ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

Enabling Smart and Secure Multi-Keyword Fuzzy Search over Encrypted Outsourced Data

Anukrishna P. R[1] , Dr Vince Paul[2], Livya George[3]

P.G. Student, Dept. of Computer Engineering, Sahrdaya College of Engineering & Technology, Kerala, India[1]

Associate Professor, Dept. of Computer Engineering, Sahrdaya College of Engineering & Technology, Kerala, India [2]

Assistant Professor, Dept. of Computer Engineering, Sahrdaya College of Engineering & Technology, Kerala, India [3]

ABSTRACT: As Cloud Computing technology becomes more mature, many organizations and individuals are interested in storing more sensitive data in the cloud for great convenience and reduced cost in data management . Such sensitive data needs to be encrypted before it is outsourced to the cloud. Typically, the cloud servers also need to support a keyword search feature for these encrypted files. This work is extending the work of Xia[1] "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data" by incorporating the approximate keyword matching. Xia's schema includes secure multi-keyword ranked search scheme over encrypted cloud data, which simultaneously supports dynamic update operations like deletion and insertion of documents. Specifically, the vector space model and the widely-used TF*IDF model are combined in the index construction and query generation. We construct a special tree-based index structure and propose a Greedy Depth-first Search algorithm to provide efficient multi-keyword ranked search. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. These schemes support only exact keyword matching. As a result, if keywords are misspelled, incorrect results are returned. To overcome this issue, this work uses the concept of proposes an algorithm that, even if the keyword was misspelled, it returns the documents which has high probability.

KEYWORDS: Searchable encryption, multi-keyword fuzzy search, cloud computing

I. INTRODUCTION

Nowadays data is created constantly, and at an ever-increasing rate, all these and more create new data must be stored somewhere for some purpose. From 2010 amount of data is getting generated has become huge as everything and everyone is leaving a digital footprint. Cloud computing provides an attractive data storage and interactive pattern with advantages, including on-demand self-services, location individualistic resource pooling and omnipresent network access. These attracting features, both users and enterprises are inspired to outsource their data to the cloud, instead of getting software and hardware to manage the data. When an organization store hosts applications or data on cloud, it prevents the ability to have physical access to the servers hosting its information. However cloud service provider claims strong protection, security and privacy are the major issues in cloud computing. The best way for data confidentiality is encrypting the data before storing it in the cloud, which become a challenging task. In recent years many cryptographic algorithms is proposed for cipher text schemas. But existing keyword based information retrieval algorithms are used on pain text that cannot be applied on encrypted documents. Downloading all the data from the cloud and decrypting is practically impossible.

General purpose solutions to address above issues are fully-homomorphic encryption or oblivious RAMs. In fully homomorphic encryption is simple, given ciphertexts that encrypt $\pi 1,..., \pi t$, it should allow anyone to output a ciphertext that encrypts $f(\pi 1,..., \pi t)$ for any specific function f, as long as that function can be efficiently computed. No information about $\pi 1,..., \pi t$ or $f(\pi 1,..., \pi t)$, or any intermediate plaintext values, should leak; the inputs, output and intermediate values are always encrypted. An Oblivious RAM (ORAM) simulator is a compiler that transforms

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

algorithms in a way that the resulting algorithms preserve the input-output behaviour of the original algorithm. And the transformed algorithm's memory access pattern is independent of the memory access pattern of the original algorithm. These methods need massive operation and high computational cost for cloud server and user. Special purpose solutions to searchable encryption schema are more efficient in terms of functionality and security. When encrypted data is outsourced, searchable encryption schema allows the user to search on ciphertext domain. Many works have been proposed to attain this various search functionality such as single keyword search, similarity search, multikeyword boolean search, ranked search, multi-keyword ranked search, etc. In current scenario multi-keyword search is more useful for its effectiveness. Dynamic multi-keyword rank search has proposed with automatic update and delete operation. Most of these schemas has not supported efficient multi-keyword search. Our contributions include (1.) Efficient and secure multi-keyword fuzzy search using tree based schema with automatic update and deletion operations. That helps to reduce the search complexity. (2.) To provide multi-keyword fuzzy search, vector space model with the commonly-used "term frequency * inverse document frequency"(TF*IDF) model are joined in the index construction and query generation. In order to obtain high search efficiency, we proposes a "Greedy depth-first Search (GDFS)" algorithm based on this index tree. (3.) KNN algorithm is utilized to encrypt Index and query vectors.

II. RELATED WORK

Searchable encryption schemes empower the clients to store the encrypted data to the cloud and execute keyword search over cipher text domain. Searchable encryption schemes can be constructed either by using public key based cryptography or by symmetric key based cryptography.

A. *SINGLE KEYWORD SEARCHABLE ENCRYTION*

Song et al [2] first introduced the method of searchable encryption. In this each word in the document has encrypted independently. This is a symmetric searchable encryption (SSE) scheme, in which scanning is from whole data collection word by word. As a result the search time of their scheme is linear and searching cost is high. Goh et al [3] defined a secure index structure and formulate a security model for index known as semantic security to avoid adaptive chosen keyword attack (ind-cka). They proposes z-idx for efficient ind-cka secure index construction using pseudo-random functions and bloom filters. The search time of this scheme is O(n), where n is the cardinality of the document collection.

Cash et al [4] recently design and implement an efficient method. Few method lacks ranking mechanism that makes the users to take a long time to select what they want when massive documents contain the query keyword. To attain rank mechanism order-preserving techniques are utilized. Wang et al [5] developed encrypted invert index to achieve secure ranked keyword search over the encrypted data. In the keyword search phase, the cloud server calculates the relevance score between documents and the query. In this approach, relevant documents are ranked according to their relevance score and outputs the op-k results. Boneh et al [6] designed searchable encryption in public key settings. In search phase anyone can use public key to write to the data stored on server but only authorized data users owning private key can search.

B. *MULTIPLE KEYWORD SEARCHABLE ENCRYPTION*

Multi-keyword boolean search allows the users to input multiple query keywords to return suitable documents. This approach uses conjunctive keyword search or disjunctive keyword search. Conjunctive keyword search schemes return the documents that contain all of the query keywords. Disjunctive keyword search schemes return all the documents that contain a subset of the query keywords. Predicate search schemes support both conjunctive and disjunctive search. All these multi-keyword search schemes retrieve search results based on the existence of keywords, which cannot provide ranking functionality.

Cao et al. [7] introduces the first privacy-preserving multi-keyword ranked search scheme. It designed the documents and queries as vectors of dictionary size. The documents are ranked according to the number of matched query keywords based on coordinate matching. However, this scheme does not consider the priority of the different keywords, and thus is not accurate enough. Also the search efficiency of this scheme is linear with the cardinality of document collection. Sun et al. [8] proposed a secure multi-keyword search scheme that supports similarity-based

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

ranking. To provide ranking, a searchable index tree based on vector space model and cosine measure together with TF*IDF are utilize to provide ranking results. Sun et al. search algorithm attains better-than-linear search efficiency but results in precision loss.

Orencik et al. [9] proposed a secure multi-keyword search method which constructed local sensitive hash (LSH) functions to cluster the similar documents. The LSH algorithm is suitable for similar search schema but cannot provide exact ranking. In [10], Zhang et al. designed a scheme to deal with secure multi-keyword ranked search in a multi-owner model. In this scheme, different data owners use different secret keys to encrypt their documents and keywords and authorized data users can query without knowing keys of these different data owners. The authors proposed an Additive Order Preserving Function to return the most relevant search results. Drawback of these works is it doesn't support dynamic operations.

Practically, most of the data owner may need to update the document collection after uploading the collection to the cloud server. Thus, the secure encryption schemes are expected to support the insertion and deletion of the documents. There are many dynamic searchable encryption schemes. Song et al. [2] proposed a method which support update operation by considering each document as a sequence of fixed length words, and is individually indexed. However this work has low efficiency. Goh [3] proposed dynamic operations by updating of a Bloom filter along with the corresponding document. However, this scheme has linear search time and suffers from false positives. Kamara and Papamanthou [11] proposed a new search scheme to handle dynamic update on document data stored in leaf nodes based on tree-based index.In [12], Cash et al. presented a data structure T-set for keyword/identity. Then, a document can be represented by a sequence of independent T-Sets. Based on this structure, Cash et al. [13] proposed different dynamic searchable encryption scheme. In their construction, newly added tuples are stored and deleted tuples are recorded. The final search result is achieved through excluding tuples in the deleted list from the ones retrieved from original and newly added tuples.

C. *FUZZY KEYWORD MATCHING*

Wang et al.'s work [5] was one of the first works to address the problem of multi-keyword fuzzy search over encrypted data problem and did not require a predefined fuzzy set (referred to as MFSE). In MFSE, a keyword was first transformed into a bi-gram set and the Euclidean distance was used to capture keywords similarity. The MFSE subsequently used LSH functions from the same hash family to generate the Bloom filter based index and query. Due to the nature of LSH, even if the keyword was misspelled, it still could be hashed into the corresponding bits in the query vectors with high probability. Finally, this method used the inner product of the index vector and the query vector as the relevance score between queries and documents. This scheme solved the problems of multi-keyword fuzzy search with high efficiency and accuracy. The most important result was that their scheme does not require a predefined fuzzy set. However, some other problems arose in this scheme. First, converting the keyword into a bi-gram set will increases the Euclidean distance. For example, for the misspelled keyword "netward", the bi-gram set is {ne, et, tw, wa, ar, rd}. Two bigram sets are different compared with original sets, which means that the Euclidean distance between two vectors is 2, and the vectors are rarely be thresholded into the same bit. Second, the scheme is not effective for other spelling mistakes.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

III.THE SYSTEM AND THE THREAT MODELS

Entities involved in the system model are data owner, data user and cloud server, as illustrated in Fig 1.

Fig 1: System model of our schema

Data owner outsource the encrypted documents to the cloud server. In our scheme, the data owner initially creates a secure searchable tree index I from document collection F, and then generates an encrypted document collection C for F. Then data owner outsource the encrypted document collection C and tree index I to the cloud server and provides the key information of trapdoor generation and document decryption details to authorized data users. Data owner generates update information and send that to the server.

Data users are those have authorization to access the documents of data owner. Let t be the query keywords, the authorized user can generate a trapdoor TD according to search mechanisms to fetch k encrypted documents from cloud server. Data user can decrypt the documents with shared secret key.

Cloud server stores the encrypted document collection C and the encrypted searchable tree index I for data owner. After receiving the trapdoor TD from the data user, the cloud server searches the index tree I, and finally returns the corresponding collection of top-k ranked encrypted documents. If the data owner updates the document collection , according to the update information from the data owner, the server needs to update the index I and document collection C.

The cloud server in the proposed scheme is considered as “honest-but-curious”, which is employed in most of the works on cloud. Two threat model proposed by Cao et al. cloud server has adopted in this. Known ciphertext model:- In this model, the cloud server only knows the encrypted document collection C, the searchable index tree I, and the search trapdoor TD submitted by the authorized user. That is to say, the cloud server can execute ciphertext-only attack(COA) in this model. Known background model:- In model is equipped with more knowledge such as term frequency statistics.

IV. DESIGN GOALS

1) Support more spelling mistakes: Our multi-keyword fuzzy search scheme should support more spelling mistakes. For example, network security related files should be found for a misspelled query netward security,netwrok security,netrwork security and netwrk security.
2) Privacy guarantee: The cloud server should be prevented from obtaining additional information from the encrypted data files and the index.
3) No Predefined Dictionary: No predefined dictionary is a great contribution as our scheme should not have predefined dictionary.
4) Support updating: The same as original scheme, our scheme should support data setup dating, such as file adding, file deleting and file modifying.
5) Ranked results according to the relevance score: To make users more satisfied with search results, the return results should be ranked according to relevance score.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

6) Efficiency and Accuracy: The efficiency of our scheme should be same as the original scheme. And our scheme should be as accurate as possible and keep high accuracy.

V. THE PROPOSAL SCHEMAS

In this we describe fuzzy keyword search on both unencrypted and encrypted schema. Unencrypted scheme is constructed on the basis of vector space model and KBB tree.

A. Fuzzy search for unencypted and encrypted schema

Trapdoor is generating based on the query keywords to fetch k encrypted documents from cloud server. If the query keywords are misspelled normally no documents will be returning. But in fuzzy search , it checkes for typos or format inconsistencies. The fuzzy-search functionality calculates the edit distance of each keyword with all the keywords. The wrong keywords in the query are replaced with keywords from dictionary. Finally, the top-K files will be returned if it is larger than the threshold.

Algorithm

Step 1: For query keywords apply the Porter Stemming Algorithm to ascertain the root of the word. For example, for the following set of words: “talk”, “talks”, “talking” and “talked” all have a similar meanings, and their stem is “talk”.
Step 2: Query keyword is matched aganist the keyword set. If there is match set corresponding bit as 1. If there is no match then it has to follow fuzzy keyword search.
Step 3: To achieve fuzzy keyword search corresponding query keyword is first transformed into the bigram-based set. For example, the bi-gram set of the keyword “secure” is {se, ec, cu, ur, re}. That will be compared aganist the bi-gram set of keyword set. Score corresponding to each keyword will be noted. This score is a good measure for evaluating the matching keywords. The keyword which has highest score will be replaced with mismatched query keyword and bit will be set as 1 in the trapdoor.
Step 4: The trapdoor is uploaded to the server and server perform the searching on corresponding owner's index tree. Top k results are returned if available.

B. Unencrypted dynamic multi-keyword Fuzzy search (UDMFS)

In this first generate a tree node for each document in the collection. These nodes are the leaf nodes of the index tree. Then, the internal tree nodes are generated based on these leaf nodes. The data structure of the tree node is defined as <ID,D,P_l, P_r, FID> where P_l is left pointer, P_r is right pointer. All keywords in all documents are stored as dictionary after applying stemming algorithm. The search process is based on the “Greedy Depth-first Search” algorithm. We build a result list represented as RList, whose element is defined as <RScore,FID> , where Rscore is the relevant score for the document FID. The cloud server computes the relevance score of node in the index tree to the query. The document with high Rsore will returns as the result.

Algorithm

Step 1: if the node u is not a leaf node then
Step 2: if RScore(Du;Q) > kthscore then
Step 3: GDFS(u:hchild);
step 4: GDFS(u:lchild);
5: else
6: return
7: end if
8: else
9: if RScore(Du;Q) > kthscore then
10: Delete the element with the smallest relevance score from RList;
11: Insert a new element ⟨ RScore(Du;Q); u:FID ⟩ and sort all the elements of RList;

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

12: end if
13: return
14: end if

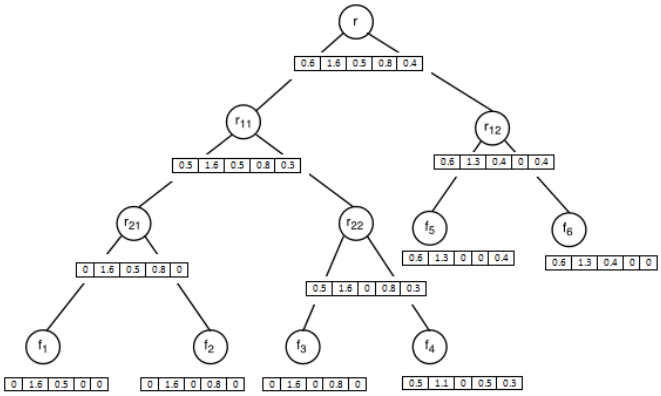


Fig 2: Index tree

This fig 2 also shows an example of search process, in which the query vector Q is equal to (0,1,0,1,0). In this example, we set the parameter k = 2 with the meaning that two documents will be returned to the user. According to the search algorithm, the search starts with the root node, and reaches the first leaf node f1 through r11 and r21. The relevance score of f4 to the query is 1.6. After that, the leaf nodes f2 and f3 are successively reached with the relevance scores 2.4 and 2.4. Next, the leaf node f4 is reached with score 1.6. Finally, the algorithm will try to search subtree rooted by r12, and find that there are no reasonable results in this subtree because the relevance score of r12 is 1.3, which is smaller than the smallest relevance score in RList.

C. Basic dynamic multi-keyword Fuzzy search (BDMFS)

In this encrypted schema data owner generates secret key set (SK) which contains m-bit vector S where m is equal to the cardinality of dictionary and m*m invertible matrices M_1 and M_2. Namely, SK = {S, M_1, M_2} .Data owner generates two random vectors {D_u', D_u''} for index vector D_u in each node , according to the secret vector S. If s[i] = 0 then value of two random vectors D_u' and D_u'' will be equal to D_u. If s[i] = 1 then D_u' and D_u'' will be set as two random values whose sum equals to D_u[i]. Finally, the encrypted index tree I is built where the node u stores two encrypted index vectors $I_u = \{M_1^T D_u', M_2^T D_u''\}$.

For generating the trapdoor the query vector Q is split into two random vectors Q' and Q''. The difference is that if S[i] = 0, Q'[i] and Q''[i] are set to two random values whose sum equals to Q[i]; else Q'[i] and Q''[i] are set as the same as Q[i]. Finally, the algorithm returns the trapdoor TD = $\{M_1^{-1}Q', M_2^{-1}Q''\}$.

Rscore is calculated as follows

$$
\begin{aligned}
& I_u \cdot TD \\
&= \left(M_1^T D_u'\right) \cdot \left(M_1^{-1} Q'\right) + \left(M_2^T D_u''\right) \cdot \left(M_2^{-1} Q''\right) \\
&= \left(M_1^T D_u'\right)^T \left(M_1^{-1} Q'\right) + \left(M_2^T D_u''\right)^T \left(M_2^{-1} Q''\right) \\
&= D_u'^T M_1 M_1^{-1} Q' + D_u''^T M_2 M_2^{-1} Q'' \\
&= D_u' \cdot Q' + D_u'' \cdot Q'' \\
&= D_u \cdot Q \\
&= \mathrm{RScore}(D_u, Q).
\end{aligned}
$$

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

D. Enhanced dynamic multi-keyword fuzzy search (EDMFS) scheme

Enhanced EDMFS schema is used to improve the security by introducing tunable randomness to protect the relevance score calculation method. Enhanced EDMFS scheme is almost the same as BDMRS scheme except few things.

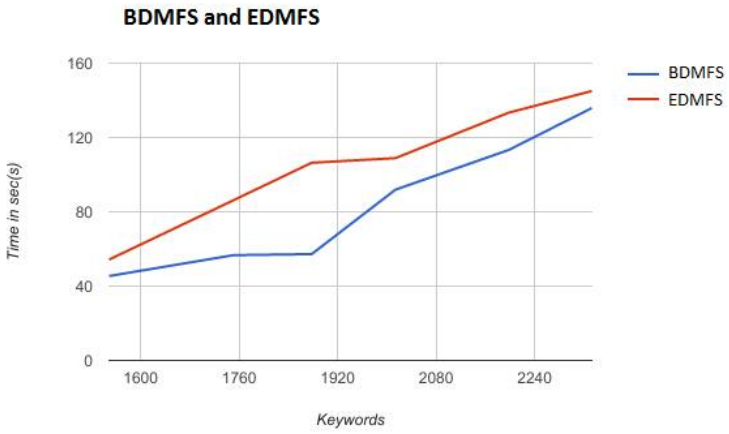
The secret vector SK = {S, M_1, M_2}in this M1 and M2 are (m + m') x (m + m')invertible matrices, where m' is the number of phantom terms. Before encrypting the index vector D_u, we extend the vector D_u to be a (m+m') dimensional vector. The query vector Q is extended to be a (m + m') dimensional vector.

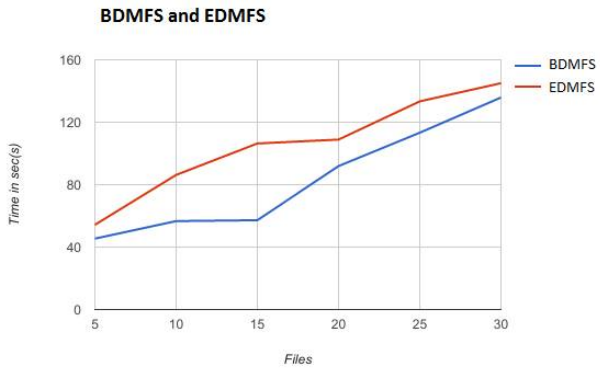
VI PERFORMANCE ANALYSIS

We implement the proposed scheme using java language in Windows 10 operation system. During the search process, if the relevance score at node u is larger than the minimum relevance score in result list RList, the cloud server examines the children of the node; else it returns. Thus, lots of nodes are not accessed during a real search. In the case of fuzzy search has high influence on the time complexity of increases when number of keyword increases.

Efficiency of index tree generation :

As shown in the graph when the number of keywords increases effiency of the index tree generation is better for EDMFS schema than BDMS. But fuzzy search time was independent of the number of query keywords to a large extent.





ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

VI. CONCLUSION AND FUTURE WORK

In this paper, we investigate the problem of multi-keyword fuzzy ranked search over encrypted cloud data. We propose a multi-keyword fuzzy ranked search scheme and develop a novel method of keyword transformation and introduce the stemming algorithm. With these two techniques, the proposed scheme is able to efficiently handle more misspelling mistake. Moreover, our proposed scheme takes the keyword weight into consideration during ranking. Our proposed scheme does not require a predefined keyword set and hence enables efficient file update too. We also give thorough security analyses and conduct experiments on real world data set, which indicates the proposed scheme's potential of practical usage.

Further, this research can be enhanced in future in the following directions: Firstly, user search experience can be further enhanced by focusing on syntactic transformations. Secondly, ranking can be improved using subject content based ranking, anaphora resolution, and plural considerations.

REFERENCES

[1] Zhihua Xia, Xinhui Wang, Xingming Sun, and Qian Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data" , in IEEE Transcations , 2016
[2] D. X. D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data", in Proc. S P 2000, Berkeley, CA , 2000.
[3] E.-J. Goh, Secure Indexes, IACR Cryptology ePrint Archive, IEEE TKDE , 2003.
[4] Cash, D., Jaeger, J., Jarecki, S., Jutla, C., Krawczyk, H., Rosu, M. C., and Steiner, M. "Dynamic searchable encryption in very large databases: Data structures and implementation.", In Proc. of NDSS , 2014.
[5] C. Wang, N. Cao, J. Li, K. Ren, and W. J. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data", in Proc. ICDCS, Genova, Italy , 2010.
[6] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search", in Proc. Eurocrypt, Interlaken, Switzerland , 2004.
[7] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data", in Proc. IEEE Infocom, 2011.
[8] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multikeyword text search in the cloud supporting similarity-based ranking", in Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. secur., 2013.
[9] C. Orencik, M. Kantarcioglu, and E. Savas, "A practical and secure multi-keyword search method over encrypted cloud data", in Proc. IEEE 6th Int. Conf. Cloud Comput., 2013.
[10] W. Zhang, S. Xiao, Y. Lin, T. Zhou, and S. Zhou, "Secure ranked multi-keyword search for multiple data owners in cloud computing", in Dependable Syst. Networks (DSN), IEEE 44th Annu.IEEE/IFIP Int. Conf., 2013.
[11 S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption", in Proc. Financ. Cryptography Data Secur., 2013.
[12] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries", in Proc. Adv. Cryptol., 2013.