PRIVACY-PRESERVING MULTI-KEYWORD TOP-K SIMILARITY SEARCH OVER ENCRYPTED CLOUD DATA

Swati Janrao[1], Prof.S.S.Shaikh[2]

Student, Department of computer Engineering, Sanjivani College of Engineering and Research, Kopargaon, Savitribai Phule Pune University, Maharashtra, India [1]

Professor, Department of computer Engineering, Sanjivani College of Engineering and Research, Kopargaon, Savitribai Phule Pune University, Maharashtra, India [2]

ABSTRACT*:* Cloud computing provides people and enterprises huge computing power and ascendible storage capacities to support a variety of big data applications in domains like health care and scientist research, therefore more and a lot of knowledge homeowner's area unit concerned to source their knowledge on cloud servers for excellent convenience in knowledge management and mining. However, data sets like health records in electronic documents sometimes contain sensitive data, which brings about privacy concerns if the documents are discharged or shared to part entrusted third-parties in cloud. A practical and widely used technique for information privacy preservation is to encode information before outsourcing to the cloud servers, that but reduces the information utility and makes several efficient knowledge analytic operators like keyword-based top-k document retrieval obsolete. In this paper, we investigate the multi-keyword top-k search downside for giant encoding against privacy breaches, and attempt to identify an efficient and secure solution to this problem. Specially, for the privacy concern of query data, we construct a special tree-based index structure and style a random traversal algorithmic program, which makes even the same query to produce completely different visiting ways on the index, and can also maintain the accuracy of queries unchanged under stronger privacy. For improving the query efficiency, we propose a group multi-keyword top-k search theme supported the thought of partition, where a group of tree-based indexes are constructed for all documents. Finally, we combine these methods together into an efficient and secure approach to deal with our projected top-k similarity search. Extensive experimental results on real-life knowledge sets demonstrate that our projected approach will considerably improve the aptitude of defensive the privacy breaches, the measurability and also the time potency of query processing over the state-of-the-art methods.

KEYWORDS: Data mining, insider attack, intrusion detection and protection, system call (SC), users' behaviours.

I. INTRODUCTION

Cloud computing has emerged as a disruptive trend in both IT industries and research communities recently, its salient characteristics like high scalability and pay-as-you-go fashion have enabled cloud consumers to purchase the powerful computing resources as services according to their actual requirements. When the companies and individuals enjoy the advantages of cloud computing, they also need to take the privacy concern of the outsourced data into account. Because data sets in many applications often contain sensitive information like e-mails, electronic health records and financial transaction records, when the data owner outsourcing such sensitive data to the cloud servers which are considered to be partially trusted, the data can be easily accessed and analyzed by cloud service providers illegally. Data encryption has been widely used for data privacy preservation in data sharing scenarios; it refers to mathematical calculation and algorithmic scheme that transform plaintext into cipher-text, which is a non-readable form to unauthorized parties. The keyword-based search is such one widely used data operator in many database and information retrieval applications, and its traditional processing methods cannot be directly applied to encrypted data. Therefore, how to process such queries over encrypted data and at the same time guarantee data privacy becomes a hot research topic. in this paper, we focus on a special type of multi-keyword ranked search, namely the multi-keyword top-*k* search, which has been a very popular database operator in many important applications, and only needs to return the *k* documents with the highest relevance scores. For supporting multi-keyword search, we introduce the vector space model which represents documents and queries as vectors.

II. RELATED WORK

- **W. Zhang, Y. Lin, S. Xiao, J. Wu, and S. Zhou, "Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing", IEEE Transactions on computers, vol. 65, no. 5, May 2016.**

The author describes a system which helps in data retrieval from the cloud. As the data can be sensitive and hence when it gets into the hands of wrong people, it can turn out to be harmful for both the owners and the receivers. Hence to overcome such a situation, encryption and decryption of data can be done for the safe exchange of any amount and type of data. There are two sets of parties which would use this technique; one being the data owners, which would own the data and the next are the data users which do not own the data but use this after getting the permission for the usage. The data

owners and the data users need to be authenticated on the cloud servers before hand to use any of the services provided. When the sensitive data is outsourced to the cloud, so as to enable the easier accessing of the data by the data owners and the data users, it is encrypted. The data encrypted has a list of keywords which are sent to an administration server. This in turn is then re-encrypted and uploaded by the administration server.

When the data users would want to access these encrypted files, they will have to get themselves authenticated. Once the data users are authenticated and verified, they would search the files using keywords. The keywords are sent to the administration server which in turn would encrypt the given keyword. The encrypted keyword is then compared to the existing keywords and the files are given to the data users after the decryption. Hence this helps in creating a secure environment for the exchange of the information among the data owners and the data users.

- **M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski,G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia,"A view of cloud computing,"Commun. ACM, vol. 53, no. 4,pp. 50–58, 2010.**

  The paper explains about the various advantages and uses of the cloud storage. It helps in a large amount of storage space without the use of the resources in the real time. The resources used to carry out storage operations are never owned by the users and hence the users have to pay per use. This strategy helps in eco friendly and green computing as well. This is possible as the resources that were used earlier for the storage, such as the servers, systems, space, cooling systems and many more are no longer required. The storage systems are virtually present for the user and logically present in a different location. The paper also discusses about the different service models; namely, Infrastructure as a service, Platform as a service and Software as a service. It also discusses the different type of clouds such as public cloud, private cloud, hybrid cloud and the community cloud.

- **D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Int. Symp. Security Privacy,Nagoya, Japan, Jan. 2000, pp. 44–55.**

  It discusses about the four of the most important concepts of provable security, query isolation, controlled searching and hidden query. The provable security helps in keeping the data secure. This is done as the untrusted server cannot understand about the plaintext from the encrypted text that is uploaded. The query isolation also helps in maintaining the secrecy.

  The untrusted server will be unable to learn anything about the plaintext present in the file through the encrypted data about which it is queried. The controlled searching explains that the untrusted server will be unable to search any query or a keyword without the authentication of the users which are already registered. The hidden query talks about the keyword or the query that is being searched. This is in a non-readable form and therefore the untrusted server does not know what the user is searching. Hence the untrusted server will never be able to guess or hack into the system and retrieve any file saved.

- **C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keywordsearch over encrypted cloud data," in Proc. IEEE Distrib.Comput. Syst., Genoa, Italy, Jun. 2010, pp. 253–262.**

  It discusses over the drawbacks of the traditional cloud file retrieval system. There are majorly two drawbacks of this system. The user when tries to retrieve a file from the cloud, the user has to download all the files that are related to the keyword or the query that the user has entered. This leads to a huge consumption of the bandwidth. As the user has downloaded a large number of files, user has to decrypt each file in order to understand whether the file is required or not. The file that is retrieved can be either useful to the user or can be an older file which has not much significance.

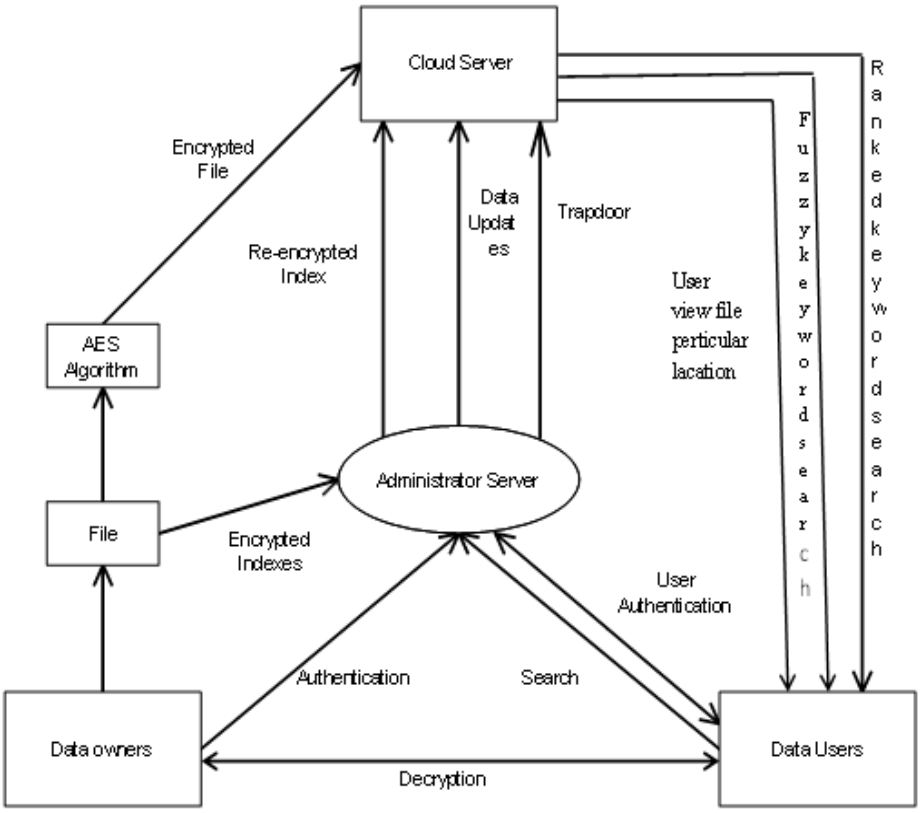
  Therefore, the authors put forward an idea about ranking of the files and documents that are saved over the cloud. This would help in retrieving the files which are recent or most downloaded. The paper also discusses the shortcomings of the Searchable Symmetric Encryption (SSE).

- **H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data", in IEEE Transaction on dependable and secure computing, vol 13, no. 3, May/June 2016.**

  It discusses about extending the existed single-owner scheme to a full-fledged multi-owner scheme will cause abundant problems. In the single-owner scheme, once a data user wants to issue a keyword search, he has to ask the data owner for secret keys to generate trapdoors (encrypted keywords). Unfortunately, when there are multiple data owners, asking different data owners for keys to generate trapdoors would be infeasible.

  First, not all data owners are always online simultaneously when a data user wants to perform a query. If data owners are offline, these owners' data can't be retrieved in time. Second, in order to search different owners' data, data user has to generate a specific trapdoor for each data owner, sending these trapdoors to the cloud server would cause considerable communication overhead. An alternative solution is to share a secret key among all data owners. However, this measure will lead to the security threat of single point of failure.

III. PROPOSED SYSTEM



Data Owner: The data owner uploads document collection *D to the cloud server, but this collection* may contain sensitive information. To protect data privacy, the data owner has to encrypt *D before outsourcing it* to the cloud server.

Data User: The data user wants to search with a query, s/he generates the trapdoor *T for this query firstly by query* encryption, and then submits the trapdoor to cloud server for query processing. After receiving *T, the cloud server* calculates the relevance scores between trapdoor *T and the* documents in index *Ie, and returns k documents with the* highest scores to the data user.

Cloud Server: The cloud server to process query efficiently over the encrypted document collection *C, the data owner constructs an encrypted* searchable index *Ie locally. Finally, the data owner* outsources both the encrypted document collection *C and* the encrypted searchable index *Ie to cloud, and shares the* secret key of trapdoor generation and document decryption to authorized data users with secure channels.

IV. ALGORITHM AND PSEUDO CODE

1. AES Algorithm for Encryption.

AES (advanced encryption standard).It is symmetric algorithm. It used to convert plain text into cipher text .The need for coming with this algorithm is weakness in DES. The 56 bit key of des is no longer safe against attacks based on exhaustive key searches and 64-bit block also consider asweak.AES was to be used128-bit block with128-bit keys.

In this drop we are using it to encrypt the data owner file.

Input:

128_bit /192 bit/256 bit input (0,1)

Secret key (128_bit) + plain text (128_bit).

Process:

10/12/14-rounds for-128_bit /192 bit/256 bit input

Xor state block (I/p)

Final round: 10, 12, 14

Each round consists: sub byte, shift byte, mix columns, add round key.

Output:

Cipher-text (128 bit)

2. MD5(Message-Digest Algorithm)

The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexa-decimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity.

Steps:

- A message digest algorithm is a hash function that takes a bit sequence of any length and produces a bit sequence of a fixed small length.
- The output of a message digest is considered as a digital signature of the input data.
- MD5 is a message digest algorithm producing 128 bits of data.
- It uses constants derived to trigonometric Sine function.
- It loops through the original message in blocks of 512 bits, with 4 rounds of operations for each block, and 16 operations in each round.
- Most modern programming languages provides MD5 algorithm as built-in functions.

3. TOP K Algorithm:

- Users specify information need via a query SQL
- Too many data objects satisfy the query
- present top-k objects
- assumes ranking according to a relevance score
- Examples: find a flat to rent according to price, location, size,
- find a flight according to price, departure and arrival time, number of stops, …

Consider the following scenario:

• Data objects have different attributes, given a query, we can obtain a ranking of the objects according to the different attributes and want to combine (aggregate) the individual rankings into a single ranking

• Top-k is obtained from the aggregate ranking and aggregator is built on top of the subsystems are viewed as middleware.

V. RESULTS AND SCREENSHOTS

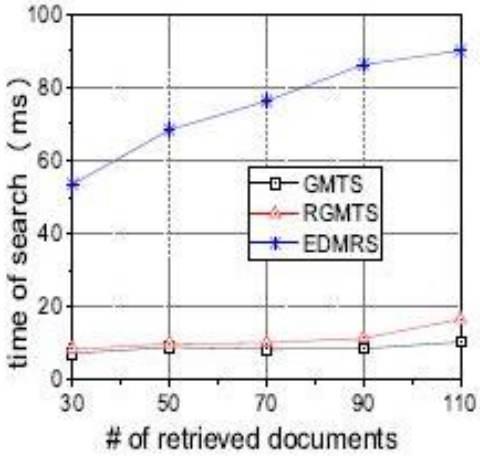


Fig a: (a) for fixed *n* and *m* with different values of *k* (where *k* is the number of documents that the data user wants to retrieve, and $n = 4000$, $m = 8000$, $t = 10$).

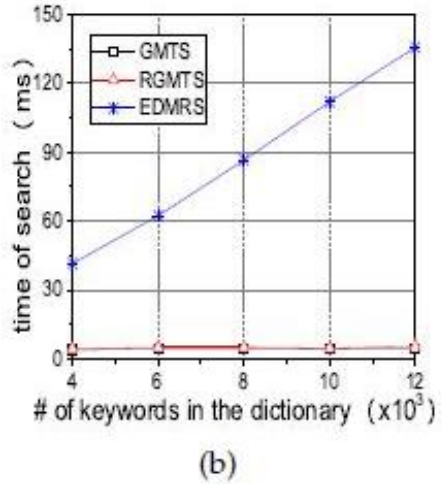
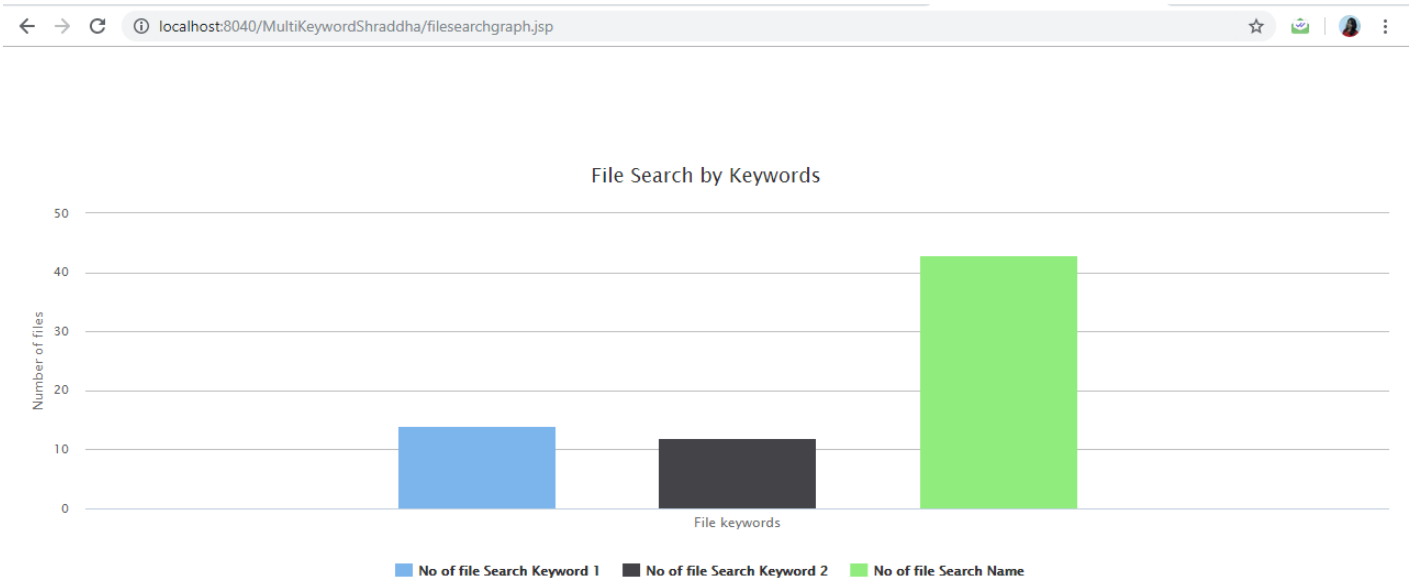


Fig b: (b) the time cost of search with different sizes of the dictionary *W*, we set the size of the document collection *D* as $m = 4000$, and $t = 10$.

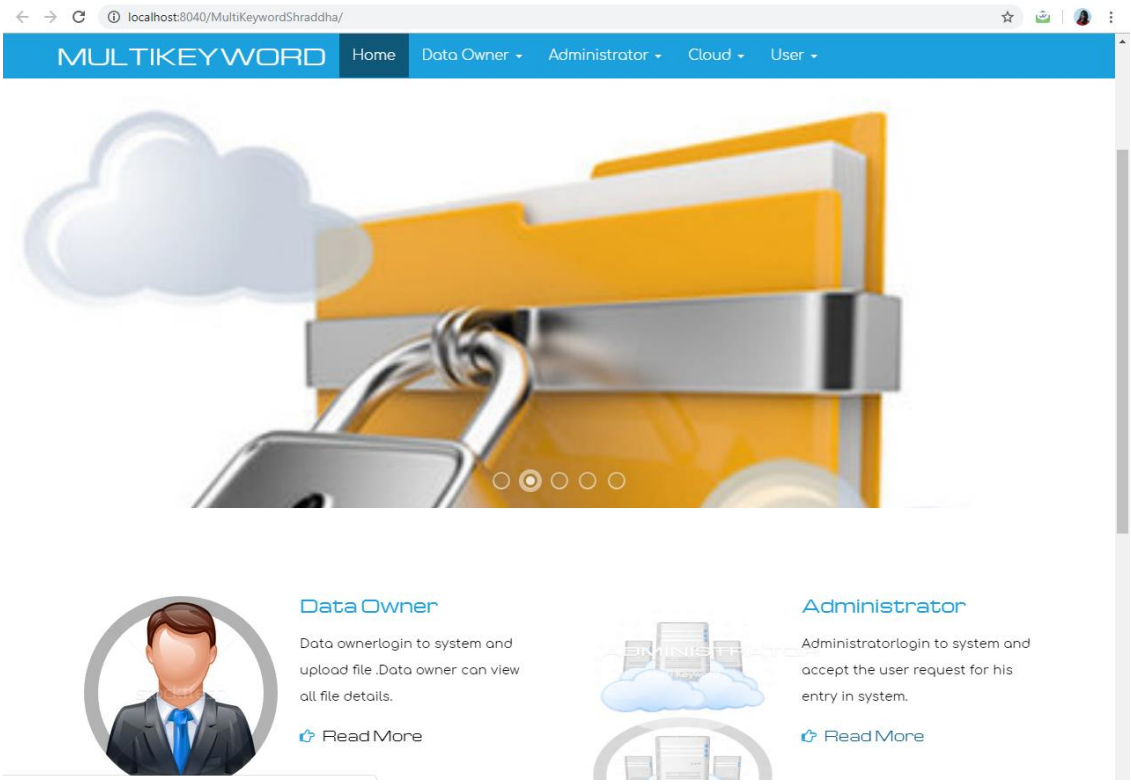
We compare the query efficiency of our methods with EDMRS under different parameter settings. In particular, we study *m* (dataset size), *t* (query size), *n* (dictionary size) and the effect of *k* (parameter *k* in our top-*k* query) on real datasets. Fig.(a) shows that the query time of each method increases with *k* since they all need more time to process the data. The time cost of query in our methods is independent of the dictionary size. So, as shown in Fig. (b), the efficiency of query in EDMRS drops sharply with the increased size *n* of dictionary, but our methods still maintain high efficiency.



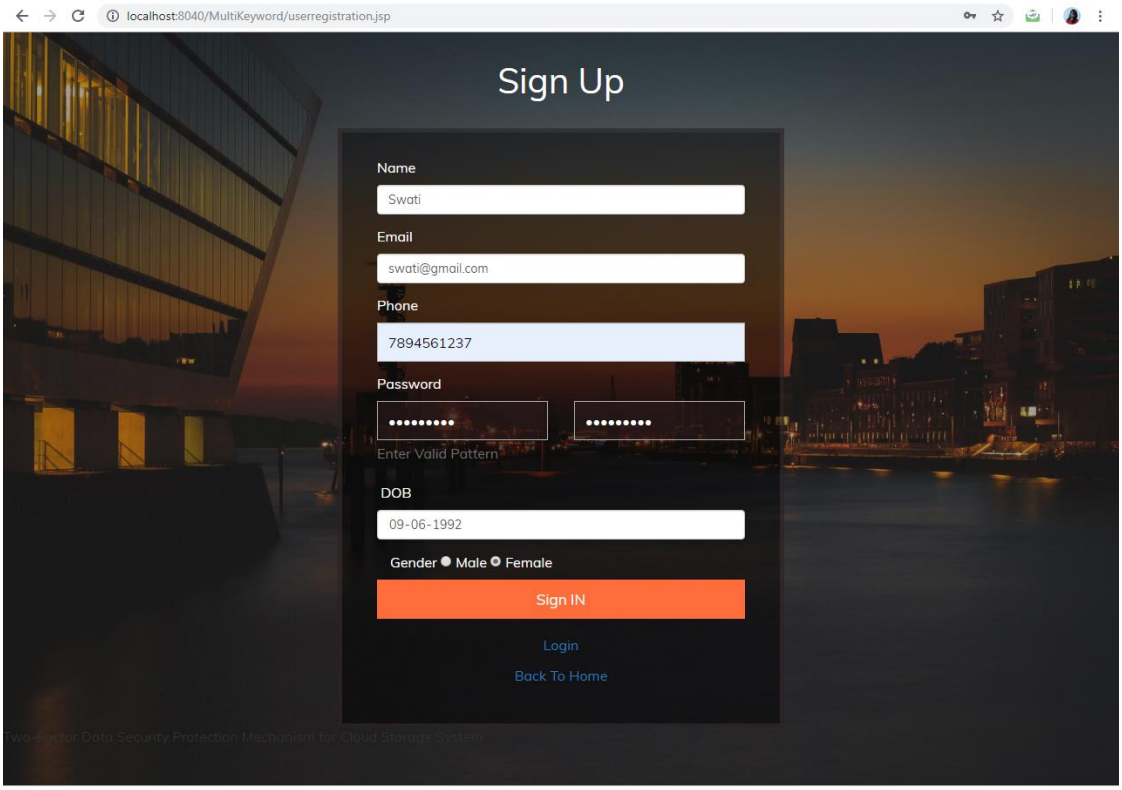
Graph 1: File graph

System screenshots

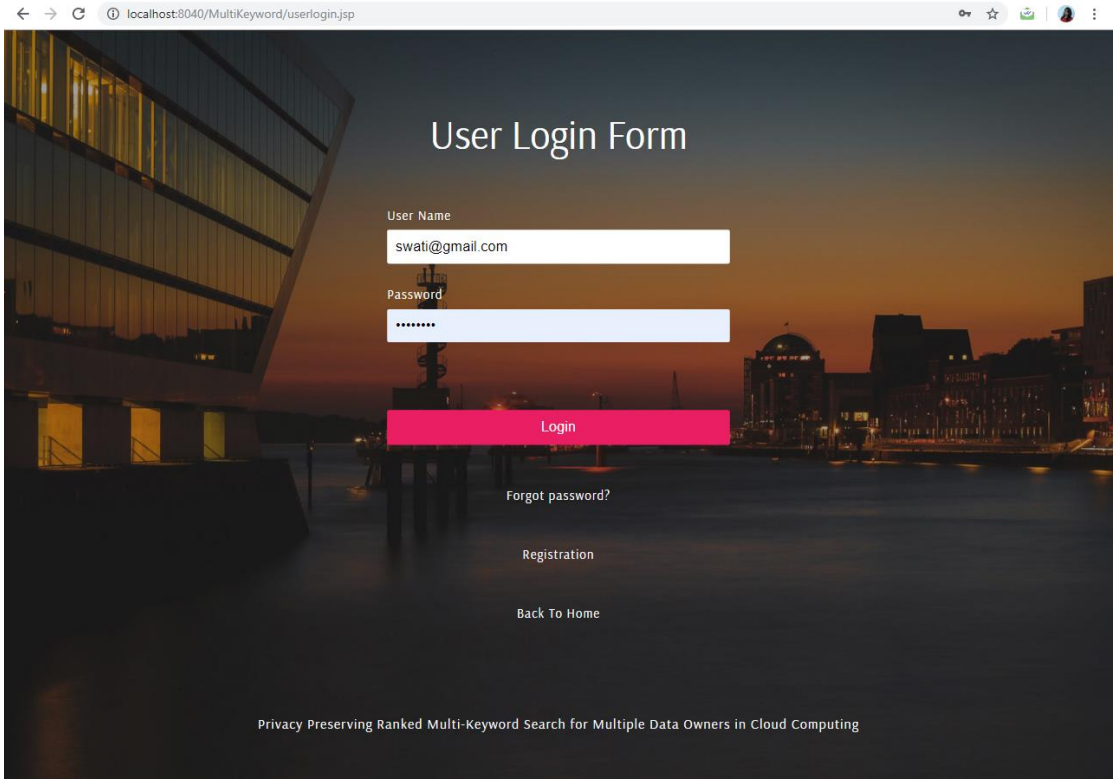
Home page –



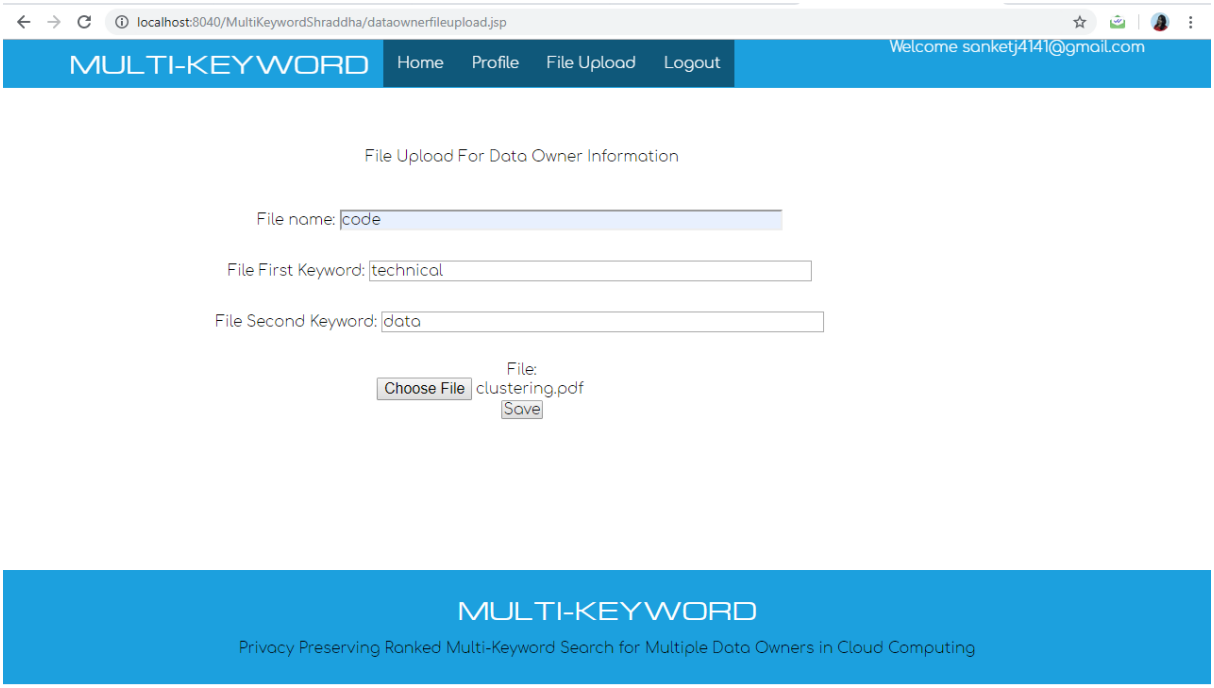
Registration -



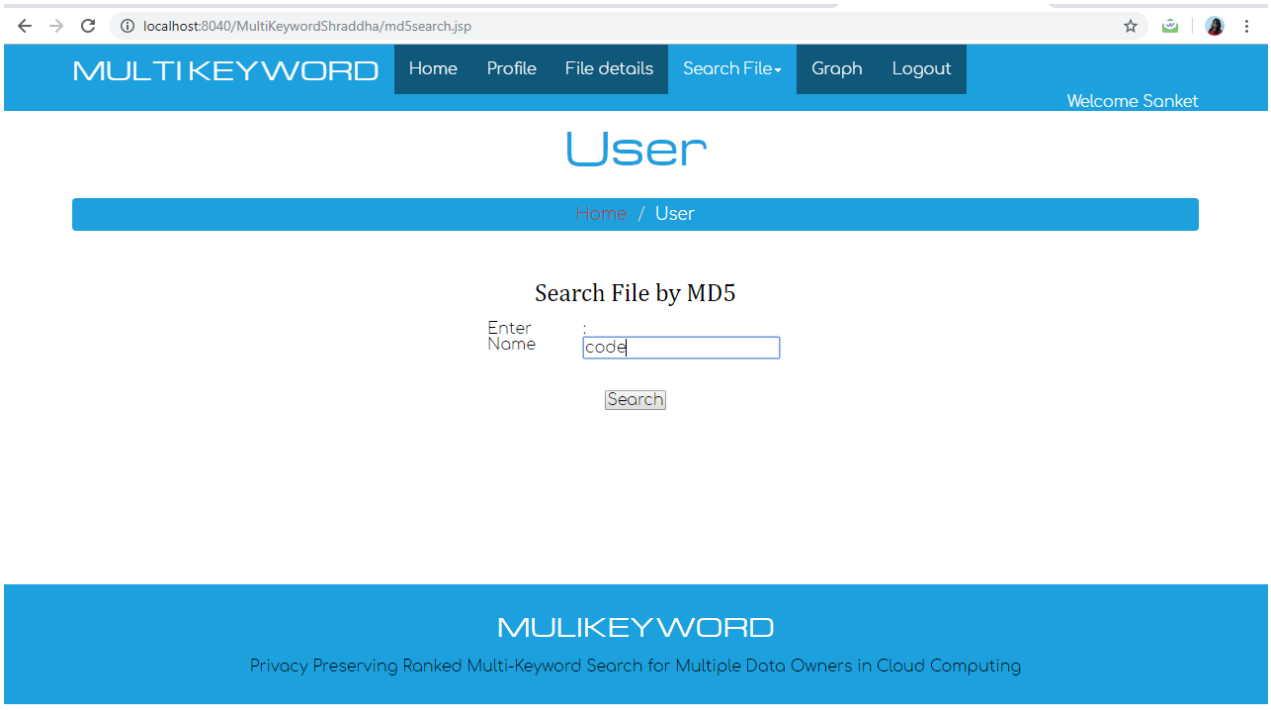
Login –

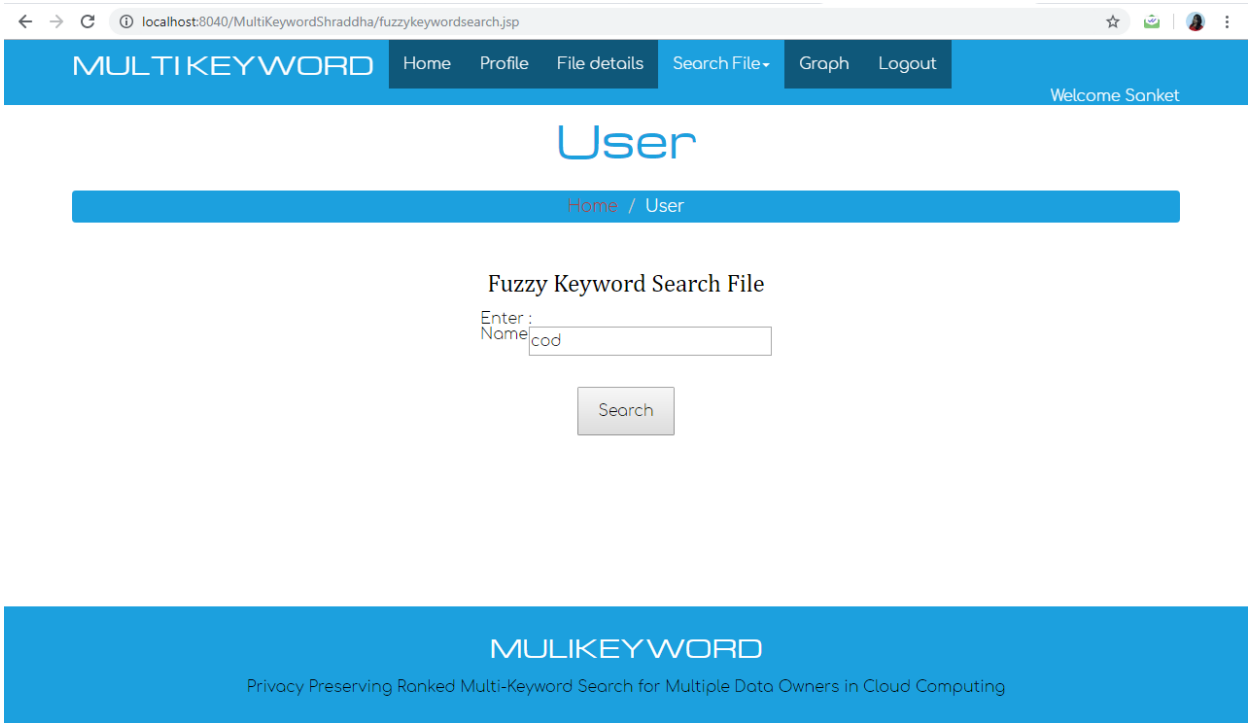


Data upload by user –



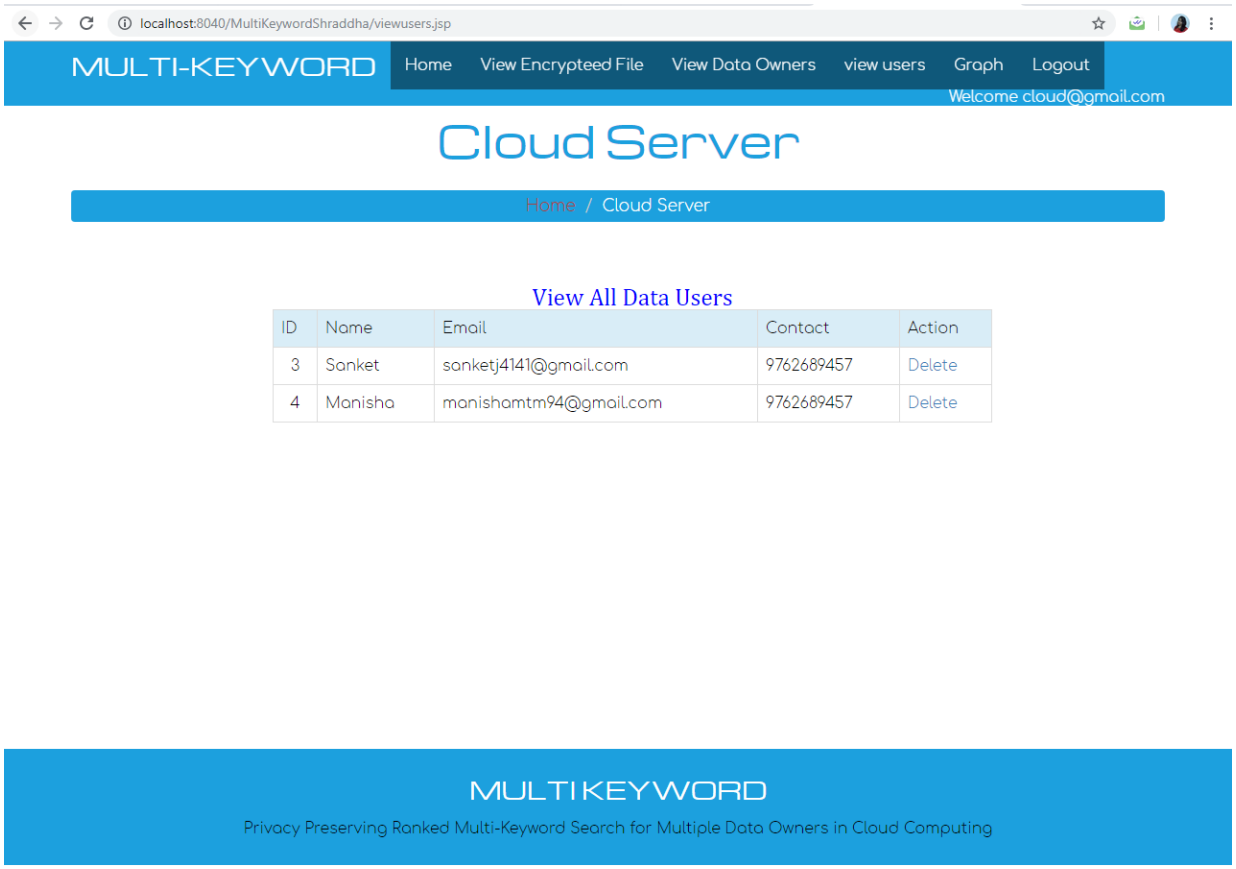
Search file -





Cloud server home –

View users –



VI. CONCLUSION AND FUTURE WORK

Proposed system focus on improving the efficiency and the security of multi-keyword top-*k* similarity search over encrypted data. At first, we propose the random traversal algorithm which can achieve that for two identical queries with different keys, the cloud server traverses different paths on the index, and the data user receives different results but with the same high level of query accuracies in the meantime. Then, in order to improve the search efficiency, we design the group multi-keyword top-*k* search scheme, which divides the dictionary into multiple groups and only needs to store the top-*ck* documents of each word group when building index. Next, to protect the query unlink ability, we apply the random traversal algorithm to get the RGMTS, which can increase the difficulty of cloud servers to conduct linkage attacks on two identical queries, and we can also tune the value of *E* to make the level of query unlink ability flexible for data owners. Finally, the experimental results show that our methods are more efficient and more secure than the state-of-the-art methods.

VII. REFERENCES

[1]. M. ARMBRUST, A. FOX, R. GRIFFITH, A. D. JOSEPH, R. KATZ, A. KONWINSKI,G. LEE, D. PATTERSON, A. RABKIN, I. STOICA, AND M. ZAHARIA,"A VIEW OF CLOUD COMPUTING,"COMMUN. ACM, VOL. 53, NO. 4,PP. 50–58, 2010.

[2]. D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Int. Symp. Security Privacy,Nagoya, Japan, Jan. 2000, pp. 44–55.

[3]. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keywordsearch over encrypted cloud data," in Proc. IEEE Distrib.Comput. Syst., Genoa, Italy, Jun. 2010, pp. 253–262.

[4]. H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data", in IEEE Transaction on dependable and secure computing, vol 13, no. 3, May/June 2016.