A Verifiable Semantic Searching Scheme by Optimal Matching over Encrypted Data in Public Cloud

Wenyuan Yang, *Student Member, IEEE,* and Yuesheng Zhu, *Senior Member, IEEE,*

***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 verifiable 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 verifiable 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.

Most of the existing secure semantic searching schemes consider the semantic relationship among words to perform query expansion on the plaintext, then still use the query words and extended semantically related words to perform exact matching with the specific keywords in outsourced documents. We can roughly divide these schemes into three categories: secure semantic searching based synonym [3], [4], secure semantic searching based mutual information model [5], [6], secure semantic searching based concept hierarchy [2], [7], [8]. We can see that these schemes only use the elementary semantic information among words. For example, synonym schemes only use synonym attributes; mutual information models only use the co-occurrences information. Although Liu et al. [9] introduce the Word2vec technique to utilize the semantic information of word embeddings, their approach damages the semantic information due to straightly aggregating all the word vectors. We think that secure semantic searching schemes should further utilize a wealth of semantic information among words and perform optimal matching on the ciphertext for high search accuracy.

In this paper, we propose a secure verifiable semantic

The authors are with the Communication and Information Security Laboratory, Shenzhen Graduate School, School of Electronics Engineering and Computer Science Peking University, Shenzhen 518055, China. (Corre-sponding author: Yuesheng Zhu.) (e-mail: wyyang; zhuys@pku.edu.cn).

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 distance 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 secure 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 duality 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 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 between 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 homomorphic 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 traditional 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. For example, Fu et al. [8] proposed a central keyword semantic extension searching scheme which calculates weights of query words based on grammatical relations, then extends the central word based on the concept hierarchy tree from WordNet. Inspired by word embedding used in plaintext information retrieval [21], [22], Liu et al. [9] introduced the Word2vec to represent both queries and documents as compact vectors. However, their approach damages

the semantic information of word embedding due to straightly aggregating all the word vectors of the words.

Verifiable Searching over Encrypted Data. Verifiable searching over encrypted data requires the searchable encryption schemes to verify the correctness of search results. Some works only verify whether all of the encrypted documents containing the single query word returned by the cloud. The first verifiable secure searching scheme was proposed in [23] that leverages a specific trie-like index. Zhu et al. [24] proposed a generic verifiable scheme, which uses Merkle Patricia Tree and Incremental Hash to build the proof index. Some works focus on verifying the correctness of ranked search results by foreseeing the ranked results. Wang et al. [25] proposed a single keyword ranked verification scheme based on the hash chain. Liu et al. [26] present a verifiable dynamic searching scheme leveraging the RSA accumulator to build a verifiable matrix for verifiable updates and searches, which fails to support multi-keyword searching. Sun et al. [27] proposed a multi-keyword ranked verifiable searching scheme via using Merkle Hash tree and cryptographic signature to create a verifiable MDB-tree. For the multi data owners scenario, Zhang et al. [28] proposed a deterrent-based scheme using anchor data to verify the correctness of search results. However, their scheme is unable to support semantic searching and introduces multiple rounds of communication between data owners.

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 $\mathrm{Initialize}\,()$ 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 trapdoor of a query to the cloud server for acquiring top-k related documents. Specifically, users input arbitrary query words q, then perform $\mathrm{BuildRLP}\,()$ to generate word transportation problems Ψ, after transform Ψ to random linear programming problems Ω and the corresponding constant terms Δ as a trapdoor. Afterward, users receive top-k encrypted documents and proofs Λ returned from the cloud. Users perform $\mathrm{VerDec}\,()$ 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 $\mathrm{SeaPro}\,()$ 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.

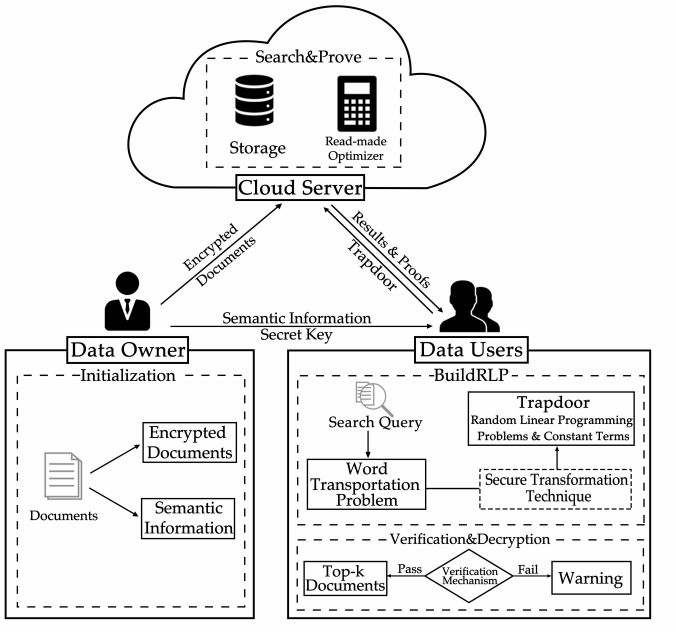

Figure 1. The system architecture of our secure verifiable semantic searching scheme.

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 erroneous search results to the users for some reasons. Formally, let q be arbitrary query words, and C be the encrypted documents. Then, let $\mathcal{T}(C, q)$ denote the correct search result, let $\mathcal{R}(C, q)$ denote the search result returned from the cloud server. In this attack, $\mathcal{R}(\mathrm{C}, q) \neq \mathcal{T}(C, q)$.*

Definition 2 (Proof Forgeries Attack). *The Proof Forgeries Attack is that a dishonest cloud server attempts to return erroneous 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 $\mathcal{V}(C, q, \Lambda) = 0$ denote the proof Λ pass the verification; otherwise $\mathcal{V}(C, q, \Lambda) > 0$. Then, let $\mathcal{C}(\Lambda)$ denote the real*

proofs, let $\mathcal{F}(\Lambda)$ denote the proofs returned from the cloud. In this attack, $\mathcal{V}(C, q, \mathcal{F}(\Lambda)) = 0$ and $\mathcal{F}(\Lambda) \neq \mathcal{C}(\Lambda)$.

As for confidentiality, we follow the widely-accepted Real/Ideal simulation [11], [24], [29] to analyze the confidentiality 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 $\mathcal{A}$, there exists a PPT stateful simulator $\mathcal{S}$, $\mathcal{L}$ is stateful leakage algorithms, consider the following probabilistic experiments:*

$\mathrm{Real}_{\mathcal{A}}(\varepsilon)$ *: The adversary $\mathcal{A}$ chooses dataset F for a challenger. The challenger runs $\{K, I, C\} \leftarrow \mathrm{Initialize}(1^{\varepsilon}, F)$, where ε is our security parameter. $\mathcal{A}$ makes a polynomial number of adaptive queries q. For any query q, the challenger acts as a data user and calls $(\Omega, \Delta) \leftarrow \mathrm{BuildRLP}(q, I, 1^{\varepsilon}, CV)$. $\mathcal{A}$ act as the cloud server and runs $\mathrm{SeaPro}()$. Finally, $\mathcal{A}$ returns a bit b as the output of the experiment.*

$\mathrm{Ideal}_{\mathcal{A},\mathcal{S}}(\varepsilon)$ *: The adversary $\mathcal{A}$ chooses a document dataset F and makes a polynomial number of adaptive queries q for a simulator $\mathcal{S}$. Given $\mathcal{L}$, $\mathcal{S}$ generates and sends C to $\mathcal{A}$, then as a data user to generate the trapdoor, namely Ω and Δ. Finally, $\mathcal{A}$ acts as the cloud server and returns a bit b, which is the output of the experiment.*

A semantic searching scheme is $\mathcal{L}$-confidential if for any PPT adversary $\mathcal{A}$, there exists a PPT simulator $\mathcal{S}$ such that:

$|Pr[\mathrm{Real}_{\mathcal{A}}(\varepsilon) = 1] - Pr[\mathrm{Ideal}_{A,S}(\varepsilon) = 1]| \leq negl(\varepsilon)$ where $negl(\varepsilon)$ 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 $F = \{f_1, f_2 \dots f_i \dots f_d\}$, where f_i denotes a document in the F.
- C: Encrypted documents $C = \{c_1, c_2 \dots c_i \dots c_d\}$, where c_i denotes a document in the C.
- Ψ: WT problems for the q and documents, and $\Psi = \{\psi_1, \psi_2 \dots \psi_i \dots \psi_d\}$, where ψ_i denotes a WT problem for the q with f_i.
- Ω: RLP problems for the q and documents, and $\Omega = \{\omega_1, \omega_2 \dots \ \omega_i \dots \omega_d\}$, 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 $\Delta = \{\delta_1, \delta_2 \dots \delta_i \dots \delta_d\}$, where δ_i denotes the constant term of the RLP problem ω_i.
- Λ: Proofs for every RLP problems, and $\Lambda = \{\lambda_1, \lambda_2 \dots \lambda_i \dots \lambda_d\}$, 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 $\Pi = \{\pi_1, \pi_2 \dots \pi_i \dots \pi_d\}$, 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 $\Xi = \{\xi_1, \xi_2, \xi_3 \dots \ \xi_i \dots \xi_d\}$, where ξ_i 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 [30], [31], [32] 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 [21] zero-shot retrieval [22] and cross-modal retrieval [33]. 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 in [34], [35] 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 identification [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.

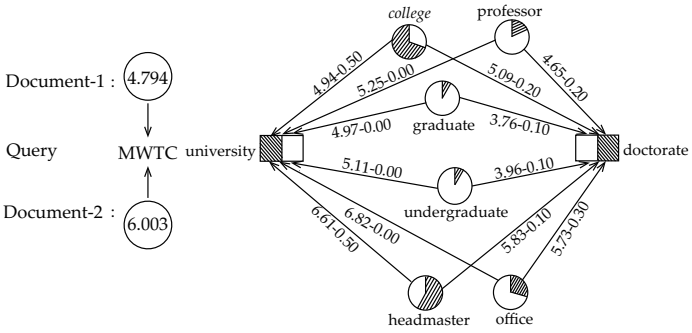

Figure 2. An example of the word transportation optimal matching. The relative area of the shadow represents the weight of a word; the length of the line segment represents the relative Euclidean distance between two connected words; as for the value M-N on the line segment, M represents the Euclidean 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.

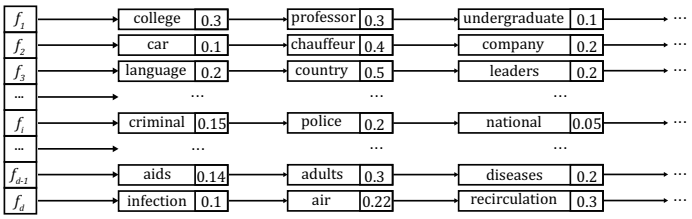

Figure 3. An example of the forward indexes of documents. Forward indexes are the data structure storing the mapping from each document to its keywords. In our scheme, each keyword carries a normalized weight representing the relevant score between the keyword and a specific document.

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):

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

where w denotes a specific keyword, f expresses a specific document, $|f_i|$ indicates the length of the document, $f_{i,w}$ is the term frequency TF of the keyword w in the f, f_w 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.

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Figure 4. An example of the matrix V, when m=3, n=2. The matrix V is used to build the constraint $\mathbf{Vx} = \mathbf{W}$ in our word transportation problem.

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 $\mathbf{f}$ and the query $\mathbf{q}$, we can formulate the WT problem as follows:

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

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 transportation value in a word transportation strategy. The m and n indicate the number of keywords in a document and the query, respectively. The e_i^f and e_j^q 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:

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

here, we still define symbol m and n as the number of keywords in a document and the query, respectively. The $\mathbf{c}^T\mathbf{x}$ denotes the total word transportation cost between the query and a document. The symbol $\mathbf{c}$ is an $mn \times 1$ cost vector whose elements are Euclidean distance values between word embeddings. The symbol $\mathbf{x}$ denotes an $mn \times 1$ decision vector, which means one of the feasible solutions for the word transportation problem. The $\mathbf{Vx} = \mathbf{W}$ is a constraint condition that requires the amount of each word transportation equal to its weight. The symbol $\mathbf{V}$ is an $(m+n) \times mn$ known matrix whose elements are 0 or 1. To facilitate the understanding, we show an example for $\mathbf{V}$ (when m=3, n=2), as illustrated in Fig 4. The symbol $\mathbf{W}$ is an $(m+n)\times 1$ weight

vector, where the first m elements are the weights of document keywords and the last n elements are the weights of query words. As the constraint $\mathbf{Vx} = \mathbf{W}$ implies the total amount of transportation is equal to the normalized total weight 1, we remove the constraint $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j} = 1$ which is in (2), where $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. The symbol $\mathbf{I}$ is an $mn \times mn$ identity matrix. We define constraint condition $\mathbf{Ix} \geq \mathbf{0}$ is equivalent to that x_i is not less than 0, where $i = 1, 2, \ldots, mn$. The $\mathbf{Ix} \geq \mathbf{0}$ is essential but easy to be ignored. In a word, we use $\psi = (\mathbf{c}, \mathbf{V}, \mathbf{W}, \mathbf{I})$ to denote the WT problem in (3). Besides, we define that β indicates the optimal value of the WT problem ψ, namely, the MWTC between the query and a document. Therefore, the document and the query are more related to each other when β is smaller.

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 problems 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 leverage any ready-made optimizer to solve the RLP problems and get the encrypted minimum word transportation cost (EMWTC) without learning sensitive information. Specifically, our secure transformation technique encrypts each WT problem $\psi = (\mathbf{c}, \mathbf{V}, \mathbf{W}, \mathbf{I})$ with a one-time secret key $K_T = (\mathbf{A}, \mathbf{Q}, \gamma, \mathbf{r}, \mathbf{R})$, where $\mathbf{A}$ is an $mn \times mn$ random invertible matrix, $\mathbf{Q}$ is an $(m+n) \times (m+n)$ random invertible matrix, γ is a real positive value, $\mathbf{r}$ is an $mn \times 1$ random vector and $\mathbf{R}$ is an $mn \times mn$ generalized permutation matrix.

We first transform the original objective function $\mathbf{c}^T\mathbf{x}$ to the encrypted form $\mathbf{c}^T\mathbf{Ay} - \mathbf{c}^T\mathbf{r}$ with $\mathbf{x} = \mathbf{Ay} - \mathbf{r}$. The symbol $\mathbf{y}$ denotes an $mn \times 1$ 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=$1, 2, \ldots, mn$. With $\mathbf{x}$ replaced by $\mathbf{Ay} - \mathbf{r}$, we transform the original WT problem ψ to (4). In (4), we define the constraint condition $\mathbf{IAy} \geq \mathbf{Ir}$ is equivalent to that the i-th element in the vector $\mathbf{T}_1 = \mathbf{IAy}$ is not less than the i-th element in the vector $\mathbf{T}_2 = \mathbf{Ir}$, where i=$1, 2, \ldots, mn$.

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{Ay} - \mathbf{c}^T\mathbf{r} \\ \text{subject to} \quad & \mathbf{VAy} = \mathbf{W} + \mathbf{Vr} \\ & \mathbf{IAy} \geq \mathbf{Ir}\,. \end{aligned} \tag{4}$$

Next, we use a random invertible matrix $\mathbf{Q}$ to encrypt the weight vector $\mathbf{W}$, and then we use a real positive value γ to protect the optimal value. Meanwhile, we leave out the identity matrix $\mathbf{I}$ due to $\mathbf{IA} = \mathbf{A}$ is established. Therefore, we transform the original WT problem ψ to (5). In (5), we define the constraint condition $\mathbf{Ay} \geq \mathbf{r}$ is equivalent to that the i-th element in the vector $\mathbf{T}_3 = \mathbf{Ay}$ is not less than the i-th element in the vector $\mathbf{r}$, where i=$1, 2, \ldots, mn$.

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

To encrypt $\mathbf{Ay} \geq \mathbf{r}$, we construct an $mn \times mn$ generalized permutation matrix $\mathbf{R}$ based on the elements in $\mathbf{r}$. Specifically, the nonzero elements in $\mathbf{R}$ are reciprocal of elements in the $\mathbf{r}$. We show an example for $\mathbf{r}$ and $\mathbf{R}$ (when $m = 3$, $n = 2$), as illustrated in Fig.5. Therefore, we transform the ψ to (6). In (6), we define the constraint condition $\mathbf{RAy} \geq \mathbf{1}$ is equivalent to that the elements in the vector $\mathbf{T}_4 = \mathbf{RAy}$ are not less than 1, where i=$1, 2, \ldots, mn$.

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

The $\delta = \gamma\mathbf{c}^{\mathrm{T}}\mathbf{r}$ does not affect the decision vector $\mathbf{y}$ since δ is a constant term when γ and $\mathbf{r}$ are appointed. Therefore, we use the (7) to express the final RLP problem via temporarily omitting the constant term δ. In (7), we define constraint condition $\mathbf{I}'\mathbf{y} \geq \mathbf{1}$ is equivalent to that the i-th element in the vector $\mathbf{T}_5 = \mathbf{I}'\mathbf{y}$ is not less than 1, where i=$1, 2, \ldots, mn$.

$$\begin{aligned} \min \quad & \mathbf{c}'^T\mathbf{y} \\ \text{subject to} \quad & \mathbf{V}'\mathbf{y} = \mathbf{W}' \\ & \mathbf{I}'\mathbf{y} \geq \mathbf{1}\,, \end{aligned} \tag{7}$$

where $\mathbf{V}' = \mathbf{QVA}$, $\mathbf{W}' = \mathbf{Q}(\mathbf{W} + \mathbf{Vr})$, $\mathbf{I}' = \mathbf{RA}$, $\mathbf{c}' = (\gamma\mathbf{c}^T\mathbf{A})^T$. We use $\omega = (\mathbf{c}', \mathbf{V}', \mathbf{W}', \mathbf{I}')$ to denote the random linear programming problem of a specific WT problem ψ. The RLP problem ω has a similar structure to the WT problem ψ as in (7) and (3). Therefore, we can get the optimal value π and the decision vector $\mathbf{y}$ of each RLP ω by solving the ω leveraging any ready-made optimizer, such as GUROBI.

We define the EMWTC as the measurement ξ=π-δ which is used to rank the documents. The smaller the ξ becomes, the smaller the gap between the query and document is. Therefore, we succeed in calculating the measurements between a query and documents in a privacy-preserving way. After solving

$$\begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{r_5} \\ 0 & 0 & \frac{1}{r_6} & 0 & 0 & 0 \\ 0 & \frac{1}{r_4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{r_2} & 0 \\ \frac{1}{r_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{r_3} & 0 & 0 \end{pmatrix}$$

$\mathbf{r}$ $\mathbf{R}$

Figure 5. An example of the generation process of the matrix $\mathbf{R}$, when m=3, n=2. The matrix $\mathbf{R}$ is an important part of secret key K_T, which is used to hide sensitive information of non-negativity constraint $\mathbf{Ix} \geq \mathbf{0}$ in our word transportation problem. The nonzero elements in $\mathbf{R}$ are reciprocal of elements in the random vector $\mathbf{r}$.

all RLP problems $\Omega = \{\omega_1, \omega_2, \omega_3 \dots \ \omega_i \dots \omega_d\}$, the cloud ranks all the measurements $\Xi = \{\xi_1, \xi_2, \xi_3 \dots \ \xi_i \dots \xi_d\}$ in ascending order and returns the top-k related encrypted documents to data users. We give a theoretical analysis of the reason why the ξ can be the measurement between the query and a document in section VII.

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:

$$\begin{aligned} \max \quad & g(\mathbf{s}, \mathbf{t}) \\ \text{subject to} \quad & \mathbf{V'}^T\mathbf{s} + \mathbf{I'}^T\mathbf{t} = \mathbf{c'} \\ & \mathbf{t} \geq \mathbf{0} \\ g(\mathbf{s}, \mathbf{t}) &= \mathbf{W'}^T\mathbf{s} + \mathbf{L}^T\mathbf{t}\,, \end{aligned} \tag{8}$$

where, $g(\mathbf{s}, \mathbf{t})$ is the objective function of the dual problem $\theta = (\mathbf{c'}, \mathbf{V'}, \mathbf{W'}, \mathbf{I'}, \mathbf{L})$, $\mathbf{L}$ is an $(m+n) \times 1$ vector whose elements are 1. In the (8), $\mathbf{s}$ and $\mathbf{t}$ are $(m+n) \times 1$ decision vectors of the dual problem θ.

In [42], the strong theorem of the LP problem demonstrates that if $\mathbf{y}$ and $(\mathbf{s}, \mathbf{t})$ are the feasible decision vectors for ω and θ respectively, $\mathbf{y}$ and $(\mathbf{s}, \mathbf{t})$ lead to the same optimal value for ω and θ, then $\mathbf{y}$ and $(\mathbf{s}, \mathbf{t})$ are the optimal decision vectors for ω and θ, respectively. Therefore, we get a corollary in (9) as the verification condition to verify whether the cloud server performs correct calculations for each RLP problem.

$$\begin{aligned} \mathbf{c'}^T\mathbf{y} =& \mathbf{W'}^T\mathbf{s} + \mathbf{L}^T\mathbf{t} \\ \mathbf{V'y} &= \mathbf{W'} \\ \mathbf{I'y} &\geq \mathbf{1} \\ \mathbf{V'}^T\mathbf{s} + \mathbf{I'}^T\mathbf{t} &= \mathbf{c'} \\ \mathbf{t} &\geq \mathbf{0}\,. \end{aligned} \tag{9}$$

In our scheme, the cloud server solves each RLP problem and its dual problem at the same time, then packs the optimal decision vectors $\mathbf{y}$ and $(\mathbf{s}, \mathbf{t})$ as a proof λ. Therefore, the users receive proofs $\Lambda = \{\lambda_1, \lambda_2, \lambda_3 \dots \lambda_i \dots \lambda_d\}$ and perform verification mechanism according to the verification condition (9). Finally, the users can verify whether the cloud server performs correct calculations for all RLP problems and determine the correctness for the search results. On the other hand, the cloud would be honest because the cloud server knows, the users would catch him when the cloud behaves dishonestly. In our verification mechanism, we do not mandate the users to calculate the encrypted minimum word transportation cost values and rank them for saving computing resources. Therefore, we make an assumption that if a rational cloud has run the complex calculation to solve RLP problems, it will perform the low computational cost ranking task. We give theoretical analysis and experimental analysis to indicate the rationality of our assumption in section VII and VIII.

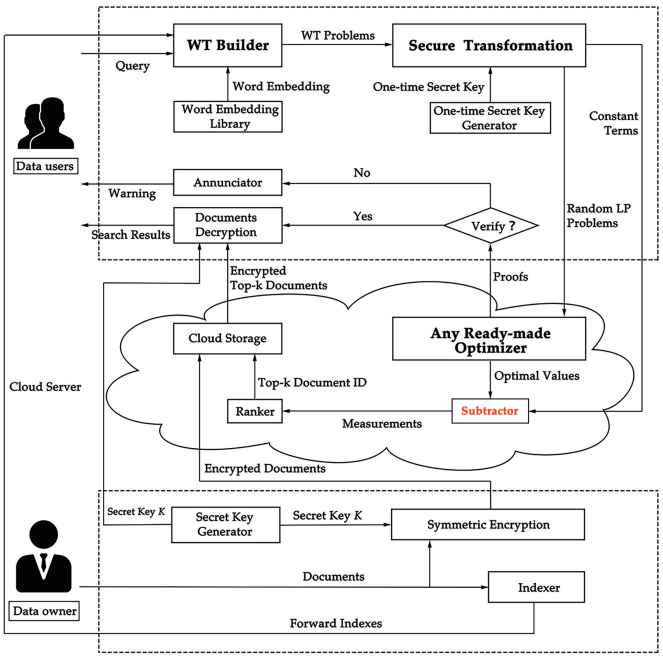

Figure 6. Overview of our secure verifiable semantic searching scheme.

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 $\mathrm{Initialize}()$ to initialize our scheme. To describe this algorithm in detail, we split it into three algorithms, as follows:

$K \leftarrow \mathrm{KeyGen}(1^{\varepsilon})$ is a probabilistic secret key generation algorithm, corresponding to the "Secret Key Generator" in Fig. 6. The data owner takes the security parameter ε as input, then generates secret key K for encrypting documents.

$C \leftarrow \mathrm{EncDoc}(K, F)$ is a deterministic algorithm, corresponding to the "Symmetric Encryption" in Fig. 6. The data owner takes the documents dataset F and the secret key K as input, then generates the ciphertext dataset C.

$I \leftarrow \mathrm{BuildIndex}(F)$ is a deterministic building index algorithm, corresponding to the "Indexer" in Fig. 6. The data owner takes F as input, then generates forward indexes I as semantic information of documents.

The data owner first calls $\mathrm{KeyGen}()$ and $\mathrm{EncDoc}()$ 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 $\mathrm{BuildRLP}\,()$ to generate trapdoor the searching query q. To describe this algorithm in detail, we split it into three algorithms, as follows:

$\Psi \leftarrow \mathrm{BuildWT}(q, I, E)$ 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.

$K_T \leftarrow \mathrm{TranKeyGen}\,(1^{\varepsilon})$ 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 $K_T = (\mathbf{A}, \mathbf{Q}, \gamma, \mathbf{r}, \mathbf{R})$ for encrypting Ψ.

$(\Omega, \Delta) \leftarrow \mathrm{SecTran}\,(\Psi, K_T)$ 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 $\mathrm{BuildWT}()$ 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 $\mathrm{TranKeyGen}()$ to generate a one-time secure key K_T for encrypting Ψ. Then, the users call $\mathrm{SecureTran}()$ to encrypt each ψ_i and get the corresponding RLP problem ω_i with its constant term δ_i, where $\psi_i \in \Psi$, $\omega_i \in \Omega$, $\delta_i \in \Delta$, and $i = 1, 2, \ldots, d$. 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 $\mathrm{SeaPro}\,()$ to search documents and generate proofs. To describe this algorithm in detail, we split $\mathrm{SeaPro}\,()$ into two algorithms, namely, $\mathrm{SolveRLP}\,()$ and $\mathrm{Rank}\,()$, as follows:

$(\Pi, \Lambda) \leftarrow \mathrm{SolveRLP}(\Omega)$ is a deterministic algorithm, corresponding 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.

$(\Gamma, \Xi) \leftarrow \mathrm{Rank}(\Pi, \Delta, C, k)$ 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 $\Xi = \{\xi_1, \xi_2, \xi_3 \ldots\ \xi_i \ldots \xi_d\}$, and $i = 1, 2, \ldots, d$.

The cloud server calls $\mathrm{SolveRLP}()$ 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 $\omega_i \in \Omega$, $\pi_i \in \Pi$, $\lambda_i \in \Lambda$, and $i = 1, 2, \ldots, d$. The cloud calls $\mathrm{RankDoc}()$ to calculate each encrypted minimum word transportation cost $\xi_i = \pi_i - \delta_i$ as measurement, where $i = 1, 2, \ldots, d$. 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 $\mathrm{VerDec}\,()$ to verify the correctness of the search results and decrypt the top-k encrypted documents. To describe this algorithm in detail, we split it into $\mathrm{Verify}\,()$ and $\mathrm{DecDoc}\,()$, as follows:

$(0 \text{ or } \alpha) \leftarrow \mathrm{Verify}(\Lambda)$ is a deterministic verification algorithm, corresponding to the "Verify ?" in Fig. 6. Data users take proofs Λ as input, then generate the result of verification 0 or α, where $\alpha \in \mathbb{N}^*$, $\mathbb{N}^*$ denotes the positive integer set.

$\Upsilon \leftarrow \mathrm{DecDoc}(K, \Gamma)$ is a deterministic decryption algorithm, 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 $\mathrm{Verify}()$ 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 $\lambda_i \in \Lambda$, and $i = 1, 2, \ldots, d$. The $\mathrm{Verify}()$ 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 $\mathrm{DecDoc}()$ 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. THEORETICAL ANALYSIS

In this section, we give theoretical analyses on the correctness, rationality, and security of our scheme.

A. Correctness Analysis

We analyze the reason why the encrypted minimum word transportation cost ξ can be used as a measurement.

Theorem 1: When $\mathbf{y}$ *is the optimal decision vector of an RLP problem* ω*, the corresponding* $\mathbf{x}$ *is the optimal decision vector of the original WT problem* ψ*, where* $\mathbf{x} = \mathbf{A}\mathbf{y} - \mathbf{r}$.

Proof: We prove Theorem 1 by using the Reductio ad ab surdum. We first suppose that $\mathbf{x}$ and $\mathbf{y}'$ are the optimal decision vectors for ψ and ω respectively, and there is no mathematical relationship $\mathbf{x} = \mathbf{A}\mathbf{y}' - \mathbf{r}$ between $\mathbf{x}$ and $\mathbf{y}'$. We can deduce $\mathbf{c}^T\mathbf{x}$ and $\gamma\mathbf{c}^T\mathbf{A}\mathbf{y}'$ are the optimal values for ψ and ω respectively. According to $\mathbf{x} = \mathbf{A}\mathbf{y} - \mathbf{r}$ and $\mathbf{x}' = \mathbf{A}\mathbf{y}' - \mathbf{r}$, we can deduce $\gamma\mathbf{c}^T\mathbf{x}' = \gamma\mathbf{c}^T\mathbf{A}\mathbf{y}' - \gamma\mathbf{c}^T\mathbf{r}$, $\gamma\mathbf{c}^T\mathbf{x} = \gamma\mathbf{c}^T\mathbf{A}\mathbf{y} - \gamma\mathbf{c}^T\mathbf{r}$, and $\gamma\mathbf{c}^T\mathbf{A}\mathbf{y}' - \gamma\mathbf{c}^T\mathbf{r} < \gamma\mathbf{c}^T\mathbf{A}\mathbf{y} - \gamma\mathbf{c}^T\mathbf{r}$, namely, $\gamma\mathbf{c}^T\mathbf{x}' < \gamma\mathbf{c}^T\mathbf{x}$. Thus, we get a corollary that the $\mathbf{x}'$ is the optimal decision vector for ψ, namely, $\mathbf{x} = \mathbf{x}' = \mathbf{A}\mathbf{y}' - \mathbf{r}$. However, this corollary contradicts the precondition. Theorem 1 is established. ■

Theorem 2: The encrypted minimum word transportation cost ξ *is* γ *times of the minimum word transportation cost* β.

Proof: According to Theorem 1, the optimal decision vector $\mathbf{x}$ and $\mathbf{y}$ meet the mathematical relationship $\mathbf{x} = \mathbf{A}\mathbf{y} - \mathbf{r}$. Therefore, we can get the final semantic difference value $\xi = \pi - \delta = \gamma\mathbf{c}^T\mathbf{A}\mathbf{y} - \gamma\mathbf{c}^T\mathbf{r} = \gamma\left(\mathbf{c}^T\mathbf{A}\mathbf{y} - \mathbf{c}^T\mathbf{r}\right) = \gamma\mathbf{c}^T\mathbf{x} = \gamma\beta$. Therefore, Theorem 2 is established. ■

As the ξ is γ times of the β, we define the encrypted minimum word transportation cost ξ as the measurement. Therefore, the cloud can calculate and rank the measurements without learning any sensitive information of documents and queries in our scheme.

B. Rationality Analysis

In this subsection, we analyze the rationality of our scheme via complexity analysis.

We still define that d denotes the number of documents in the dataset, m denotes the number of keywords in a document and n denotes the number of query words. For the data users, the most time-consuming operations in our secure transformation technique are the matrix-matrix operations. The complexity of each step is analyzed as follows. Encrypting the cost vector $\mathbf{c}$ in the word transportation problem ψ to obtain $\mathbf{c}' = (\gamma \mathbf{c}^T \mathbf{A})^T$, its complexity is $\mathcal{O}\left(dm^2n^2\right)$. Then, encrypting the constraints in the ψ to get $\mathbf{V}' = \mathbf{QVA}$, $\mathbf{W}' = \mathbf{Q}(\mathbf{W} + \mathbf{Vr})$, and $\mathbf{I}' = \mathbf{RA}$, its complexity is $\mathcal{O}\left(\max\left\{d(m+n)^2mn, d(mn)^3\right\}\right)$. Finally, obtaining the constant term $\delta = \gamma \mathbf{c}^{\mathrm{T}}\mathbf{r}$, its complexity is $\mathcal{O}(dmn)$. To summarize, the proposed secure transformation technique requires computational complexity of $\mathcal{O}\left(\max\left\{d(m+n)^2mn, d(mn)^3\right\}\right)$ at the user side. We adopt $\mathcal{O}\left(dm^3n^3\right)$ as the computational complexity of the user side due to $m > 2$ and $n > 2$ usually in our scheme.

For the cloud, the time-consuming operations are solving the RLP problems and dual problems, and the ranking task. Most of the ready-made optimizers solve both the primal problems and dual problems at the same time via using the interior point method. Therefore, the cloud server solves the RLP problems and dual problems with computation of $\mathcal{O}\left(dm^{3.5}n^{3.5}L\right)$, where L denotes the input length of the problem, i.e., the total binary length of the numerical data specifying the problem instance. The ranking task includes calculating all of the minimum word transportation cost values Ξ and ranking these values, resulting in the computational complexity of $\mathcal{O}\left(d + d\log_2 d\right)$. We can get the following conclusions and further demonstrate these conclusions with experimental results in section VIII.

1. It is rational that data users outsource the retrieval task to the cloud server due to the complexity $\mathcal{O}\left(dm^3n^3\right) \ll \mathcal{O}\left(dm^{3.5}n^{3.5}L\right)$.
2. A rational cloud server would perform the ranking task honestly if the cloud ran the complex calculation honestly for solving RLP problems to obtain the encrypted minimum word transportation cost due to the complexity of $\mathcal{O}\left(dm^{3.5}n^{3.5}L\right) \gg \mathcal{O}\left(d + d\log_2 d\right)$ at the cloud side.

C. Security Analysis

In this subsection, we elaborate on the security analysis the proposed verifiable semantic searching scheme in two aspects, i.e., verifiability and confidentiality.

*1) **Verifiability**:* The verifiability means that the proposed scheme can verify the correctness of the search results and proofs returned from the cloud. We adopt the game-based security definition to prove the verifiability of our scheme.

Definition 4(Verifiability). *Let the proposed scheme be verifiable and consider the following probabilistic experiment on our scheme, where $\mathcal{A}$ is a stateful adversary:* $\mathbf{Vrf}_{\mathcal{A}}(\varepsilon)$:

1. *The challenger calls* $\mathrm{KeyGen}\left(1^{\varepsilon}\right)$ *to generate the symmetric secret key* K.
2. *The adversary* $\mathcal{A}$ *chooses a document dataset* F *for the challenger.*
3. *The challenger calls* $\mathrm{EncDoc}(K, F)$ *to generate ciphertext dataset* C *and calls* $\mathrm{BuildIndex}(F)$ *to generate forward indexes* I.
4. *Given* $\{C, I\}$ *and oracle access to* $\mathrm{BuildWT}(q, I, E)$, $\mathrm{TranKeyGen}\left(1^{\varepsilon}\right)$, $\mathrm{SecTran}\left(\Psi, K_T\right)$, *the adversary* $\mathcal{A}$ *generates a set of RLP problems* Ω_q *and the corresponding constant terms* Δ_q *according to a query* q, *then outputs a sequence of documents* C'_q *and the proofs* Λ'_q. *However,* $C'_q \neq C_q$, *where* C_q *denotes the real search result of the query* q.
5. *The challenger calls* $\mathrm{Verify}(\Lambda'_q)$ *to generate a bit* η
6. *Finally, this probabilistic experiment outputs the bit* η

Our scheme is verifiable if for any probabilistic polynomial-time adversary $\mathcal{A}$, $Pr\left[\mathbf{Vrf}_{\mathcal{A}}(\varepsilon) = 1\right] \leq \mathrm{negl}(\varepsilon)$.

Therefore, the verifiability of our scheme means that it prevents the Result Forgeries Attack and Proof Forgeries Attack. We prove it via theorem 3.

Theorem 3: Our secure semantic searching scheme guarantee the verifiability if the probability $Pr[Misbehavior]$ *that any probabilistic polynomial time (PPT) adversary* $\mathcal{A}$ *successfully forge search result* $\mathcal{R}(C, q)$ *and proofs* $\mathcal{F}(\Lambda)$, $Pr[Misbehavior]$ *is negligible.*

Proof: According to the definitions of the Result Forgeries Attack and Proof Forgeries Attack, we can deduce the successfully probability of $\mathcal{A}$ as follow: $Pr[Misbehavior] = Pr[(\mathcal{R}(\mathrm{C}, q) \neq \mathcal{T}(C, q)) \cap (\mathcal{V}(C, q, \mathcal{F}(\Lambda)) = 0)]$, where $\mathcal{R}(C, q)$ denotes the search result from the cloud server, $\mathcal{T}(C, q)$ denotes the correct search result, $\mathcal{F}(\Lambda)$ denotes the proofs returned from the cloud, $\mathcal{V}(C, q, \mathcal{F}(\Lambda)) = 0$ denotes the proofs $\mathcal{F}(\Lambda)$ pass the verification. To prove $Pr[Misbehavior] \leq negl(\varepsilon)$, we just prove that $\mathcal{A}$ has the successfully probability $Pr[Mis]$ of forging each proof $\mathcal{F}(\lambda)$ in $\mathcal{F}(\Lambda)$, $Pr[Mis] \leq neg(\varepsilon)$, since $negl(\varepsilon) = \sum_{i=1}^{d} neg(\varepsilon)$ based on applying union bound for all proofs.

We prove $Pr[Mis] \leq neg(\varepsilon)$ according to duality theorem [42]. We first suppose that $\mathcal{F}(\lambda) \neq \mathcal{C}(\lambda)$ and $\mathcal{F}(\lambda)$ could pass our verification mechanism, namely, $\mathcal{F}(\lambda) = \{\mathcal{F}(\mathbf{y}), \mathcal{F}(\mathbf{s}), \mathcal{F}(\mathbf{t})\}$ meets (9), where $\mathcal{C}(\lambda)$ denotes the real proof for a specific problem ω. We split the (9) into (10), (11) and (12), as follows:

$$\begin{aligned} v_1 &= \mathbf{c}^T \mathcal{F}(\mathbf{y}) \\ \mathbf{V}' \mathcal{F}(\mathbf{y}) &= \mathbf{W}' \\ \mathbf{I}' \mathcal{F}(\mathbf{y}) &\geq \mathbf{1}\,, \end{aligned} \tag{10}$$

$$\begin{aligned} v_2 = \mathbf{W}'^T \mathcal{F}(\mathbf{s}) + \mathbf{L}^T \mathcal{F}(\mathbf{t}) \\ \mathbf{V}'^T \mathcal{F}(\mathbf{s}) + \mathbf{I}'^T \mathcal{F}(\mathbf{t}) = \mathbf{c}' \\ \mathcal{F}(\mathbf{t}) \geq \mathbf{0}\,, \end{aligned} \tag{11}$$

$$v_1 = v_2\,. \tag{12}$$

According to (10) and (11), we can deduce that $\mathcal{F}(\mathbf{y})$ and $\{\mathcal{F}(\mathbf{s}), \mathcal{F}(\mathbf{t})\}$ are the feasible decision vectors for a RLP problem ω and its dual problem θ, respectively. We also get that v_1 and v_2 are the optimal values of ω and θ, respectively. Therefore, there is $v_1 \geq v_2$ between v_1 and v_2 according to the weak duality lemma [42]. Next, we can deduce that $\mathcal{F}(\mathbf{y})$ and $\{\mathcal{F}(\mathbf{s}), \mathcal{F}(\mathbf{t})\}$ make the problems obtain the same optimal values $v_1 = v_2$ according to (12). The strong dual theorem [42] claims that if x^* and y^* are feasible for a linear programming and its dual problem, respectively, and if x^* and y^* make the problems obtain the same optimal values, then x^* and y^* are optimal for their respective problems. Therefore, we can get a corollary that $\mathcal{F}(\lambda) = \{\mathcal{F}(\mathbf{y}), \mathcal{F}(\mathbf{s}), \mathcal{F}(\mathbf{t})\}$ meets the strong theorem of the LP problem, thus $\mathcal{F}(\lambda) = \mathcal{C}(\lambda)$. Therefore, for any PPT adversary $\mathcal{A}$, there exits:$Pr[Mis] = 0 \leq neg(\varepsilon)$.

In a word, $Pr[Misbehavior] \leq negl(\varepsilon)$. Therefore, the proposed scheme is verifiable. ■

*2) **Confidentiality**:* The confidentiality means that the proposed scheme can guarantee that any adversary $\mathcal{A}$ cannot any useful sensitive information about documents F and query q through the trapdoor (the corresponding RLP problems Ω and constant terms Δ) and proofs Λ. We demonstrate that our scheme guarantees the confidentiality defined by Definition 4. To elaborate the confidentiality of our scheme, we first define leakage function $\mathcal{L}(F) = (\Omega_q, \Delta_q, \Lambda_q)$ which refers to the maximum information that is allowed to learn by the adversary $\mathcal{A}$, where F denotes the documents, Ω_q and Δ_q denotes the RLP problems and constant terms which are adaptively generated from q as the trapdoor, Λ_q denotes the proofs.

Theorem 4: Our verifiable semantic searching scheme is $\mathcal{L}$-confidential if the proposed secure transformation technique is secure.

Proof: We prove it by describing a polynomial-time simulator $\mathcal{S}$ such that for any PPT adversary $\mathcal{A}$, the output between $\mathrm{Real}_{\mathcal{A}}(\varepsilon)$ and $\mathrm{Ideal}_{\mathcal{A},\mathcal{S}}(\varepsilon)$ are computationally indistinguishable:

$|Pr[\mathrm{Real}_{\mathcal{A}}(\varepsilon) = 1] - Pr[\mathrm{Ideal}_{A,S}(\varepsilon) = 1]| \leq negl(\varepsilon)$.

Given the leakage function $\mathcal{L}(F) = (\Omega_q, \Delta_q, \Lambda_q)$, the simulator $\mathcal{S}$ can simulate the search trapdoor (random linear programming problems and constant terms) and proofs. for every query q, $\mathcal{S}$ simulate and generate randomly the trapdoor (problems $\tilde{\Omega}$ and terms $\tilde{\Delta}$) and proofs $\tilde{\Lambda}$. Our scheme is secure if any PPT adversary $\mathcal{A}$ cannot differentiate the real trapdoor and proofs from the simulated trapdoor and proofs. Therefore, we just need to prove that any PPT $\mathcal{A}$ is able to deduce each ψ_i in Ψ through the ω_i in Ω and the corresponding Δ_i in Δ, and proof λ_i in Λ_i with negligible success probability, where $\forall i \in [1, d]$, d is the number of document in dataset. To prove it, we formally analyze that our secure transformation technique guarantees the input/output privacy to original word transportation problem $\psi = (\mathbf{c}, \mathbf{V}, \mathbf{W}, \mathbf{I})$.

First, we prove that the proposed technique guarantees the output privacy of ψ, namely, provides the security to the optimal decision vector $\mathbf{x}$ and the real semantic difference value β of ψ. As the random positive number γ hide the real semantic difference value β, we only need to prove the security of $\mathbf{x}$. In our scheme, $\mathbf{x}$ is protected by the random invertible matrix $\mathbf{A}$ and random vector $\mathbf{r}$. As $\mathbf{x} = \mathbf{A}\mathbf{y} - \mathbf{r}$, we can use $\mathbf{y} = \mathbf{A}^{-1}(\mathbf{x} + \mathbf{r})$ to indicate $\mathbf{y}$. Values of elements in $\mathbf{x}$ belong to $[0, 1]$. Besides, the vector $\mathbf{r}$ whose elements are uniformly sampled from a numerical interval $(0, 2^{\varepsilon}]$, where ε is security parameter. We prove the privacy of the $\mathbf{x}$ via that $\mathbf{x}_i + \mathbf{r}_i$ and $\mathbf{r}'_i$ are computationally indistinguishable for any $\mathcal{A}$, where $i \in [1, mn]$. From the view of $\mathcal{A}$, the best strategy is that guesses η from $0, 1$ with equal probability if $\mathbf{x}_i + \mathbf{r}_i \in (0, 2^{\varepsilon}]$, and output 1 if $\mathbf{x}_i + \mathbf{r}_i$ comes from a specific range $(2^{\varepsilon}, 2^{\varepsilon} + 1]$. Therefore, the success probability of the distinguisher $\mathcal{A}$ is:

$$\begin{aligned} Pr[\mathcal{A}(\mathbf{x_i} + \mathbf{r_i}) = 1] &= \frac{1}{2} Pr[0 < \mathbf{x_i} + \mathbf{r_i} \leq 2^{\varepsilon}] \\ &\quad + Pr[2^{\varepsilon} < \mathbf{x_i} + \mathbf{r_i} \leq 2^{\varepsilon} + 1] \\ &\leq \frac{1}{2} + \frac{1}{2^{\varepsilon}} \\ &= \frac{1}{2} + negl(\varepsilon) . \end{aligned}$$

Meanwhile, when the input is $\mathbf{r}'_i$, $\mathcal{A}$ obviously has the success probability which is:

$$Pr[\mathcal{A}(\mathbf{r}'_i) = 1] = \frac{1}{2} .$$

Therefore, we can deduce that:

$$|Pr[\mathcal{A}(\mathbf{x_i} + \mathbf{r_i}) = 1] - Pr[\mathcal{A}(\mathbf{r}'_i)] = 1| \leq ne(\varepsilon) .$$

In the end, we can apply union bound to deducing the conclusion that any probabilistic polynomial-time $\mathcal{A}$ distinguishes $\mathbf{x} + \mathbf{r}$ from $\mathbf{r}'$ with negligible success probability:

$$|Pr[\mathcal{A}(\mathbf{x} + \mathbf{r}) = 1] - Pr[\mathcal{A}(\mathbf{r}')] = 1| \leq neg(\varepsilon) = \sum_{i=1}^{mn} ne(\varepsilon) .$$

Second, we prove that the proposed technique guarantees the input privacy of ψ, namely, provides the security to the $\mathbf{c}$, $\mathbf{V}$, $\mathbf{W}$, $\mathbf{I}$ in the word transportation problem ψ. For example, the $\mathbf{W}$ is encrypted into $\mathbf{W}' = \mathbf{Q}(\mathbf{W} + \mathbf{V}\mathbf{r})$ in our scheme. We have $\mathbf{W}' = \mathbf{Q}\mathbf{V}(\mathbf{x} + \mathbf{r})$ according to $\mathbf{V}\mathbf{x} = \mathbf{W}$. We have proved any $\mathcal{A}$ distinguishes $\mathbf{x} + \mathbf{r}$ from $\mathbf{r}'$ with negligible success probability. Therefore, we can deduce that $\mathbf{Q}\mathbf{V}\mathbf{r}'$ and $\mathbf{Q}\mathbf{V}(\mathbf{x} + \mathbf{r})$ are statistically indistinguishable. Any $\mathcal{A}$ distinguishes $\mathbf{Q}\mathbf{V}\mathbf{r}'$ from $\mathbf{Q}\mathbf{V}(\mathbf{x} + \mathbf{r})$ with negligible success probability:

$$\begin{aligned} &|Pr[\mathcal{A}(\mathbf{Q}\mathbf{V}(\mathbf{x} + \mathbf{r})) = 1] - Pr[\mathcal{A}(\mathbf{Q}\mathbf{V}\mathbf{r}')] = 1| \\ &\leq neg(\cdot) = \sum_{i=1}^{mn} neg(\varepsilon) , \end{aligned}$$

where m and n are the number of keywords in a document and the query, respectively. The matrixes $\mathbf{c}$, $\mathbf{V}$ and $\mathbf{I}$ in the ψ is encrypted into $\mathbf{c}'$, $\mathbf{V}'$ and $\mathbf{I}'$ with random invertible matrixes. For example, the element values and the structure pattern and values of $\mathbf{c}$ are hidden effectively via multiplying the random matrix $\mathbf{A}$ and random positive number γ as proved [43].

We can deduce the proof λ is unable to reveal information about original WT problem ψ since $\mathcal{A}$ is unable to learn any sensitive information of $\psi = (\mathbf{c}, \mathbf{V}, \mathbf{W}, \mathbf{I})$ from $\omega = (\mathbf{c}', \mathbf{V}', \mathbf{W}', \mathbf{I}')$.

In a word, we can claim the proposed secure transformation technique is secure and our verifiable semantic searching

scheme is $\mathcal{L}$-confidential, since the output between $\mathrm{Real}_{\mathcal{A}}(\varepsilon)$ and $\mathrm{Ideal}_{\mathcal{A},\mathcal{S}}(\varepsilon)$ are computationally indistinguishable. ■

*3) **Further discussion for security**:*

Access Pattern & Search Pattern. Most of the symmetric searchable encryption schemes resort to the weakened security guarantee [11] revealing the access pattern and the search pattern. Informally speaking, the access pattern implies that the search results are used to derive some information, such as the identifiers of the documents returned for some specific trapdoors. The search pattern refers to that the cloud server can derive whether two arbitrary searches were performed for the same words. Formal definitions of these two patterns can be found in [11]. These information leakages could be hidden by using Oblivious RAMs [44], [45], but it will bring massive computation and communication burdens. Therefore, liking prior works, we do not consider these leakage issues in our scheme. Note that, the proposed secure transformation technique can encrypt a set of word transportation problems into two different sets of RLP problems by using one-time secret keys in two times of query, which reduce information leakage and increase the difficulty of linking the two identical queries. Furthermore, we also recognize that the order information of documents will be leaked unavoidably during the searching phase in ranked search schemes.

Collusion Attacks. We further discuss two types of collusion attacks launched by the dishonest cloud server and attackers.

The cloud server may collude with attackers who steal one user's secret key K which is used to decrypt the documents C. This attack exposes the entire database contents. From another view, we regard this attack as the cloud colluding with a malicious user. Therefore, multi-user schemes [46], [47], [48] supporting access control can resist this attack. The research focus of this paper is on secure verifiable semantic searching, which is orthogonal to the current research aiming for providing access control. The proposed scheme can also integrate with some popular techniques such as Attribute-based Encryption (ABE), to provide a fine-grained access control service. For example, we can learn from some schemes [47], [48] based on ABE, then design the data owner formulating access policy over an attribute set and encrypting the secret key K under the policy according to attribute public keys.

The cloud server may collude with attackers who maliciously delete or tamper with the C. This attack is similar to the attack in a security model in which the cloud server is malicious. Our scheme supports verification for the search results returned from the dishonest cloud server, but is unable to provide integrity verification for documents retrieved. The main challenge is that our flexible semantic searching scheme should not like other verifiable schemes in which the data owner predicts the fixed search results in advance. We left this problem to be addressed in future works.

VIII. EXPERIMENTS

In this section, we conduct empirical experiments to present the search accuracy and performance of the proposed scheme.

Table II
STATISTICS OF THE ROBUST04 AND CLUEWEB-09-CAT-B

	Robust04	Clueweb-09-Cat-B
Document Count	528,155	50,220,423
Document Mean Length	318	981
Query Count	250	150
Title Topic Mean Length	3	3
Desc Topic Mean Length	16	10

A. Experimental Settings

In this subsection, we present the experimental settings that include the experimental environment, datasets, evaluation measures, and baselines.

1) Experimental Environment: The overall experiments ran on the computer with the following parameters: Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz with 32 GB of RAM. We developed our scheme and other schemes with the Java.

2) Datasets: To evaluate the accuracy, we conducted experiments on two TREC collections, i.e., Robust04 and ClueWeb-09-Cat-B. The statistics of the collections are provided in Table II. Robust04 is a news dataset. ClueWeb-09-Cat-B is a large Web collection, which is filtered to the set of documents with spam scores in the 60^{th} percentile. The topics in both Robust04 and ClueWeb-09-Cat-B are chosen from TREC Tracks. Each topic contains different lengths of queries, namely, short-text title (Title Topic) and long-text description (Desc Topic). Here the Robust04-Title, Robust04-Desc, ClueWeb-Title, and ClueWeb-Desc mean that the title or description of the topics are used as query. The relevant judgment files are contained in both collections, presenting the relevance assessments among topics and documents, which is a benefit compared with other datasets used in other schemes.

3) Evaluation Measures: The precision and normalized discounted cumulative gain (NDCG) as evaluation measures used in our experiments. We evaluate the accuracy of schemes via comparing the top-k ranked documents using precision at rank 20 (P @20) and normalized discounted cumulative gain at rank 20 (NDCG @20).

The precision P @k is defined to measure the accuracy of a set of relevant documents from a given cutoff rank (top-k) retrieved documents, which is defined as follows:

$$P@k = \frac{|F_{relevant} \cap F_{retrieved}|}{|F_{retrieved}| = k}, \tag{13}$$

where $F_{retrieved}$ represents the top-k retrieved documents and $F_{relevant}$ represents the relevant documents of the query, $|F_{retrieved}|$ denotes the number of $F_{retrieved}$, $|F_{relevant} \cap F_{retrieved}|$ denotes the number of really relevant documents in $F_{retrieved}$.

The normalized discounted cumulative gain (NDCG) considers the ranking orders and relevance scores of retrieval results. The NDCG is accomplished by dividing the query's discounted cumulative gain (DCG) with the ideal DCG (IDCG). As a result of top-k retrieved documents, NDCG @k

Table III
ACCURACY COMPARISON OF DIFFERENT SECURE SEMANTIC SEARCHING SCHEMES

	Robust04				ClueWeb-09-Cat-B			
	Robust04-Title		Robust04-Desc		ClueWeb-Title		ClueWeb-Desc	
	P@20	NDCG@20	P@20	NDCG@20	P@20	NDCG@20	P@20	NDCG@20
SSERS [3]	0.052	0.058	0.027	0.029	0.049	0.046	0.057	0.070
SSSMIM-Single [6]	0.128	0.142	0.025	0.023	0.041	0.046	0.028	0.041
SSSMIM-Multi [6]	0.123	0.134	0.028	0.030	0.043	0.049	0.032	0.046
CKSER-1 [8]	0.051	0.117	0.032	0.086	0.049	0.077	0.026	0.081
CKSER-2 [8]	0.050	0.092	0.031	0.083	0.033	0.076	0.019	0.053
VKSS [2]	0.049	0.087	0.030	0.068	0.018	0.019	0.022	0.024
SSSW-1 [9]	0.107	0.192	0.070	0.128	0.060	0.086	0.027	0.060
SSSW-2 [9]	0.106	0.186	0.067	0.122	0.037	0.082	0.021	0.054
Ours	**0.148**	**0.271**	**0.136**	**0.255**	**0.061**	**0.103**	**0.041**	**0.102**

is computed as follows:

$$\begin{aligned} \mathrm{NDCG}\ @k &= \tfrac{DCG@k}{IDCG@k} \\ \mathrm{DCG}\ @k &= \textstyle\sum_{i=1}^{|real|} \tfrac{rel_i}{\log_2(i+1)} \\ \mathrm{IDCG}\ @k &= \textstyle\sum_{j=1}^{|ideal|} \tfrac{rel_j}{\log_2(j+1)} \end{aligned} \quad , \tag{14}$$

where $\mathrm{DCG}\ @k$ indicates the truth accumulated from the real ranking permutation at a particular rank position k, $\mathrm{IDCG}\ @k$ represents the ideal DCG at k. rel_i and rel_j denote relevance assessments between the query and documents, these relevance assessments can be got from relevant judgment file in our datasets. $|real|$ represents the top-k documents in the result of real ranking order for a query, $|ideal|$ represents the top-k documents in the result of ideal ranking order.

4) Baselines: Secure synonym extension ranked searching scheme (SSERS): SSERS is a semantic searching scheme extending the query words from synonym thesaurus. We implemented the SSERS proposed in [3].

Secure semantic searching based mutual information model (SSSMIM): SSSMIM is a ciphertext extension scheme, which extends the query words from the mutual information model. We implemented the single-keyword searching (SSSMIM-Single) proposed in [6]. We also extended it to a multi-keyword searching scheme (SSSMIM-Multi).

Central keyword semantic extension ranked searching (CKSER): CKSER is a secure semantic searching scheme based concept hierarchy. CKSER selects a central query word and extends it to get semantically related words from WordNet. We implemented CKSER-1 and CKSER-2 proposed in [8].

Verifiable Keyword-based Semantic Searching (VKSS): VKSS extends the query words according to WordNet and searches the words on a symbol-based trie index for supporting verifiability. We implemented the VKSS proposed in [2].

Secure searching scheme based on word2vec (SSSW): SSSW is a secure searching scheme that uses the word vectors trained on Word2vec. We implemented both SSSW-1 and SSSW-2 proposed in [9].

B. Performance Evaluation of Accuracy

In this subsection, we compare the proposed scheme with other secure semantic searching schemes over the two benchmark datasets and analyze the effectiveness of different word embeddings on our scheme.

We used title topics and description topics as the queries in our experiments. We did pre-process as follows: both documents and query words were white-space tokenized, lowercased, and removed the stopword. We adopted a re-ranking strategy for efficient computation. We used Indri to perform initial retrievals for obtaining the top 1, 000 ranked documents of each query. We applied schemes to re-rank these documents and used the results of re-ranked to evaluate accuracy.

1) Compared with baselines: The experiments results show that the accuracy of our proposed scheme is better than that of other schemes in terms of all the evaluation measures on both Robust04 and ClueWeb-09-Cat-B dataset, as illustrated in Table III. Taking the Robust04 dataset as an example, the relative improvement of our scheme over the second-highest ones in other schemes are about 15.62%, 41.14%, 94.28%, and 99.21% when using Robust04-Title and Robust04-Desc as queries under $\mathrm{P}\ @20$ and $\mathrm{NDCG}\ @20$. The results demonstrate the effectiveness of our secure verifiable semantic searching scheme based on word transportation optimal matching.

Overall, the SSERS scheme is inferior to other schemes except using description topics as queries search on the Clueweb-09-Cat-B. The accuracy of SSSMIM schemes when using title topics as queries on both datasets is higher than the case when using description topics. As for schemes based on concept hierarchy, CKSER-1, CKSER-2, and VKSS schemes usually are not competitive to other schemes. In particular, VKSS is inferior to CKSER-1 and CKSER-2. The reason is that CKSER-1 and CKSER-2 schemes consider the grammatical relationship among query keywords and introduce a word weighting algorithm to show the importance of the distinction among them. From Table IV, we can see that CKSER-2 is inferior to CKSER-1 under different evaluation measures on both datasets, which also be observed between SSSW-1 and SSSW-2. This finding is not surprising since the CKSER-2 and SSSW-2 adopt enhanced Secknn algorithm introducing more random numbers, leading to limited accuracy. We take a look at the SSSW-1 and SSSW-2 schemes using word embeddings and the Secknn algorithm to build secure indexes. Overall, we can see that the SSSW schemes obtain very limited improvement compared with query expansion schemes. The results demonstrate that using a compact vector representation

Table IV
ACCURACY COMPARISON OF OUR SCHEME USING DIFFERENT WORD EMBEDDINDS OVER ROBUST04

	Robust04-Title		Robust04-Desc	
	P@20	NDCG@20	P@20	NDCG@20
Ours-GloVe100	0.137	0.239	0.126	0.244
Ours-GloVe200	0.146	0.259	0.127	0.254
Ours-GloVe300	0.147	0.264	0.128	0.252
Ours-Word2vec300	0.145	0.254	0.122	0.176
Ours-Fasttext300	0.148	0.271	0.136	0.255

of documents/queries may damage the semantic information of word embeddings.

We can see that the accuracy of other schemes using the title topics as queries is usually larger than that using the description topics across different datasets, which is consistent with the previous findings in plaintext information retrieval [49]. A reason is that the description topics are usually one or two sentences containing about 16 words so that these semantic searching schemes are challenging to analyze the semantic relationship among the query words. For example, the SSSMIM-Single, CKSER-1 and CKSER-2 schemes are facing the challenge to select an accurate central keyword from a long-text description topic. The accuracy of the proposed scheme using the description topics is still higher than that of other schemes. A reason is that the word transportation optimal matching is beneficial to analyze the semantic relationship between the words and the importance of the distinction among words in long-text queries.

2) Effect of Word embeddings: As word embedding is an essential component in our scheme, we used two groups of word embeddings to conduct experiments for studying the effect of word embeddings on our scheme. Table IV lists the experimental results of our scheme on Robust04 dataset. In the first experiment, we used different word embeddings with 100, 200, and 300 dimensions trained by GloVe over same corpuses, namely, GloVe100, GloVe200, GloVe300. From Table IV, we can see that the proposed scheme using word embedding with 300 dimensions usually obtains the best accuracy except under $\mathrm{NDCG}\,@20$ using the description topics. The result of the second experiment demonstrates that the higher dimensionality may help our scheme capture the accurate semantic information. In the second experiment, we used two types of word embeddings with 300 dimensions trained by Word2vec and Fasttext over same corpus, namely, Word2vec300 and Fasttext300. We can see that our scheme gets higher accuracy when using the word embeddings trained by Fasttext compared with using Word2vec300.

C. Performance Evaluation of Time Cost

In this subsection, we present the performance evaluation of the proposed secure verifiable semantic searching scheme.

We report the experimental results of our scheme over the Robust04 dataset using title topics as queries due to limited space. The performance evaluation of time cost at the owner, the users, and the cloud server in our scheme is as follows:

The data owner is the initiator who initializes the secure searching scheme. Not like in other schemes, the data owner in our scheme does not need to perform massive cryptographic operations, such as order-preserving encryption and homomorphic encryption. For the owner, the main steps including (1) generating symmetrical encryption secret key with t_{O_Key}; (2) encrypting documents with t_{O_Enc}; (3) building forward indexes with t_{O_Index}. We define that the total time of a user is $t_{Owner} = t_{O\_Key} + t_{O\_Enc} + t_{O\_Index}$. From Table V, we can see that it takes the data owner $t_{Owner} = 6.289s$ to initialize the proposed secure verifiable semantic searching scheme.

The data users are the searching requesters that send the trapdoor of a query for acquiring top-k related documents. In our scheme, data users need to perform the main steps including (1) building word transportation problems with t_{U_WTP}; (2) generating the one-time secret key with t_{U_Key}; (3) transforming the WT problems with t_{U_Tran}. We define that the total time of a user is $t_{User} = t_{U\_WTP} + t_{U\_Key} + t_{U\_Tran}$. From Table V, we can see that the total time of generating the trapdoor in our scheme is $t_{U\_Trapdoor} = t_{User} = 1.106s$. It is acceptable in many real-world scenarios in which the users want to search for essential materials with high accuracy. In addition, once receiving the proofs and search results, the users need to spend $t_{U\_Verify} = 0.033s$ using the proposed verification mechanism to check the proofs and spend $t_{U\_Dec} = 0.005s$ decrypting the top-k related documents.

The cloud server is an intermediate service provider that performs the retrieval process. In our scheme, the cloud needs to perform the main steps including (1) performing optimal matching on ciphertext and generating proofs in the matching process with t_{C_Match}; (2) calculating and ranking the measurements with t_{C_Rank}. We define that $t_{Cloud} = t_{C\_Match} + t_{C\_Rank}$ denotes the total time of the cloud. From Table V, the cloud in our scheme needs $t_{Cloud} = 14.036s$ to perform the retrieval process. At first glance, such time may seem a little long for ordinary users. However, It is worth spending more time to get higher search accuracy for the applications in practical scenarios, such as secure searching for medical and financial information. In addition, the cloud server providers usually have enormous computing resources to reduce retrieval time in the real world.

According to the experimental results, we further demonstrate the Rationality Analysis in section VII. We define that $t_{Outsource} = t_{U\_Key} + t_{U\_Tran}$ denotes the time-consuming for outsourcing the retrieval tasks to the cloud,$\rho = \frac{t_{C\_Match}}{t_{Outsource}}$ denotes the savings of the computational costs when the user outsources retrieval tasks to the cloud. We obtain $\rho = 93.546$ from the experimental results in Table V. It demonstrates the rationality that data users want to outsource the retrieval task to the cloud server. We define $\mu = \frac{100 t_{C\_Match}}{t_{C\_Rank}}\%$ to denote the possibility of the cloud honestly performs the ranking task. We can obtain $\mu = 350800\%$ from the experimental results in Table V. It demonstrates a rational cloud performs the ranking task honestly if he/she honestly solved the RLP problems.

D. Performance Comparisons with VKSS

In this subsection, we present performance comparisons between the proposed scheme and the VKSS scheme.

Table V
TIME COST STATISTICS OF OUR SCHEME

	t_{O_Key}	t_{O_Enc}	t_{O_Index}	t_{U_WTP}	t_{U_Key}	t_{U_Tran}	t_{U_Verify}	t_{U_Dec}	t_{C_Match}	t_{C_Rank}
Ours	0.001s	0.335s	5.953s	0.956s	0.008s	0.142s	0.033s	0.005s	14.032s	0.004s

Table VI
PERFORMANCE COMPARISONS BETWEEN OUR SCHEME AND VKSS

Time Cost	t_{O_Pre}	$t_{O\_BuildIndex}$	$t_{U\_WordTree}$	$t_{U\_Trapdoor}$	t_{U_Verify}	t_{U_Rank}	t_{C_Match}	t_{C_Rank}
VKSS	4.879s	17.778s	614.312s	0.034s	0.011s	0.035s	1.416ms	\
Ours	**4.879s**	**1.074s**	\	**1.106s**	**0.033s**	\	**14.032s**	**0.004s**

Accuracy	Robust04-Title		Robust04-Desc		ClueWeb-Title		ClueWeb-Desc	
	P@20	NDCG@20	P@20	NDCG@20	P@20	NDCG@20	P@20	NDCG@20
VKSS	0.049	0.087	0.030	0.068	0.018	0.019	0.022	0.024
Ours	**0.148**	**0.271**	**0.136**	**0.255**	**0.061**	**0.103**	**0.041**	**0.102**
Growth Rate	202%	211%	353%	275%	238%	442%	86%	325%

We conducted extensive experiments to evaluate the performance of the time cost and accuracy between our scheme and the VKSS scheme which is the only one supports secure verifiable semantic searching in prior works found. We report the results of time cost experiments over the Robust04 dataset using title topics as queries due to limited space. We do not consider the time cost of encrypting documents and decrypting documents since the time cost is the same in both schemes. Table VI lists the results of comparison experiments.

From the performance evaluation of the time cost, we can see that the owner and users in our scheme can ease the computational burden by paying the time cost in cloud compared with the VKSS. Specifically, the owner in our scheme spends $t_{O\_Pre} = 4.879s$ and $t_{O\_BuildIndex} = 1.074s$ preprocessing documents and building the forward indexes when generating the indexes for the documents, where $t_{O\_Pre} + t_{O\_BuildIndex} = t_{O\_Index}$. However, the owner in VKSS spends much time $t_{O\_BuildIndex} = 17.778s$ on building the trie index since the owner needs to forecast the results from predefined document keywords for verifiable searching. Moreover, the users in VKSS take less time than our scheme to generate a trapdoor with $t_{U\_Trapdoor}$, but the users need to spend $614.312s$ for generating the word similarity tree to expand query words from document keywords as the query. Although part of the work constructing the word similarity tree can be performed in an off-line way, it is unfriendly to the resource-constrained users. In other words, the owner and users in VKSS need to perform burdensome tasks to realize secure verifiable semantic searching. While in our scheme, the owner does not need to forecast the results since the verifiable searching can be realized by using the intermediate data produced in the optimal matching process; and the users build the encrypted word transportation problems and outsource them to cloud for secure semantic searching rather than performing query expansion on plaintext. Moreover, after receiving the proofs and search results, the users in VKSS spend less time on verifying the correctness of results than our scheme but need to spend additional $t_{U\_Rank} = 0.035s$ on calculating the relevance scores between the query and documents and ranking the scores for getting the top-k related documents. The time cost in the cloud in VKSS is small, which is benefit from the efficiency of the trie index. Another reason is that VKSS is unable to support verifying the ranked results return from the cloud. To obtain high accuracy in our scheme, the optimal matching on ciphertext is performed and the documents are ranked in the cloud.

As for performance evaluation of accuracy, the accuracy of our scheme is much higher than that of VKSS on both Robust04 and ClueWeb-09-Cat-B dataset. Taking the Robust04 dataset as an example, the relative improvements of our scheme over the VKSS scheme under $\mathrm{NDCG}@20$ are about 211% and 275% using title and description topics, respectively. Overall, our scheme spends more time on performing match in the cloud to obtain much more accuracy improvement than VKSS. It is worth spending more time to get higher search accuracy for the applications in practical scenarios. Take the medical profile as an example, a correct personal medical profile of a patient is essential and useful to help the doctor make a precise disease diagnosis and health evaluation.

In summary, the proposed scheme outsources the complex computational of performing proof generation task and semantic matching task to the cloud. Therefore, our scheme is more in line with the outsourcing computation characteristics of the cloud computing paradigm. The main reason why our scheme spends more time in the cloud is that the computation of optimal matching on ciphertext is related to the size of linear programming problem. Therefore, in the future, we plan to design other schemes to reduce the time cost of optimal matching on ciphertext according to this finding.

IX. 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 transform 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.

ACKNOWLEDGMENT

We thank Shaocong Wu for valuable discussions and his help in experiments. This work was supported in part by the Shenzhen Municipal Development and Reform Commission (Disciplinary Development Program for Data Science and Intelligent Computing), in part by Shenzhen International cooperative research projects GJHZ20170313150021171, and in part by NSFC-Shenzhen Robot Jointed Founding (U1613215).

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.

[19] S. K. Kermanshahi, J. K. Liu, R. Steinfeld, and S. Nepal, "Generic multi-keyword ranked search on encrypted cloud data," in *Proc. Springer Eur. Symp. Res. Comput. Secur.*, 2019, pp. 322–343.

[20] L. F. Lai, C. C. Wu, P. Y. Lin, and L. T. Huang, "Developing a fuzzy search engine based on fuzzy ontology and semantic search," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2011, pp. 2684–2689.

[21] A. Imani, A. Vakili, A. Montazer, and A. Shakery, "Deep neural networks for query expansion using word embeddings," in *Proc. Springer Eur. Conf. Inf. Retrieval*, 2019, pp. 203–210.

[22] Y. Long, L. Liu, Y. Shen, and L. Shao, "Towards affordable semantic searching: Zero-shot retrieval via dominant attributes," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7210–7217.

[23] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in *Proc. IEEE Int. Conf. Commun.*, 2012, pp. 917–922.

[24] J. Zhu, Q. Li, C. Wang, X. L. Yuan, Q. Wang, and K. Ren, "Enabling generic, verifiable, and secure data search in cloud services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 8, pp. 1721–1735, 2018.

[25] C. Wang, N. Cao, K. Ren, and W. J. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 8, pp. 1467–1479, 2011.

[26] Q. Liu, X. Nie, X. Liu, T. Peng, and J. Wu, "Verifiable ranked search over dynamic encrypted data in cloud computing," in *Proc. IEEE/ACM Int. Symp. Qual. Serv.*, 2017, pp. 1–6.

[27] W. H. Sun, B. Wang, N. Cao, M. Li, W. J. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in *Proc. ACM SIGSAC Symp. Inf. Comput. Commun. Secur.*, 2013, pp. 71–82.

[28] W. Zhang, Y. P. Lin, and G. Qi, "Catch you if you misbehave: Ranked keyword search results verification in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 74–86, 2015.

[29] K. Kurosawa and Y. Ohtaki, "UC-secure searchable symmetric encryption," in *Proc. Int. Conf. Financial Cryptography Data Secur.* Springer, 2012, pp. 285–298.

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Represent.*, 2013, pp. 1532–1543.

[31] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, 2014, pp. 1532–1543.

[32] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Ling.*, vol. 5, pp. 135–146, 2017.

[33] M. Cornia, L. Baraldi, H. R. Tavakoli, and R. Cucchiara, "Towards cycle-consistent models for text and image retrieval," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 687–691.

[34] Y. Rubner, L. J. Guibas, and C. Tomasi, "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval," in *Proc. ARPA Workshop*, vol. 661, 1997, pp. 668–675.

[35] Z. H. Xia, Y. Zhu, X. M. Sun, Z. Qin, and K. Ren, "Towards privacy-preserving content-based image retrieval in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 276–286, 2018.

[36] C. Wang, Z. Liu, and S. C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 29–39, 2014.

[37] I. Goienetxea, J. M. Martínez-Otzeta, B. Sierra, and I. Mendialdua, "Towards the use of similarity distances to music genre classification: A comparative study," *PloS One*, vol. 13, no. 2, p. e0191417, 2018.

[38] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 957–966.
[39] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "Semantic matching by non-linear word transportation for information retrieval," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 701–710.
[40] S. Nabavi and A. H. Beck, "Earth mover's distance for differential analysis of heterogeneous genomics data," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2015, pp. 963–966.
[41] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2011, pp. 820–828.
[42] D. G. Luenberger and Y. Y. Ye, *Linear and Nonlinear Programming*. Reading, MA: Springer International Publishing, 2016.
[43] A. Li, W. Du, and Q. Li, "Privacy-preserving outsourcing of large-scale nonlinear programming to the cloud," in *Proc. Springer Int. Conf. Secur. Privacy Commun. Syst.*, 2018, pp. 569–587.
[44] X. S. Wang, K. Nayak, C. Liu, T. H. Chan, E. Shi, E. Stefanov, and Y. Huang, "Oblivious data structures," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 215–226.
[45] D. S. Roche, A. Aviv, and S. G. Choi, "A practical oblivious map data structure with secure deletion and history independence," in *Proc. IEEE Symp. Secur. Privacy*. IEEE, 2016, pp. 178–197.
[46] S. K. Kermanshahi, J. K. Liu, R. Steinfeld, S. Nepal, S. Lai, R. Loh, and C. Zuo, "Multi-client cloud-based symmetric searchable encryption," *IEEE Trans. Dependable Secure Comput.*, 2019.
[47] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.
[48] Y. Xue, K. Xue, N. Gai, J. Hong, D. S. Wei, and P. Hong, "An attribute-based controlled collaborative access control scheme for public cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 11, pp. 2927–2942, 2019.
[49] J. Guo, Y. X. Fan, Q. Y. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 55–64.

Wenyuan Yang is pursuing his P.h.D degree in School of Electronics Engineering and Computer Science, Peking University. He received his B.S. degree in Software Engineering from University of Electronic Science and Technology of China in 2016. His research interests include Searchable Encryption and Trusted Computing.

Yuesheng Zhu received his B.Eng. degree in radio engineering, M. Eng. degree in circuits and systems and Ph.D. degree in electronics engineering in 1982, 1989 and 1996, respectively. He is currently working as a professor at the Lab of Communication and Information Security, Shenzhen Graduate School, Peking University. He is a senior member of IEEE, fellow of China Institute of Electronics, and senior member of China Institute of Communications. His interests include digital signal processing, multimedia technology, information security.