RANKED PHRASE SEARCH USING ORDER PRESERVING ENCRYPTION FOR CLOUD DOCUMENTS

N.Jayashri, T.Chakravarthy.

Abstract - Cloud Computing has been considered as the next-generation architecture of IT Enterprise. This exclusive paradigm produces many new security issues, which have not been well recognized. This paper presents a new framework for privacy preserving multikeyword rank-ordered search and retrieval over large document collections. The proposed framework not only protects document/query privacy against an outside intruder, but also prohibits an untrusted data centre from learning information related to the query and the document collection. We introduce practical methods for proper combination of relevance scoring methods and cryptographic techniques, such as order preserving encryption, to protect data collections and indices and provide efficient and accurate search capabilities to securely rank-order documents in response to a query and related document collection. The proposed methods thus form the steps to bring together advanced information retrieval and secure search capabilities for a wide range of applications including managing data in government and business operations, enabling scholarly study of sensitive data, and facilitating the document discovery process in litigation.

Index Terms - Cloud Computing, Data Storage, Document Retrieval, Order Preserving Encryption, Ranked Retrieval, Searchable Encryption, Secure Index.

1 INTRODUCTION

Cloud computing is receiving a great deal of attention, both in publications and among users. Yet it is not always clearly defined [1]. Cloud computing is a subscription-based service where you can obtain networked storage space and computer resources. One way to think of cloud computing is to consider our experience with email [2]. The cloud makes it possible for you to access your information from anywhere at any time. The cloud provides convenient, on-demand network access to a centralized pool of configurable computing resources that can be rapidly deployed with great efficiency and minimal management overhead.[3].

Although cloud computing's benefits are tremendous, security and privacy concerns are the primary obstacles to wide adoption[4]. Because cloud service providers (CSPs) are separate administrative entities, moving to the commercial public cloud divests users of direct control over the systems that manage their data and applications.[5]. Cloud users can face severe constraints in moving their data from one cloud provider to another and find themselves locked in. [6]. As individuals and enterprises produce more and more data that must be stored and utilized, they're motivated to outsource their local complex data management systems to the cloud owing to its greater flexibility and cost-efficiency.[5]. Ultimately, the cloud is neither good nor bad: it's just a new paradigm with its own advantages and disadvantages.[6]. However, once users no longer physically possess their data, its confidentiality and integrity can be at risk[7].

Traditionally, data centers consist of large collections of server farms implementing perimeter-security measures including firewalls, demilitarized zones, intrusion-detection-and-prevention systems, and network-monitoring tools.[8] Administrative access typically is through a LAN to limit external access[10]. However, virtualization has provided the mechanism to shrink this configuration. A single server can now provide multitenant services; in a public-cloud environment, the concept of the network perimeter evaporates.[9]. The public cloud offers user access via the Internet, and cloud subscribers conduct administrative activities in this environment. This paradigm in itself introduces security risks because this remote access provides exposure to potential cyberattackers.[10]

For the former concern, data encryption before outsourcing is the simplest way to protect data privacy and combat unsolicited access in the cloud and beyond[5]. But encryption also makes deploying traditional data utilization services — such as plaintext keyword search over textual data or query over database — a difficult task. The trivial solution of downloading all the data and decrypting it locally is clearly impractical, due to the huge bandwidth cost resulting from cloudscale systems[11]. Moreover, aside from eliminating local storage management, storing data in the cloud serves no purpose unless people can easily search and utilize that data.[12] This problem on how to search encrypted data has recently

* N.Jayashri. Research Scholar. Dept. of Computer Science. AVVM Sri Pushpam College. Thanjavur, India. jayashri_13@yahoo.co.in.
* T.Chakravarthy. Asso. Professor. Dept. of Computer Science. AVVM Sri Pushpam College. Thanjavur, India. tcvarthy@gmail.com.

gained attention and led to the development of *searchable encryption* techniques[5].

Song et al. [13] first introduced the notion of searchable encryption. They proposed a scheme in the symmetric key setting, where each word in the file is encrypted independently under a special two-layered encryption construction. Thus, a searching overhead is linear to the whole file collection length. At a high level, a searchable encryption scheme employs a prebuilt encrypted search index that lets users with appropriate tokens securely search over the encrypted data via keywords without first decrypting it[5].In this context, numerous interesting yet challenging problems remain, including similarity search over encrypted data, secure ranked search over encrypted data, secure multikeyword semantic search, secure range query, and even secure search over nontextual data such as graph or numerical data.

Our Contributions can be summarized below

In this work we assume that the document and index are encrypted with the OPE method. We just concentrate on effective searching methodology to locate the specific document with minimal overhead and low time consume.

a. Early method perform exact keyword match, in this work we are trying to implement ranked phrase search with help of Natural Language Processing-Information Retrieval (NLP-IR) system.
b. Instead of TF x IDF we use new term weighting scheme to calculate relevance score for improving the document retrieval accuracy.

The rest of the paper is organized as follows: Existing works in Order Preserving Encryption in single keyword search is discussed in Section 2. Section 3, explain about our proposed work. Discussion about present work is done in section 4. Section 5 List, some of the Searchable Encryption techniques. Finally Section 7 gives the conclusion of the whole work done in this paper.

2 EXISTING WORK

Searchable encryption is still far from providing the same search usability, functionality, and flexibility as in plaintext search. How to create the same search experiences over encrypted cloud data for users, while providing the security and privacy guarantees? To enable semantic –rich encrypted search over largescale cloud data. Order Preserving Encryption(OPE) can be viewed as a tool somewhat similar to fully-homomorphic encryption, in that it can repeatedly operate on encrypted data. It is weaker than FHE since the manipulation primitive is limited to equality checking and comparisons.[14]

2.1 Order Preserving Symmetric Encryption

The OPSE is a deterministic encryption scheme where the numerical ordering of the plaintexts gets preserved by the encryption function. Boldyreva et al. [15] gives the first cryptographic study of OPSE primitive and provides a construction that is provably secure under the security framework of pseudorandom function or pseudorandom permutation. Namely, considering that any order-preserving function g(.) from domain D={1,....M} to range R={1,.....,N}can be uniquely defined by a combination of M out of N ordered items, an OPSE is then said to be secure if and only if an adversary has to perform a brute force search over all the possible combinations of M out of N to break the encryption scheme. If the security level is chosen to be 80 bits, then it is suggested to choose M = N/2 > 80 so that the total number of combinations will be greater than 2^{80}. Their construction is based on an uncovered relationship between a random order-preserving function (which meets the above security notion) and the hypergeometric probability distribution, which will later be denoted as HGD. We refer readers to [15] for more details about OPSE and its security definition. At the first glance, by changing the relevance score encryption from the standard indistinguishable symmetric encryption scheme to this OPSE, that efficient relevance score ranking can be achieved just like in the plaintext domain.

2.2 Ranked Keyword Search

C.Wang et. al.[16] try to solve the problem of supporting efficient ranked keyword search for achieving effective utilization of remotely stored encrypted data in Cloud Computing. We first give a basic scheme and show that by following the same existing searchable encryption framework, it is very inefficient to achieve ranked search. Then appropriately weaken the security guarantee, resort to the newly developed crypto primitive OPSE, and derive an efficient one-to-many Orderpreserving mapping function. [16] also investigate some further enhancements of our ranked search mechanism, including the efficient support of relevance score dynamics, the authentication of ranked search results. Through thorough security analysis, C.Wang et. al. [16] show that their solution is secure and privacy preserving, while correctly realizing the goal of ranked keyword search. Extensive experimental results demonstrate the efficiency of their solution.

3 PROPOSED WORK

We are selecting Order Preserving Symmetric Encryption (OPSE) scheme as our methodology to secure Inverted Index which contain terms, rank scores, and posting list which contain corresponding document IDs. Steps involved in creating Inverted Index is explained below:

3.1. Phrase Selection

Syntactic phrases obtained from the parse structures are denoted as head-modifier pairs. The head in such a pair is a vital element of a phrase, whereas the modifier is one of the adjuncts or arguments of the head. In the TREC researches described here we obtained head-modifier word pairs only, i.e., nested pairs were not used though this was accepted by the size of the database.

Figure 1 show all steps involved in initial linguistic analysis of a sample sentence from the database. From this structure, we obtain head-modifier pairs that develop into candidates for compound terms. In common, the following kinds of

pairs are taken: (i) a head noun of a noun phrase and its left adjective or noun adjunct, (ii) a head noun and the head of its right adjunct, (iii) the main verb of a clause and the head of its object phrase, and (iv) the head of the subject phrase and the main verb.

These kinds of pairs report for most of the syntactic alternatives for linking two words into pairs holding compatible semantic substance. For example, the pair retrieve + information will be obtained from one of the following piece of sentences: retrieval of information from databases; information retrieval system; and information that can be accessed by a user-controlled interactive search process. We also tried to recognize and purge any terms which were openly invalid in order to avoid matches against their positive counterparts, either in the database or in the queries.

3.2. Term Weighting Issues

Selecting a suitable term weighting scheme is difficult in term-based retrieval while the rank of a document is decided by the weights of the terms it shares with the query. One standard term weighting method is known as tf.idf. In the official TREC they used the normalized tf.idf weights for all terms identical: single 'ordinary-word' terms, correct names, in addition to phrasal terms consisting of 2 or more words. Each time phrases were added in the term set of a document, the size of this document was raised consequently. This had the impression of reducing tf factors for 'regular' single word terms. A traditional tf.idf weighting scheme may be unsuitable for mixed term sets, containing of proper names, ordinary concepts, and phrases, because:

(i) It supports terms that appear fairly repeatedly in a document, which favours only general-type queries. Such queries were not usual in TREC. (ii) It appends low weights to uncommon, extremely precise terms, such as phrases and names, whose only appears in a document are often significant for relevance. Note that such terms cannot be consistently differentiated using their circulation in the database as the exclusive factor, and therefore syntactic and lexical information is expected.
(iii) It does not solve the issue of inter-term dependencies occurring when phrasal terms and their piece of single word terms are all incorporated in a document illustration, i.e., launch + satellite and satellite are not unrelated, and it is uncertain whether they should be calculated as two terms. Systematically, the new weights for phrasal and extremely precise terms are acquired by using the below mentioned formula, while weights for most of the single-word terms reside unaffected:

$$weight\ (W_i) = (\ C_1 * log\ (tf\ ) + C_2 * \alpha(N, i)\ ) * idf \text{----------} (1)$$

In the above, $\alpha(N,i)$ is 1 for i <N and is 0 otherwise. Table 1 demonstrate the outcome of differential weighting of

```
INPUT SENTENCE
The former Soviet president has been a local hero
ever since a Russian tank
invaded Wisconsin.

TAGG ED SENTENCE
The/dt former/jj Soviet/jj president/nn has/vbz
been/vbn a/dt local/jj hero/nn ever/rb since/in
a/dt Russian/jj tank/nn invaded/vbd Wisconsin/np

TAGGED & STEMMED SENTENCE
the/dt former/jj soviet/jj president/nn have/vbz
be/vbn a/dt local/jj hero/nn ever/rb since/in a/dt
russian/jj tank/nn invade/vbd wisconsin/np ./per


PARSED SENTENCE
{assert
  {{perf {HAVE}} {{verb {BE}}
    {subject
    {np {n PRESIDENT} {t_pos THE}
      {adj {FORMER}} {adj {SOVIET}}}}}
    {object
      {np {n HERO} {t_pos A} {adj
       {LOCAL}}}}
    {adv EVER}
    {sub_ord{ SINCE
        {{verb {INVADE}}
          {subject {np {n TANK} {t_pos A{
            {adj {RUSSIAN}}}}}
          {object {np {name
            {WISCONSIN}}}}}}}}}

EXTRACTED TERMS & WEIGHTS
president              2.623519
tank                   6.848128
president+former       14.594883
soviet                 5.416102
hero                   7.896426
tank+russian           16.030809
russian                7.383342
president+soviet       11.556747
invade                 8.435012
hero+local             14.314775
tank+invade            17.402237
Wisconsin              7.785689
```

Fig 1. Stages of Sentence Processing Algorithm.

phrasal terms using concept 101 and a relevant document as an example. Table 2 presents how ranks of the relevant documents vary when phrasal terms are used with the new weighting method. Modifying the weighting method for compound terms has directed to an overall raise of precision of more than 20% on official TREC-2 ad-hoc results. Table 3 reviews statistics of the executions for queries 101-150 against the WSJ database, both with new weighting method and with the traditional tf.idf weighting.

3.3. Retrieving Hot Spot

One more complexity with frequency-based term weighting occurs when a lengthy document wants to be accessed on the basis of some short relevant sentences. If the immensity of the document is not completely relevant to the query, then there is a intense possibility that the document will obtain low score in the final ranking, although some robustly relevant material in it. Topic 101 matches WSJ870226-0091 duplicate terms not shown

TERM	*TF.IDF*	*NEW WEIGHT*
weapon	1639	1639
missile	872	872
laser	1456	1456
interceptor	2075	2075
space+base	2641	2105
system+defense	2846	2219
reentry+vehicle	1879	3480
exoatmospheric	1879	3480
system+interceptor	2526	3118
initiative+defense	1646	2032
DOC RANK	**30**	**10**

Table 1. The result of differential term weighting method.

DOC ID	OLD RANK	NEW RANK
WSJ891004-0119	7	1
WSJ880609-0061	53	50
WSJ891005-0005	15	4
WSJ891005-0001	283	72
WSJ870213-0053	10	12
WSJ891009-0009	35	18
WSJ870723-0064	8	8
WSJ891009-0188	73	46
WSJ890928-0184	40	61

Table 2. Rank variations for related documents for Subject 104 when phrasal terms are employed in retrieval.

This difficulty can be handled with by subdividing lengthy documents at paragraph breaks, or into roughly equal length fragments and indexing the database with regard to these (e.g., [18]). While such methodologies are efficient, they also cause to be expensive because of enlarged index size and more problematical access methods.

Effectiveness considerations have brought us to examine a different method to acquire **hot spot** which would not demand re-indexing of the old database or any modifications in document retrieval. In this method, the highest number of terms on which a query is allowed to match a document is restricted to N highest weight terms, where N can be identical for all queries or may differ from one query to another. Note that this is not as similar as just taking the N top terms from every query. Rather, for each and every document for which there are M matching terms with the query, simply min (M, N) of them, namely those which have maximum weights, will be taken when calculating the document score. Moreover, only the overall significance weights for terms are considered, while local in-document occurrence is hidden by either getting a log or substituting it with a constant.

3.4 Inverted Index Creation,

The above explained NLP-IR system use head-modifier word pairs as a index terms, new weighting scheme (Eq.1.) calculate the relevance score between query terms and documents, we also find Hot-Spots to optimize ranking score. Actually this is the first step for our proposed work, now we just retrieve index terms, relevance score. According to the relevance score documents are also ranked. Based on these details we are create posting entries to link documents and their corresponding terms and relevance score. All these items are organized into index. Below mentioned Build-index algorithm is used to construct Index.

BuildIndex()

a. Initialization

i) Scan document extract the distinct words.
ii) Find head-modifier pairs.
iii) Find Hot-Spot

b. Posting the Entries.

For each term

i) Derive the relevance score between document and term according to equation(1).

ii) Apply OPE encryption over relevance score , find corresponding Document identifier (Document contain the query term), posting these entries in the Index.

c. Output Index I

Finally documents which are encrypted using traditional encryption schemes and inverted index encrypted using OPE are placed in the cloud server for further accessing.

4 DISCUSSION

One complexity in acquiring head-modifier pairs of maximum precision is the disreputable insecurity of insignificant compounds. The pair extractor looks at the distribution informations of the compound terms to choose

whether the combination of any two words in a noun phrase is both semantically significant and syntactically formal. Besides, phrases with a significant number of incidences across various documents, comprising those for which no obvious disambiguation into pairs can be acquired, are added as a third level of index.

Rather than take effort to determine anaphoric references, we altered the weighting method so that the phrases were more solidly weighted by their idf scores while the in-document occurrence scores were substituted by logarithms multiplied by suitably big constants. As well as, the top most N idf matching terms were calculated more toward the document score than the remaining terms. The outcome of 'hot spot' accessing is shown in Table 4 in the ranking of related documents within the top 30 retrieved documents for subject 72. The final ranking is acquired by appending the scores of documents in regular tf.idf ranking and in the hot-spot ranking. The hot spot weighting is denoted with the α factor in the term weighting formula given in the section3.2.Although some of the recall may be gave up the joint ranking precision has been steadily better than in either of the original rankings: an average perfection is 10-12% above the tf.idf run precision.

Compared to the original SSE, the OPE scheme appends the encrypted relevance scores in the searchable index in addition to document ID. Thus, the encrypted scores are the only supplementary information that the intruder can utilize against the security guarantee, i.e., keyword privacy and document confidentiality. Due to the security strength of the document encryption scheme, the document content is clearly well secured. Thus, we only need to concentrate on keyword privacy. We know that as long as data owner properly chooses the range size R sufficiently large, the encrypted scores in the searchable index will only be a sequence of order-preserved numeric value with extremely low duplicates. Though adversary may learn partial information from the duplicates, the fully randomized score-to-bucket assignment and the highly flattened one-to-many mapping still makes it difficult for the adversary to predict the original plaintext score distribution[16] Thus, the keyword privacy is also well preserved in our work.

5 RELATED WORK

Goh [19], which presents a construction that uses per-document indexes derived from Bloom filters [20]. There, each word in the document is processed using a pseudo-random function and then inserted into a Bloom filter. The client then provides a trapdoor consisting of an indicator of which bits in the filter should be tested, thereby resulting in constant per-document search time. Moreover, Goh's work also introduced the notion of semantic security in opposition to chosen-keyword attacks (called IND-CKA), which is the first formal notion of security defined for searchable encryption.

As discussed earlier, neither of these schemes allow users to perform Boolean keyword searches securely and efficiently. This shortcoming was first addressed by Golle, Staddon and Waters in [21], where they present two solutions that achieve the desired level of security. The first is provably secure under the Decision Diffie-Hellman assumption [22] and requires two modular exponentiations per document for searching. Additionally, the size of the trapdoors is linear in the number of documents being searched. Park, Kim and Lee proposed the first public-key searchable encryption schemes [22,23,24] that allow for secure conjunctive keyword searches [25]. Boolean systems were first developed and marketed over 30 years ago at a time when computing power was minimal compared with today. Because of this, these systems require the user to provide sufficient syntactical restrictions in their query to limit the number of documents retrieved, and those retrieved documents are not ranked in order of any relationship to the user's query. Although the Boolean systems offer very powerful on-line search capabilities to librarians and other trained intermediaries, they tend to provide very poor service to end-users, particularly those who use the system on an infrequent basis (Cleverdon 1983)[26].

To apply the searchable encryption to cloud computing, some researchers have been studying further on how to search over encrypted cloud data efficiently. Li et al. [27] firstly proposed a fuzzy keyword search scheme over encrypted cloud data, which combines edit distance with wildcard-based technique to construct fuzzy keyword sets, to address problems of minor typos and format inconsistence. Wang et al. [15] proposed a secure ranked search scheme, in which through giving each keyword weight by TF-IDF, under the help of the order preserving symmetric encryption, the cloud server can rank relevant data files with no knowledge of specific keyword weigh. But this scheme supports only single keyword search. Then Cao et al. [29] proposed a privacy preserving ranked scheme supporting multi-keyword, which uses vector space model and characteristics of matrix to realize trapdoor unlinkablility and thereby preserves data privacy. Sun et al. also propose a secure multi-keyword ranked search scheme based on vector space model (VSM). The VSM can measure the similarity between document index vector and query vector and hence support more accurate ranked search result.

6 CONCLUSION

Most relevant document retrieval is the biggest overhead today. Compare with existing approaches they are using TF X IDF score to rank keywords. Both single and multikeyword search verifying the same TF X IDF score belongs to given word(s). In our work we are try to achieve phrase search over encrypted documents. For that concern first we are retrieve head_modifier pairs, in the next step hot-sopts are retrieved, according to these two factors we are calculating the relevance score instead of common TF

and IDF. We demonstrated that natural language processing can now be done on a fairly large scale and that its speed and robustness can match those of traditional statistical programs such as key-word indexing or statistical phrase extraction. We suggest moreover that when properly used natural language processing can be very effective in improving retrieval precision. In particular, we show that in term-based document representation, term weighting is at least as important as their selection. In order to achieve optimal performance terms obtained primarily through the linguistic analysis must be weighted differently than those obtained through traditional frequency-based methods. So the proposed scheme perform well and also increase accuracy of retrieval over encrypted data ,with the help of new NLP_IR term weighting scheme in our environment.

REFERENCES

1. Lewis, Grace. Cloud Computing: Finding the Silver Lining, Not the Silver Bullet. http://www.sei.cmu.edu/newsitems/cloudcomputing.cfm (2009).
2. A.Huth and J.Cebula. "The Basics of Cloud Computing". 2011. Carnegie Mellon University. Produced for US-CERT, a government organization.
3. P. Mell and T. Grance, "The NIST Definition of Cloud Computing," US Nat'l Inst. of Science and Technology, 2011; http://csrc.nist.gov/publications/nistpubs/800-45/SP800-145.pdf.
4. "Security Guidance for Critical Areas of Focus in Cloud Computing," Cloud Security Alliance, Dec. 2009; https:// cloudsecurityalliance.org/csaguide.pdf.
5. K.Ren, C.Wang, and Q.Wang. "Security Challenges for the Public Cloud". Illinois Institute of Technology. January/February 2012. Published by the IEEE Computer Society.
6. P.Hofmann. "Cloud Computing: The Limits of Public Clouds for Business Applications". SAP Labs. Published by the IEEE Computer Society.
7. C. Wang et al., "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. 30th IEEE Int'l Conf. Computer Communications (INFOCOM 10), IEEE Press, 2010, pp. 525–533.
8. J.W. Rittinghouse and J.F. Ran- some, "Cloud Security Challenges," Cloud Computing: Implementation, Management, and Security, CRC Press, 2009, pp. 158–161; www. infosectoday.com/Articles/Cloud _Security_Challenges.htm.
9. T.Micro. "Cloud Computing Security: Making Virtual Machines Cloud-Ready".Aug. 2009.
10. Lori M. Kaufman. " Can Public-Cloud Security Meet Its Unique Challenges?". *BAE Systems.* July/August 2010. Copublished By The IEEE Computer And Reliability Societies.
11. S.Kamara, and K.Lauter. Cryptographic cloud storage in RLCPS, Jan 2010, LNCS.Springer. Heidelberg.
12. Z.Zhang, Q.Guan and S.Fu. "An Adaptive Power Management Framework for Autonomic Resource Configuration in Cloud Computing Infrastructure. In Proceedings of IEEE 31st International Performance Computing and Communications Conference(IPCCC), 2012,pp. 51 – 60.
13. D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy, 2000.
14. . V.Kolesnikov and A.Shikfa. "On The Limits of Privacy Provided by Order- Preserving Encryption". *Bell Labs Technical Journal.*
15. . A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In *Proceedings of the 28th International Conference on Advances in Cryptology, EUROCRYPT, 2009.*
16. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS '10), 2010.
17. Lewis, David D. and W. Bruce Croft. 1990. "Term Clustering of Syntactic Phrases". Proceedings of ACM SIGIR-90, pp. 385-405.
18. Kwok, K.L., L. Papadopoulos and Kathy Y.Y. Kwan. 1993. "Retrieval Experiments with a Large Collection using *PIRCS."* Proceedings of TREC-1 conference, NIST special publication 500-207, pp. 153-172.
19. E-J. Goh. Secure indexes. Technical Report 2003/216, IACR ePrint Cryptography Archive, 2003. See http://eprint.iacr.org/2003/216.
20. B. Bloom. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 13(7):422–426, 1970.
21. P. Golle, J. Staddon, and B. Waters. Secure conjunctive keyword search over encrypted data. In Applied Cryptography and Network Security Conference (ACNS), volume 3089 of Lecture Notes in Computer Science, pages 31–45. Springer, 2004
22. D. Boneh. The Decision-Diffie Hellman Problem. In Third International Symposium on Algorithmic Number Theory (ANTS-III), volume 1423 of Lecture Notes in Computer Science, pages 48–63. Springer-Verlag, 1998.
23. B. Waters, D. Balfanz, G. Durfee, and D. Smetters. Building an encrypted and searchable audit log. In Network and Distributed System Security Symposium (NDSS 2004). The Internet Society, 2004.
24. D. Davis, F. Monrose, and M. Reiter. Time-scoped searching of encrypted audit logs. In 6th International Conference on Information and Communications Security (ICICS 2004), volume 3269 of Lecture Notes in Computer Science, pages 532–545. Springer-Verlag, 2004.
25. D. Park, K. Kim, and P. Lee. Public key encryption with conjunctive field keyword search. In 5th International Workshop on Information Security Applications (WISA 2004), volume 3325 of Lecture Notes in Computer Science, pages 73–86. Springer-Verlag, 2004.
26. CLEVERDON, C. 1983. "Optimizing Convenient Online Access to Bibliographic databases." Information Services and Use, 4(1/2), 37-47.
27. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM '10, 2010.
28. Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data Ning Caoy, Cong Wangz, Ming Li, Kui Ren, and Wenjing Lou.