Preserving Privacy of Co-occurring Keywords over Encrypted Data

D. Siva Kumar, P. Santhi Thilagam

Preserving Privacy of Co-Occurring Keywords over Encrypted Data

D.V.N. Siva Kumar[1] and P. Santhi Thilagam[2]

[1] Department of CS&IS, BITS Pilani, Hyderabad Campus,
dvnsivakumar@gmail.com
[2] Department of CSE, National Institute of Technology Karnataka, Surathkal, India.
santhisocrates@gmail.com

Abstract. The indexes of ranked searchable encryption contain encrypted keywords and their encrypted relevance scores. The encryption scheme of relevance scores must preserve the plaintext order after encryption so as to enable the cloud server to determine ranks of the documents directly from the encrypted keywords' scores for a given trapdoor. Existing schemes such as Order Preserving Encryption (OPE) and One-to-Many OPE preserve the plaintext order. However, they leak the distribution information, i.e., the frequency of ciphertext values, due to the insufficient randomness employed in these schemes. The cloud server uses frequency analysis attack to infer plaintext keywords of the indexes based on the frequency leakage. In this paper, an Enhanced One-to-Many OPE scheme is proposed to minimize the frequency leakage. The proposed scheme reduces not only the frequency leakage of individual keywords but also the co-occurring keywords of the phrases like "computer network", and "communication network".

Keywords: OPE · Frequency leakage · Index keywords' confidentiality.

1 Introduction

In spite of numeric benefits with cloud computing services, privacy, and confidentiality are significant concerns for the data owners who store sensitive documents at the cloud servers. The stored documents are accessible to the snooping administrators, who could misuse the sensitive information for their own benefits. Therefore, storing documents in plaintext form at cloud servers poses a threat to the confidentiality of data owners' sensitive information. Although the encryption guarantees confidentiality, it makes the retrieval process more complicated. A cryptographic paradigm called Searchable Encryption (SE) facilitates searching over encrypted documents by uploading encrypted indexes, i.e., searchable indexes corresponding to the uploaded encrypted documents.

The information in searchable indexes includes unique keywords of the dataset and their corresponding keywords' relevance scores,e.g., Term Frequency (TF), and Term Frequency-Inverse Document Frequency (TF-IDF), which convey keywords' distribution information of the documents and the dataset, respectively.

This information should not be direclty exposed to the cloud servers. Hence, these relevance scores must be encrypted alongside the index keywords. The relevance scores should be encrypted so that the cloud server could still perform rank-ordering of the documents directly from the keywords' encrypted relevance scores. The encryption scheme must maintain the plaintext order of relevance scores after their encryption to meet this requirement. Various encryption schemes exist such as Order Preservation Encryption (OPE) [2], and One-to-Many OPE [17]. However, they leak frequency information due to the insufficient randomness in mapping plaintext scores to encrypted values. The frequency information is leaked especially when two or more keywords contain the same plaintext score in the same document.

Various attacks such as correlation attack [1] and frequency analysis attack [11,3] exploit frequency information from the encrypted scores to infer index keywords of the dataset. Hence, it is required to minimize or prevent the frequency information leakage in order to use it for encrypting sensitive information. There are many schemes like Paillier encryption for preventing frequency information leakage, but they do not preserve plaintext order due to which the cloud server cannot perform rank-ordering of the documents. As a result, only OPE schemes should be used but they should not leak frequency information. An Enhanced One-to-Many OPE scheme is proposed in this paper, which reduces the frequency leakage of individual keywords and also the co-occurring keywords of the dataset. The summary of our contributions are provided below.

- An Enhanced One-to-Many OPE scheme is proposed to return a ciphertext value for a given plaintext relevance score.
- A thorough analysis of the proposed approach and its demonstration that it is safer in frequency leakage than the conventional OPE and One-to-Many OPE schemes.

The rest of the paper is presented as follows: The preliminary details are presented in Section 2. Section 3 presents the related work, followed by the proposed approach in Section 4, results and discussion in Section 5, and the conclusion in Section 6.

2 Preliminaries

(a) Term Frequency (TF): The TF value of a keyword captures its relevance in the corresponding document. Another measure, TF-IDF, could also be used, but TF is more suitable for demonstrating the frequency leakage. The normalized TF value a of keyword kw of a document d_i can be measured using the below equation.

$$TF(kw, d_i) = \frac{1}{|d_i|}(1 + \log(tf_i)) \tag{1}$$

where, $|d_i|$ denotes the length of the document, i.e., the total number of unique keywords of the document d_i and tf_i denotes the number of a keyword kw occurs in document d_i.

(b) Frequency Analysis Attack: This attack aims to deduce the plaintext index keyword from the encrypted TF values of the encrypted index [11,7]. It is based on the premise that the distribution details of plaintext TF values and the corresponding encrypted TF values will be the same for some specific keywords. The cloud server infers frequently occurring keywords of the dataset from the encrypted index by relating the frequency of the same encrypted TF values to the frequency of plaintext TF values of frequently occurring keywords of the publicly available datasets. Sometimes, index keywords also are guessed by observing the frequency information of some encrypted TF values. This attack is more likely to succeed With the help of background knowledge of the dataset, which could be about what data the data owners stored at the cloud server, and what probable keywords that are most likely to occur. For example, assume that the data owners store Request for Comments [13] dataset in encrypted form at the cloud server. This dataset is all about how information can be transmitted from one host to another over the internet using different "computer network" protocols. It can be assumed that some of the specific keywords like "computer", "network" and "communication" may appear in most of the documents of this dataset. The cloud server then infers one or all of these specific keywords by generating frequency histograms (i.e., showing distribution information) for all encrypted TF values of each encrypted index keyword. The cloud server then examines the histograms and distinguishes those of a much higher frequency (i.e., same encrypted TF values) than the others. The corresponding encrypted keywords of those histograms are noted. The encrypted keywords of histograms may be related to one or all of those specific keywords of the dataset. The cloud server thus infers plaintext index keywords by observing the frequency information of encrypted TF values.

(c) Coordinate Matching: It is a similarity metric that uses the inclusion of query keywords in the document to assess the document's relevance score [18]. The score of a document for a trapdoor can be determined by adding the encrypted TF values of each trapdoor's keywords in that document.

3 Related Work

Order Preserving Encryption (OPE) [2] and Order Revealing Encryption (ORE) [4,8] support ranked search as they preserve plaintext order after encryption. The problem with ORE schemes is the output of these schemes is not numerical, due to which an additional public function is required that will let the cloud server know the order of the given ciphertexts. Swaminathan et al. [16] proposed an OPE scheme based approach, but it leaks frequency information, i.e., returns the same ciphertext value for the same plaintext value. Wang et al. [17] proposed using the One-to-Many OPE scheme, an extension of the OPE scheme, but leaks frequency information when two or more keywords contain the same TF values in the same document's index.

Kerschbaum proposed a frequency hiding OPE scheme [6] to conceal the frequency information. However, this approach still leaks frequency information

in the form of the tree's linear depth, where each new ciphertext value is stored in a new node for the same plaintext value. This depth precisely reflects the frequency of plaintext scores. Several works like Grubbs et al. [5] and Maffei et al. [9] demonstrated the inference of plaintext information from the encrypted values of frequency-hiding OPE scheme.

Orencik et al. [12] proposed a multi-keyword ranked SE scheme using TF-IDF and Forward indexing techniques. They used the Paillier Encryption (PE) scheme to encrypt TF-IDF values. A fully homomorphic Encryption (FHE) and a partially homomorphic encryption (PHE) [14] also can be used in ranked search approaches and for encrypting TF-IDF values. The PE, FHE and PHE prevent the frequency leakage, but they do not preserve the plaintext order after the encryption. Roche et al. [15] proposed a partial order preserving encryption (POPE) approach for encrypting keywords' relevance scores. However, all the ciphertext values in this approach do not preserve the plaintext order, and it is maintained only for some relevance scores. Hence, this approach is also not completely suitable for ranked search approaches.

4 Proposed Approach

4.1 Objectives

It is aimed to meet the following two objectives:

1. *Relevance Score Encryption Scheme:* A novel approach is designed based on the existing OPE schemes to minimize frequency leakage.
2. *Efficiency:* The proposed approach should encrypt the keyword's relevance scores efficiently as the existing approaches.

4.2 Proposed Methodology

It consists of the Initialization Phase and the Retrieval Phase.

Initialization Phase: It includes the following activities to be done by data owner, who owns the dataset:

1. *Build the Encrypted Index:* The steps required to generate the encrypted index $\widetilde{I}$ for all the documents of dataset D is explained as follows:
   (a) *Building Dictionary, W:* Construct the dictionary W by extracting all the unique keywords kw from the input file dataset D.
   (b) *Building Plaintext Index I:* For each keyword kw in W, determine its TF value (1) for each document D_i if it is present in D_i and store it as $I[kw]= [D_i][TF]$. Otherwise, set its TF value to 0, i.e., $I[kw]= [D_i][0]$.
   (c) *Generate Encrypted Index, $\widetilde{I}$:* The generated index I is encrypted as follows:
      – Each keyword in I is hashed by using a one-way secure hash algorithm SHA-2 with a 256-bit key.

- The TF value of each keyword kw of every document D_i is encrypted using the proposed Enhanced One-to-Many OPE scheme, which is explained in Section 4.3.
- The document Ids are not necessary to encrypt as they do not convey any information. The encrypted index generated would then be: $\widetilde{I} = \widetilde{[TF]}[D_i]$.

2. *Encrypting Documents:* After generating the encrypted index, the data owner also encrypts his/her documents D using AES algorithm with 128 bit key size.

Retrieval Phase: It includes the following activities to be done by a data user, who will be allowed by the data owner to retrieve documents of his/her interest.

1. *Query_Masking:* The masked query $\widetilde{M_{Q_{kw}}}$ is generated by hashing each keyword of his/her query Q_{kw} using a secure SHA-256 hash function. Then, the data user sends the masked query and the parameter k to the cloud server to retrieve only the relevant top-k documents.
2. *Searching:*
   (a) The cloud server then utilizes the encrypted index $\widetilde{I}$ and adopt the coordinate matching similarity measure [18] to assign the scores to each document D_{id}. Then it sorts the scores of the documents in descending order and sends the top-k of them to the user.
   (b) The data user then uses the corresponding secret key shared by the data owner to decrypt the retrieved documents.

4.3 Enhanced One-to-Many OPE

Algorithm 1: Enhanced One-to-Many OPE

Input: K (Key), D (domain), R (range), $pscore$, $id(doc)$, kw
Output: Cipher text c

```
while |D|! = 1 do
    {D, R} = Binary_Search(K, D, R, pscore)
coin ← TapeGen(K,(D,R,1||pscore, id(doc), kw)
c ⟸^coin R return c
```

The proposed Enhanced One-to-Many OPE scheme returns a possible unique ciphertext value c for each keyword kw's TF value, i.e., a plaintext relevance score $pscore$ by mapping it to one of the output range of values. The mapping procedure is explained in Algorithm 1. It takes Key (K), input domain (D), output range (R), $pscore$, document identity $id(doc)$, and keyword (kw) and returns a possible unique ciphertext value c. During mapping, the range R is divided into some non-overlapping interval buckets each with different size. Each

bucket contains some range of values. For the given *pscore*, the random-sized bucket is determined using a Binary_Search(.) procedure, which is explained in Algorithm 2. Binary_Search(.) is a recursive procedure, which returns a new domain and a new range of values for a given *pscore* based on an HYGEINV(.) function. HYGEINV(.) is a hypergeometric sampling process that returns an integer value based on the initial domain, range, and middle value. This integer helps Binary_Search(.) in choosing a new domain, and a new range of values for the given *pscore*. In each iteration of binary search, the size of domain D and range R will be reduced by an integer value. Binary_Search(.) stops when the size of the domain becomes 1 where it contains only the given *pscore*. The *pscore* then will be mapped to one of the values in a new range R using a TapeGen(.) function, which is a random coin generator, which generates the seed value. This seed value helps in choosing one of the values in the new range as the ciphertext value c for the given *pscore*.

Algorithm 2: Binary_Search

Input: K, D, R, *pscore*
Output: D,R

$M \leftarrow length(D)$; $N \leftarrow length(R)$
$d \leftarrow min(D) - 1$; $r \leftarrow min(R) - 1$
$y \leftarrow r + \text{ceil}(\frac{N}{2})$
$coin \xleftarrow{R}$ TapeGen(K,(D,R,0||y))
$x \xleftarrow{R} d +$ HYGEINV($coin$,M,N,$y - r$)
if $pscore \leq x$ **then**
    $D \leftarrow \{d+1, ..., x\}$
    $R \leftarrow \{r+1, ..., y\}$
else
    $D \leftarrow \{x+1, ..., d+M\}$
    $R \leftarrow \{y+1, ..., r+N\}$
return D,R

5 Results & Discussion

The proposed approach has been implemented using Python 3.6 version and tested on Intel i7-4770 CPU system. The experiments are conducted on a Requests for Comments (RFC) [13] dataset. The effectiveness of the proposed approach is compared in terms frequency leakage with the OPE and One-to-Many OPE schemes for different keywords and phrases of the dataset.

5.1 Frequency Analysis Attack

The proposed Enhanced One-to-Many OPE scheme is an extension of One-to-Many OPE [17], which in turn is an extension of OPE [2]. In our experiments, the input domain is the actual plaintext relevance scores, and the output range is set between 0 and 2^{45}-1. In OPE, the plaintext score *pscore*, .i.e., the TF value of the keyword is mapped to a ciphertext, which is a value within the new range, R (also can be treated as a bucket), which is determined by using the binary search algorithm 2. The selection of ciphertext value within the bucket values for the given *pscore* is based on the seed value generated by the TapeGen(.). In OPE, this seed value is entirely dependent on the plaintext score *pscore* due to which the same *pscore* is mapped to the same ciphertext value. For the given dataset, if the plaintext score is repeated n number of times, then there will be a ciphertext c that will also be repeated n number of times. Thus, frequency information is leaked from OPE encrypted values.

From the RFC dataset, it is found that the keywords "computer", "network", and "communication" are the frequently occurring keywords. The plaintext distribution information of these three keywords is respectively shown in figures Fig.1.(a), 1.(b) and 1.(c). The values on the x-axis represent the actual plaintext TF values and values on y-axis represent the frequency, i.e., the number of points having the same TF value. As the OPE maps the same plaintext score to the same ciphertext, the distribution of ciphertext values of these keywords would be the same. To infer the plaintext keyword, the cloud server plots the graphs for each encrypted keyword's relevance scores in index. Then, it will note down the encrypted or masked keyword of the index for which the frequency (the repetition of the same encrypted score) is higher than the frequency of scores of other encrypted keywords. This encrypted keyword is more likely to be the frequently occurring keyword, e.g., "computer". Thus, frequency leakage from OPE scores allows the cloud server to deduce plaintext keyword.

In One-to-Many OPE scheme [17], for the given plaintext score *pscore*, it uses both the *pscore*, and document identity $id(doc)$ for generating a different seed for the same *pscore*. Due to this seed, it maps the same plaintext to different ciphertext value within the values of the bucket. There is a scope for frequency leakage with this scheme when two or more keywords have the same score in the same document. This is possible especially when some keywords equally co-occur in the same document, e.g., the keywords of the phrases like {"computer", "network"} or {"communication", "network"}. The cloud server first plots the histogram for each phrase and then identifies the histogram of the phrase in which the frequency of relevance score is higher than the frequency of other possible phrases. The cloud server notes the histogram of this phrase, and the encrypted keywords of this histogram are more likely to be the most co-occurring keywords of the phrases. Thus, the cloud server could infer the phrases' plaintext keywords from the One-to-Many OPE encrypted scores.

The proposed Enhanced One-to-Many OPE scheme, explained in Section 4.3, minimizes the frequency leakage of the phrases caused by the One-to-Many-OPE scheme. This approach also uses the same algorithm 2 (Binary search)

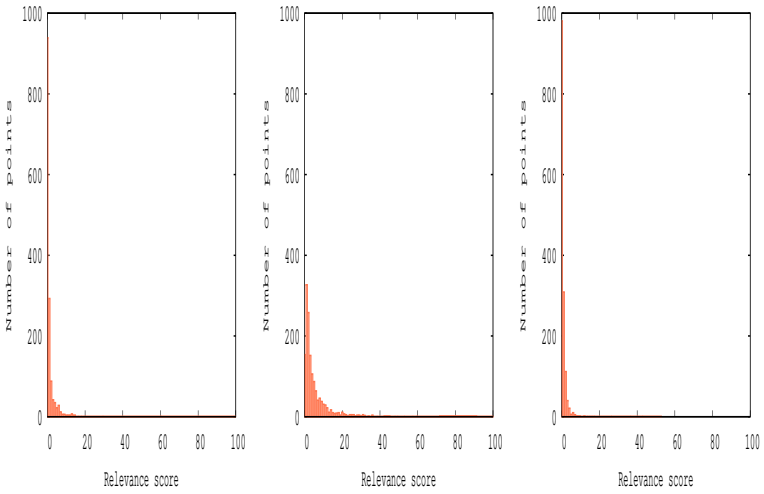

Fig. 1: Plaintext keyword relevance score distribution for keywords: (a) computer (b) network (c) communication

to select the bucket, but the TapeGen(.) here uses *pscore*, document identity, *id*(*doc*) and keyword *kw* for generating a different seed value for the same plaintext score *pscore*. Due to this seed, the same *pscore* of co-occurring keywords will be mapped to different values within the values of the bucket. Thus, it minimizes the frequency leakage of co-occurring keywords of the phrases. This minimization is due to the improvement of the randomness in generating the seed value. Due to this seed, this approach reduces not only the frequency leakage of phrases but also the individual keyword's. To demonstrate the reduction of frequency leakage for an individual keyword, we compare the frequency leakage of One-to-Many OPE encrypted scores, and the proposed Enhanced One-to-Many OPE encrypted scores for the keyword "computer". The distribution information (frequency) of One-to-Many encrypted scores and the proposed Enhanced One-to-many encrypted scores for the keyword "computer" is shown in Fig.2.(a) and 2.(b) respectively. The values on x-axis represent the normalized encrypted scores using the min-max normalization approach [10]. The values of the y-axis represent information about the frequency, i.e., the number of points having the same encrypted TF value. Figure 2.(b) demonstrates that the frequency leakage for some encrypted scores of the keyword "computer" using the proposed Enhanced One-to-Many OPE scheme is lower than the frequency leakage of One-to-Many OPE scores, shown in Fig.2.(a). As it is already mentioned, the proposed scheme also minimizes the frequency leakage of phrases due to the usage of plaintext score *pscore*, document identity *id***(doc)** and keyword *kw* for generating a different seed value. Due to this seed, different value within the bucket will be chosen as the ciphertext value for the same plaintext relevance score

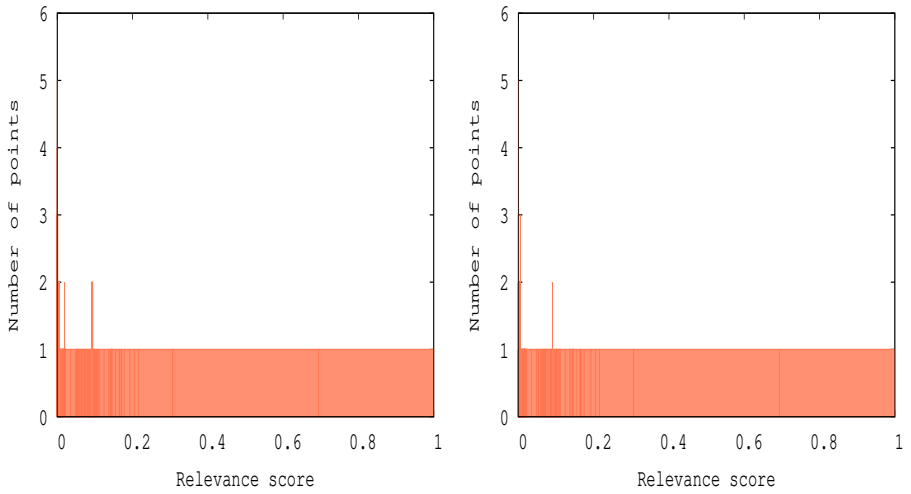

Fig. 2: Encrypted score distribution for "computer": (a)One-to-Many OPE (b)Enhanced One-to-Many OPE

pscore. In RFC dataset, "computer network" and "communication network" are the most occurring phrases. The proposed Enhanced One-to-Many OPE scheme's frequency leakage is compared with the One-to-Many-OPE scheme to demonstrate the reduction of frequency leakage for the phrases. The distribution information of One-to-Many-OPE encrypted scores and the proposed Enhanced One-to-Many-OPE encrypted scores for the phrase "computer network" is shown in figures Fig. 3 and 4, respectively. It can be observed from Fig.4 that the frequency leakage of the proposed approach is much lesser than the frequency leakage of the One-to-Many OPE scheme, shown in Fig.3. Similarly, Fig. 5 and 6 respectively show the distribution information for the phrase "communication network". Fig.6 shows that the frequency leakage of the proposed approach is

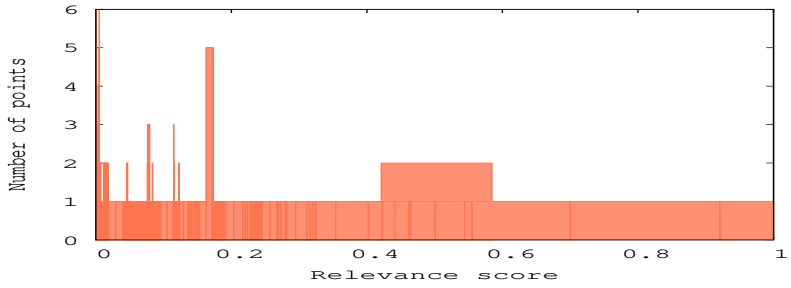

Fig. 3: One-to-Many OPE scores distribution for "Computer Network"

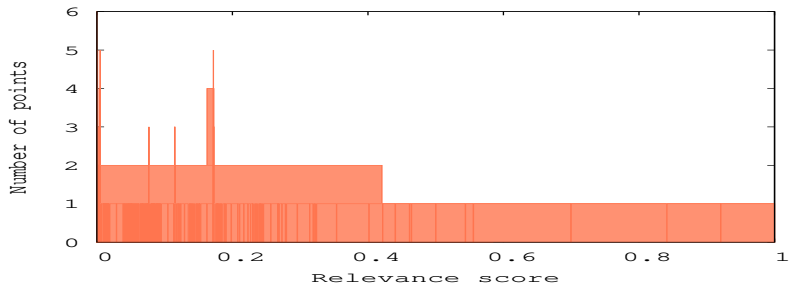

Fig. 4: Enhanced-One-to-Many OPE scores distribution for "Computer Network"

much lesser than the frequency leakage of the One-to-Many OPE scheme, shown in Fig.5. Thus, the proposed approach makes it difficult for the cloud server to infer plaintext keywords of the phrases from the Enhanced One-to-Many OPE scores.

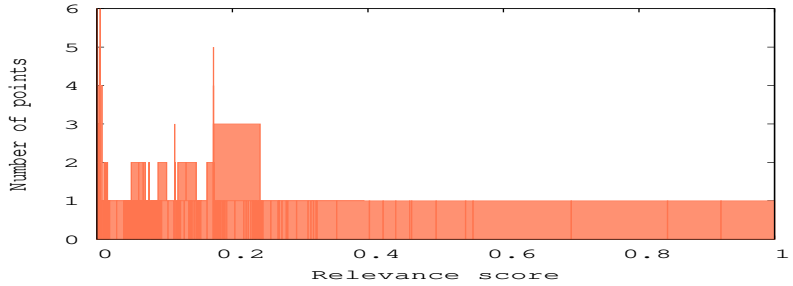

Fig. 5: One-to-Many OPE scores distribution for "communication network"

5.2 Efficiency

The time complexity of mapping a plaintext score *pscore* to a ciphertext c is *O(log n)*, i.e., *log n* times the Binary_Search(.) process will be called during the mapping. The time cost comparison of mapping a plaintext score, *pscore* to a ciphertext value, c using the proposed Enhanced One-to-Many OPE and One-to-Many OPE schemes is shown in Fig.7. The x-axis values represent domain size and the values on y-axis represent the amount of time taken to map *pscore* to c. Y-axis represents the average time of 100 trails for a single mapping operation over different domain sizes and a range $|R| = 2^{45}$. From the Fig.7, it can be observed that efficiency of both the schemes is almost same since both of them call the Binary_Search procedure recursively same number of times.

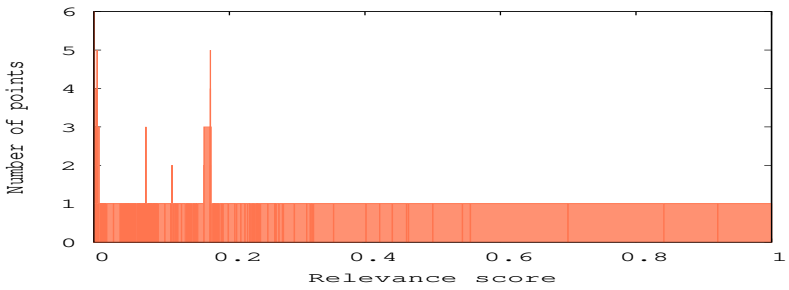

Fig. 6: Enhanced One-to-Many OPE scores distribution for "communication network"

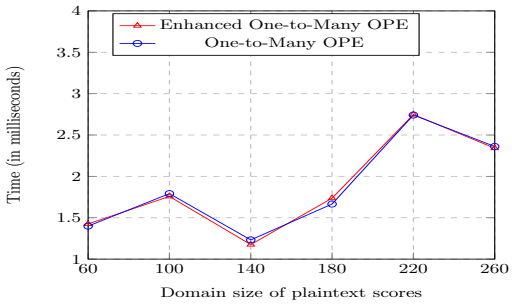

Fig. 7: The time cost comparison of single mapping operation using Enhanced One-to-Many OPE and One-to-Many OPE over a range $|R| = 2^{45}$.

6 Conclusion

In this paper, an Enhanced One-to-Many OPE scheme is proposed that maps the same plaintext relevance score to a different ciphertext value even when multiple keywords have the same plaintext score within the same document. The proposed scheme may prevent complete frequency leakage if it is used for encrypting information that is moderately distributed. Our future work would be to extend the proposed scheme to minimize the frequency leakage further with the help of constrained random numbers as the seed value in mapping process.

References

1. Bindschaedler, V., Grubbs, P., Cash, D., Ristenpart, T., Shmatikov, V.: The tao of inference in privacy-protected databases. Proc. VLDB Endow. **11**(11), 1715–1728 (Jul 2018)
2. Boldyreva, A., Chenette, N., Lee, Y., O'Neill, A.: Order-preserving symmetric encryption. In: Proceedings of the 28th Annual International Conference on Advances in Cryptology: The Theory and Applications of Cryptographic Techniques. pp. 224–241. EUROCRYPT '09, Springer-Verlag, Berlin, Heidelberg (2009)

3. Cash, D., Grubbs, P., Perry, J., Ristenpart, T.: Leakage-abuse attacks against searchable encryption. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security. pp. 668–679. CCS '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2810103.2813700
4. Chenette, N., Lewi, K., Weis, S.A., Wu, D.J.: Practical order-revealing encryption with limited leakage. In: Revised Selected Papers of the 23rd International Conference on Fast Software Encryption - Volume 9783. pp. 474–493. FSE 2016, Springer-Verlag New York, Inc., New York, NY, USA (2016)
5. Grubbs, P., Sekniqi, K., Bindschaedler, V., Naveed, M., Ristenpart, T.: Leakage-abuse attacks against order-revealing encryption. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 655–672 (May 2017)
6. Kerschbaum, F.: Frequency-hiding order-preserving encryption. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security. pp. 656–667. CCS '15, ACM, New York, NY, USA (2015)
7. Kumar, D.S., Thilagam, P.S.: Searchable encryption approaches: attacks and challenges. Knowledge and Information Systems (2018)
8. Liu, Z., Li, J., Lv, S., Huang, Y., Guo, L., Yuan, Y., Dong, C.: Encodeore: Reducing leakage and preserving practicality in order-revealing encryption. IEEE Transactions on Dependable and Secure Computing pp. 1–1 (2020)
9. Maffei, M., Reinert, M., Schröder, D.: On the security of frequency-hiding order-preserving encryption. In: CANS (2017)
10. Margae, S.E., Sanae, B., Mounir, A.K., Youssef, F.: Traffic sign recognition based on multi-block lbp features using svm with normalization. In: 2014 9th International Conference on Intelligent Systems: Theories and Applications (SITA-14). pp. 1–7 (May 2014). https://doi.org/10.1109/SITA.2014.6847283
11. Naveed, M., Kamara, S., Wright, C.V.: Inference attacks on property-preserving encrypted databases. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security. pp. 644–655. CCS '15, ACM, New York, NY, USA (2015)
12. Orencik, C., Kantarcioglu, M., Savas, E.: A practical and secure multi-keyword search method over encrypted cloud data. In: 2013 IEEE Sixth International Conference on Cloud Computing. pp. 390–397 (June 2013)
13. RFC: Request for comments database. `https://www.rfc-editor.org/retrieve/bulk/` (2016)
14. Ristic, M., Noack, B., Hanebeck, U.D.: Secure fast covariance intersection using partially homomorphic and order revealing encryption schemes. IEEE Control Systems Letters **5**(1), 217–222 (2021). https://doi.org/10.1109/LCSYS.2020.3000649
15. Roche, D.S., Apon, D., Choi, S.G., Yerukhimovich, A.: Pope: Partial order preserving encoding. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1131–1142. CCS '16, ACM, New York, NY, USA (2016)
16. Swaminathan, A., Mao, Y., Su, G.M., Gou, H., Varna, A.L., He, S., Wu, M., Oard, D.W.: Confidentiality-preserving rank-ordered search. In: Proceedings of the 2007 ACM workshop on Storage security and survivability. pp. 7–12. ACM (2007)
17. Wang, C., Cao, N., Ren, K., Lou, W.: Enabling secure and efficient ranked keyword search over outsourced cloud data. IEEE Transactions on parallel and distributed systems **23**(8), 1467–1479 (2012)
18. Witten, I.H., Bell, T.C., Moffat, A.: Managing Gigabytes: Compressing and Indexing Documents and Images. John Wiley & Sons, Inc., New York, NY, USA, 1st edn. (1994)