IEICE TRANS. INF. & SYST., VOL.E95–D, NO.12 DECEMBER 2012

LETTER *Special Section on Parallel and Distributed Computing and Networking*

Secure Ranking over Encrypted Documents

Jiuling ZHANG[†a)], **Beixing DENG**[†], **Xing LI**[†], ***Nonmembers***, ***and*** **Xiao-lei ZHANG**[†], ***Student Member***

SUMMARY Ranking the encrypted documents stored on secure cloud computing servers is becoming prominent with the expansion of the encrypted data collection. In our work, order preserving encryption is employed to pre-rank the encrypted documents. Paillier's additive homomorphic encryption is used to re-rank the top pre-ranked documents of some considerate scale.

key words: *order preserving encryption, paillier public key encryption, okapi BM25*

1. Introduction

Security is a fundamental issue to be solved in the emerging cloud computing. Searching over sensitive data out-sourced to cloud servers is becoming urgent with the expansion of the collection. A secure service discovery protocol that allows multiple keywords search is proposed in [1]. In another work [2], query generalizer is introduced to hide user's sensitive information. In this work, we step further to securely rank the encrypted documents stored on cloud servers. With the expansion of the encrypted documents collection stored on servers, the documents containing some given encrypted keywords is also increasing. Thus it is imperative to rank encrypted documents and return the most relevant encrypted documents to users.

Existing attempts on ranking encrypted documents generally employ the order preserving encryption [3]. In [4] order preserving encryption is used to encrypt the term frequency to rank the documents. In the following work term weights are encrypted by the scheme to perform ranking [5]. Order preserving encryption works well in ranking the retrieved encrypted documents of single keyword query. However if multiple keywords occur in a query, the ranking will be depreciated. Homomorphic encryption such as Paillier's public key encryption [6] allow addition over the ciphertext, thus it is able to give the exact relevant scores given the encrypted term weights. Nevertheless, the computation results are still in ciphertext form, in order to perform ranking users should download the encrypted scores, decrypt them and rank them on their own. The communication and computation cost will be unaffordable if tens of thousands or even millions of ciphertext form scores are to be handled.

Manuscript received January 6, 2012.
Manuscript revised March 28, 2012.

[†]The authors are with the Dept E.E., Tsinghua University, China.
a) E-mail: zhang-jl07@mails.tsinghua.edu.cn

DOI: 10.1587/transinf.E95.D.2954

We propose to combine both the order preserving encryption and Paillier's public key encryption to rank the encrypted documents. Firstly cloud servers pre-rank the retrieved documents by the order preserving encryption. The documents are ranked by the sum of order preserving encryptions of term weights, the top K documents are deemed the most relevant ones. Then users download K Paillier encryptions of scores, decrypt them and re-rank the plaintext scores locally.

This paper is organized as follows. Section 2 gives the order preserving encryption and Paillier's encryption based ranking method. Experiment result is presented in Sect. 3. A conclusion is drawn in Sect. 4.

2. Proposition of Ranking Scheme Using both order Preserving Encryption and Paillier's Cryptosystem

In this work the Okapi BM25 [7], one of the probabilistic models, is applied to ranking encrypted documents. In this model, the relevance score between a document and a query is given by 1.

$$Score(Q, D) = \sum_{q_i \in Q} w_{q_i} \tag{1}$$

In Eq. 1, term weights $w_{q_i} = F(f(q_i, D), n(q_i))$ is a function of the term frequency $f(q_i, D)$ and the number of documents that contains q_i in the whole collection $n(q_i)$.

The order preserving encryption method employed here is a modification of the one proposed by Ozsoyoglu in 2003 [3]. In this scheme a plaintext x is encrypted by an iteration of the simple function $t_{i+1} = a_i t_i + b_i + r_i$, setting $t_0 = x$ and let $y = t_I$. The coefficients a_i and b_i are randomly generated. For guaranteeing the encryption order preserving or almost order preserving, the variable r_i is confined in the near interval centered of 0.

In Paillier's additively homomorphic public key encryption cryptosystem [6], the plaintext x is encrypted as $y = x^m r^n \bmod n^2$, where r is pseudorandomly generated. While the decryption is as $x = L(c^\lambda \bmod n^2)/L(g^\lambda \bmod n^2) \bmod n$, in which $L(u) = (u-1)/n$. The homomorphism is reflected by $Dec(y_1 \times y_2 \bmod n^2) = x_1 + x_2$.

Since the terms should also be protected, they are hash transformed here. We use the hashes of the terms to index the documents. For each document, the Okapi BM25 weights are pre-computed by the user, then the weights are encrypted by the order preserving encryption and the Paillier's public key encryption respectively. To put it simply,

Table 1 Encrypted document representation.

Terms	t_1	t_2	t_3	$\cdots$	t_N
Hash (Terms)	$H(t_1)$	$H(t_2)$	$H(t_3)$	$\cdots$	$H(t_N)$
Order preserving	$O(w_{t_1})$	$O(w_{t_2})$	$O(w_{t_3})$	$\cdots$	$O(w_{t_N})$
Paillier	$P(w_{t_1})$	$P(w_{t_2})$	$P(w_{t_3})$	$\cdots$	$P(w_{t_N})$

the document is represented by Table 1. Here the $H(\cdot)$, $O(\cdot)$ and $P(\cdot)$ denote hash function, order preserving encryption and the Paillier's public key encryption respectively.

Given an encrypted document D and a query Q, for each $q_i \in Q$, the relevant score is given by the order preserving encryption. The score of the relevance under the order preserving encryption is as Eq. 2.

$$O(Score(Q, D)) = \sum_{q_i \in Q} O(w_{q_i}) \tag{2}$$

The documents are ranked according to the scores in encryption form $O(Score(Q, D))$. However, while existing order preserving encryption do rank the documents well for single term queries, they don't support addition over the ciphertext well. There may be fluctuation in the ranking. Thus, order preserving encryption is helpful only in pre-ranking the documents. After obtaining the top K document IDs, the scores of Paillier's encryption form can also be obtained by performing operations over the ciphertext. The score in Paillier's form is given by Eq. 3.

$$P(Score(Q, D)) = \prod_{q_i \in Q} P(w_{q_i}) \tag{3}$$

Due to additive homomorphic property of Paillier's public key encryption, the scores in encryption form as Eq. 3 are exactly the ciphertext of the plaintext scores $\sum_{q_i \in Q} w_{q_i}$. The encryption form scores are then decrypted and ranked again by users.

3. Experiment

Experiment is carried out on TREC test collection ClueWeb09 Cat.B. The documents are ranked according to the proposition in Sect. 2. The extra communication cost is downloading the $K < Doc_{ID}, Score >$ pairs, and the extra computation involved includes the decryption of the Paillier ciphertext, the ranking of the K recovered scores. Since generally only first 10 or 20 retrieved results are to be viewed by the users, the number K can be fixed at 2 or 3 order of magnitude higher than 10, such as 100 or 1000. The communication and computation cost is considerably small and acceptable in the general scenario.

Table 2 demonstrates that the time cost in Paillier's encryption is $T_e = 0.0174\,(\mathrm{s})$ per document. Since the timeliness in encryption and ciphertext uploading is not very strict, the cost is acceptable. While the searching response should be instantaneous, as the decryption cost is $T_d = 0.438\,(\mathrm{s})$ when $K = 100$ and $T_d = 4.38\,(\mathrm{s})$ when $K = 1000$ per search, so the proposed scheme is feasible. Meanwhile, the ciphertext size is $11.14(kB)$ per document.

Table 2 Time and storage cost.

Cost	$T_e(s)$	$T_{d,K=100}(s)$	$T_{d,K=1000}(\mathrm{s})$	Cipher(kB)
Paillier	0.0174	0.438	4.38	11.14

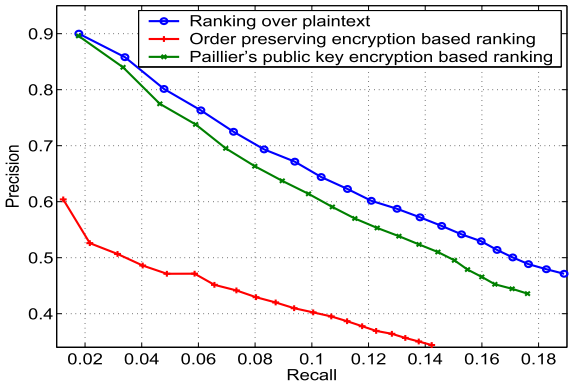

Fig. 1 Precision recall curve for 3 different sceanrio.

Figure 1 indicates that by refining the ranking using the Paillier's additively homomorphic encryption, precision and recall can be greatly enhanced in comparison with utilizing only the order preserving encryption scenario, and get very close to the plaintext scenario.

4. Conclusion

In this paper, order preserving encryption is used to pre-rank the encrypted documents, then Paillier's additively homomorphic encryption are employed to perform refined ranking. Experimental result shows that the precision and recall of the encrypted document ranking have been greatly improved with the help of duplex encryption.

References

[1] J. Kim, J. Baek, J. Zhou, and T. Shon, "An efficient and secure service discovery protocol for ubiquitous computing environments," IEICE Trans. Inf. & Syst., vol.E95-D, no.1, pp.117–125, Jan. 2012.

[2] Y. Oh, H. Kim, and T. Obi, "Privacy-enhancing queries in personalized search with untrusted service providers," IEICE Trans. Inf. & Syst., vol.E95-D, no.1, pp.143–151, Jan. 2012.

[3] G. Ozsoyoglu, D. Singer, and S. Chung, "Anti-tamper databases: Querying encrypted databases," Proc. 17th Annual IFIP WG, pp.4–6, 2003.

[4] A. Swaminathan, Y. Mao, G. Su, H. Gou, A. Varna, S. He, M. Wu, and D. Oard, "Confidentiality-preserving rank-ordered search," Proc. 2007 ACM workshop on Storage security and survivability, pp.7–12, ACM, 2007.

[5] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on, pp.253–262, 2010.

[6] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," Advances in Cryptology łEUROCRYPT99, pp.223–238, Springer, 1999.

[7] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford, "Okapi at trec-3," NIST SPECIAL PUBLICATION SP, pp.109–109, 1995.