DRSIG: Domain and Range Specific Index Generation for Encrypted Cloud Data

Raghavendra S*, Geeta Mara*, Rajkumar Buyya†, Venugopal Kuppanna Rajuk*, Sitharama Iyengar‡ and L M Patnaik§
*University Visvesvaraya College of Engineering, India
e-mail: raghush86@gmail.com.
†The University of Melbourne, Australia.
‡Florida International University, USA
§National Institute of Advanced Studies, Bangalore, India.

***Abstract*—One of the most fundamental services of cloud computing is Cloud storage service. Huge amount of sensitive data is stored in the cloud for easy remote access and to reduce the cost of storage. The confidential data is encrypt before uploading to the cloud server in order to maintain privacy and security. All conventional searchable symmetric encryption(SSE) schemes enable the users to search on the entire index file. In this paper, we propose the Domain and Range Specific Index Generation($DRSIG$) scheme that minimizes the Index Generation time. This scheme adopts collection sort technique to split the index file into D Domains and R Ranges. The Domain is based on the length of the keyword; the Range splits within the domain based on the first letter of the keyword. A mathematical model is used to encrypt the indexed keyword that eliminates the information leakage. The time complexity of the index generation is $O(N_T \times 3)$ where N_T - Number of rows in index document and 3 is Number of columns in index document. Experiments have been conducted on real world dataset to validate proposed $DRSIG$ scheme.It is observed that DRSIG scheme is efficient and provide more secure data than Ranked Searchable Symmetric Encryption($RSSE$) Scheme.**

***Index Terms*—Index Generation; Searchable Encryption; Data Security; DRSIG; Cloud Computing.**

I. Introduction

Cloud computing has become a important deployment platform for distributed applications especially as data storage and information management service due to its enormous potential in computing, storage and various applications [1]. From a joint pool of computing resources that are configurable, it allow storage of remote data, on-demand usage. An elastic and financial plan is provided by Cloud Computing infrastructure for managing information and sharing resources [2], [3]. System maintenance overhead and hardware-software expenditure is reduced by it. It offer appropriate communication path to share resources between data owners and data users. The popularity of cloud services, such as Microsoft Azure, AWS Amazon Services, Apple iCloud, Google AppEngine, has enabled companies to shift their data onto the cloud. The data owner can deploy the personal information onto public cloud and data user can access the information anytime and anywhere. Particularly, huge amount of information and workloads can be deployed by end-user to the cloud. Usage of unlimited computing in a pay-per-use resource sharing model serve a one of the benefits and this permits the user to pay only for the amount of service used.

The highly challenge tasks faced by Cloud Computing infrastructure, data confidentiality, reliability and safety concerns occupy the main position. In practical, the public cloud which are away from the trusted domains contain the confidential data. The data uploaded by the data owners to the cloud bring concern of possible data loss, dishonest utilization of confidential data as the owners do not possess any direct control over the sensitive information. Generally, cloud servers are labelled as curious and untrusted entities. Data owner hinder to implement cloud technologies when a case of breach of information to third party or cloud provider itself is possible. Hence providing ample security and confidentiality protection to information that is susceptible to breach is of high importance. This gets employed in application designed for healthcare[4], financial and government data. So as to prevent the breach of more confidential data that is uploaded to the cloud, information is encrypted beforehand and then uploaded to the cloud server. To retrieve data files, traditional searchable symmetric encryption (SSE)[5] technique depends on keyword search mechanism but they support only Boolean keyword search without any assurance of n file retrieval accuracy. This mechanism is inefficient in retrieval; it demands a large amount of post-processing overhead and incurs unnecessary network traffic.

To resolves the problems of data breach in cloud, current solutions use the following approaches to provide searching ability on cloud data on basis of keywords[6]. A collection of keywords are identified and stored on the index file. For every file an index vector is designed. After the creation of index vectors, all the index vector are merged in an index file and produced. The index file thus produced and the data file are uploaded to cloud servers after encryption. Now, the information is prepared to allow queries from the datauser. Cipher-text is supported by cloud servers based n queries as follows. A keyword based search query on the cloud containing the encrypted data is sent by the data user and

keywords that are encrypted are sent to the cloud. After receiving the query, the cloud server implements as search on the encrypted index and returns a results of the list of relatable files. The data user then makes a choice of the files that are necessary and are retrieved from the cloud server. With the help of the authorized secret key, the user decrypts the required encrypted files that were retrieved from the cloud. This way, protection of data from breach and data confidentiality is safeguarded. During the entire procedure, plaintext information or keywords are invisible to the cloud servers.

*Motivation:*In the existing schemes, single keyword and multikeyword search are used to search query over encrypted cloud data. The major obstacles in achieving these search schemes are: How to perform resourceful and secure search over encrypted data. In the previous schemes database-centre does not give supportable protection over encrypted cloud data [7]. Current results over encrypted cloud data support only for linear search. However, given enormous amount of outsourced data, linear search is inefficient for huge data. This paper fouses on secure searching technique with resourceful and flexible search over encrypted data. We propose a new scheme that incorporates Domain and Range concept which depends on the length and starting letter of the keyword.

Contribution: In this scheme, we present a novel Index generation and a queried keyword search technique *Domain and Range Specific Multi-keyword Search* $(DRSIG)$, that supports accurate search over encrypted cloud data. $DRSIG$ provides secure, efficient and effective search results within a short time and it protects confidentially of data from the cloud service provider and unauthorised users. $DRSIG$ scheme reduces Index Storage Space by arranging keywords in an array format discussed in section VI. Specifically, our contribution can be summarised as follows:

1) We proposed a information retrieval technique Domain and Range Specific Index Generation $(DRSIG)$ scheme that support for accurate and minimum Index Generation time over large dataset.
2) The time complexity of $DRSIG$ scheme is reduced to $O(N_T \times 3)$ for index building.
3) A mathematical model is developed that provides security. The proposed scheme prevents leakage of sensitive information to achieve privacy of keywords.

Organisation: The rest of the paper is structured as follows: First, Literature survey is reviewed in Section 2. Ranked Searchable Symmetric Encryption $(RSSE)$ scheme and its drawbacks are discussed in Section 3. System architecture and design goals are defined in Section 4. Section 5 gives the detailed description of our domain and range specified Index generation scheme. Section 6 discussed performance analysis and conclusions are presented in Section 7.

II. Related work

Lu et al., [8] designed a novel cryptographic primitive - range predicate encryption - to build a Logarithmic Search over Encrypted Data $(LSED)$ system. This scheme is provably secure with regard to plaintext confidentiality, predicate privacy and supports logarithmic search over encrypted data, query authentication and secure data update. The $LSED$ system reveals the access patterns of cipher texts to the cloud server. Moreover, all database update operation and query authorization relies on the database owner i.e., a single point of failure.

Xia et al., [9] proposed a scheme for basic similarity search over encrypted images based on a secure transformation method that protected the information about features, and did not degrade the result accuracy. The proposed scheme protected the confidentiality of image database, feature vectors, and user's query. Moreover, the image owner could update the encrypted image database as well as the secure index quite easily. This scheme assured the confidentiality of the data, result accuracy and query unlinkability. The time complexity of query on invert index is $O(n)$, which can be further enhanced by using better indexing technique to reduce search time.

Pang et al., [10] presented a general framework for multi-user noisy-keyword-based searchable symmetric encryption in a fault-tolerant manner. Existing efforts on multi-user searchable symmetric encryption (SSE) have focused on exact keyword search, but these results are not applied to the situation where the keywords associated with the files are noisy data. It combines a single-user noisy-keyword-based SSE scheme with a private-key dynamic broadcast encryption scheme. This scheme permits dataowner to efficiently and dynamically revoke the users. It allows the authorised users to search the encrypted document set using their chosen noisy keywords with the assistance from an honest-but-curious server.

Wang et al., [11] designed a novel Fuzzy Keyword Search Scheme $(F2SE)$. Top-k fuzzy keyword search is achieved with extracted fingerprint and kNN encryption algorithm. It require less storage space and search time. The fingerprint extraction algorithm can be optimized to improve Searching Accuracy Rate and match it with other symbols or languages.

Raghavendra et al., [12] addressed the issue of split keyword search over encrypted cloud data. Split keyword index generation algorithm is used to search keyword based on fuzzy and synonyms keywords. The advantage of split keyword is to reduce the storage space and variation in index generation time is less compare to B-Tree scheme. communication cost is high.

Buyrukbilen et al., [13] designed a Ranked Privacy Preserving Search technique on Public Key Encrypted Data. Homomorphic encryption, private information retrieval protocols and indexing structure is applied to achieve queries

TABLE I
NOTATIONS

Symbols	Definition
F	The collection of plain-text file are outsourced as a set of n information documents $F = (f_1, f_2, f_3, \ldots, f_n)$.
W	Distinct keywords extracted from the file collection F, is a set of m keywords $W = (w_1, w_2, \ldots, w_m)$.
I	The collection of F files generates searchable index, denoted as $(I_1, I_2, \ldots, I_m)$ where each sub-index I_i built from F_i.
t_w	The trapdoor generated by a user for search request of keyword W.
ID_{list}	The queried keyword w_i presents in a set of ranked identifiers in F files.
$ID(f_i)$	The file identifier in F_i which helps to locate the actual file.
Q	User interested Queried multi-keyword.
q_i	Individual queried keyword.
S	Score is computed by term frequency TF.
a	ASCII value of each letter in the keyword.
$\alpha(w_i)$	Extracted keyword computation results by using Equation-2.
$\alpha(q_i)$	Queried keyword computation results by using Equation-2.
D	Number of files.
T	Total terms in each file.
N_T	Total rows in index document.
C	Total columns in index document (i.e., C=5).
L	Same length of the word is grouped as Domain D_i.
FLW	Same First Letter of the Word is grouped as Range R_j with in the Domain D_i.
w_i	Individual extracted term.
F'	Encrypted n documents.
I'	Computed $\alpha(w_i)$ is stored in the Index document.
I''	Encrypted I'
C_s	Bucket start position
C_e	Bucket end position

in a secure manner. The query response time reduces by several orders of magnitude but has storage overhead and increased computation cost.

Wang et al., [14] established Static Index (SI) and Dynamic Index (DI) for Public-key Encryption with Keyword Search $(PEKS)$ to make search secure and efficient. SI and DI help $PEKS$ to decrease the load respectively in two parts: If data users are searching queried keyword for first time, SI is used or else, DI is used, SI and DI are concurrently functional with $PEKS$ and enhanced as Secure Hybrid Indexed Search $(SHIS)$ scheme that uses deterministic encryption (DE). $SHIS$ is improved further for multiple-receiver applications but this extension, supports only one keyword searchable ciphertext.

III. Background Work

Wang et al., [15] designed a statistical measure approach known as Ranked Searchable Symmetric Encryption(RSSE). RSSE scheme introduced Information Retrieval and text mining to embed the weight information i.e., relevance score of each file. RSSE scheme generates searchable index before outsourcing the encrypted file by using inverted index. RSSE scheme adopts one-to-many Order-Preserving mapping technique integrated with crypto primitive and Order-Preserving Symmetric Encryption $(OPSE)$ for security.

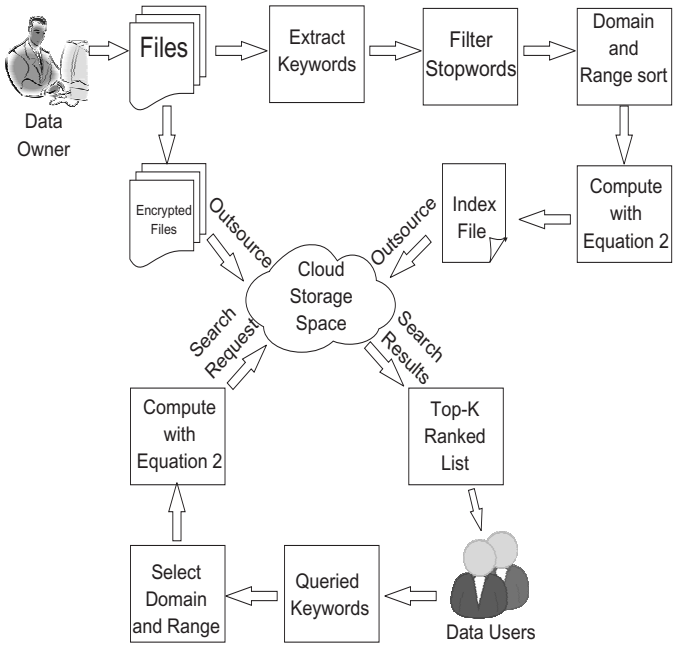

Fig. 1. system Architecture

Inverted index has storage overhead (50% - 150%) on large scale datasets and high maintenance costs on updates, insertions and deletions. The processing cost increases with the number of weights in the $(m \times n)$ matrix is shown in Figure 2. Even though a term is not contained in the document, it still allocates memory to store term frequency zero on the matrix. Large number of zero's appears on the index that increases the search and computation time. In $RSSE$, the analysis is performed on single keyword (w_i). The $OPSE$ technique allocates extra memory to store cipher text of each relevance score. Inverted index is not compatible for large datasets. The $RSSE$ Order-Preserving Mapping (OPM) is achieved on single keyword search with the support of Domain and Range over encrypted cloud data but not on multi keyword search. Here the Domain is distribution of relevance score for keyword and Range is $(OPSE)$ distribution score for the similar keyword.

IV. Problem Definition and System Model

A. Problem Definition

Given that n files are encrypted and uploaded onto the cloud. The major objectives are:

- To reduce the index generation time.
- To reduce search time over encrypted cloud data.
- To provides security and privacy without learning any extra information from the attackers

B. System Model

The system architecture has three modules: *DataUsers*, *DataOwner* and *Cloud Server* as shown in Figure 1.

The data owner is a collection of n files represented by $F = (f_1, f_2, \ldots, f_n)$ to be outsourced on the cloud space. The terms are extracted before outsourcing a file and an index file is built. The index file contains: *term, file* $ID(f_i)$ and *frequency* in the form of domain and range. It is easy to find the keyword over encrypted index file. Further the index file and collection of n files are encrypted before outsourcing to the cloud.

The cloud server communicates with the data owner and data user. It hosts storage and retrieval services for the third party. The cloud provider is not involved in any deletion or modification of data. In the SSE schemes, the cloud server is considered as honest-but-curious, status to learn information from stored data.

The secret key is used to generate the trapdoor t_w between the *Data User* and *Cloud Server*. The authorised user can send queried keyword to the cloud server to search the top-k files. The queried keyword search depends on domain and range of the index. After receiving the search keyword, the cloud server returns the relevant files as fast as possible related to the queried keyword. The optimal k value is used to reduce the communication cost by data user. The top-k results are returned to the datauser from the cloud server.

V. Score Calculation Method

The Score S is computed by calculating occurence of individual term in each file. The expression for standardized Score estimation is as per the following:

$$S = \frac{freq}{max_freq}. \tag{1}$$

where $freq$ - recurrence of each term in a record, max_freq - most extreme recurrence in the wake of considering each documents in the folder and S - is Score acquired by $\frac{freq}{max_freq}$.

Another scientific model for encrypting the keyword is given below:

$$\alpha(w_i) = (a_0 x^k + a_1 x^{k-1} + \ldots\ldots + a_n x^{k-n}) \tag{2}$$

$$\alpha(w_i) = \sum_{p=0}^{n} a_b x^{k-p} \tag{3}$$

where x is a real number, k represents the length of the keyword and p is the postion of the letter in a keyword. For example, if the keyword is $sciences$ then the length is 8 and position of letter e is 4.

VI. Proposed Scheme

In this paper, we consider Domain D and Range R specific secure keyword search over encrypted cloud data for the outsourced text data. In this structure, the data owner have insufficient resources for storing confidential data to the semi trusted cloud server. Cloud server preserves the confidentiality of the outsourced data except from the search and access pattern. Sensitive information is used to generate searchable index securely. The authorised data user can perform search on the encrypted cloud data by utilising searchable index file which returns the matched files related to the queried keyword. During these process the cloud server does not learn anything from the encrypted stored data. The data user decrypts the top-k selected document using decryption key shared by the data owner.

The frame work has two parts: *retrieval phase* and *Initialization phase*. The initialization phase consists the functions $Setup(\lambda)$ and $IndexGeneration(K, C)$. The function $Setup(\lambda)$ generates the keys SK and PK for communcation among the data owner, data user and the remote cloud server. The index generation function involves operation on the plaintext and it extract words from the set of plaintext documents C and generates a secure searchable index I from the extracted words. The searchable Index is a $(n \times 3)$ matrix that involves file IDs, word w_i and frequency S of w_i for convenient retrieval of data (see Algorithm 1). Most of the work process on data owner side for security reason. The detail description of function build index is given below.

Initialisation Phase:

- The data owner initiates the DRSIG scheme by calling the function $Setup$ to generate the Secret Key(SK) and the Public Key(PK). The authorised data users access cloud data files using the secret key provided by data owner.
- The data owner calls the function $IndexGeneration$ to build index. The algorithm build index scans sensitive set of files F and then extracts n distinct keywords set, $W = (w_1, w_2, w_3, \ldots, w_n,)$ for each file f_i. After extracting a set of distinct keywords $w_i = (w_i | 1 \leq j \leq n)$ the stop-words are removed. The normalisation is computed for each file frequency S according to Equation 1. The data owner stores the extracted keywords in the index file I. Index file I stores parameters $\langle L||FLW||ID(f_i)||w_i||S\rangle$ in a $(n \times 5)$ matrix format. The $(n \times 5)$ matrix sorted according to Domain D and Range R by using a collection sort as shown in figure 3. Domain D_n sort is based on the length L of each keyword and after sorting stores $(n \times 4)$ except the L column. Each domain start and end positions are stored in the index I for reference of Domains. The domain D_i is split into range R_{ij} based on alphabetical FLW and is stored in a specific bucket B_{ij}. Each bucket range R_{ij} start and end positions are stored in the index I for reference of different Ranges.
- The index file I stores the reference of Domains, reference of Ranges and $\langle ID(f_i)||\alpha(w_i)||S\rangle$ in a $(n \times 3)$ matrix. After completing the process of Domain and Range, then $\alpha(w_i)$ is computed for each keyword w_i in the index file I and values are stored in I' according to Equation 2. The computed $\alpha(w_i)$ is stored with File $ID(f_i)$ and frequency score S. The searchable index file I' is encrypted partially.
- The data owner encrypts both the searchable index file I' into I'' and the collection of sensitive files $F = (f_1, f_2, \ldots, f_n)$ into $F' = (f'_1, f'_2, \ldots, f'_n)$ using cryptology techniques. The encrypted searchable index I'' and

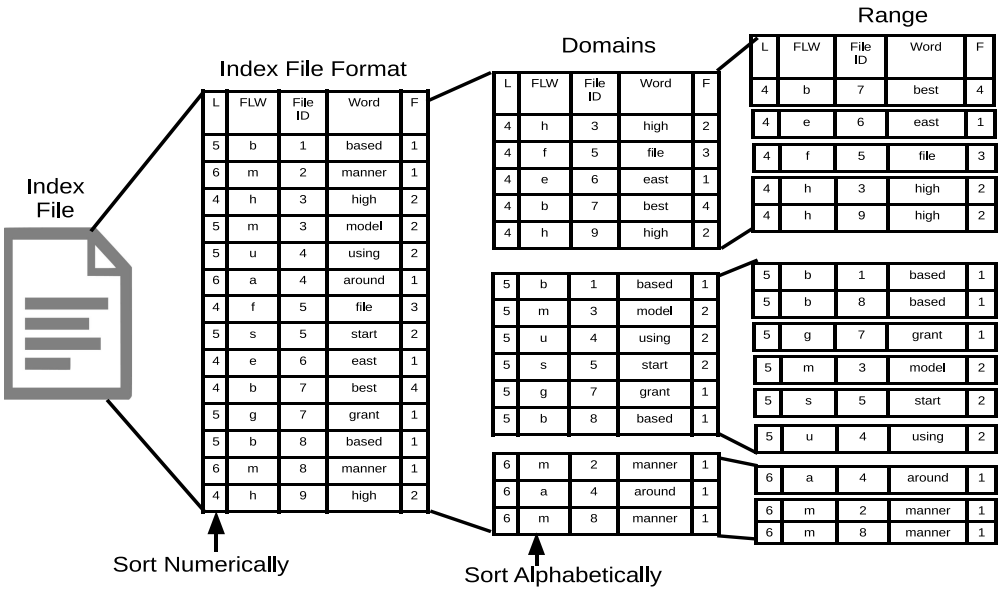

Fig. 2. Numerical Sort Splits the Index into Domains and Alphabetical Sort Splits each Domain into Ranges

encrypted files $F' = (f'_1, f'_2, \ldots, f'_n)$ are uploaded to the cloud server.

Algorithm 1: Build Index

input : A Collection of n Data Files $F = (f_1, f_2, \ldots, f_n)$
output : Domain and Range sorted Index file I''
procedure: Build Index(K, F)

for $f_i \leftarrow 1$ **to** F **do**
  each file $f_i \in F$;
  Scan F and Extract each term in f_i, denoted as a $W = (w_1, w_2, w_3, \ldots, w_n,)$;
  Normalised and remove the stopwords from W;

for $i \leftarrow 1$ **to** W **do**
  count frequency of each word in f_i;
  store the $\langle L||FLW||ID(f_i)||w_i||S\rangle$ in I;

- Index I sort based on length L of keyword and store in specific Domain D_i;
- Each Domain D_i sort based on alphabets FLW and store in Specified Bucket B_{ij};
  for $i \leftarrow 1$ **to** W **do**
  Compute $\alpha(w_i)$ for each keyword w_i in B_{ij}according to Equation 2;
  Each computed results stores $\langle ID(f_i)||\alpha(w_i)||S\rangle$ in ascending order of index I';
- I'' = encryption of Index file I';
  return I;

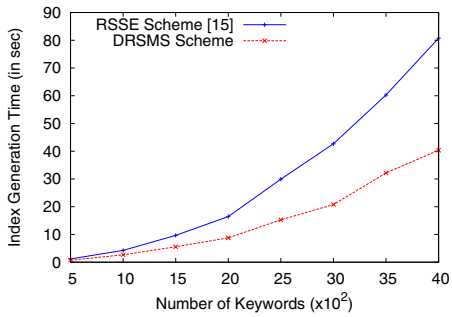

Fig. 3. Index Generation Time Comparison with Keywords.

VII. Performance

National Science Foundation Research Award Abstracts 1990-2003 [16] is used to evaluate the $DRSIG$ Scheme. The abstracts have huge amount of unique technical keywords. The entire system is implemented in Java language. The Data owner and the data user use a windows platform with Intel(R) Core(TM)2 Duo CPU T6400 @2.00GHz, 3072 MB of RAM and commercial public cloud Amazon S3 service to store the Index file I'' and encrypted collection of files C'. We analyse the overall Index generation cost and per keyword index storage cost of DRSIG scheme.

A. Indexing cost

The index generation time over $RSSE$ and $DRSIG$ scheme is shown in Figure 3 and Figure 4. The time complexity of index generation in $RSSE$ scheme is $O(T \times D)$. If D=5 and T=500, then $T \times D$ = 500 $\times$ 5 = 2500 elements takes 1174 milliseconds to generate index file for $RSSE$ scheme.

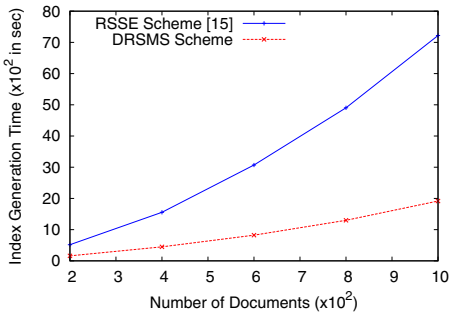

Fig. 4. Index Generation Time Comparison with Documents.

The time complexity of index generation in $DRSIG$ scheme is $O(N_T \times C)$. If N_T=660 and C=5 (i.e., $\langle L||FLW||id(f_i)||W_{ij}||F_{ij}\rangle$) then $N_T \times C = 660 \times 5 = 3300$ elements takes 708 milliseconds to generate index file for $DRSIG$ scheme. $OPSE$ takes $O(n)$ for frequency score calculation and storage space can takes the range between 2^{30} to 2^{80} bits. In the proposed $DRSIG$ scheme have emphasised on keyword security with minimum storage space, the number of elements in $DRSIG$ scheme is more than $RSSE$ scheme; therefore, index generation time of $DRSIG$ scheme is less than $RSSE$ scheme. This is due to a large computation required by $RSSE$ scheme to generate a large encrypted number (i.e., 2^{30} to 2^{80} bits range) in comparsion to $DRSIG$ scheme. The score calculation time for 5 files and 500 distinct keywords in $RSSE$ scheme is 103 milliseconds while in $DRSIG$ scheme it is 3.4 milliseconds. The time complexity of the score calculation is $O(T \times D)$ for $RSSE$ scheme and $O(N_T)$ for $DRSIG$. If T=500 and D=5 then $T \times D = 500 \times 5 = 2500$ frequency elements; score calculation time is 103 milliseconds in $RSSE$ scheme. When N_T=660 keyword elements, score calculation time is 3.4 milliseconds for $DRSIG$ scheme. It is observed that, 2500 frequency elements of $RSSE$ scheme score calculation time reduces for 660 keyword elements score calculation time in $DRSIG$ scheme. The computation time of $DRSIG$ is reduced by 99.95% compare to $RSSE$ scheme. For 1000 files D=1000 and T=20387, $T \times D = 20387 \times 1000 = 20.4$ million elements and the index generation time is 7221.4 seconds for $RSSE$ scheme. In $DRSIG$ scheme N_T=311800, C=5 then $N_T \times C = 311800 \times 5 = 1.5$ million elements and the index generation time is 1916.8 seconds. The index generation time in $DRSIG$ improves by 39.7% for 5 files and 73.47% for 1000 files.

VIII. Conclusions

In this paper, we introduce a new Index Generation technique for achieving effective data utilisation through Internet over remotely stored encrypted cloud data. $RSSE$ scheme is inefficient to achieve Index Generation on a large dataset. To overcome this drawback in $RSSE$ scheme, we propose a new Domain and Range Specific Index Generation($DRSIG$) algorithm that supports more efficient and accurate Index for search. $DRSIG$ algorithm performs effective and efficient indexing and searching on encrypted data. Indexing phase reduces index computation time and index storage space up-to 90%. Domain D is sort based on length of the keyword and Range R is sorted based on first character of the keyword. The $DRSIG$ scheme requires a small amount of additional time to sort on index file. In future, search time need to be reduced on encrypted data set.

References

[1] Rajkumar Buyya, Rodrigo N Calheiros, Jungmin Son, Amir Vahid Dastjerdi, and Young Yoon. Software-Defined Cloud Computing: Architectural Elements and Open Challenges. *in Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1–12, 2014.

[2] S. Raghavendra, S Girish, C. M. Geeta, Rajkumar Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik. MSIGT: Most Significant Index Generation Technique for Cloud Environment. In *12th IEEE India International Conference on E^3-C^3(INDICON 2015)*. IEEE, IEEE, 2015.

[3] Sugam Sharma. Expanded Cloud Plumes Hiding Big Data Ecosystem. *Future Generation Computer Systems*, 59:63–92, 2016.

[4] S Vishwa Kiran, Ramesh Prasad, J Thriveni, KR Venugopal, and LM Patnaik. Cloud Enabled 3D Tablet Design for Medical Applications. In *in Proceedings of the 9th International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6. IEEE, 2014.

[5] Yap Joe Earn, Raed Alsaqour, Maha Abdelhaq, and Tariq Abdullah. Searchable Symmetric Encryption: Review and Evaluation. *in Journal of Theoretical & Applied Information Technology*, 30(1), 2011.

[6] S Raghavendra, C M Geeta, K Shaila, Rajkumar Buyya, K R Venugopal, S S Iyengar, and L M Patnaik. "MSSS: Most Significant Single-keyword Search over Encrypted Cloud Data". *in Proceedings of the 6th Annual Intrernational Conference on ICT: BigData, Cloud and Securit*, 2015.

[7] Hakan Hacigümüş, Bala Iyer, Chen Li, and Sharad Mehrotra. Executing SQL over Encrypted Data in the Database-Service-Provider Model. *in Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 216–227, 2002.

[8] Yanbin Lu. Privacy-preserving Logarithmic-time Search on Encrypted Data in Cloud. *in 19th Annual Network and Distributed System Security Symposium, (NDSS)*, 2012.

[9] Zhihua Xia, Yi Zhu, Xingming Sun, and Jin Wang. A Similarity Search Scheme over Encrypted Cloud Images based on Secure Transformation. *in International Journal of Future Generation Communication & Networking*, 6(6):71–80, 2013.

[10] Xiaoqiong Pang, Bo Yang, Mingwu Zhang, and Hongzhen Wang. Multi-user Noisy Keyword Search over Encrypted Data. *in Journal of Computational Information Systems*, 9(5):1973–1981, 2013.

[11] Dongsheng Wang, Shaojing Fu, and Ming Xu. A Privacy-Preserving Fuzzy Keyword Search Scheme over Encrypted Cloud Data. *in Proceedings of the 5th International Conference on Cloud Computing Technology and Science (CloudCom)*, 1:663–670, 2013.

[12] S. Raghavendra, S Girish, C. M. Geeta, Rajkumar Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik. IGSK: Index Generation on Split Keyword for Search Over Cloud Data. In *2015 International Conference on Computing and Network Communications (CoCoNet'15)*, pages 380–386. IEEE, December 16-19 2015.

[13] Sahin Buyrukbilen and Spiridon Bakiras. Privacy-Preserving Ranked Search on Public-Key Encrypted Data. *in Proceedings of the 2013 IEEE International Conference on High Performance Computing and Communications*, pages 165–174, 2013.

[14] Wei Wang, Peng Xu, Hui Li, and Laurence Tianruo Yang. Secure Hybrid-Indexed Search for High Efficiency over Keyword Searchable Ciphertexts. *Future Generation Computer Systems*, 2014.

[15] Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou. Enabling Secure and Efficient Ranked Keyword Search over Outsourced CloudData. *in IEEE Transactions on Parallel and Distributed Systems*, 23(8):1467–1479, 2012.

[16] National Science Foundation Research Awards Abstracts 1990-2003. *http://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html*, 2013.