ISSN (e): 2250 – 3005 || Volume, 08 || Issue, 9|| September – 2018 ||
International Journal of Computational Engineering Research (IJCER)

Fast Efficient Keyword Search (FEKS) For Cloud Data Search and Retrieval

Rohan Kindalkar[1], Prasanna Shete[2]
[1]MTech Scholar, Department of Computer Engineering, K.J. Somaiya College of Engineering (KJSCE), Vidyavihar-400077, Maharashtra, India.
[2]Associate Professor, Department of Computer Engineering, K.J. Somaiya College of Engineering (KJSCE), Vidyavihar-400077, Maharashtra, India.
Corresponding Author:Rohan Kindalkar

ABSTRACT:Cloud has become the most popular way of storing the data in the cloud due to huge expansion of Internet. Since the data may contain confidential information, it is encrypted before keeping it in the cloud. But, the user experience is reduced drastically if the user needs to search over the cloud data which is encrypted. In this paper, a framework known as Fast Efficient Keyword Search (FEKS) is developed. It is used for the secure cloud storage and retrieval that enables the searching of encrypted data via keywords. The first part of FEKS uses the Boolean logic operations such as AND, OR and NOT to retrieve relevant and efficient results. The second part uses the n-gram search to return faster and precise results.
KEYWORDS: Cloud Computing, Searchable Encryption, Fast Efficient Keyword Search (FEKS), N-gram Search.

Date of Submission: 06-09-2018 Date of acceptance: 22-09-2018

I. INTRODUCTION

Cloud is broadly utilized all over the world by people and organizations. In Cloud computing, data is stored in the cloud using the centralized server which is handled by the cloud provider. This makes the data less secure. So, the data/information is encrypted to make it secure. Due to the importance of cloud models and storage facilities, the count of users is increasing exponentially. Since, increase in users directly equates to the increase in storage of bulk data, the search and return of particular file becomes difficult. As the data is encrypted in cloud, the search also needs to happen in encrypted form only. To meet the required needs of encrypted searching, a single query or multi-keyword query can be used to get the results to the user. There are numerous searching techniques available which are:

- Searchable encryption [1] helps the users to search the encrypted data in cloud using single or multiple keywords. The symmetric and secret key is used to return the documents to the user.
- Fuzzy Keyword Search [2] is used to return the documents when the keyword input is same as the predefined keywords.
- Ranked Keyword Search [3] is used to return the ranked results according to the frequency of keywords in the documents.

But these types of searching techniques do not help to attain the user experience as Google search. The two main reasons are the inefficiency and slower response time. Also, the keywords used for the documents are not fully secure from attacks. To address these issues, a framework is proposed which is known as Fast Efficient Keyword Search (FEKS). Boolean operations and n-gram search are used in FEKS over encrypted data. The objective of this work is to construct a keyword searching scheme for storage and retrieval of encrypted data in the private cloud. The remainder of this paper is organized as follows. In Section II, we studied the findings of the literature survey. We present the proposed methodology in Section III. Section IV are the results analysed and Section V concludes the paper.

II. LITERATURE SURVEY

Searchable encryption includes a trusting data owner, a semi-trusting server and a search user. The various tasks of each of modules are as follows:

- Data Owner: The data owner browses the document from the computer to upload to the cloud server. The keywords are assigned to the corresponding document and both the keywords and the document are encrypted. The encrypted document is then uploaded to the cloud.

- Search User: The search user writes a keyword query and generates a trapdoor to avoid the deduction of keywords while sending the query to the cloud server. Then, the documents are retrieved from the cloud server.
- Cloud Server: The cloud server stores the encrypted documents. When it receives the trapdoors from search user, it searches over the encrypted index and then the related documents are returned to the search user.

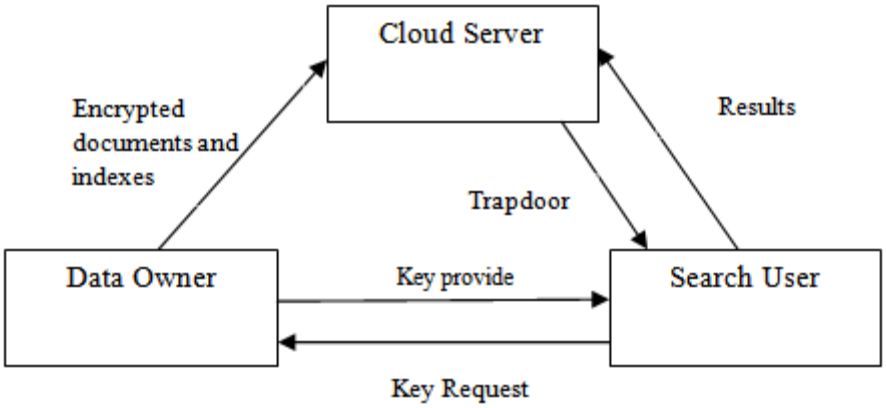

Figure 1: Searchable Encryption [1]

The security requirements of the searchable encryption are as follows:

- Data Confidentiality: Here, if keywords which are searched matches the keywords assigned to the documents, then the document is sent to the search user. So, the documents which are outsourced must not be identifiable to preserve the confidentiality.
- Protecting the privacy of index and trapdoor: Based on the assigned keywords by data owner and search keywords, the index and trapdoors are created respectively. If the server is able to identify the index or trapdoor and then is further able to associate the relation between the documents and keywords, it may be able to learn the context of document. So, the privacy of index and trapdoor should be protected.
- Trapdoor unlinkability: The search user may search a particular document using the keywords multiple times. The server must not be able to deduce any kind of relationship using the search keywords. So, the generation of trapdoor should be random rather than sequential. Even if the set of keywords are same, the trapdoor generated must be different every time.

To ensure the retrieving the documents using keywords, various keyword searching techniques are used. Cao et al. [4] proposed a single keyword search and co-ordinate matching which shows as many results as possible to capture the importance of data to the search query.Sun et al. [5] tells the notion of privacy preserving multi-keyword text search using similarity based ranking. It has proposed to build frequency-based search index which uses vector space model with cosine similarity to obtain high accuracy on search result. Örencik et al. [6] proposes a method utilizing hash functions and using multi-keyword search in a single query. The writers have formulated a technique to fulfill security semantic definition by combining an effective ranking scheme based on term frequency-inverse document frequency (tf-idf) values of keyword document pairs. Zhang et al. [7] proposes a scheme which uses conjunctive keyword or a technique which uses 'AND' operation for searching. It returns the documents which is matching all the query keywords with the document keywords. Li et al. [8] uses a multi-keyword search known as FMS which consists of two frameworks: FMS-I and FMS-II. FMS-I uses the relevance score and preference factors of keywords while FMS-II uses the Boolean operations such as AND, OR and NOT together on keywords.

III. PROPOSED SYSTEM

The various searching schemes proposed depend on the weight of the keywords to retrieve the results. The weight of the keywords is the frequency of the keyword appearing in the document. Due to the keyword weights, the various file formats which can be stored in the cloud becomes limited. To eliminate the dependency of the keyword weights and to make it faster, a framework known as Fast Efficient Keyword Search (FEKS) is proposed. The FEKS has two types of frameworks. First framework uses the Boolean operations for returning the efficient results. Second type consists of the n-gram search for retrieving the results in a faster and accurate manner. An n-gram is a continuous chronology of ‘n’ items from a provided text sample. They are a sub-string of the words. Using n-gram, it overcomes the limitations of logic operations search technique by retrieving

faster and accurate results to the search controller. To accomplish the security requirements of FEKS, it can be achieved by taking care of:

- Data must be encrypted before uploading them.
- The index which is to be searched must also be encrypted.
- The keywords which the user uses to search the documents must also be encrypted before searching it.
- A pre-existing trust between a particular data client and search controller before sharing the keywords and details of data between them.

The properties of the proposed system are as follows:

- The proposed scheme introduce low overhead on computation and communication cost.
- To support extra search semantics and dynamic data operations, it uses Boolean search mechanism.
- It is more secure and efficient mechanism.
- It helps to prevent the association of index with the files and from finding the connection between the search query entered for searching and retrieving files.

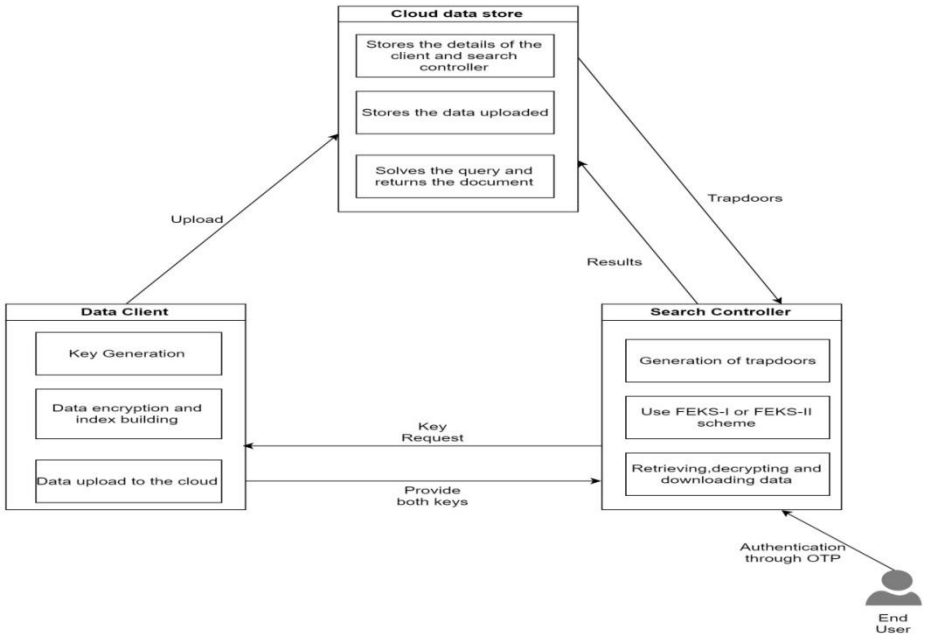

Figure 2: FEKS Architecture

The FEKS framework consists of three modules:

Module 1: Data Client

The data client needs to generate the symmetric key and secret key to send to authorized search controller for securely searching the data from data store. The data which can be encrypted includes the documents (.doc/.txt) and image files (.jpg/.png). The keywords assigned to the data are encrypted using secret key and stored as index. Then, the encrypted data is sent to the cloud data store.

Module 2: Search Controller

The controller searches the data using the particular keywords assigned to the required data by the data client. The trapdoors of the keywords are generated to secure them from the threats. The Boolean operations such as AND, OR and NOT is used in FEKS-I. For FEKS-II, n-gram search is used to retrieve precise results. Then the trapdoor is sent to the cloud data store. If the result is returned by the data store, the controller can decrypt and download the data.

Module 3: Cloud Data Store

The data store stores the details of the data client and search controller. A 4-digit OTP is sent to their registered email-ids for authentication of the user. It also stores the encrypted data, indexes uploaded by data client and trapdoors sent by the search controller. The data store solves the query which is sent by the search controller to return the accurate results.

The FEKS consists of two types of frameworks for keyword searching:

FEKS-I

As shown in Figure 3, the first scheme uses logic or Boolean operations such as AND, OR and NOT. For AND operation, all the keywords should be entered by the search controller to retrieve the documents in the proper order. If one of the keywords is wrong or not written, the query fails. For OR operation, if search

controller enters at least one of the keyword matching the assigned keywords, the documents containing the matching keywords will be returned. For NOT operation, the documents containing the keywords entered by the search controller will not be returned. The remaining documents in the collection will be returned.

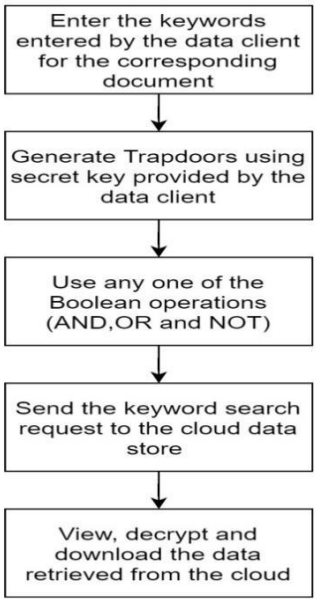

Figure 3: FEKS-I scheme

FEKS-II

An n-gram [9] is a continuous chronology of 'n' items from a provided text sample. They are a sub-string of the words. An n-gram of length 1 is a uni-gram, length 2 is bi-gram and length 3 is tri-gram. In FEKS-II, using n-gram technique, it overcomes the limitations of FEKS-I scheme by retrieving faster and accurate results to the search controller. It performs the AND, OR operation without the need to select it.

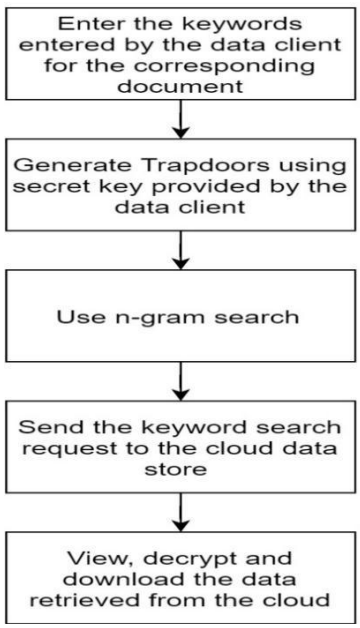

Figure 4: FEKS-II scheme

The system and the data are made secured by implementing the following things:

- Key Encryption- The keys generated by the data client are the symmetric key and the secret key. The symmetric key is needed to encrypt the data with AES-256 algorithm. The secret key is used for building the indexes of keywords at data client side and generation of trapdoors at search controller side. The symmetric key is generated using kNN and secret key is generated using random number function.

- Data Encryption- The data is encrypted using symmetric key of the data client and AES-256 algorithm to convert it into ciphertext. The same symmetric key is needed for document decryption.
- OTP Generator- One Time Password Generator (OTP) acts as an extra security for the system. OTP is used to provide more security to the system. The advantage of the OTP Generator is that each time a new password is generated due to this extra security added to the system. In our model, a 4-digit OTP is generated by the Cloud Server and sent to registered email-id of the user. Then OTP Generator is used at the time of login.
- Keyword Encryption- The keywords are either assigned to the corresponding documents at the data client side or used by the search controller to retrieve the documents. For both the processes, the keywords need to be encrypted for secure computation. At the data client side, the keywords are encrypted using the secret key generated using random number function. The keywords at the search controller side are encrypted as trapdoors using the secret key of the data client so that while sending the keywords to the cloud data store, the keywords cannot be deduced by other unauthorized users.
- Image Encryption- The image in Joint Photographic Experts Group (JPEG) or Portable Network Graphics (PNG) format can be encrypted by converting it into binary image by increasing the threshold of the image. Then, the image can be uploaded to the cloud.

IV. RESULTS

In FEKS, the weight of the keywords is not considered. The weights can be used to retrieve the document from the cloud. Consider the dataset consisting of 10 documents which have keywords assigned to it and stored in the cloud by the data client. The detailed description of the dataset is shown in Table 1.

No	Name of the document	Set of Keywords	Weights (Case-II)
1	attacks.txt	(passive, active, attack)	(10, 8, 9)
2	backdoor.txt	(program, access)	(4, 4)
3	ddos.txt	(resource, attack)	(5, 22)
4	kerberos.txt	(user, service)	(12, 4)
5	standards.txt	(internet, industry)	(22, 2)
6	deadlockprevention.txt	(calls, resource)	(2, 5)
7	osdesign.txt	(system, design)	(13, 5)
8	systemcalls.txt	(calls, file)	(15, 20)
9	ipsec.txt	(key, internet, facility)	(5, 8, 2)
10	linkencryption.txt	(communications, link)	(3, 13)

Table 1: Detailed Dataset Description

There will be two cases:

CASE-I (Where weights are not considered)

Here the weight is negligible for both the frameworks. For FEKS-I, there are three Boolean operations AND, OR and NOT. It is shown in Table 2.

- In AND operation, only one document is returned to the search controller. The keywords should be present in the keyword set and in the same order assigned by the client in order to be retrieved. If the condition is not followed, then no document is returned. In Set-1, if the user wants to retrieve 'backdoor.txt' file, then the keywords should be 'program, access' and not interchanged or incomplete. The keyword in Set-2 does not return the document since the keyword 'lock' is not assigned to any file in the dataset.
- For OR operation, no order is needed. The documents containing the keywords entered by controller will be retrieved. The keywords (user, service) in Set-3 return 'kerberos.txt' because the client has assigned those keywords to that document. In Set-4, the keywords return two documents because two files are assigned the same keyword 'calls' and thus data store returns two documents.
- For NOT operation, the documents which do not have the keywords entered by search controller will be returned. The keywords (attack, internet) in Set-5 are assigned to four documents by the data client. So, other than those four documents, the six documents will be returned. In Set-6, only the keyword 'passive' is assigned to 'attacks.txt' by the client. So, other than that text file, the remaining documents will be retrieved by the search controller.

Table 2: FEKS-I without weights

FEKS-I			
Logic	**Set**	**Search Keywords**	**Expected Results**
AND	1	(program, access)	backdoor.txt
	2	(service, lock)	No documents returned
OR	3	(user, service)	kerberos.txt
	4	(calls, file)	deadlockprevention.txt systemcalls.txt
NOT	5	(attack, internet)	backdoor.txt kerberos.txt deadlockprevention.txt osdesign.txt systemcalls.txt linkencryption.txt
	6	(passive, dead)	backdoor.txt ddos.txt kerberos.txt standards.txt deadlockprevention.txt osdesign.txt systemcalls.txt ipsec.txt linkencryption.txt

For FEKS-II, n-gram search is used. It gives lesser and precise results than the FEKS-I. Also, searching can be performed without assigning a specific operation. The Set-1 keywords return a single document because of the assignment of those keywords to only one file. In Set-II, since the keyword 'service' is assigned to 'kerberos.txt', it is returned to the controller and does not take into consideration the keyword 'lock' which is not assigned. In Set-3, Set-4 and Set-5, the files are returned according to the keywords assigned by the client. The keyword 'calls' returns two files in Set-4, 'attack' and 'internet' returns two documents each, hence a retrieval of four documents in Set-5. The keyword 'passive' returns 'attacks.txt' and does not take the keyword 'dead' which is not assigned to any file. The results are shown in Table 3.

FEKS-II		
Set	**Search Keywords**	**Expected Results**
1	(program, access)	backdoor.txt
2	(service, lock)	kerberos.txt
3	(user, service)	kerberos.txt
4	(calls, file)	deadlockprevention.txt systemcalls.txt
5	(attack, internet)	attacks.txt ddos.txt standards.txt ipsec.txt
6	(passive, dead)	attacks.txt

Table 3: FEKS-II without weights

CASE-II (where weights of keywords are considered)

The weight of the keyword is the frequency of the keyword appearing in the document.

For FEKS-I (shown in Table 4),

- For AND operation, all the keywords entered should be present in the set otherwise the document will not be returned. If the keyword is not present (the weight is zero), retrieval is not possible. The Set-1 keywords have a non-zero weight and assigned to one file. Hence, it returns a single document. One of the keywords in Set-2 has a zero weight, so no retrieval is possible.
- For OR operation, the document with the highest keyword weight will be returned. If the weights are same, then the first keyword in the set will be preferred and that document will be retrieved. The keywords in Set-3 and Set-4 having maximum weights are 'user' and 'file' respectively. Hence 'kerberos.txt' and 'systemcalls.txt' are retrieved in both the sets.
- For NOT operation, the documents with a non-zero keyword weight specified by the search controller will be returned to the client. The keywords in Set-5 have non-zero weights. So, the keywords assigned to the files by the client will be returned. Since, in Set-6, the keyword 'dead' has a zero weight, only 'attacks.txt' is returned because the keyword 'passive' is assigned to it.

FEKS-I				
Logic	**Set**	**Search Keywords**	**Weights**	**Expected Results**
AND	1	(program, access)	(4, 4)	backdoor.txt
	2	(service, lock)	(4, 0)	No documents returned
OR	3	(user, service)	(12, 4)	kerberos.txt
	4	(calls, file)	{(2, 15), (20)}	systemcalls.txt
NOT	5	(attack, internet)	{(9, 22), (22, 8)}	attacks.txt ddos.txt standards.txt ipsec.txt
	6	(passive, dead)	(10, 0)	attacks.txt

Table 4: FEKS-I with weights

In FEKS-II, since n-gram technique is used and no operations such as AND, OR and NOT are performed, the results will be same as FEKS-II without keyword weights (Table 3).
By analyzing the results, it can be concluded that using keyword weights in FEKS-I, it returns less number of results to the user. But without weights, multiple and relevant results are returned to the client. Also, since weights cannot be calculated for other files such as images and pdf, the search results becomes much more limited.
In FEKS-II, as no logic operations are to be performed, the documents retrieved are same with or without using keyword weights

V. CONCLUSION AND FUTURE SCOPE

In this paper, a multi-keyword search scheme which is a safe, effective and works in dynamic run-time has been proposed and implemented which supports insertion and deletion of documents. Secure kNN computation has been used for symmetric key generation. The symmetric key is needed for data encryption and decryption. For multi-keyword search, two schemes are proposed viz. namely Fast Efficient Keyword Search; FEKS-I and FEKS-II. FEKS-I uses various logic operations i.e. AND, OR and NO operations to enhance the efficiency of document searching. FEKS-II uses n-gram search to reduce the number of operations and faster than FEKS-I. FEKS not only allows searching and retrieval of text documents but also facilitates searching of image files (.jpg/.png). The 4-digit OTP feature is added for extra security to protect against unauthorized access.

In the future, the system could further be improved by storing other file formats including PDF and Office documents in the cloud.

REFERENCES

[1]. Wang, Yunling, Jianfeng Wang, and Xiaofeng Chen. "Secure searchable encryption: a survey." Journal of Communications and Information Networks 1, no. 4 (2016): 52-65.
[2]. Li, Jin, Qian Wang, Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou. "Fuzzy keyword search over encrypted data in cloud computing." In Infocom, 2010 proceedings ieee, pp. 1-5. IEEE, 2010.
[3]. Wang, Cong, Ning Cao, Jin Li, Kui Ren, and Wenjing Lou. "Secure ranked keyword search over encrypted cloud data." In Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on, pp. 253-262. IEEE, 2010.
[4]. Cao, Ning, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou. "Privacy-preserving multi-keyword ranked search over encrypted cloud data." IEEE Transactions on parallel and distributed systems 25, no. 1 (2014): 222-233.
[5]. Sun, Wenhai, Bing Wang, Ning Cao, Ming Li, Wenjing Lou, Y. Thomas Hou, and Hui Li. "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking." In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, pp. 71-82. ACM, 2013.
[6]. Örencik, Cengiz, and Erkay Savaş. "Efficient and secure ranked multi-keyword search on encrypted cloud data." In Proceedings of the 2012 Joint EDBT/ICDT Workshops, pp. 186-195. ACM, 2012.
[7]. Yu, Jiadi, Peng Lu, Yanmin Zhu, Guangtao Xue, and Minglu Li. "Toward secure multikeyword top-k retrieval over encrypted cloud data." IEEE transactions on dependable and secure computing 10, no. 4 (2013): 239-250.
[8]. Li, Hongwei, Yi Yang, Tom H. Luan, Xiaohui Liang, Liang Zhou, and Xuemin Sherman Shen. "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data." IEEE Transactions on Dependable and Secure Computing 13, no. 3 (2016): 312-325.
[9]. Wikipedia contributors, "N-gram," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=N-gram&oldid=835900923 (accessed September 8, 2018).

Rohan Kindalkar "Fast Efficient Keyword Search (FEKS) For Cloud Data Search and Retrieval "International Journal of Computational Engineering Research (IJCER), vol. 08, no. 09, 2018, pp 01-07