004, July). Computationally efficient incremental transitive closure of sparse fuzzy binary relations. In *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on* (Vol. 3, pp. 1561-1565). IEEE. doi:10.1109/FUZZY.2004.1375408

Whang, S. E., Marmaros, D., & Garcia-Molina, H. (2013). Pay-as-you-go entity resolution. *IEEE Transactions on Knowledge and Data Engineering, 25*(5), 1111–1124. doi:10.1109/TKDE.2012.43

Xu, P., Jin, H., Wu, Q., & Wang, W. (2013). Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack. *IEEE Transactions on Computers, 62*(11), 2266–2277. doi:10.1109/TC.2012.215
M. Ambika completed her Ph.D in Faculty of Information and Communication Engineering, from Anna University, Chennai, Tamil Nadu, India. She graduated as a B.E (CSE) from Bharathidasan University, Thiruchirappalli and as an M.E (CSE) from Anna University Thiruchirappalli, being placed in first class with distinction. Her research interest includes Data Mining, Web Mining, and Semantic web. She got 6 years of teaching experience. She has published 5 articles in International Journals, 6 in the proceedings of International Conferences and 11 in the proceedings of national conferences /seminars. Ms. Ambika is a lifetime member of The Indian Society for Technical Education and lifetime member of International Association for Engineers. Research Scholar Associate Professor Currently working as Visiting Faculty in CEG Campus, Anna University, India. His current area of research is Data mining, Security in WSN’s, Cloud Database and Personalized Image Retrieval.

Selvakumar K. is currently working as an Assistant Professor in National Institute of Technology, Trichy. He has completed Ph.D., Degree from Anna University, Chennai. He has completed his Master of Engineering in Computer Science from Anna University, Chennai. Bachelor of engineering from University of Madras. His field of specialization is Mobile Ad-hoc Networking with soft computing approaches. He is also interested in Internetworking Technologies, Network Protocols and Design, Group Communication, Distributed Computing, and Web Mining. He has published articles in International and National Journals and Conferences.

Ms. N. Mangayarkarasi is currently pursuing her Ph. D in the Faculty of Information and Communication in Anna University. She completed her B. E and M. E in the specialization of Computer Science and Engineering. She has published research articles in international conferences and reputed journals. Her area of interests are Image Processing and Information Retrieval. She is a Life member of Indian Society for Technical Education (ISTE).

Dr. G. Raghuraman, Associate Professor in the Department of Computer Science and Engineering has 15 years of teaching including 4.5 years of research experience in Image Processing. He received his BE and M.E(First Class with Distinction) degrees from Madurai Kamaraj University and Ph.D in Content Based Image Retrieval from Anna University. He has published over 20 research publications in referred journals and international conferences. His research interests are Image Processing, Multi Agent Systems and Cloud computing. He has guided Undergraduate and Post Graduate students in their projects. He is a Life member of Computer Society of India (CSI) and Indian Society for Technical Education (ISTE).

Dr. L. SaiRamesh is currently working as a Faculty in Department of Information Science and Technology, Anna University, Chennai, India. He completed his M.E. in CSE in 2007 and Ph.D. in 2015 from Anna University, Chennai, India. He has thirteen years of teaching and research experience. He published more than 50 research articles in reputed journal and international conferences. His are of research is Wireless Networks, Cloud computing and Machine Learning.