Fine Grained Synonym Multi-Keyword Search Based on Hierarchical Clustering

Pratima Pawar[1], K. N. Shedge[2]
[1]Student, ME Computer, SVIT College Nasik, Maharashtra, India
[2]Professor, Computer, SVIT College Nasik, Maharashtra, India

ABSTRACT

Many business organizations outsourced their data to cloud in encrypted format. Data encryption is performed before outsourcing data to cloud due to security and privacy issues. Cloud releases the burden of heavy management of data. To retrieve data from cloud server, keyword search approach is preferred. A traditional approach of data retrieving does not preserve the relationship between encrypted and plain text documents. In this paper, MRSE-HCI technique is proposed to preserve relationships between various plain text documents over an encrypted domain. While uploading documents to cloud, document index is generated and it is used to generate clusters of documents index. QHC algorithm is implemented for index cluster creation. With the proposed approach, complex logical search operations using AND, OR, NOT keywords and synonym search is contributed.

Keywords: - Cloud computing, ciphertext search, ranked search, multi-keyword search, hierarchical clustering, security

I. INTRODUCTION

Data encryption is the traditional way to preserve data from unconditional leakages. Even if cloud based services offer many benefits, data privacy is the big concern in it. To address this issue, it is adorable to outsourced data in encrypted format. At the other user's end they retrieve data using keyword searching strategy, but there is challenge in the process of encryption is documents will be buried which affected on searching performance and hence, accuracy of searching is degraded. Another challenge is data centers have impressive growth which is more challenging to design ciphertext search schemes. Ciphertext search scheme provides an efficient and decent way of online information retrieval on volume of encrypted data. There are several techniques to reduce information leakages from those techniques data encryption is the most popular approach. Searching over an encrypted data become a very challenging task as, it will makes the server-side utilization. In recent studies, there are many ciphertext search schemes have been proposed by organizing cryptographic techniques. These methods provide undoubted security but heavy operations required by them which resulting into high time complexity. Therefore, prior methods were not suitable in the scenario of huge data; also online data processing is required by them. Properties of documents are represented by their relationship which required appropriate maintenance of document is required to express documents. Categories of documents are also decided by relationship of documents. Data encryption is random procedure which takes place blindly at the time of data outsourcing hence, document categorization is the best way to determine or retrieve particular category wised document form cloud server. It also helps to increase document searching speed.

Another issue is noticed during analysis of literature survey is the data corruption which is occurred due to damage software or hardware. Data corruption or leak returns incomplete data as result of searching. To recover data damage, a data verifiable mechanism is required for complete data verification. In this research, a vector space model is introduced, in this each and every document is represented by vector. To maintain relationship between documents, they are divided into multiple categories such as, sports, entertainment, techniques, social network etc. Specific short categories are designed for minimum distance in high dimensional space. Document categorization can be also referred as "document clustering" due to this document search time will be reduced by selecting category of document. Generally, documents searched by user is very small, hence categories are divided into sub-categories such as, main category: - sports can be divided into subcategories like, "Indoor" and "Outdoor". For efficient document search "Backtracking" algorithm is also proposed in this work. It produces only targeted documents. Hash function is constructed to check integrity of search result documents. Every document is hashed and hash result is used to represent the document. In this research work, multi-keyword ranked search over encrypted data based on hierarchical clustering index (MRSE-HCI) is proposed to preserve the relationship between different plain documents over the encrypted domain[14]. It will help to improve efficiency of searching approach. Searching time is linear to exponential growth of documents. It is an efficient and better technique to address the problem of maintaining semantic relationship between plaintext and encrypted documents. Proposed system will contribute Synonym Based Fined-grained Multi-keyword Search Using Hierarchical Clustering.

II. RELATED WORK

A. Single keyword Searchable encryption

M. Bellare, et al. discussed about public key encryption scheme. This scheme is useful to achieve provable privacy in encryption algorithm. Encrypt-with-hash technique is introduced for secure deterministic encryption scheme. It replaces the coins used by standard encryption schemes with the hash of messages. An efficient searchable encryption primitive is defined which allows more efficient search ability ESE. Provable encryption gives the solution for sequential scan queries; it is one type of extension to the hash function [1].

Song et al. discussed about the concept of searchable encryption. They represented remote searching approach for encrypted data using untrusted server. In an encryption approach a single word is independently encrypted therefore cost required for scanning complete data collection word by word is more [2].

E. Goh [3], construct an efficient IND-CKA secure index known as Z-IDX. It uses pseudo-random functions and Bloom filters. It depicts the way of how to use Z-IDX to implement searches on encrypted data. The proposed scheme is more efficient for encrypted data search as it required O (1) search time / document. IND-CKA is efficient for handling compressed data, variable length words and regular expression queries. Search indexes are the natural expansion for the problem of developing data structures with privacy guarantee.

A searchable symmetric encryption discussed in [4]-[5]. It is used to outsource symmetric private information retrieval. They were expand OXT protocol from MC-SSE settings, in order to any doubtful relation in original clear-text database randomization of data locations is called in EDB i.e. encrypted database.

W. Sun, et al. [6], proposed privacy-preserving similarity based text retrieval technique. A vector space model is used. Also cosine measure is utilized in proposed work for secure searching result with accuracy. To achieve privacy meets two secure index schemes are used in two threat model. Tree based index search scheme is implemented for refinement of search efficiency. Finally, author represents the performance of their system with BMTS and EMTS in terms of search effectiveness, efficiency and privacy.

B. Multiple Keywords Searchable Encryption

F. Li, M. Hadjieleftheriou al. [7] discussed about outsource database i.e. ODB. They proposed a comprehensive evaluation of authenticated index structure based on variety of cost metrics. They extended their work to dynamic environment. Query formulation technique is used for query formulation. In future work they are planning to extend their idea to multidimensional structure with more types of queries. A secure KNN computation on encrypted databases is introduced in [8]. SCONEDB model is introduced in this paper to capture activities of users and attacker on encrypted database. Existing techniques such as, OPE for range queries also introduced in it. Symmetric encryption approach ASPE is used to preserve special type of scalar products. Security goal are also included as another component into SCONEDB model for future evaluation. Author C. Gentry represents a complete homomorphic encryption using ideal lattices [9]. In this paper author proposed a randomize algorithm due to lack of space. Introduced lattice based cryptosystem is typically used for decryption as it have ability of decryption. This scheme is not boots trappable i.e. i.e., the depth that the scheme can correctly evaluate can be logarithmic in the lattice dimension, just like the depth of the decryption circuit, but the latter is greater than the former. D. Boneh and G. Crescenzo proposed PEKS scheme for public key encryption with keyword search [10].

This scheme is related to IBE scheme i.e. Identity Based Encryption. But the problem is PEKS scheme is complicated to design. Author showed that PEKS implies Identity Based Encryption, but the converse is currently an open problem. There PEKS scheme required IBE construction to prove its security by exploiting more attributes.

Confidentiality-Preserving Rank-Ordered Search approach discussed in [11] by A. Swaminathan and Y.Mao. They construct a framework to maintain data confidentiality in ranked order search in large scale document. The proposed mechanism extracts the most relevant document from an encrypted collection based on the encrypted search queries. This technique attempts to bring together advanced information retrieval capabilities and secure search capabilities. In this paper author plan to focus on securing indices, other important security issues include protecting communication links and combining traffic analysis in future work. A secret key can produce tokens for testing any promoted query signify. Without analyzing any other information about plaintext the query significance token anyone test the predicate on a given ciphertext [12].

Analyzing security of searching on encrypted data a general framework is represented in this paper. There are protocols introduced in [13], for conjunctive search which is provably difficult for server to differentiate between encrypted keywords.

III. PROBLEM STATEMENT

"To design and develop synonym based fined-grained multi-keyword search using hierarchical clustering."

- To maintain semantic relationship between different plain documents over related encrypted document and improve the semantic search performance. Also provide fined-grained and synonym search over an encrypted documents.

IV. PROPOSED SYSTEM

In past few years there is rapid growth in the use of cloud storage. Before storing data on cloud data owner firstly encrypts it. Other user can download documents from cloud by performing keyword search over cloud. Previously there are certain approaches are available for searching such as, single keyword i.e. Boolean keyword search that does not maintain relationships between original and encrypted document. Correctness and privacy for search result is again another problem in keyword searching. Therefore MRSE-HCI can be better solution for earlier discussed problem. It also support for multikeyword synonym search.

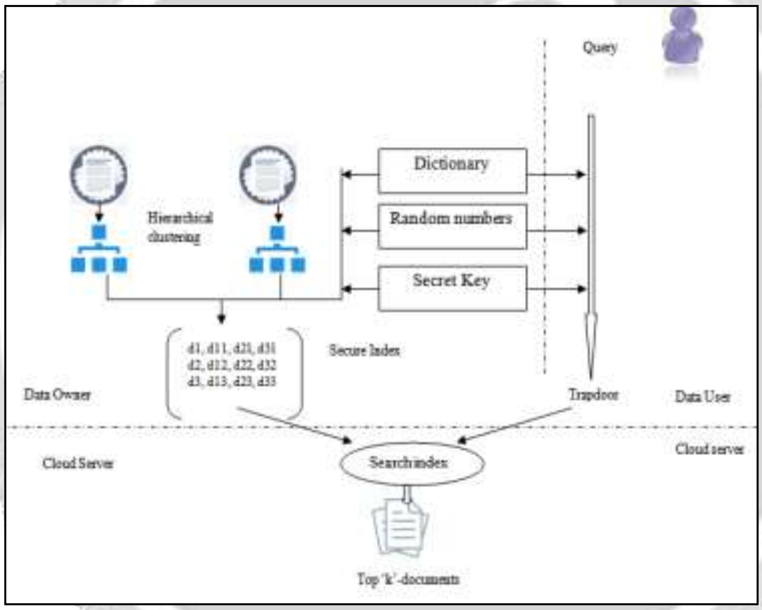

Figure.1: System Architecture

Following are system modules:

1. **File encryption and upload:**
   - Cloud user selects file to upload on cloud server. Using AES encryption algorithm file get encrypt and upload to server.
   - In proposed system, 128-bit AES algorithm is implemented.
   - After file encryption, it gets uploaded on cloud server.
2. **Index generation:**
   - While uploading file to cloud server index generation algorithm is used to extract index terms from file.
   - Document vector 'dv' is constructed for every document 'D'.
   - Hierarchical index is generated using QHC algorithm
3. **Cluster creation:**
   - For document index cluster creation QHC algorithm is implemented.
   - It is "Quality Hierarchical Clustering" algorithm

- With hierarchical clustering, times required for document searching get reduced and search efficiency gets improved.

4. **Download document:**
   - User request to download document by sending keyword trapdoor to server. As per search keywords, system performs searching over an encrypted document. It searches document into sub-category specified in search keywords.
   - Search result is displayed into inbox panel.
   - User can select file and download it by providing secret key.
5. **Decryption:** Before uploading data to cloud server, it get encrypted using AES256 () encryption algorithm. In document decryption, reverse of encryption is performed. To decrypt document AES256() decryption algorithm is used.
6. **Search:** There are three types of searching strategies are provided in proposed system such as:

   6.1. Complex logistic search: In this logical AND, OR and No operation query results are generated by system.

   AND: Result is produced which satisfies the conditions formed using AND operator.

   OR: Result is produced in which any one condition in OR-query is satisfied.

   No: Except the condition present in No-query results are generated.
7. **Synonym search:**
   - In synonym search approach, files which contain synonyms of search keywords are also displayed to end user.
   - A wordnet dictionary is used to find synonym from cloud stored data.

V. ALGORITHMS

1. Indexing Algorithm:

INPUT:

-Dataset D

-Secret key {Sk, k}

- PROCESSING:

Steps:

```
1. Select dictionary from given dataset D.
2. Construct document vector ‘dv’ for each document of D
3. Construct hierarchical index using QHC
4. Extend dimension of V in the DV and CCV i.e. document vector and cluster collection vector.
5. if $i^{th}$ of S=0
   then
   $V_i^{'}= V_i^{''}= V_i^{''}$
6. else
   $V_i^{'}$ is set to any random number
7. Encrypt index as, $\{M_1^T V^{'}, M_2^T V^{''},DC\}$
```

OUTPUT:

- Encrypted index.

2. Algorithm for QHC(Quality Hierarchical Clustering)

INPUT:

-set of documents

-set of threshold TH

PROCESSING:

Steps:

```
1. Construct cluster set $C_0$ in k-means cluster creation.
2. While there is new cluster set $C_i$
3. For every cluster $C_{i,j}$
4. if size of $C_{i,j}$>TH
5. Split this cluster into sub-cluster $C_{i+1}$ until all cluster match the size constraint.
```

OUTPUT:

QHC: quality hierarchical clustering

3. AES algorithm

3.1. AES encryption algorithm:

Input:

Plain text message m in Byte [] , Key k

Output:

Cipher text message in byte []

Processing:

```
1. Define 4 * 4 state array
2. Define constant Nr = 4, R=16
3. Copy m in state[]
4. Add each byte of state[] to key k using ⊕
5. For Nr-1 rounds
       Replace every byte in state[] with new value using lookup table
       Shift last 3 rows of state[] upside cyclically
       Combine last 4 columns of state[]
       Add each byte of state[] to key k using ⊕
   end For
6. Shift last 3 rows of state[] upside cyclically
7. Add each byte of state[] to key k using ⊕
8. Copy State[] to output[]
```

3.2. AES decryption algorithm:

Input:

Cipher text message C in byte [], Key k

Output:

Plain text message m in Byte []

Processing:

```
1. Define 4 * 4 state array
2. Define constant Nr = 4, R=16 ,
3. Copy C in state[]
4. Add each byte of state[] to key k using ⊕
5. For Nr-1 rounds Inverse Replace every byte in state[] with new value using lookup table Inverse Shift last 3 rows of
   state[] downside cyclically combine last 4 columns of state[] Add each byte of state[] to key k using L end For
6. Inverse Shift last 3 rows of state[] down word cyclically
7. Inverse Add each byte of state[] to key k using ⊕
8. Copy State[] to output[]
```

VI. MATHEMATICAL MODEL

S = {O, U, C} WHERE,
O = {OI, OO, OF} Data owner system
OI = {OI1,O I2,OI3, OI4} Data owner system input,
OI1 = Details of data owner
OI2 = Master Key
OI3= Documents
OI4 = Sharing Rights
OO = {OO1, OO2} Output of Owner system
OO1 = Encrypted document
OO2= Document Index
OF= {OF1, OF2, OF3, OF4, OF5, OF6} Functions in Owner System
OF1=Registration

OF2=Login
OF3=Upload document
OF4= Index Generation
OF5= Encrypt document
OF6=Save document on cloud
U = {UI, UO, UF} Users System
UI = {UI1, UI2} User system input,
UI1 = User details UI2=Search keywords
UO= {UO1, UO2, UO3} , Output of User System
UO1=Trapdoor
UO3 = Search Result i.e. synonym search and fine-grained search
UO1 = Decrypt file
UF = {UF1, UF2, UF3, UF4, UF5, UF6} Function in user system,
UF1 = Register
UF2 = login
UF3=Search document
UF3 = download document
UF4 = Trapdoor key
UF5 = decrypt document
UF6 = Save document
C = {CI, CO, CF} Cloud system
CI= {CI1, CI2, CI3, CI4} WHERE,
CI1 = {D1, D2....., Dn} Set of encrypted document
CI2= DI1, DI2…. .DIn set of Document Index
CI3 = Access Rights
CI4 = Search Keywords
CO= {CO1, CO2, CO3} WHERE,
CO1= {DC1, DC2 ...DCm} Document Cluster where each cluster DC1 contains {D1, D2 ...Dk} documents CO2=Index storage
CO3 = Search Result containing set of documents = {D1, D2 ...Di} top I matched document set
CF= {CF1, CF2, CF3, CF4, CF5, CF6, CF7, CF8} Function in user system,
CF1 = Register user
CF2 = Validate user
CF3=Save Index
CF4 = Dynamic Cluster Creation
CF5 = Save document
CF6=Index search using MRSE-HCI
CF7= Validate users rights
CF8 = Apply fine grained search

VII. EXPERIMENTAL SETUP

Windows 7 having Java platform with jdk 1.7.0 is used for system implementation. Client system is build using swing components. HTTP client facility is used for communication. Mysql database is used to store data. Apache tomcat server is used at server end. Eclipse and netbean-8.1 IDE are used to develop a system. Min 4GB RAM with min i3 processor configuration is required for system performance testing at client as well as server side.

Dataset: IEEE paper (PDF) dataset [15] is using to test the system performance. It is collected from the official website of IEEE. It contains PDF files of various domains.
At this initial level almost 300 IEEE papers downloaded.

VIII.RESULT TABLE AND DISCUSSION

TABLE I: CLUSTER CREATION

PDF Document	Cluster Creation Time (in milisec.)
200	0.35

400	0.39
600	0.43
800	0.47
1000	0.51

Table I represents the document index cluster generation time in milisec. To test the performance of cluster creation we have uploaded 200, 400, 600, 800 and 1000 PDF documents and take readings of it resp. As per cluster creation readings given in table 1, there is minor time difference for each group of PDF documents.

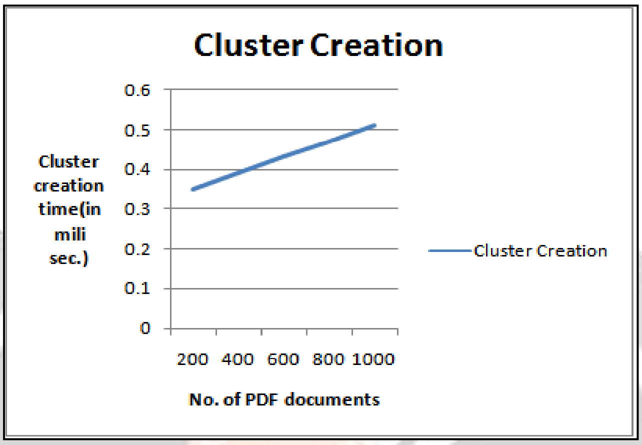

Figure.2: Graph of cluster creation

Figure 2 depicted the graphical form of cluster creation. X-axis represents No. of PDF documents taken as input for testing whereas, Y-axis represents the Cluster creation time in milliseconds.

TABLE II: SEARCH PERFORMANCE

	No. of Results			
	Existing system	**Proposed system**		
No. of documents	**OR**	**Synonym**	**AND**	**NO**
200	14	21	8	13
400	25	34	11	17
600	28	46	15	19
800	32	52	16	24
1000	42	62	23	31

Table II represents the search performance of proposed system. The existing system only outputs results for OR operation which is less as compared to proposed system (i.e. synonym search). In proposed system, complex logistic search with synonym search is contributed. Therefore, we test the search performance for logical AND & OR operation and synonym search. In synonym search, more PDF documents can retrieve as compared to results of logical AND & OR search.

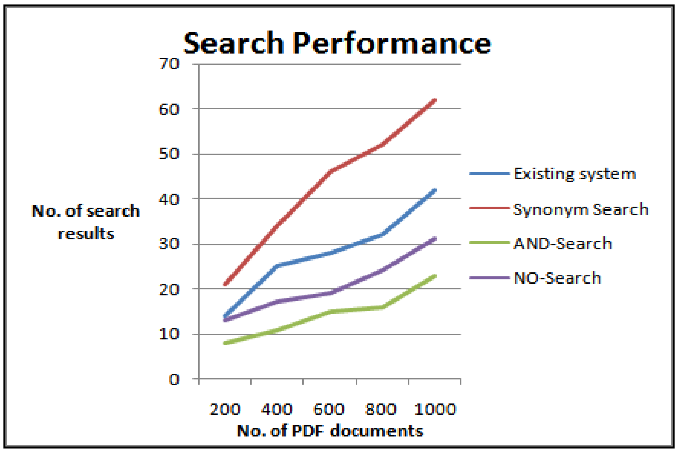

Figure 3: Graph of search performance

Figure 3 depicted the graph of search performance. In this, X-axis represents the no. of PDF documents and Y-axis represents the No. of search results.

TABLE III: SEARCH EFFICIENCY

No. of PDF documents	Existing System (in milisec.)	Proposed System (in milisec.)
200	0.23	0.25
400	0.27	0.28
600	0.31	0.32
800	0.34	0.37
1000	0.37	0.39

Table III represents comparative analysis of existing and proposed systems. Search time required for proposed system is bit increased as synonym and logistic search is provided in it.

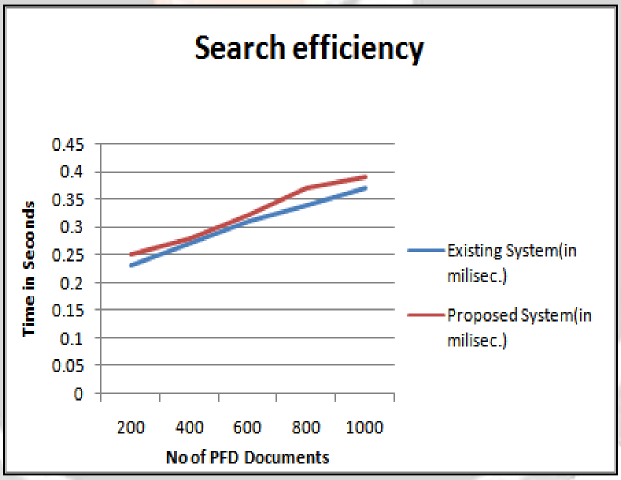

Figure 4: Graph of search efficiency

Figure 4 depicted the graph of search efficiency. Proposed system required more time for searching the documents as compared to existing system. X-axis represents the No. of PDF documents retrieved in searching whereas, Y-axis represents the time (in sec.) required for document search.

TABLE IV: EFFICIENCY EVALUATION

File size (in MB)	Encryption time (in sec.)	Index Generation time (in sec.)	Upload time (in sec.)	Download time (in sec.)	Decryption time (in sec.)
1	1.511	1.32	4.26	3.837	0.97
2	3.3	2.5	5.12	4.28	1.02
3	4.2	3.1	6.24	4.96	1.53

4	5.6	3.9	6.75	5.67	2.23
5	6.1	4.7	7.1	6.95	2.78

Table IV represents the performance of system efficiency evaluation. It represents the time (in sec.) required for document encryption, upload, download, decryption and index generation. For system efficiency testing 1MB to 5MB documents are taken. From the efficiency table 4, time required for file uploading is more than the other operations.

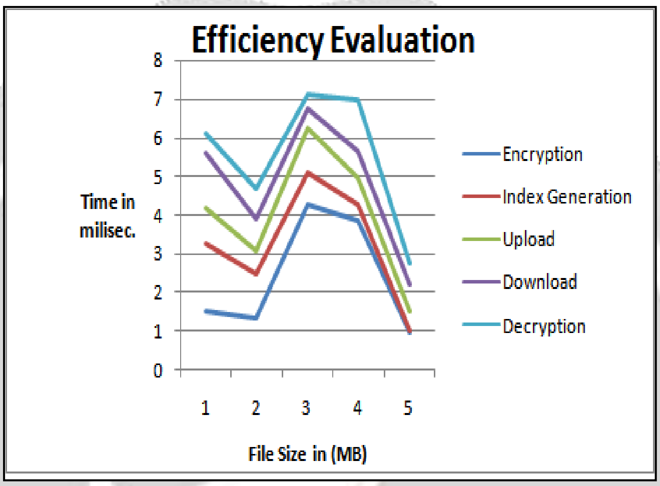

Figure 5 represents the graph of efficiency evaluation.

IX. CONCLUSION

We proposed MRSE-HCI technique for efficient searching over an encrypted data. It is also known as co-ordinate matching. In coordinate matching, multiple matches can possible by searching more relevant data documents to search query. As aggressive size increased in data documents searching phase reach to linear computational complexity. MRSE method can provide efficient search result than traditional search approach. As a part of contribution complex logistic with synonym based fined-grained searching facility is provided.

REFERENCES

[1] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and effi- ciently searchable encryption," in Proc. 27th Annu. Int. Cryptol. Conf. Adv. Cryptol., Santa Barbara, CA, 2007, pp. 535–552.

[2] D. X. D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Symp. Security Priv., BERKELEY, CA, 2000, pp. 44–55.

[3] E.-J. Goh, Secure Indexes, IACR Cryptology ePrint Archive, vol. 2003, pp. 216. 2003.

[4] S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, "Outsourced symmetric private information retrieval," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Nov. 2013, pp. 875–888.

[5] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M. C. Rosu, and M. Steiner, "Dynamic searchable encryption in very large databases: Data structures and implementation," in Proc. Netw. Distrib. Syst. Security Symp., vol. 14, 2014, Doi: http://dx.doi.org/ 10.14722/ndss.2014.23264.

[6] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proc. 8th ACM SIGSAC Symp. Inform., Comput. Commun. Security, Hangzhou, China, 2013, pp. 71–82.

[7] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, "Dynamic authenticated index structures for outsourced databases," in Proc. ACM SIGMOD, Chicago, IL, 2006, pp. 121–132.

[8] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in Proc. ACM SIGMOD Int. Conf. Manage. Data, Providence, RI, 2009, pp. 139–152.

[9] G. Craig, "Fully homomorphic encryption using ideal lattices," in Proc. 41st Annu. ACM Symp. Theory Comput., 2009, vol. 9, pp. 169–178

[10] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in Proc. 1st Int. Conf. Pairing-Based Cryptography, Tokyo, JAPAN, 2007, pp. 2–22.

[11] A. Swaminathan, Y. Mao, G. M. Su, H. Gou, A. Varna, S. He, M. Wu, and D. Oard, "Confidentiality-preserving rank-ordered search," in Proc. ACM ACM Workshop Storage Security Survivability, Alexandria, VA, 2007, pp. 7–12.

[12] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Proc. 4th Conf. Theory Cryptography, Amsterdam, NETHERLANDS, 2007, pp. 535–554

[13] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Proc. Proc. 2nd Int. Conf. Appl. Cryptography Netw. Security, Yellow Mt, China, 2004, pp. 31–45.

[14] C. Chen, X. Zhu,. P. Shen, J. Hu, "An Efficient Privacy-Preserving Ranked Keyword Search Method," IEEE Trans. Parallel Distrib. Syst., vol. 27, no.4, pp., Apr. 2016.

[15]ieeexplore.ieee.org