Two-Step-Ranking Secure Multi-Keyword Search Over Encrypted Cloud Data

Jun Xu, Weiming Zhang, Ce Yang, Jiajia Xu, Nenghai Yu
Electronic Engineering and Information Science University of Science and Technology of China Anhui, China
Email: {junxlcustc, zwmshu, cn.yang.ce}@gmail.com, xujiajia@mail.ustc.edu.cn, ynh@ustc.edu.cn

***Abstract*—To protect privacy of users, sensitive data need to be encrypted before outsourcing to cloud, which makes effective data retrieval a very tough task. In this paper, we proposed a novel order-preserving encryption(OPE) based ranked search scheme over encrypted cloud data, which uses the encrypted keyword frequency to rank the results and provide accurate results via two-step ranking strategy. The first step coarsely ranks the documents with the measure of coordinate matching, i.e., classifying the documents according to the number of query terms included in each document. In the second step, for each category obtained in the first step, a fine ranking process is executed by adding up the encrypted score. Extensive experiments show that this new method is indeed an advanced solution for secure multi-keyword retrieval.**

I. Introduction

Computable cloud is now prevalent in our daily life here and there, where customers can remotely store their data so as to enjoy the convenient and effective services on-demand [1]. More and more sensitive information such as e-mails and finance data are professionally maintained in data centers. While the fact that data owners and cloud server are no longer in the same trusted domain may put the plain data in risk [2]. So it comes that sensitive data has to be encrypted prior for data privacy and combating unsolicited access.

However encrypted data obsoletes the traditional data utilization service based on plain text keyword search. The simplest solution of downloading all the data and decrypting locally is clearly impractical, due to the huge amount of data and the super bandwidth cost in cloud scale systems. Taking the potentially great amount of data users and mass data in the cloud into consideration, searchable encryption schemes are necessary. The requirements of balancing the privacy and confidentiality with efficiency and accuracy is now challenging the design of searchable encryption schemes.

Traditional searchable encryption [3]- [8] can securely search through a single keyword and retrieve documents of interest in symmetric key setting or public key setting. However, they are not suitable for the large scale cloud data utilization system as they can not provide high service-level such as system usability and user searching experience.

Some methods supporting Boolean keyword search [9]- [13] are designed to enrich the search flexibility such as conjunctive and disjunctive search. Conjunctive keyword search returns those documents including all interested keywords, while disjunctive returns every document that contains even only one keyword of interest. Obviously, they are not adequate to provide acceptable results ranked according to relevance.

In practice, to realize effective data retrieval in the large amount of documents of cloud storage, it is necessary to perform result relevance ranking. Ranked search can also significantly save network traffic by sending back only the most relevant data. Wang et al. [14], [15] proposed secure ranked search which utilizes keyword frequency to rank results, and key word frequency is protected using order-preserving encryption (OPE) [16], [17]. In fact, the OPE keeps the order between the plain texts and cipher texts, so the cloud server can rank the relevant document according to the encrypted keyword frequency. However this method [14], [15] only supports secure ranked search for a single keyword.

Single keyword search often returns coarse results, so it is necessary to support multiple keywords search for such ranking system. Cao et al. [18] proposed a scheme for multi-keyword ranked search over encrypted cloud data. Based on secure inner product computation, the authors [18] exploit the similarity measure of "coordinate matching", i.e., as many matches as possible, to capture the relevance of data documents to the search query. In other words, the returned results are ranked according to the number of interested keywords included in the documents. However, if two documents include the same number of query terms, they will not be differentiated. Swaminathan et al. [19] extend the OPE-based scheme to multi-keyword setting by simply adding up the scores of all terms. Because the OPE expands the plain data into cipher data with much greater range, adding up the encrypted scores can not completely preserve the order and make the scheme [19] inaccurate. For instance, considering that the expansion of plain score often makes the weight of each single keyword deviate from its original value, which obviously should be avoided in multi-keyword search.

Intuitively, accurate ranked search should depend on the information from keyword frequency, so OPE-based scheme is promising, which enables the utility of keyword frequency for search and protects the distribution of keyword frequency at the same time. For single keywords search, OPE-based scheme can achieve the same accuracy for encrypted data as for plain data by accurately identifying the target. For multi-keyword search, how to develop accurate scheme based on OPE is still a problem.

To solve the problem mentioned above, in this paper, we propose a novel OPE-based ranked search scheme over

978-0-7695-4910-1/12 $26.00 © 2012 IEEE
DOI 10.1109/CSC.2012.26

TABLE I
EXAMPLE OF A MODIFIED INVERTED INDEX TABLE(RELEVANCE SCORE)

File ID	F_{i_1}	F_{i_2}	F_{i_3}	F_{i_4}	F_{i_5}
w_1	1.5	0	1.6	0	1.2
w_2	0	2.5	0	0	1.5

encrypted cloud data, which uses the encrypted keywords frequency to rank the results and provide accurate results via two-step ranking strategy. The first step coarsely ranks the documents with the measure of coordinate matching, i.e., classifying the documents according to the number of query terms including in each document. In the second step, for each category obtained in the first step, a fine ranking process is executed by adding up the encrypted score. On the one hand, compared with the original OPE-based retrieval in [19], our scheme takes coordinate matching into consideration, which modify the error caused by expansion of data in OPE. Additionally, our new idea can free the range of OPE, which makes a significant difference to the retrieval system. On the other hand, our method can sort the documents with the same degree of coordinate matching, which obviously can do better than the mere boolean retrieval. Extensive experiments show that this new method can greatly improve the accuracy of searching results when comparing with the secure inner product based method [18] and original OPE-based method [19].

II. PREVIOUS SEARCH SCHEMES BASED ON OPE

A. Basic retrieval scheme

We now introduce some necessary information retrieval background.

Index Building. In information retrieval, inverted index (posting table) is a widely used indexing structure that stores a list of mappings from keywords to the corresponding set of files that contain this keyword, which allows full text search. The inverted table is then employed to define the relevance score for rank-ordering documents in a data collection by a scoring function $Score(i, j)$. However, in retrieval scheme based on OPE, it's necessary to list all original scores even when they equal to zero. Example of a modified inverted index table are shown in Table. I, where each row is determined by the frequency of a certain keyword in all documents. And such kind of inverted index structure will be used in original OPE-based scheme and our scheme.

Scoring Function. Consider a data collection that contains N^D documents, where there are N^T unique terms. Here we choose the $CW(i, j)$ [20] as the scoring function to keep in accord with [19], which is defined as:

$$CW(i,j) = \frac{CFW(i)TF(i,j)(K+1)}{K(1-b+b\cdot NDL(j)) + TF(i,j)} \quad (1)$$

Where $CW(i, j)$ stands for score that the j^{th} document gets based on the i^{th} term; $TF(i, j)$ stands for the frequency of i^{th} term in the j^{th} document; $NDL(j) = L(j)/L_{avg}$ represents

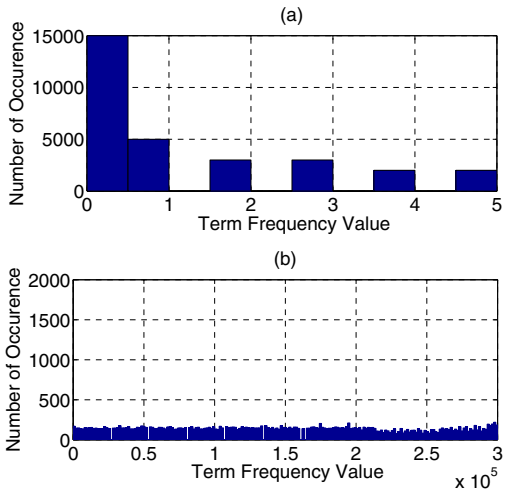


Fig. 1. (a): Original-data distribution; (b):Encrypted-data distribution.

the normalized length of the j^{th} document and is obtained by dividing the length of the j^{th} document, $L(j)$, by the average document length L_{avg}; if N_i is the number of documents containing the i^{th} term, then $CFW(i)$ can be denoted as:

$$CFW(i) = log(N^D/N_i) \quad (2)$$

What's more, K and b are constants chosen to achieve the best retrieval effect. We also choose $K = 2, b = 0.75$ as [19].

B. Brief Introduction of OPE

Since we are not focused on the detail of order-preserving encryption (OPE) [16], [17], we might as well just talk about the main property of OPE:

Order-Preserving. Given a set of data $\{n_1, n_2, \ldots, n_k\}$ with the sorted sequence $\{n_{i1}, n_{i_2}, \ldots, n_{ik}\}$, where $n_{i1} < n_{i2} < \cdots < n_{ik}$, if we encrypt $\{n_1, n_2, \ldots, n_k\}$ with OPE into $\{sn_1, sn_2, \ldots, sn_k\}$ then there must be $sn_{i1} < sn_{i2} < \cdots < sn_{ik}$.

Target-Distribution. In order to keep OPE secure, the distribution of the encrypted data must be a designated distribution independent of the distribution of plain data. In this paper, we take the uniform distribution as target distribution. An example on a set of data about 10K integers is shown in Fig. 1 , whose actual histogram is Fig. 1(a) and the encrypted histogram is Fig. 1(b).

C. Single Keyword Search with OPE

Considering the difference between traditional retrieval and secure retrieval, we illustrate a framework for confidentiality-preserving ranked search in Fig. 2, which includes the data owner, the data user, the cloud server.

In the secure retrieval scheme, we store the hash-values instead of keywords in the cloud server, which can keep the server away from the plain text of keywords. Then we encrypt the inverted index table with OPE and store it in cloud server,

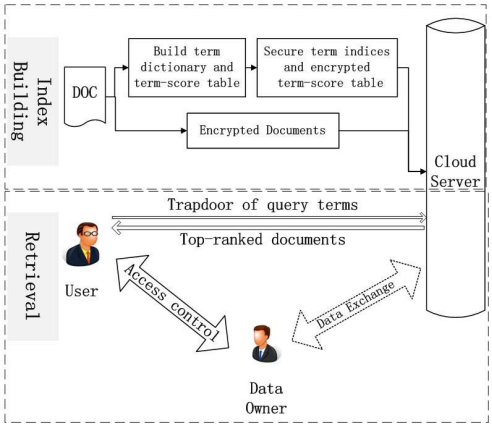


Fig. 2. Architecture for search over encrypted cloud data.

too. Encrypted examples corresponding to Table. I are shown in Table. II.

Wang et al. [14], [15] proposed single keyword search scheme based OPE, which is outlined as follows.

1) The data owner builds a privacy-preserving index from a dataset of documents. After the index construction, the document can be independently encrypted and outsourced.
2) User calculates the hash-value of the interested single keyword w_i to get $Trapdoor(w_i)$ and designate the number(top-k) of documents wanted, sending both of them to the cloud server.
3) After receiving the $Trapdoor(w_i)$, the cloud server looks up the matched $HASH(w_i)$ according to $Trapdoor(w_i)$. Then the server sorts all the documents based on their order-preserving encrypted scores.
4) After getting the sorted files ID of documents, the server returns the top-k documents to user in encrypted format and user can decrypt them to get the needed documents.

D. Encrypted Scores Summing (ESS) Scheme for multi-keyword search

Swaminathan et al. [19] proposed a multi-keyword search based on OPE, which simply adds the encrypted score of every single keyword of each documents with Eq. 3 and then sorts the documents by the sum of the scores. Given a multi-keyword query $Q = \{q_1, q_2, ..., q_n\}$, the sum of scores can be defined as:

$$Score(Q, j) = \sum_{i \in Q} OPE(CW(i, j)) \tag{3}$$

Where $OPE(CW(i, j))$ is the encrypted value of $CW(i, j)$ with OPE. We refer to the method in [19] as Encrypted Scores Summing (ESS) Scheme. However, there are two ways to encrypt the inverted index table with OPE, which will make a great difference to the result in ESS. The first way is to encrypt

TABLE II
EXAMPLE OF ENCRYPTED INDEX TABLE(ENCRYPTED SCORE)

File ID	F_{i_1}	F_{i_2}	F_{i_3}	F_{i_4}	F_{i_5}
$HASH(w_1)$	12	3	16	6	9
$HASH(w_2)$	3	16	6	9	12

the modified inverted index table row by row with different key, where each row in the table stands for the scores of all documents corresponding to a certain keyword. Another way is to encrypt the whole index table by a key. We call the first way "row by row OPE" and the second way "global OPE". As recommended in [19], it'd better to apply row by row OPE. We also strongly suggest the row by row OPE rather than global OPE and we will discuss the drawbacks of global OPE later.

However, in ESS scheme [19], adopting row by row OPE will greatly reduce the accuracy in retrieval. Taking Table. I and Table. II as an example, we can easily know the row by row OPE will cause inaccuracy of ESS: First, the largest cipher data encrypted from 0 of w_2 is 9, which equals to the smallest cipher data encrypted from nonzero value of w_1. Such a problem will result in errors in retrieval. Second, considering the multi-keyword query just contains w_1 and w_2, the sum of original scores of F_{i_2} ,$F_{i_3}$and F_{i_5} are 2.5, 1.6 and 2.7. However, the sum of encrypted scores of F_{i_2}, $F_{i_3}$and F_{i_5} are 19, 22 and 21, which obviously upsets the original rank of documents.

To evaluate the retrieval accuracies of the ESS schemes with different OPE, we conduct experiments on 7594 documents from CERC document collection, 31 queries and standard related results used for the TREC 2007 Enterprise Track topics [21]. Any document that is judged partially relevant or relevant is taken to be relevant in our test. The recall-precision results for all 31 queries are collected and the average is shown in Fig. 3(a). It can be easily noticed that the performance of ESS with global OPE is relatively better than the performance of ESS with row by row OPE. However, the drawbacks of global OPE are more obvious.

The first drawback of global OPE is lower security. From the perspective of linguistics, the server can know how terms are distributed across documents according to Zipf Law [22], which states that the frequency of any term is inversely proportional to its rank in the frequency table. For example, assume the frequency of the i-th most frequent term is $freq_i$, then there is $freq_i \propto \frac{1}{i}$. Global OPE may create an access to the original terms for the server because the server can just arrange or add up all the cipher data in a row to speculate the true frequency of a certain term by comparing with other rows. Also due to the global OPE, the encrypted nonzero-value are always greater than the encrypted zeros, which gives the server a chance to sort the whole encrypted table and to know that the collection of smaller values are encrypted from zero-values with the help of statistical information in corpus of human language.

The second drawback of global OPE is the expansion of

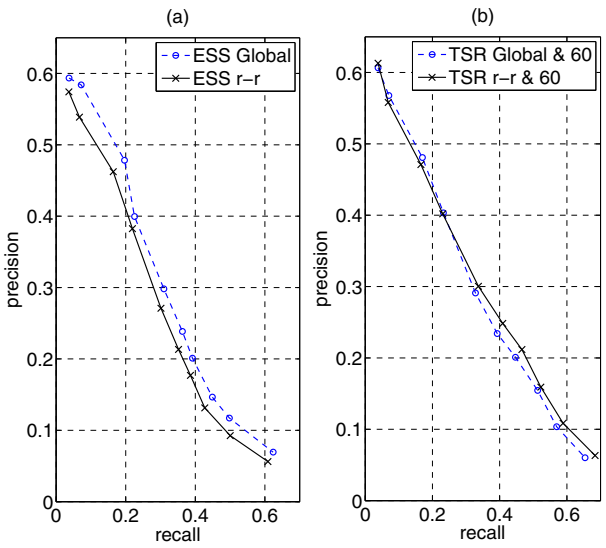


Fig. 3. (a)Recall-precision graphs for ESS with two kind of OPE;(b)Recall-precision graphs for TSR with two kind of OPE. ("r-r" or "Global" stands for scheme with row by row OPE or global OPE; "& 60" stands for TSR of α at 60. All pictures are labeled in this way)

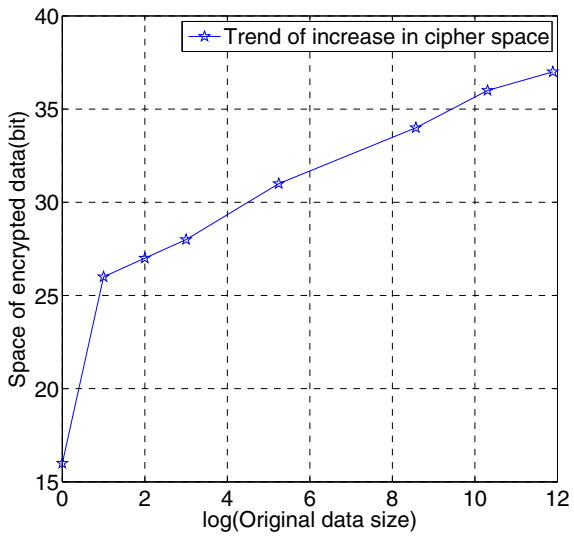


Fig. 4. The expansion of encrypted space with the increase of original data size.

inverted index table. To keep secure, OPE must homogenize the distribution of plain data into a uniform distribution, which means that the range of cipher data will increase with the size of plain data as shown in Fig. 4. Therefore, the space cost of global OPE will greatly larger than that of row by row OPE.

III. Proposed Scheme

A. Two Step Ranking (TSR) scheme

To improve the searching accuracy of row by row OPE, we develop a two step ranking (TSR) scheme. In the first step, classifying the documents according to coordinate matching, i.e., ranking according to the number of interested keywords included in the documents. In the second step, we sort the documents in each class in the same way like ESS with score summing.

Coordinate matching. After encryption, the inverted index table cannot tell whether a document contains a certain word or not, which makes the direct coordinate matching or boolean multi-keyword search impossible. However, the OPE preserves the order of plain data. It's natural to think that in a row of encrypted scores corresponding to a certain keyword, documents with greater encrypted scores are more likely to contain the keyword.

Based on that, In TSR, we develop a new way to calculate the degree of coordinate matching. In our scheme, in the row corresponding to a certain keyword, if the encrypted score of a document is greater than or equal to the K-th greatest encrypted score, we assume that the document contains the keyword and we say that this document is K-related with this keyword. In multi-keyword searching, as shown in Fig. 5, in first step, the server classifies and sorts the documents according to how many K-related keywords they contain. Denoting the K-th largest encrypted score for the i-th keyword (i.e., the K-th largest encrypted scores in the i-th row) by $S(K,i)$, and we can define the coordinate matching function of the j document as $MatchScore(Q,j)$:

$$MatchScore(Q,j) = \sum_{i \in Q} I(OPE(CW(i,j)) \geq S(K,i)) \tag{4}$$

where $I(OPE(CW(i,j)) \geq S(K,i))$ is a indicative function. If $OPE(CW(i,j)) \geq S(K,i)$, then $I(OPE(CW(i,j)) \geq S(K,i)) = 1$, otherwise $I(OPE(CW(i,j)) \geq S(K,i)) = 0$. In other words, Eq. 4 records the number of interested keywords that is K-related with the j document. The documents with the same MatchScore are classified into the same set, and the sets are sorted in descending order of MatchScores.

Summing scores in class. As shown in Fig. 5, in the second step, the server sort the documents in the same class by adding up their scores as Eq. 3. After two steps of sorting, the server can return the first k ranked documents, where k is the number of documents that user wants to get.

Here we go back to Table. I and Table. II again. We assume $K = 3$ and $Q = \{w_1, w_2\}$. According to the first step of TSR: $S(3,1) = 9, S(3,2) = 9$. Now we know $MatchScore(Q,5) = 2$ and the $MatchScore$ of all other four documents are 1 with the help of Eq. 4. The result of classification will be $\{F_5\}, \{F_{i_1}, F_{i_2}, F_{i_3}, F_{i_4}\}$. Now we need to sort documents in the same class based on Eq. 3: $\{Score(Q,5) = 21\}$,$\{Score(Q,3) = 22, Score(Q,2) = 19, Score(Q,1) = 15, Score(Q,4) = 15\}$. The final rank will be $F_{i_5}, F_{i_3}, F_{i_2}, F_{i_1}, F_{i_4}$. To make a comparison, we give out the result of ESS: $F_{i_3}(22), F_{i_5}(21), F_{i_2}(19), F_{i_1}(15), F_{i_4}(15)$, where the value followed the document's id is the sum of

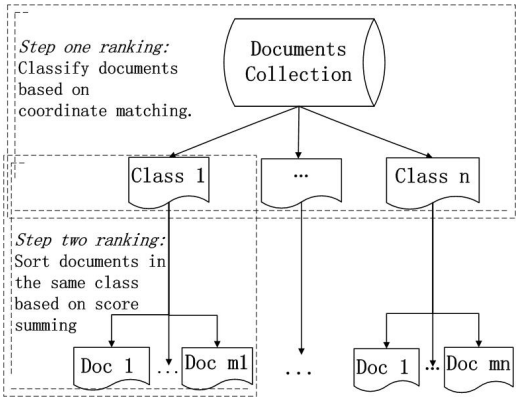


Fig. 5. Model of Two Step Ranking(TSR). The upper comment box shows how TSR classifies the documents in the first step; the lower comment box shows how TSR sorts the documents in the same class.

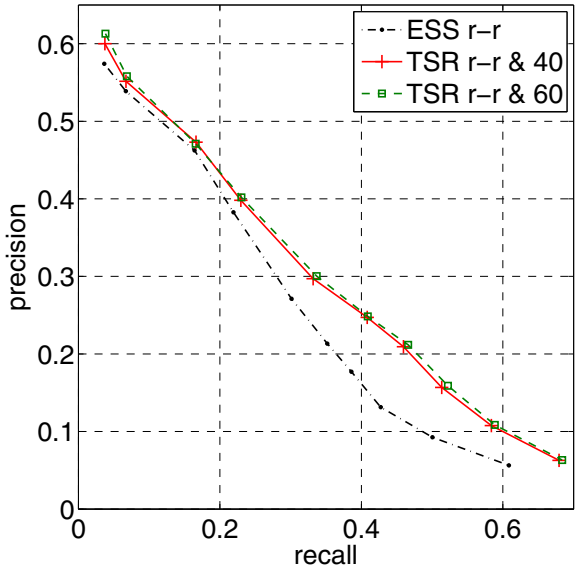
encrypted scores. Obviously the result of F_{i_3}, F_{i_5} in TSR is the right one and TSR truly realizes coordinate matching, considering that F_{i_5} contains two keywords while F_{i_3} just contains one and the sum of original scores of F_{i_5} is greater than F_{i_3}.

B. Choose the Parameter K

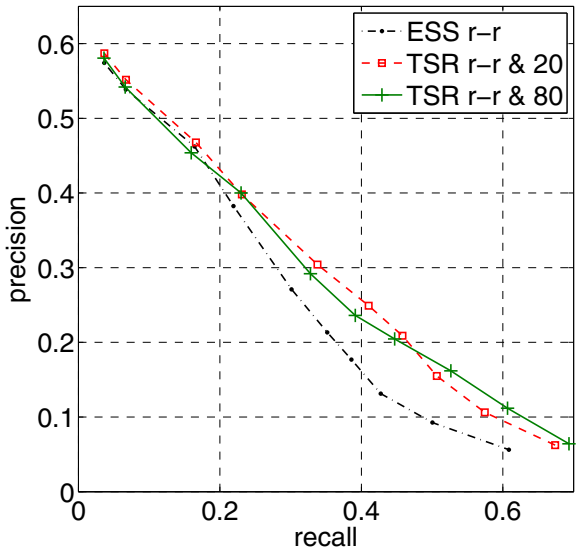
The purpose of coordinate matching is to enable the documents containing more keywords to occupy a higher rank. The K-related can roughly catch this idea, and the search accuracy will be influenced by the value of K. If we randomly choose a K, in worst cases, all documents will be assigned to one class, and thus the TSR scheme will degenerate to the ESS scheme. Obviously, the goodness of fit for the K is related to the number of nonzero-frequency of terms in query. For each term w_i in modified inverted index table, we define NZ_i as the numbers of nonzero-frequency of w_i in all documents, i.e.,

$$NZ_i = \#\{TF(i,j) | TF(i,j) > 0, 1 \leq j \leq N^D\}, \quad (5)$$

and denote the collection of NZ_i's as $NonZero = \{NZ_1, NZ_2, ..., NZ_{N^T}\}$, which is secret information hold by the user. Denote $T_{NZ}(\alpha)$ as the $\alpha\%$ quantile ($0 \leq \alpha \leq 100$) of $NonZero$. The user can randomly select a $\alpha \in [0, 100]$ and $K = T_{NZ}(\alpha)$. Note that this strategy will leak a little information to the server, because the server will know that $\max\{NonZero\} \geq K$ and $\min\{NonZero\} \leq K$. However, it is hard for the server to get more other information from such information leakage. In fact, because the invert index table is encrypted with OPE, the server can always guess that the small encrypted values are zeros and large encrypted values are larger than zero.



(a) $\alpha = 40$ *and* $\alpha = 60$



(b) $\alpha = 20$ *and* $\alpha = 80$

Fig. 6. Recall-precision graphs for TSR with row by row OPE at different α.

IV. Result and Comparison

A. Result of TSR

To show how TSR works and how α influences TSR, we conducted experiments on TSR scheme with the same TREC data, standard query and evaluation scheme. In the experiments, we choose 4 different α: 20,40,60 and 80, which are uniformly selected from $[0, 100]$. 0 and 100 are ignored because these two α will classify almost all the documents

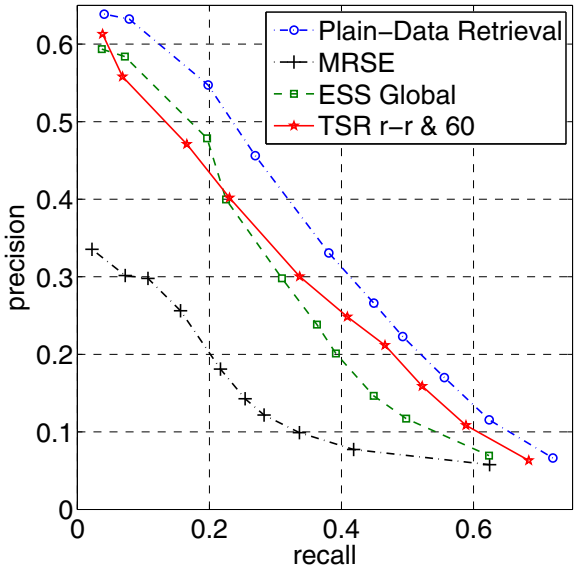


Fig. 7. Recall-precision graphs for plain data retrieval, MRSE scheme, ESS scheme with global OPE and TSR scheme with row by row OPE of α at 60.

into an identical class, in which TSR degenerates to ESS.

As shown in Fig. 6(a), we know that when α locates at a more reasonable range such as $[40, 60]$, TSR with row by row OPE can always perform better than ESS with row by row OPE. While when α is some extreme value like those larger than 80 or smaller than 20, TSR will degenerate gradually. In Fig. 6(b), when α is 20 or 80, TSR sometimes works no better than ESS especially in the first several results. So when α is deviant, the performance of TSR is not so satisfying. However, as illustrated in Fig. 6(a) and Fig. 6(b), the results of α at 20 or 80 actually are close to that of α at 40 or 60, which exactly shows that TSR with row by row OPE works better than ESS averagely even when α varies from 20 to 80.

In Fig. 3(b), we show the result of TRS with global OPE with result of TSR with row by row OPE and α at 60. Compared with Fig. 3(a), the difference is much smaller in TSR scheme than in ESS, which exactly indicates that TSR scheme is independent of the range of OPE while the ESS scheme is not.

Now we discuss why extreme α decreases TSR's advantage over ESS. First, if α is so small that very close to 0, the corresponding K almost equals to 1, considering that averagely one in 250 words is extreme rare according to [23]. Consequently, nearly all documents will be classified into one class. For example, if a multi-keyword query Q contains three terms and K is 1, then there are at most 3 documents can get $MatchScore$ 1, while that of all other documents will be 0. Likewise, if α is extreme great, for example 100, the K related to such α probably gets close to N^D, for which the most frequent terms "the" is a good example. Therefore, such great α causes same problem as extremely small α. These deviant α will make TSR degenerate to ESS.

TABLE III
RETRIEVAL ACCURACY MEASURES FOR DIFFERENT SCHEME

Metric	Plain	MRSE	ESS Global	ESS r-r	TSR r-r 60
MAP	0.3033	0.1830	0.2353	0.2304	0.2648
R-PREC	0.3615	0.2373	0.2922	0.2900	0.3242
P@5	0.6322	0.3870	0.5935	0.5742	0.6129
P@10	0.6096	0.3580	0.5839	0.5387	0.5581
P@50	0.4483	0.3290	0.3896	0.3826	0.4019
P@100	0.3225	0.2080	0.2881	0.2710	0.3003
P@500	0.1160	0.0914	0.1072	0.0926	0.1085

B. Comparison

We compared the result of TSR with MRSE scheme in [18] and ESS scheme in [19]. In Fig. 7, we clearly see that TSR scheme with row by row OPE of α at 80 do much better than MRSE scheme. Besides, TSR with row by row OPE can do as well as or better than ESS with global OPE. However, our scheme is based on row by row OPE, whose advantage over global OPE has been discussed in detail before. To compare the search accuracy for multi-keyword one more step, we compute the Mean Average Precision(MAP) and r-prec for different scheme. According to Table. III, the MAP for ESS with global OPE and TSR scheme with row by row OPE are 0.2353 and 0.2648, which is 12.5% improved. And r-prec for the two scheme are 0.2922 and 0.3242, which is %10.1 enhanced. It's reasonable to conclude that our scheme is better in secure multi-keyword retrieval over encrypted cloud data.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a two step ranking based on Order-Preserving Encryption(OPE) to achieve effective yet secure ranked multi-keyword search over encrypted cloud data. Our scheme takes both of coordinate matching and relevance score into consideration in multi-keyword search. Extensive experimental results demonstrate the efficiency of our scheme. However, to realize our scheme, the parameter K is a key factor of accuracy and a hidden danger of security. Our future work will be focused on developing a better solution of K. We are trying to work out a scheme to enable the server to modify K automatically only based on encrypted inverted index table in each multi-keyword search. In this way, server will return better results self-adaptively and no information about K will be leaked.

ACKNOWLEDGEMENT

This work was supported in part by the Natural Science Foundation of China under Grant 61170234 and Grant 60803155, by the Strategic and Piloted Project of CAS under Grant XDA06030601, and by the National Science and Technology Major Project of China under Grant 2010ZX03004-003.

REFERENCES

[1] P. Mell and T. Grance, *Draft Nist Working definition of Cloud Computing*. http://csrc.nist.gov/groups/SNS/cloud computing/ index.html,Jan,2010

[2] K. Ren, C. Wang, and Q. Wang, *Security Challengs for the public Cloud*. IEEE InternetComputing, vol.16,no.1,pp.69-73, 2012.

[3] D. Song, D. Wagner, and A. Perrig, *Practical techniques for searches on encrypted data*. in Proc. of S&P, 2000.

[4] E.-J. Goh, *Secure indexes*. Cryptology ePrint Archive, 2003, http:// eprint.iacr.org/2003/216.

[5] Y.-C. Chang and M. Mitzenmacher, *Privacy preserving keyword searches on remote encrypted data*. in Proc. of ACNS, 2005.

[6] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, *Searchable symmetric encryption: improved definitions and effiient constructions*. in Proc. of ACM CCS, 2006.

[7] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, *Public key encryption with keyword search*. in Proc. of EUROCRYPT, 2004.

[8] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. S. III, *Public key encryption that allows pir queries*. in Proc. of CRYPTO, 2007.

[9] D. Boneh and B. Waters, *Conjunctive, subset, and range queries on encrypted data*. in Proc. of TCC, 2007, pp. 535-554.

[10] R. Brinkman, *Searching in encrypted data*. in University of Twente, PhD thesis, 2007.

[11] Y. Hwang and P. Lee, *Public key encryption with conjunctive keyword search and its extension to a multi-user system*. in Pairing, 2007.

[12] J. Katz, A. Sahai, and B. Waters, *Predicate encryption supporting disjunctions, polynomial equations, and inner products*. in Proc. Of EUROCRYPT, 2008.

[13] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, *Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption*. in Proc. of EUROCRYPT, 2010.

[14] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, *Secure ranked keyword search over encrypted cloud data*. in Proc. of ICDCS'10, 2010.

[15] C. Wang, N. Cao, K. Ren, and W. Lou, *Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data*. IEEE Trans. on Parallel and Distributed Systems, vol. 32, 2012.

[16] R. Agrawal, A. Evfimievski, R. Srikant, and Y. Xu, *Order Preserving Encryption for Numeric Data*. Proc. ACM SIGMOD Int'l Conf. Management of Data, pp.563-574, 2004.

[17] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, *Order-Preserving Symmetric Encryption*. Proc. Int'l Conf. Advances in Cryptology (Eurocrypt '09), 2009.

[18] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, *Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data*. in INFOCOM'11, IEEE.

[19] A. Swanminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M. Wu, and D. W. Oard, *Confidentiality-Preserving Rank-Ordered Search*. Proc. Workshop Storage Security and Survivablity, 2007.

[20] S. E. Robertson and K. S. Jones, *Simple Proven Approaches to Text Retrieval*. Thchenical Report TR356, Cambridge University Computer Laboratory,1997.

[21] P. Bailey, N. Craswell, I. Soboroff, and A. P. Vries, *The CSIRO Enterprise Search Test Collection*. SIGIR Forum 41(2), pp. 42-45.2007.

[22] George K. Zipf , *The Psychobiology of Language*. Houghton-Mifflin,1935.

[23] H. Williams and J. Zobel, *Searchable words on the Web*, International Journal on Digital Libraries, pp. 99-105, 2005.