International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET2310233

A Proveable Contextual Retrieval Strategy in the Public Cloud Using Optimal Matching Over Secured Data

Prof. Rahul Bhandekar[1], Pranay Kumbhalkar[2]
[1]HOD – AIDS, Wainganga College of Engineering & Management, Nagpur, India
[2]M. Tech. Student (Software System), Wainganga College of Engineering & Management, Nagpur, India

ARTICLEINFO

Article History:

Accepted: 01 April 2023
Published: 15 April 2023

Publication Issue
Volume 10, Issue 2
March-April-2023

Page Number
467-478

ABSTRACT

Semantic searching over encrypted data is a crucial task for secure information retrieval in public cloud. It aims to provide retrieval service to arbitrary words so that queries and search results are flexible. In existing semantic searching schemes, the verifiable searching does not be supported since it is dependent on the forecasted results from predefined keywords to verify the search results from cloud, and the queries are expanded on plaintext and the exact matching is performed by the extended semantically words with predefined keywords, which limits their accuracy.

In this paper, we propose a secure verifiable semantic searching scheme. For semantic optimal matching on ciphertext, we formulate word transportation (WT) problem to calculate the minimum word transportation cost (MWTC) as the similarity between queries and documents, and propose a secure transformation to transform WT problems into random linear programming (LP) problems to obtain the encrypted MWTC. For verifiability, we explore the duality theorem of LP to design a verification mechanism using the intermediate data produced in matching process to verify the correctness of search results. Security analysis demonstrates that our scheme can guarantee verifiability and confidentiality. Experimental results on two datasets show our scheme has higher accuracy than other schemes.

Index Terms - public cloud, results verifiable searching, secure semantic searching, word transportation.

I. INTRODUCTION

Inherent scalability and flexibility of cloud computing make cloud services so popular and attract cloud customers to outsource their storage and computation into the public cloud. Although the cloud computing technique develops magnificently in both academia and industry, cloud security is becoming one of the critical factors restricting its development. The events of data breaching in cloud computing, such as the Apple Fappening and the Uber data breaches, are increasingly attracting public attention. In principle, the cloud services are trusted and honest, should ensure data confidentiality and integrity according to predefined protocols. Unfortunately, as the cloud server providers take full control of data and execute protocols, they may conduct dishonest behavior in the real world, such as sniffing sensitive data or performing incorrect calculations. Therefore, cloud customers should encrypt their data and establish a result verification mechanism before outsourcing storage and computation to the cloud. Since Song et al. [1] proposed the pioneering work about the searchable encryption scheme, searchable encryption has attracted significant attention. However, the traditional searchable encryption schemes require that query words must be the predefined keywords in the outsourced documents, which leads to an obvious limitation of these schemes that similarity measurement solely base on the exact matching between keywords in the queries and documents. Therefore, some works proposed semantic searching schemes to provide retrieval service to arbitrary words, making the query words and search results flexible and uncertain. However, the verifi- able searching schemes are dependent on forecasting the fixed results of predefined keywords to verify the correctness of the search result returned by the cloud. Therefore, the flexibility of semantic schemes and the fixity of verifiable schemes enlarge the gap between semantic searching and verifiable searching over encrypted data. Although Fu et al. [2] proposed a verifi- able semantic searching scheme that extends the query words to get the predefined keywords related to query words, then they used the extended keywords to search on a symbol-based trie index. However, their scheme only verifies whether all the documents containing the extended keywords are returned to users or not, and needs users to rank all the documents for getting top-k related documents. Therefore, it is challenging to design a secure semantic searching scheme to support verifiable searching.

In this paper, we propose a secure verifiable semantic searching scheme that treats matching between queries and documents as an optimal matching task. We treat the document words as “suppliers,” the query words as “consumers,” and the semantic information as “product,” and design the minimum word transportation cost (MWTC) as the similarity metric between queries and documents. Therefore, we introduce word embeddings to represent words and compute Euclidean dis- tance as the similarity distance between words, then formulate the word transportation (WT) problems based on the word embeddings representation. However, the cloud server could learn sensitive information in the WT problems, such as the similarity between words. For semantic optimal matching on the ciphertext, we further propose a secure transformation to transform WT problems into random linear programming (LP) problems. In this way, the cloud can leverage any ready- made optimizer to solve the RLP problems and obtain the encrypted MWTC as measurements without learning sensitive information. Considering the cloud server may be dishonest to return wrong/forged search results, we explore the duality theorem of linear programming (LP) and derive a set of necessary and sufficient conditions that the intermediate data produced in the matching process must satisfy. Thus, we can verify whether the cloud solves correctly RLP problems and further confirm the correctness of search results. Our new ideas are summarized as follows:

1. Treating the matching between queries and documents as an optimal matching task, we explore the fundamental theorems of linear programming (LP) to propose a se- cure verifiable semantic searching scheme that performs semantic optimal matching on the ciphertext.
2. For secure semantic optimal matching on the ciphertext, we formulate the word transportation (WT) problem and propose a secure transformation technique to transform WT problems into random linear programming (LP) problems for obtaining the encrypted minimum word transportation cost as measurements between queries and documents.
3. For supporting verifiable searching, we explore the dual- ity theorem of LP and present a novel insight that using the intermediate data produced in the matching

process as proof to verify the correctness of search results.

II. RELATED WORK

Since Song et al. [1] proposed the notion of searching over encrypted cloud data, searchable encryption has received significant attention for its practicability in the past 20 years. Therefore, many works have made efforts on the security as well as functionality in the searchable encryption field.

Along the research line about security, many works formulate the definitions of security as well as novel attack pattern against the existing schemes. Goh et al. [10] formulated a security model for document indexes known as semantic security against adaptive chosen keyword attack (IND-CKA), which requires the document indexes not to reveal contents of documents. However, we note that the definition of IND-CKA does not indicate that the queries must be secure. Curtmola et al. [11] further improved security definitions for symmetric 2 searchable encryption, then put forth chosen-keyword attacks and adaptive chosen-keyword attacks. Besides, Islam et al. [12] first introduced the access pattern disclosure used to learn sensitive information about the encrypted documents, then Liu et al. [13] presented a novel attack based on the search pattern leakage. Stefanov et al. [14] introduced the notions of forward security and backward security for the dynamic searchable encryption schemes that support data addition and deletion. Along another research line about functionality, many works introduced practical functions to meet the demand in practice, such as ranked search and semantic searching for improving search accuracy. Additionally, some works proposed verifiable searching schemes to verify the correctness of search results. Ranked Search over Encrypted Data. Ranked search means that the cloud server can calculate the relevance scores be- tween the query and each document, then ranks the documents without leaking sensitive information. The notion of single-keyword ranked search was proposed in [15] that used a modified one-to-many order-preserving encryption (OPE) to encrypt relevance scores and rank the encrypted documents. Cao et al. [16] first proposed a privacy-preserving multi- keyword ranked search scheme (MRSE), which represents documents and queries with binary vectors and uses the secure kNN algorithm (SeckNN) [17] to encrypt the vectors, then use the inner product of the encrypted vectors as the similarity measure. Besides, Yu et al. [18] introduced homomorphic encryption to encrypt relevance scores and realize a multi- keyword ranked search scheme under the vector space model. Recently, Kermanshahi et al. [19] used various homomor- phic encryption techniques to propose a generic solution for supporting multi-keyword ranked searching schemes that can resist against several attacks brought by OPE-based schemes. Secure Semantic Searching. A general limitation of tradi- tional searchable encryption schemes is that they fail to utilize semantic information among words to evaluate the relevance between queries and documents. Fu et al. [3] proposed the first synonym searchable encryption scheme under the vector space model to bridge the gap between semantically related words and given keywords. They first extended the keyword set from the synonym keyword thesaurus built on the New American Roget’s College Thesaurus (NARCT), then used the extended keyword set to build secure indexes with SeckNN. Using the order-preserving encryption algorithm, [5] and [6] presented secure semantic searching schemes based on the mutual information model. Xia et al. [6] proposed a scheme that requires the cloud to constructs a semantic relationship library based on the mutual information used in [20]. However, any schemes based on the inverted index can calculate the mutual information model. Using the SeckNN algorithm, [7], [8], [2] proposed secure semantic searching schemes based on the concept hierarchy.

III. PROBLEM FORMULATION

In this section, we define the system architecture, the security model, and the main notations used in this paper.

A. System Architecture

As illustrated in Fig. 1, there are three entities involved in our system: the data owner, data users, and the cloud server. The data owner has a lot of useful documents, but only has limited resources on the local machines. Therefore, the owner is highly motivated to perform Initialize () for initializing the proposed scheme. The owner encrypts documents F to get ciphertext documents C with secret key K, then outsources C to the cloud server. The data owner builds forward indexes I, then sends indexes I and K to data users.

Data users are the searching requesters that send the trap- door of a query to the cloud server for acquiring top-k related documents. Specifically, users input arbitrary query words q, then perform BuildRLP () to generate word transportation problems Ψ, after transform Ψ to random linear programming problems Ω and the corresponding constant terms Δ as a trap-door. Afterward, users receive top-k encrypted documents and proofs Λ returned from the cloud. Users perform VerDec () to decrypt documents when Λ passes our verification mechanism. The cloud server is an intermediate service provider that stores the encrypted document dataset C and performs the retrieval process. Once receiving the trapdoor, the cloud server performs SeaPro () for leveraging any ready-made optimizer to solve the Ω, then obtains the encrypted minimum word transportation cost values with Δ. The cloud ranks the values in ascending order and returns the top-k encrypted documents to users. In the process, the cloud server also provides proofs Λ for proving the correctness of the search results.

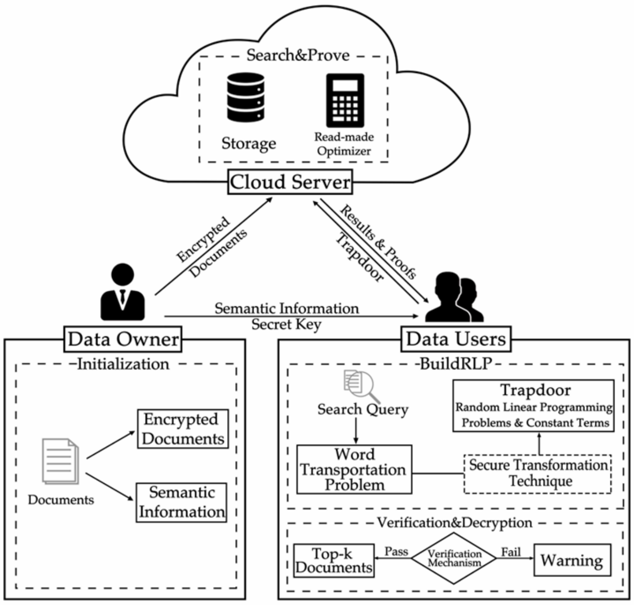

Figure 1. The system architecture of our secure verifiable semantic searching scheme.

B. Security Model

We assume that the data owner is trusted, and the data users are authorized by the data owner. The communication channels between the owner and users are secure on existing security protocols such as SSL, TLS.

With regard to the cloud server, our scheme resists a more challenging security model which is beyond the “semi-honest server” used in other secure semantic searching schemes [3], [4], [5], [6], [7], [8], [9]. In our model, the dishonest cloud server attempts to return wrong/forged search results and learn sensitive information, but would not maliciously delete or tamper with the outsourced documents. Therefore, our secure semantic scheme should guarantee the verifiability, and confidentiality under such a security model. As for verifiability, we first re-formalized the definitions of the Result Forgeries Attack and Proof Forgeries Attack in [24], then adopt a game-based security definition to analyze the verifiability of the proposed scheme in Section VII. Definition 1 (Result Forgeries Attack). The Result Forgeries Attack is that a dishonest cloud server attempts to return erro- neous search results to the users for some reasons. Formally, let q be arbitrary query words, and C be the encrypted documents. Then, let T (C, q) denote the correct search result, let R(C, q) denote the search result returned from the cloud server. In this attack, R(C, q) 6 = T (C, q). Definition 2 (Proof Forgeries Attack). The Proof Forgeries Attack is that a dishonest cloud server attempts to return erro- neous search results and forged proofs to the users. The cloud must generate some forged proofs at a small computational cost for passing the result verification mechanism. Formally, let q be arbitrary query words, C be the encrypted documents. Next, let V(C, q, Λ) = 0 denote the proof Λ pass the verification; otherwise V(C, q, Λ) > 0. Then, let C(Λ) denote the real proofs, let F(Λ) denote the proofs returned from the cloud. In this attack, V(C, q, F(Λ)) = 0 and F(Λ) 6 = C(Λ). As for confidentiality, we follow the widely-accepted Real/Ideal simulation [11], [24], [29] to analyze the confi- dentiality of symmetric searchable encryption schemes. Below we give the definition of confidentiality with respect to the verifiable semantic searching scheme we are going to propose. Definition 3 (Confidentiality). Our verifiable secure semantic searching scheme is secure against adaptively chosen query attack, if for any PPT stateful adversary A, there exists a PPT stateful simulator S, L is stateful leakage algorithms, consider the following probabilistic experiments: Real A (ε) : The adversary A chooses dataset F for a chal- lenger. The challenger runs {K, I, C} ← Initialize (1 ε , F), where ε is our security parameter. A makes a polynomial num- ber of adaptive queries q. For any query q, the challenger acts as a data user and calls (Ω, Δ) ← BuildRLP (q, I, 1 ε , CV). A act as the cloud server and runs SeaPro (). Finally, A returns a bit b as the output of the experiment. Ideal A,S (ε) : The adversary A chooses a document dataset F and makes a polynomial number of adaptive queries q for a simulator S. Given L, S generates and sends C to A, then as a data user to generate the trapdoor, namely Ω and Δ. Finally, A acts as the cloud server and returns a bit b, which is the output of the experiment. A semantic searching scheme is L-confidential if for any PPT adversary A, there exists a PPT simulator S such that: |P r [Real A (ε) = 1] − P r [Ideal A,S (ε) = 1]| ≤ negl(ε) where negl(ε) is a negligible function.

C. Notations

The main notations used in this paper are shown as follows:

• q: The query inputted from a data user.
• d: The number of documents in the dataset.
• m: The number of keywords in a document.
• n: The number of query words in the query.
• F : Plaintext documents dataset F = {f 1 , f 2 . . . f i . . . f d }, where f i denotes a document in the F.
• C: Encrypted documents C = {c 1 , c 2 . . . c i . . . c d }, where c i denotes a document in the C.
• Ψ: WT problems for the q and documents, and Ψ = {ψ 1 , ψ 2 . . . ψ i . . . ψ d }, where ψ i denotes a WT problem for the q with f i.
• Ω: RLP problems for the q and documents, and Ω = {ω 1 , ω 2 . . . ω i . . . ω d }, where ω i denotes a RLP problem for the q with f i.
• θ: The dual problems of the RLP problem ω.
• Δ: Constant terms of every RLP problems, and Δ = {δ 1 , δ 2 . . . δ i . . . δ d }, where δ i denotes the constant term of the RLP problem ω i.
• Λ: Proofs for every RLP problems, and Λ= {λ 1 , λ 2 . . . λ i . . . λ d }, where λ i denotes the proof for ω i.
• β: The minimum word transportation cost value of a WT
problem.
• Π: Optimal values of RLP problems, and Π = {π 1 , π 2 . . . π i . . . π d }, where π i denotes the optimal value of the RLP problem ω i.

TABLE I

THE EUCLIDEAN DISTANCE VALUES BETWEEN WORDS

	university	college	professor	office
university	0	4.94	5.25	6.82
college	4.94	0	5.11	5.18
professor	5.25	5.11	0	5.48
office	6.82	5.18	5.48	0

Ξ: The encrypted minimum word transportation cost values as measurements between q and documents, and $\Xi = \{\xi 1 , \xi 2 , \xi 3 \ldots \xi i \ldots \xi d \}$, where ξi denotes the measurement between q and f i.

IV. PRELIMINARIES

A. Word Embedding

Word embedding is a representative method for words in vector space, through which we can preserve the fundamental properties of words and the semantic relations between them. Neural language models are trained to minimize the prediction error to learn vector representations for words. Therefore, we can perform algebraic operations with word embeddings to probe semantic information between words. As illustrated in Table I, take "university, college, professor, and office" as an example, the Euclidean distance values are just in line with our intuition that the more relevant the words are, the smaller the Euclidean distance is. Word embedding has been studied in plaintext information retrieval tasks, such as query expansion zero-shot retrieval and cross-modal retrieval. In this paper, we use word embeddings to capture semantic information between words without revealing semantic information to the cloud server.

B. Earth Mover's Distance

Earth Mover's Distance (EMD) is introduced as a metric in computer vision to capture the signatures distribution differences between images. The name of EMD comes from its intuitive interpretation: Given two distributions, we regard one as a mass of earth spread properly in space, the other as a collection of holes in that same space. Then, EMD is the result that the minimum amount of work cost to fill the holes with earth. As EMD has advantages in representing problems involving multifeatured signatures, it has been applied to some practical scenarios, such as gesture recognition [36], music genre classification [37], document classification [38], plaintext retrieval [39] and gene identifica- tion [40]. We observe that EMD is a particular case of linear programming problems. Therefore, in this paper, we explore the fundamental theorems of linear programming and security algorithms to design our scheme for realizing secure semantic optimal matching on the ciphertext.

V. PROPOSED APPROACHES

In this section, we present the proposed core approaches in Fig. 1, namely, the word transportation problem, the secure transformation technique, and the verification mechanism.

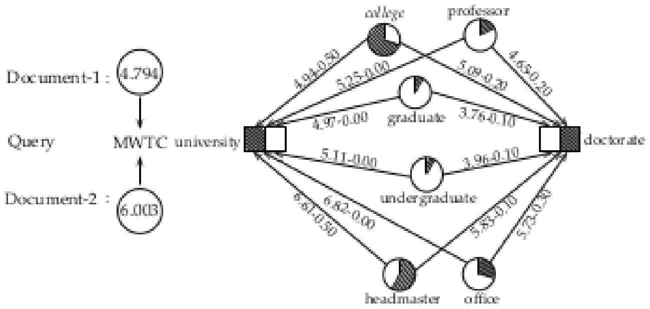

Figure 2. An example of the word transportation optimal matching. The relative area of the shadow represents the weight of a word; the length of the line segment represents the relative Euclidean distance between two connected words; as for the value M-N on the line segment, M represents the Euclidean distance between two words, N represents the amount of transportation between them. In this example, the MWTC between document-1 and the query is 4.794; the MWTC between document-2 and the query is 6.003, so document-1 is more relevant to the query compared with document-2.

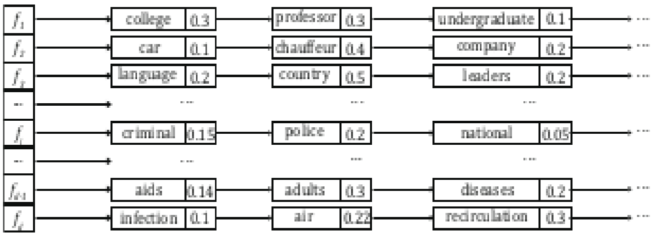

Figure 3. An example of the forward indexes of documents. Forward indexes are the data structure storing the mapping from each document to its keywords. In our scheme, each keyword carries a normalized weight representing the relevant score between the keyword and a specific document.

A. Word Transportation Problem for Optimal Matching

Treating the matching between queries and documents as an optimal matching task, we formulate the word transportation (WT) problem following the optimal transportation problem of linear programming. We utilize WT problems to calculate the minimum word transportation cost (MWTC) as the similarity metric between queries and documents, as illustrated in Fig 2. To represent the documents in WT problems, we introduce the forward indexes as semantic information of documents. An example of forward indexes, as illustrated in Fig. 3. We define each keyword and its weight in the forward index of a document as the keywords distributions for the document. Therefore, we need to select keywords for each document and calculate the weight of each keyword in a specific document. Without loss of generality, we use TF-IDF (term frequency- inverse document frequency) as a criterion to select keywords in our scheme. Besides, we calculate weights via using (1):

$$weight(w,f) = \frac{1}{|f_i|} \cdot \ (1 + In\, f_{i.w}) \cdot In\left(1 + \frac{d}{f_w}\right), \quad (1)$$

where w denotes a specific keyword, *f* expresses a specific document, |*fi*| indicates the length of the document, *f i,w* is the term frequency TF of the keyword w in the *f* , *fw* denotes the number of documents that contain the keyword *w* and *d* is the number of documents in the dataset. We adopt the same method to represent the query and define the weights of query words are equivalent. In this work, we normalize the amount of weight of each document/query to 1. Given forward indexes of documents and the query, we treat the document words as "suppliers," the query words as "consumers," and the semantic information as "product." Therefore, given the forward index of a document f and the query q, we can formulate the WT problem as follows:

$$\begin{aligned} \mathrm{WT}(f,q) = \min & \sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}d_{i,j} \\ \text{subject to} \quad & \sum_{j=1}^{n} f_{i,j} = e_i^f, i = 1,2,\ldots,m \\ & \sum_{i=1}^{m} f_{i,j} = e_j^q, j = 1,2,\ldots,n \qquad (2) \\ & f_{i,j} \geq 0 \\ & \sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j} = 1\,, \end{aligned}$$

where the d i,j represents the transportation cost of each movement, namely, the Euclidean distance values between word embeddings in this work. The f i,j denotes the trans- portation value in a word transportation strategy. The m and n indicate the number of keywords in a document and the query, respectively. The e fi and e qj denote the weight of each word in the document and the query, respectively. Next, we use the matrixes expression method to express (2), as follows:

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & Vx = W \qquad (3) \\ Ix \geq 0\,, & \end{aligned}$$

here, we still define symbol m and n as the number of keywords in a document and the query, respectively. The c T x denotes the total word transportation cost between the query and a document. The symbol c is an mn × 1 cost vector whose elements are Euclidean distance values between word embeddings. The symbol x denotes an mn × 1 decision vector, which means one of the feasible solutions for the word transportation problem. The Vx = W is a constraint condition that requires the amount of each word transportation equal to its weight. The symbol V is an (m + n) × mn known matrix whose elements are 0 or 1. To facilitate the understanding, we show an example for V (when m=3, n=2), The symbol W is an (m+n)×1 weight.

In this work, we calculate the semantic difference between the queries and documents via the word transportation optimal matching. In this way, we can observe that the document is more semantically related to the query when there is less transportation cost between them.

B. Secure Transformation Technique

Word transportation problems can not be applied directly to the secure semantic searching scheme due to that the original WT problem can reveal sensitive information. Therefore, we propose a secure transformation technique to realize semantic optimal matching on the ciphertext so that the confidentiality and integrity of the information in word transportation prob- lems can be guaranteed.

In our scheme, the users utilize our secure transformation technique to transform the WT problems into random linear programming (RLP) problems so that the cloud can lever- age any ready-made optimizer to solve the RLP problems and get the encrypted minimum word transportation cost (EMWTC) without learning sensitive information. Specifi- cally, our secure transformation technique encrypts each WT problem $\psi = (c, V, W, I)$ with a one-time secret key $K\,T = (A, Q, \gamma, r, R)$, where A is an mn × mn random invertible matrix, Q is an $(m + n) \times (m + n)$ random invertible matrix, γ is a real positive value, r is an $mn \times 1$ random vector and R is an $mn \times mn$ generalized permutation matrix. We first transform the original objective function

We first transform the original objective function c T x to the encrypted form $cT\,Ay - cTr$ with $x = Ay - r$. The symbol y denotes an mn × 1 decision vector, which denotes one of the feasible solutions for the RLP problem. Note that, we require each r i is no less than 0, where $i=1, 2,..., mn$. With x replaced by $Ay - r$, we transform the original WT problem ψ to (4). In (4), we define the constraint condition $IAy \geq Ir$ is equivalent to that the i-th element in the vector $T1 = IAy$ is not less than the i-th element in the vector $T2 = Ir$, where $i=1, 2,..., mn$.

$$\begin{aligned} &\min && cTAy - cTr \\ &\text{subject to} && VAy = W + Vr \qquad (4) \\ &IAy \geq Ir\,. \end{aligned}$$

Next, we use a random invertible matrix Q to encrypt the weight vector W, and then we use a real positive value γ to protect the optimal value. Meanwhile, we leave out the identity matrix I due to $IA = A$ is established. Therefore, we transform the original WT problem ψ to (5). In (5), we define the constraint condition $Ay \geq r$ is equivalent to that the i-th element in the vector $T\,3 = Ay$ is not less than the i-th element in the vector r, where $i=1, 2,..., mn$.

$$\begin{aligned} &\min && \gamma c\,T\,Ay - \gamma c\,T\,r \\ &\text{subject to} && QVAy = Q(W + Vr) \qquad (5) \\ &Ay \geq r\,. \end{aligned}$$

To encrypt $Ay \geq r$, we construct an mn × mn generalized permutation matrix R based on the elements in r. Specifically, the nonzero elements in R are reciprocal of elements in the r. We show an example for r and R (when m = 3, n = 2), as illustrated in Fig.5. Therefore, we transform the ψ to (6). In (6), we define the constraint condition RAy ≥ 1 is equivalent to that the elements in the vector T 4 = RAy are not less than 1, where $i=1, 2,..., mn$

$$\begin{aligned} &\min && \gamma c\,T\,Ay - \gamma c\,T\,r \\ &\text{subject to} && QVAy = Q(W + Vr) \qquad (6) \\ &RAy \geq 1. \end{aligned}$$

C. Result Verification Mechanism

To verify the correctness of search results, we design a result verification mechanism using the intermediate data produced in the matching process.

As the optimal matching on the ciphertext is a linear programming (LP) task, we further explore the duality theorem of LP and use the strong theorem of LP problem to design our verification mechanism, inspired by [41]. We first construct the dual programming problem of each RLP problem ω. Given the (7) of ω, we adopt Lagrange multipliers to construct its dual problem θ, as follows:construct the dual programming problem of each RLP problem ω. Given the (7) of ω, we adopt Lagran

max $g(s, t)$

subject to $Vs + I^{T}t = c\,0$

$t \geq 0$ (8)

$g(s, t) = W^{T}\,s + L^{T}t$,

where, g(s, t) is the objective function of the dual problem $\theta = (c\,0,\ V0,\ W0,\ I0,\ L)$, L is an $(m + n) \times 1$ vector whose elements are 1. In the (8), s and t are $(m + n) \times 1$ decision vectors of the dual problem θ.

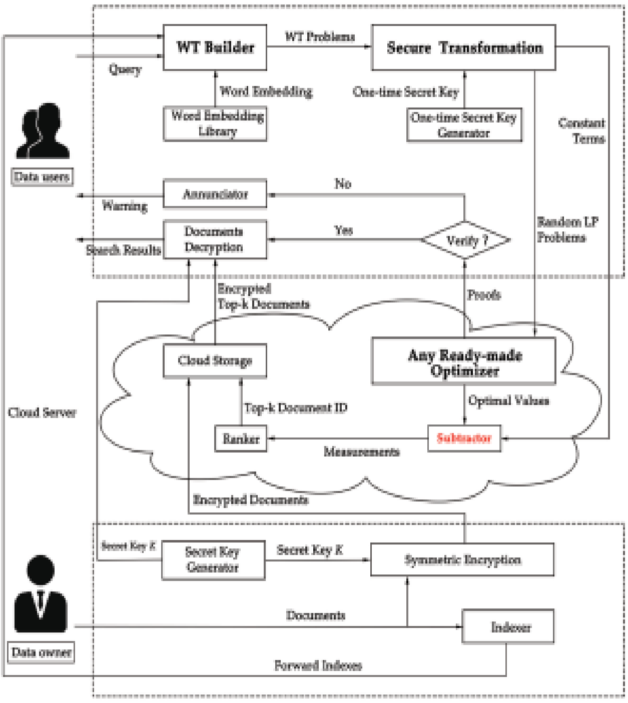

Figure 4. Overview of our secure verifiable semantic searching scheme.

VI. OUR SCHEME

In this section, we present the detailed design of our scheme that consists of four phases, namely, Initialization, BuildRLP, Search&Prove, Verification&Decryption. The overview of our scheme, as illustrated in Fig. 6.

A. Initialization

In this phase, the data owner performs Initialize () to initialize our scheme. To describe this algorithm in detail, we

split it into three algorithms, as follows:

K KeyGen (1ε) is a probabilistic secret key generation algorithm, corresponding to the "Secret Key Generator" in Fig.4. The data owner takes the security parameter ε as input, thengenerates secret key K for encrypting documents.

C EncDoc(K, F) is a deterministic algorithm, corresponding to the "Symmetric Encryption" in Fig. 4. The data owner takes the documents dataset F and the secret key K as input, then generates the ciphertext dataset C.

I BuildIndex(F) is a deterministic building index algorithm, corresponding to the "Indexer" in Fig. 4. The data owner takes F as input, then generates forward indexes I as semantic information of documents.

The data owner first calls KeyGen() and EncDoc() to generate a secret key K for encrypting documents dataset F and get the ciphertext dataset C, then outsources C to the cloud server. Afterward, the owner calls BuildIndex() to build forward indexes I. In this algorithm, the data owner extracts keywords and calculates weights for building forward indexes as semantic information of documents. Finally, the owner sends the secret key K and indexes I to data users.

B. BuildRLP

In this phase, data users perform BuildRLP () to generate trapdoor the searching query q. To describe this algorithm in detail, we split it into three algorithms, as follows:

Ψ BuildWT(q, I, E) is a deterministic algorithm, corresponding to the "WT Builder" in Fig. 6. The users take query q, forward indexes I and word embedding library E as input, then generate word transportation problems Ψ for each pair of query and each document.

K T ← TranKeyGen (1 ε) is a probabilistic transformation key generation algorithm, corresponding to the "One-Secret Key Generator" in Fig. 6. The user takes the security parameter ε as input, then generates one-time transformation secret key K T = (A, Q, γ, r, R) for encrypting Ψ.

(Ω, Δ) ← SecTran (Ψ, K T) is a deterministic algorithm, corresponding to the "Secure Transformation" in Fig. 6. The users take WT problems Ψ and transformation key K T as input, then generate random linear programming problems Ω and the corresponding constant terms Δ.

The users first call BuildWT() to build WT problems Ψ for the query and forward index of each document. Specifically, The users use word embeddings to represent all words and calculate Euclidean distance values between word embeddings, then build word transportation problems Ψ according to the proposed approach. After building WT problems Ψ, the data users call TranKeyGen() to generate a one-time secure key K T for encrypting Ψ. Then, the users call SecureTran() to encrypt each ψ i and get the corresponding RLP problem ω i with its constant term δ i , where ψ i ∈ Ψ, ω i ∈ Ω, δ i ∈ Δ, and i = 1, 2, . . . , d. Finally, the user sends all RLP problems Ω and the corresponding constant terms Δ to the cloud server.

C. Search&Prove

In this phase, the cloud server performs SeaPro () to search documents and generate proofs. To describe this algorithm in detail, we split SeaPro () into two algorithms, namely, SolveRLP () and Rank (), as follows:

(Π, Λ) ← SolveRLP(Ω) is a deterministic algorithm, cor- responding to the "Any Ready-made Optimizer" in Fig. 6. The cloud server takes RLP problems Ω as input, then generates the optimal values Π and proofs Λ for RLP problems.

(Γ, Ξ) Rank(Π, Δ, C, k) is a deterministic ranking algorithm, corresponding to the "Subtractor" and "Ranker" in Fig. 6. The cloud server takes optimal values Π, the constant terms Δ, the ciphertext dataset C and the number k as input, first calculates all the measurements Ξ, then generates the top-k related encrypted documents Γ, where Ξ = {ξ 1 , ξ 2 , ξ 3 . . . ξ i . . . ξ d }, and i = 1, 2, . . . , d.

The cloud server calls SolveRLP() to solve RLP problems. The cloud can leverage any ready-made optimizer to solve each RLP ω i and get the corresponding optimal value π i and proof λ i , where ω i ∈ Ω, π i ∈ Π, λ i ∈ Λ, and i = 1, 2, . . . , d. The cloud calls RankDoc() to calculate each encrypted minimum word transportation cost ξ i = π i – δ i as measurement, where i = 1, 2, . . . , d. Then, the cloud ranks measurements Ξ in ascending order and obtains the top-k related encrypted documents Γ. Finally, the cloud returns the top-k related encrypted documents Γ and proofs Λ to the users.

D. Verification&Decryption

In this phase, data users perform VerDec () to verify the correctness of the search results and decrypt the top-k en- crypted documents. To describe this algorithm in detail, we split it into Verify () and DecDoc (), as follows:

(0 or α) ← Verify(Λ) is a deterministic verification algo- rithm, corresponding to the "Verify ?" in Fig. 6. Data users take proofs Λ as input, then generate the result of verification 0 or α, where α ∈ N ∗ , N ∗ denotes the positive integer set.

ϒ DecDoc(K, Γ) is a deterministic decryption algo- rithm, corresponding to the "Documents Decryption" in Fig. 6. The users take the top-k related encrypted documents Γ and secret key K as input, then generate the top-k related plaintext documents ϒ for the query q.

The users first call Verify() to verify the correctness of the search results. The users verify the correctness of each proof λ i according to (9), thus verifying whether the cloud performs the correct calculations for each RLP problem and determining the correctness of the search result, where λ i ∈ Λ, and i = 1, 2, . . . , d. The

Verify() will output 0 when the verification pass; otherwise, this algorithm calls “Annunciator” to output α as the warning which denotes the number of failing proofs. The users call DecDoc() to decrypt the top- k encrypted documents Γ with the secret key K and obtains the top-k related documents ϒ if the proofs Λ pass our result verification mechanism.

VII. CONCLUSIONS

We propose a secure verifiable semantic searching scheme that treats matching between queries and documents as a word transportation optimal matching task. Therefore, we investigate the fundamental theorems of linear programming (LP) to design the word transportation (WT) problem and a result verification mechanism. We formulate the WT problem to calculate the minimum word transportation cost (MWTC)as the similarity metric between queries and documents, and further propose a secure transformation technique to trans- form WT problems into random LP problems. Therefore, our scheme is simple to deploy in practice as any ready-made optimizer can solve the RLP problems to obtain the encrypted MWTC without learning sensitive information in the WT problems. Meanwhile, we believe that the proposed secure transformation technique can be used to design other privacy-preserving linear programming applications. We bridge the semantic-verifiable searching gap by observing an insight that using the intermediate data produced in the optimal matching process to verify the correctness of search results. Specifically, we investigate the duality theorem of LP and derive a set of necessary and sufficient conditions that the intermediate data must meet. The experimental results on two TREC collections show that our scheme has higher accuracy than other schemes. In the future, we plan to research on applying the principles of secure semantic searching to design secure cross-language searching schemes.

VIII. ACKNOWLEDGMENT

We thanks department of Software System from Wainganga Collge of Engineering and management for help in experiment. We successfully achived the objectves of the paper.

IX. REFERENCES

[1]. D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. IEEE Symp. Secur. Privacy, 2000, pp. 44–55.

[2]. Z. Fu, J. Shu, X. Sun, and N. Linge, “Smart cloud search services: verifiable keyword-based semantic search over encrypted cloud data,” IEEE Trans. Consum. Electron., vol. 60, no. 4, pp. 762–770, 2014.

[3]. Z. J. Fu, X. M. Sun, N. Linge, and L. Zhou, “Achieving effective cloud search services: multi-keyword ranked search over encrypted cloud data supporting synonym query,” IEEE Trans. Consum. Electron., vol. 60, no. 1, pp. 164–172, 2014.

[4]. T. S. Moh and K. H. Ho, “Efficient semantic search over encrypted data in cloud computing,” in Proc. IEEE. Int. Conf. High Perform. Comput. Simul., 2014, pp. 382–390

[5]. N. Jadhav, J. Nikam, and S. Bahekar, “Semantic search supporting similarity ranking over encrypted private cloud data,” Int. J. Emerging Eng. Res. Technol., vol. 2, no. 7, pp. 215–219, 2014.

[6]. Z. H. Xia, Y. L. Zhu, X. M. Sun, and L. H. Chen, “Secure semantic expansion based search over encrypted cloud data supporting similarity ranking,” J. Cloud Comput., vol. 3, no. 1, pp. 1–11, 2014.

[7]. Z. Fu, L. Xia, X. Sun, A. X. Liu, and G. Xie, “Semantic- aware searching over encrypted data for cloud computing,” IEEE Trans. Inf. Forensics Security, vol. 13, no. 9, pp. 2359–2371, Sep. 2018.

[8]. Z. J. Fu, X. L. Wu, Q. Wang, and K. Ren, "Enabling central keyword- based semantic extension search over encrypted outsourced data," IEEE Trans. Inf. Forensics Security., vol. 12, no. 12, pp. 2986–2997, 2017.

[9]. Y. G. Liu and Z. J. Fu, "Secure search service based on word2vec in the public cloud," Int. J. Comput. Sci. Eng., vol. 18, no. 3, pp. 305–313, 2019.

[10]. E. J. Goh, "Secure indexes." IACR Cryptology ePrint Archive, vol. 2003, pp. 216–234, 2003

[11]. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," J. Comput. Secur., vol. 19, no. 5, pp. 895–934, 2011

[12]. M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation." in Proc. ISOC Network Distrib. Syst. Secur. Symp., vol. 20, 2012, pp. 12–26.

[13]. C. Liu, L. H. Zhu, M. Z. Wang, and Y. A. Tan, "Search pattern leakage in searchable encryption: Attacks and new construction," Inf. Sci., vol. 265, pp. 176–188, 2014

[14]. E. Stefanov, C. Papamanthou, and E. Shi, "Practical dynamic searchable encryption with small leakage." in Proc. ISOC Network Distrib. Syst. Secur. Symp., vol. 71, 2014, pp. 72–75.

[15]. C. Wang, N. Cao, J. Li, K. Ren, and W. J. Lou, "Secure ranked keyword search over encrypted cloud data," in Proc. Int. Conf. Distrib. Comput, Syst., 2010, pp. 253– 262.

[16]. N. Cao, C. Wang, M. Li, K. Ren, and W. J. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," IEEE Trans.Parallel Distrib. Syst., vol. 25, no. 1, pp. 222–233, 2013.

[17]. W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in Proc. ACM Symp. Int. Conf. Manage. Data, 2009, pp. 139–152.

[18]. J. D. Yu, P. Lu, Y. M. Zhu, G. T. Xue, and M. L. Li, "Toward secure multikeyword top-k retrieval over encrypted cloud data," IEEE Trans.Dependable Secure Comput., vol. 10, no. 4, pp. 239–250, 2013.

Cite this article as :

Prof. Rahul Bhandekar, Pranay Kumbhalkar, "A Proveable Contextual Retrieval Strategy in the Public Cloud Using Optimal Matching Over Secured Data", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 10 Issue 2, pp. 467-478, March-April 2023. Available at doi : https://doi.org/10.32628/IJSRSET2310233
Journal URL : https://ijsrset.com/IJSRSET2310233