Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6617959, 12 pages
https://doi.org/10.1155/2021/6617959

Research Article

Privacy-Guarding Optimal Route Finding with Support for Semantic Search on Encrypted Graph in Cloud Computing Scenario

Bin Wu,[1] **Xianyi Chen**,[2] **Zongda Wu**,[3] **Zhiqiang Zhao**,[1] **Zhuolin Mei**,[1] **and Caicai Zhang**[1]

[1]*School of Computer and Big Data Science, Jiujiang University, Jiujiang, Jiangxi 332005, China*
[2]*School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, Jiangsu 210044, China*
[3]*School of Mathematical Information, Shaoxing University, Shaoxing, Zhejiang 312000, China*

Correspondence should be addressed to Bin Wu; wubcst@163.com

Received 26 November 2020; Revised 22 December 2020; Accepted 4 March 2021; Published 17 March 2021

Academic Editor: Jun Cai

The arrival of cloud computing age makes data outsourcing an important and convenient application. More and more individuals and organizations outsource large amounts of graph data to the cloud computing platform (CCP) for the sake of saving cost. As the server on CCP is not completely honest and trustworthy, the outsourcing graph data are usually encrypted before they are sent to CCP. The optimal route finding on graph data is a popular operation which is frequently used in many fields. The optimal route finding with support for semantic search has stronger query capabilities, and a consumer can use similar words of graph vertices as query terms to implement optimal route finding. Due to encrypting the outsourcing graph data before they are sent to CCP, it is not easy for data customers to manipulate and further use the encrypted graph data. In this paper, we present a solution to execute privacy-guarding optimal route finding with support for semantic search on the encrypted graph in the cloud computing scenario (PORF). We designed a scheme by building secure query index to implement optimal route finding with support for semantic search based on searchable encryption idea and stemmer mechanism. We give formal security analysis for our scheme. We also analyze the efficiency of our scheme through the experimental evaluation.

1. Introduction

With the rapid progress of electronic devices and communication technologies, it promotes the advent of the era of cloud computing that has an important impact and value on all walks of life [1, 2]. Cloud computing also speeds up data outsourcing service and makes it an important and convenient application [3, 4]. Graph is a structure that is often used in various fields, such as traffic graph [5], social graph [6], and molecular structure graph [7]. Due to the powerful processing capability of cloud computing and the problem of cost saving, the enormous graph data are usually outsourced to the cloud computing platform (CCP) which is responsible for storing, managing, and processing these data. But the server on CCP is not completely honest and trustworthy, the privacy and security issues of the outsourcing graph data need to be considered and handled. Encrypting the outsourcing graph data is an effective and commonly used method before they are outsourced to CCP [8]. However, it is not easy for data customers to manipulate and further use the encrypted outsourcing graph data. Therefore, it is an extremely meaningful work to implement privacy-guarding optimal route finding with support for semantic search on the encrypted graph in the cloud computing scenario.

The optimal route finding on graph is an operation that is frequently used in many fields, and related applications include shortest path query [9], path planning [10], and minimum spanning tree [11]. The optimal route finding with support for semantic search has stronger query capabilities, and a customer can use similar words of graph vertices as query terms to implement optimal route finding. For instance, in a traffic network graph, the graph vertices

represent locations, and the weight on the edge represents the route cost between two locations. The optimal path finding is to query for the path with the least cost between two locations. To realize semantic search, the porter stemmer mechanism commonly used in information retrieval [12] is adopted in this paper. The graph vertex set is transformed into a new set by the stemmer mechanism, and the new set is similar to the original set of vertices in semantics. When performing optimal route finding, the new set serves as the source set of query terms. Our work researches optimal route finding over encrypted graph data, and we also take into account cost savings; it is expensive in cost to download all the graph data from the remote server. In view of this, it is of great significance to implement optimal route finding with support for semantic search on encrypted graph. However, it is not an easy job to carry out the optimal route finding in consideration of the security and privacy issues in the cloud computing scenario.

To implement query operations on the remote server in the cloud computing scenario, the idea of searchable encryption is very effective [13–17]. The remote server performs query through encrypted query terms, and the server cannot obtain the privacy information of the query terms and query results. The searchable encryption is a research hotspot in the area of information security, and the research progress in this field has also been advancing. Soon after, some dynamic and extended searchable encryption schemes have emerged [18–22], but the above searchable encryption schemes cannot be used to implement route finding with support for semantic search on encrypted graph. Recently, some researchers have studied and implemented some query schemes on the encrypted graph [23–27]. Chase et al. studied the query problem on the encrypted graph and proposed the structured encryption method [23]. Privacy preserving subgraph query problems were researched in the literatures [24, 25]. Shen et al. studied the problem of cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection [26]. Ciucanu et al. presented a secure framework for graph outsourcing and SPARQL evaluation in the literature [27]. However, these methods cannot address the problem of optimal route finding with support for semantic search on encrypted graph.

To solve the problem on CCP, we present a solution to execute privacy-guarding optimal route finding with support for semantic search on the encrypted graph in the cloud computing scenario (PORF). In the PORF scheme, the server on CCP implements the privacy-guarding optimal route finding with the help of the index we build. Firstly, we use porter stemmer mechanism to transform the graph vertices into a new set to achieve semantic search. Then, based on the new set, we build the chain tables, and each node of which contains optimal route information. Finally, we build a secure index on the basis of all the chain tables, and the index is stored on the server of CCP, and the server executes optimal route finding by the index and the encrypted query terms. The server cannot learn the privacy contents of the query results and query terms. The security analysis and the experimental evaluation show that our proposed PORF scheme is secure and efficient.

The contributions of our paper are described below.

(1) We present a scheme to address the problem of optimal route finding with support for semantic search on the encrypted graph in the cloud computing scenario

(2) We give the formal security analysis of the scheme to ensure the privacy and security of query results and query terms

(3) We conduct an experimental analysis to demonstrate the efficiency of our scheme

The rest of our paper is organized below. Section 2 presents the related work. Section 3 gives the design and analysis process of our PORF scheme. Section 4 analyzes the security of the PORF scheme. Section 5 demonstrates the PORF scheme through experiment and comparison. Finally, section 6 summarizes our paper.

2. Related Work

With the rapid development of computer technology and the huge increase of people's demands [28–31], privacy and security issues have increasingly become an important consideration [32–35]. In the field of data outsourcing security, searchable encryption plays an important role and can realize the query of outsourcing data without disclosing privacy information [36, 37]. Generally speaking, there are two kinds of searchable encryption types: searchable symmetric encryption and searchable asymmetric encryption [16, 17]. Usually, the querying of symmetric encryption is more efficient than that of asymmetrical encryption. Thus, we use symmetric encryption idea in our scheme.

Searchable encryption becomes an efficient cryptographic primitive in remote data query which plays an important value in data outsourcing [13–16]. The concept and idea of searchable symmetric encryption was first presented in the literature [13]. Goh came up with the secure index strategy in which the bloom filter was used to address the search question on outsourced data in the literature [14]. Curtmola et al. put forward the conception of nonadaptive searchable symmetric encryption and adaptive searchable symmetric encryption in the literature [16]. Chang et al. adopted pseudorandom functions to solve the problem of privacy preserving keyword searches on remote encrypted data and solved the update problem [17]. Thereafter, some researchers in the field of security proposed a lot of outspread searchable encryption solutions [18–22]. Wang et al. investigate the problem of secure and efficient similarity search over outsourced cloud data and proposed a new symbol-based trietraverse searching mechanism [18]. The questions of the secure and efficient ranked search over encrypted outsourcing data were studied in the literatures [19, 20]. Du et al. presented a dynamic multiclient searchable symmetric encryption scheme supporting Boolean queries which allowed the data owner to authorize multiple users to excute Boolean queries over the encrypted data [21]. Li et al. used the k-nearest neighbor and attribute-based encryption

Table 1: Summary of notations.

Notations	Denotations
G	The outsourcing graph
$\mathcal{I}$	The optimal route finding index
n	The number of vertices of graph G
V	The vertices set of graph G, $V = \{v_1, \cdots, v_n\}$
m	The number of optimal routes of graph G
$x\Delta y$	The words x and y are concatenated as a whole
c	The maximum number of optimal routes about querying vertices in graph G
$\mathcal{Q}_i$	The encrypted query term, where $1 \le i \le m$
$R_{\mathcal{Q}_i}$	The set of optimal route finding results of query term $\mathcal{Q}_{i,j}$
$\mathrm{Enc}_{\mathrm{key}}(\cdot)$	The symmetric encryption algorithm
$\mathrm{Dec}_{\mathrm{key}}(\cdot)$	The symmetric decryption algorithm

techniques to present a dynamic searchable symmetric encryption scheme and addressed the key sharing problem [22], but all the searchable encryption solutions cannot be used to perform optimal route finding with support for semantic search over encrypted graph data.

In recent years, the secure query questions on encrypted graph were studied, and some relative research achievements have been acquired [23–27]. Chase et al. presented the idea of structured encryption and proposed the application of controlled disclosure in the literature [23]. Cao et al. adopted the "filtering-and-verification" principle to study and address the problem of subgraph query on the encrypted graph [24]. Fan et al. proposed a private subgraph query solution for large graphs, and the query subgraph needed to be protected while the data graph did not [25]. Shen et al. proposed a graph encryption scheme to achieve the constrained shortest distance query and presented a tree-based ciphertext comparison protocol [26]. Ciucanu et al. designed and implemented a secure framework to perform the quey with SPARQL evaluation on outsourcing graphs in the literature [27]. However, all the existing solutions of encrytped graph search have not addressed the issue of privacy-guarding optimal route finding with support for semantic search on encrypted graph.

In this paper, we come up with a solution on the strength of searchable encryption idea and porter stemmer mechanism to perform optimal route finding with support for semantic search. We first transform graph vertices into new word set and build the chain tables based on the new set. We next build an index which is sent to the server on CCP, and the server excutes optimal route finding through the index and the encrypted query terms. We finally analyze and evaluate our solution both from security and experiment.

3. PORF Scheme Construction

3.1. Preliminaries. Goldwasser et al. presented the concepts of semantic security and indistinguishability in the literature [38]. A system is semantically secure if whatever an adversary can compute about the plaintext given the ciphertext, he can also compute without the ciphertext [38]. In this paper, we use the set $(K\mathrm{ge}, \mathrm{Enc}, \mathrm{Dec})$ containing three polynomial-size algorithms to represent a semantically secure encryption mechanism [39]. Kge is a secret key generating algorithm. Enc and Dec represent the encryption algorithm and decryption algorithm, respectively.

To implement semantic search in our PORF scheme, we adopt the porter stemmer mechanism in information retrieval [12], but not limited to only this method can achieve. We use $V = \{v_1, \cdots, v_n\}$ to represent the vertices set of outsourcing graph, and the new word set after transformation through porter stemmer mechanism is represented as $A = \{a_1, \cdots, a_u\}$, where $|A| \le |V|$. The main notations used in our paper follow in Table 1.

3.2. PORF Overview. In the cloud computing scenario, the architecture about outsourcing query illustrated in Figure 1 is mainly composed of three entities: the server on CCP, the data owner, and data customers. The server provide data owners and customers with storage, management, and query services. To enable the server to implement optimal route finding on the encrypted outsourcing graph, we construct an index and encrypt the query request and then put them on the server. In this work, our main task is to study and achieve optimal route finding with support for semantic search. With respect to the query control and authentication of data customers, we adopt the thought of preexisting searchable encryption such as broadcast encryption [16].

Our PORF scheme can perform optimal route finding with support for semantic search over the encrypted graph on CCP, and the following design targets can be achieved.

(1) Privacy-guarding optimal route finding functionality. The customer can achieve privacy-guarding optimal route finding with support for semantic search by means of the server on CCP

(2) Security guarantee. We can give the security guarantee for our scheme through formal analysis, and it

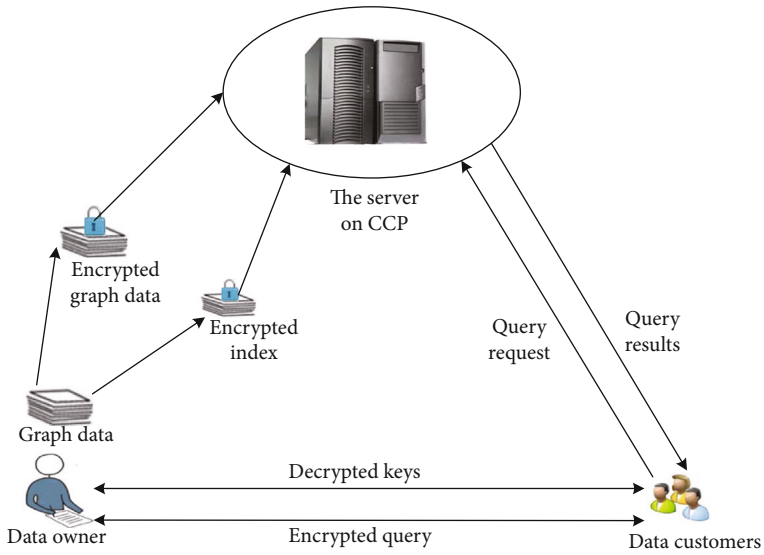

FIGURE 1: Architecture of optimal route finding on the encrypted graph.

prevents the server from getting the privacy of the query results and query terms

(3) Efficiency. We can implement our scheme of optimal route finding with less overhead

In the PORF scheme design, we make use of index and chain table on the part of data structures. Our work considers the undirected graph as the research object, and the processing operation of directed graph is similar. We adopt the chain tables to store the optimal route information. Query vertices need to be processed and encrypted before being sent to the server, and the index is then built to complete privacy-guarding optimal route finding via the server on CCP.

To implement the optimal route finding in the cloud computing scenario, we try to deal with it in three steps. Firstly, we build the chain tables, and each node of the chain tables consists of the vertices and optimal route information which need to be encrypted. Secondly, we are going to build a secure index to randomly store all the nodes of the chain tables. Finally, the query vertices of a customer are delivered to the server after security processing, and the server executes optimal route finding with support for semantic search on CCP with the help of the built index. Consistent with the thought of existing symmetric searchable encryption schemes [16, 18, 19], we assume that the server on CCP will adopt the adaptive attack model, and the query customers have the mutual request authentication and query control mechanisms with the data owner [16].

3.3. Scheme Design and Implementation. In the PORF scheme, our main work is to build the index and how to implement optimal route finding by means of the index on CCP. The several used algorithms in our scheme are described below.

(i) *Genkeys* (1^{ℓ}): symmetric secret key generation algorithm. The parameter ℓ is taken as the input, and the symmetric secret key k is used as the output

(ii) *Chaintablebuilding* $(G, \mathcal{K})$: building the chain tables of graph vertices to store the contents of optimal route. The graph data G and the symmetric secret key set $\mathcal{K}$ serve as inputs, and the outputs are the work set A with similar meanings of graph vertices, the set W of compound terms about work set, and the chain table set $\mathcal{F}$

(iii) *Indexbuilding* $(W, \mathcal{F}, \mathcal{K}, \mathcal{K}')$: building query index algorithm. The inputs are the set W, chain table set $\mathcal{F}$, and the key sets $\mathcal{K}$ and $\mathcal{K}'$, and the output is the query index $\mathcal{I}$

(iv) *Querybuilding* $(w_i, \mathcal{K})$: building query term algorithm. The inputs include the word w_i from the set W and the key set $\mathcal{K}$. The encrypted query term set $\mathcal{Q}_i$ serves as the output

(v) *Queryperforming* $(\mathcal{I}, \mathcal{Q}_i)$: implementing optimal route finding on CCP. The index $\mathcal{I}$ and the query term set $\mathcal{Q}_i$ serve as the inputs, and the set of optimal route is the output

In our work, we use ℓ to be the security parameter and adopt $(\mathrm{Kge}, \mathrm{Enc}, \mathrm{Dec})$ as a secure symmetric encryption scheme in our optimal route finding solution. The building process of our proposed PORF scheme is as follows.

3.3.1. Building Chain Table. To implement the semantic search, we adopt the porter stemmer mechanism to turn graph vertices set V into a new set $A = \{a_1, \cdots, a_u\}$, where $u \leq n$. We combine two arbitrary inequable words in the set A to form a new compound term, and new term set is denoted as $W = \{w_1, w_2, \cdots, w_m\}$. For every member w_i of the set W $(0 \leq i \leq m)$, we build its chain table $\mathcal{F}_i$, and we use elem to represent the node of the chain table. The creation process of chain tables is described in Algorithm 1. The content of each node in the chain table $\mathcal{F}_i$ consists of optimal route information, and we write it in terms of the

Input: The graph data G, and the symmetric key set $\mathcal{K}$.
Output: The chain table set $\mathcal{F}$
1: The set of graph vertices is $V = \{v_1, v_2, \cdots, v_n\}$, and the symmetric key set is $\mathcal{K} = \{k_1, k_2, \cdots, k_m\}$
2: **for all** $i \in [1, n]$ **do**
3: Each member v_i in the set V is turned into a new word after conversion via porter stemmer mechanism, and the new word set is represented as $A = \{a_1, \cdots, a_u\}$, where $u \leq n$
4: **end for**
5: **for all** $i, j \in [1, u]$ **do**
6: Two arbitrary inequable words a_i and a_j in the set A are combined into a new compound term, and all the new terms set is denoted as $W = \{w_1, w_2, \cdots, w_m\}$; and $|w_i|$ $(1 \leq i \leq m)$ represents the number of optimal route about the compound term w_i
7: **end for**
8: **for all** $i \in [1, m]$ **do**
9: **for all** $j \in [1, |w_i|]$ **do**
10: $w_{i,j} = Enc_{k_i}(optrvalue)$;
11: $\mathcal{F}_i \rightarrow elem_j = w_{i,j}$;
12: **end for**
13: **end for**
14: $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \cdots \mathcal{F}_m\}$
15: **return** $\mathcal{F}$.

Algorithm 1: *Chaintablebuilding*.

symbol optrvalue. To protect the security of the chain table contents, we need to encrypt the chain table nodes. The two used symmetric key sets in our scheme are represented as $\mathcal{K} = \{k_1, k_2, \cdots, k_m\}$ and $\mathcal{K}' = \{k_1', k_2', \cdots, k_m'\}$. The set of all chain tables that are built related to the set W is denoted as $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \cdots \mathcal{F}_m\}$.

In the algorithm *Chaintablebuilding*, the time complexity of new word conversion via porter stemmer is $O(n)$, and the time complexity of building set W is $O(n)$. The time complexity of building every chain table $\mathcal{F}_i$ is $O(\max(|\mathcal{F}_i|, c) = O(c)$ (c is the maximum number of optimal routes of compound terms), and the time complexity of building m chain tables is $O(m \cdot c)$. Therefore, the total time complexity of the algorithm *Chaintablebuilding* is $O(n)+O(m \cdot c)$.

3.3.2. Building Optimal Route Finding Index. To enable the server to perform optimal route finding in the cloud computing scenario, we propose to implement this by building an index. The process of creating our index is described in Algorithm 2. For each chain table $\mathcal{F}_i$ $(0 \leq i \leq m)$, the contents of each node contain optimal route information, and the number of nodes (that is, the number of optimal routes about the compound term $|w_i|$) is denoted as $|\mathcal{F}_i|$. For $1 \leq j \leq |\mathcal{F}_i|$, we generate a tag for the term w_i by concatenating w_i and j, and the tag is represented as $w_i \Delta j$. Then, all the tags about the term w_i are denoted as a set $\gamma_{w_i} = \{w_i \Delta 1, \cdots, w_i \Delta |\mathcal{F}_i|\}$. The matching optimal route informaton of each member in the set γ_{w_i} is placed in the index. Performing optimal route finding of the term w_i is equivalent to seeking for the corresponding members in the index $\mathcal{I}$ via all the correlative tags in the set γ_{w_i}. To prevent the server on CCP from getting information about the number of optimal routes of each member w_i, we need to add the extra elements (We call them the disturbing values) to pad the index such that the number of optimal routes of each member in the set W is the same; that is, it is the maximum number of optimal routes c in the graph G. If $|\mathcal{F}_i| < c$, we need to add $c - |\mathcal{F}_i|$ extra elements.

In the algorithm *Indexbuilding* of generating the index, we need to calculate the storage location of index members and then assign values to each member of the index. The chain table set $\mathcal{F}$ contains m chain tables, and each table has at most c nodes. All of the nodes in the chain table set $\mathcal{F}$ are stored in the index $\mathcal{I}$. As a result, the time complexity of the algorithm *Indexbuilding* is $O(m \cdot c)$.

3.3.3. Performing Optimal Route Finding. After the index is built, we will consider how to execute optimal route finding through the server on CCP. To accomplish this operation, for the element w_i from set W $(0 \leq i \leq m)$, we need to build the query term $\mathcal{Q}_i = (\psi_{i1}, \cdots, \psi_{ic})$. More specifically, the query term is created by the symmetric encryption algorithm $\mathrm{Enc}_{\mathrm{key}}(\cdot)$, that is $\mathcal{Q}_i = (\psi_{i1}, \cdots, \psi_{ic}) = (\mathrm{Enc}_{k_i}(w_i \Delta 1), \cdots, Enc_{k_i}(w_i \Delta c))$. When a customer is going to execute optimal route finding about the word w_i, the query term $\mathcal{Q}_i$ is delivered to the server on CCP. With the help of the query term and the index, the server completes the operation of optimal route finding in the cloud computing scenario through Algorithm 3.

In the algorithm *Queryperforming*, for the query term $\mathcal{Q}_i$ $(0 \leq i \leq m)$, if $\mathcal{I}[\psi_{ij}]$ is not a disturbingvalue $(1 \leq y \leq c)$, we will put $\mathcal{I}[\psi_{ij}]$ into R_{Q_i}. Therefore, the time complexity of the algorithm *Queryperforming* is $O(c)$.

In the paper, we propose a solution to solve the problem of privacy-guarding optimal route finding with support for semantic search on the encrypted graph in the cloud computing scenario. By the aid of encrypted query terms and a secure index, the server of CCP executes the privacy-guarding optimal route finding and returns the encrypted query results to the query customer. Our scheme satisfies the efficiency and the query security, and the server cannot obtain the privacy information of the query terms and the retrieval results.

Input: $W, \mathscr{F}, \mathscr{K}, \mathscr{K}'$
Output: $\mathscr{I}$
1: The compound terms set is $W = \{w_1, w_2, \cdots, w_m\}$, and the chain table set is $\mathscr{F} = \{\mathscr{F}_1, \mathscr{F}_2, \cdots \mathscr{F}_m\}$; The two secret key sets are $\mathscr{K} = \{k_1, k_2, \cdots, k_m\}$ and $\mathscr{K}' = \{k_1', k_2', \cdots, k_m'\}$;
2: **for all** $\mathscr{F}_i \in \mathscr{F}$ $(0 \le i \le m)$ **do**
3: **for all** $j \in [1, |\mathscr{F}_i|]$ **do**
4: $loc = Enc_{k_i'}(w_i \Delta j)$;
/* *loc is the location of members in the index* */
5: $\mathscr{I}[loc] = \mathscr{F}_i \to elem_{loc}$
6: **end for**
7: **end for**
8: **for all** $\mathscr{F}_i \in \mathscr{F}$ $(0 \le i \le m)$ **do**
9: **if** $|\mathscr{F}_i| < c$ **then**
10: **for all** $t \in [1, c - |\mathscr{F}_i|]$ **do**
11: $loc = Enc_{k_i'}(|\mathscr{F}_i| + t)$;
12: $\mathscr{I}[loc] = Enc_{k_i}(optrvalue + t)$;
13: **end for**
14: **end if**
15: **end for**
16: **return** $\mathscr{I}$.

Algorithm 2: *Indexbuilding*.

Input: $\mathscr{Q}_i, \mathscr{I}$
Output: R_{Q_i}
1: Generating the query term of the member w_i from the set W $(0 \le i \le m)$, and is denoted as $\mathscr{Q}_i = (\psi_{i1}, \cdots, \psi_{ic})$;
2: **for all** $j \in [1, c]$ **do**
3: **if** $\mathscr{I}[\psi_{ij}] \ne disturbingvalue$ **then**
4: put $\mathscr{I}[\psi_{ij}]$ into R_{Q_i};
5: **end if**
6: **end for**
7: **return** R_{Q_i};

Algorithm 3: *Queryperforming*.

4. Security Analysis

Now, we analyze the security of the PORF scheme. We first give several concepts used in security analysis of our scheme [10].

(i) *History*: the interaction between the server on CCP and a query customer, containing the graph data G and the set of query terms, expressed as $H_q = (G, \mathscr{Q}_1, \cdots, \mathscr{Q}_q)$. The partial history is expressed as $H_q^t = (G, \mathscr{Q}_1, \cdots, \mathscr{Q}_t)$, where $t \le q$

(ii) *View*: existing the history H_q about the key k, a view is defined as $V_k(H_q) = (\mathrm{Enc}_k(G), \mathscr{I}, \mathscr{Q}_1, \cdots, \mathscr{Q}_q)$. The partial view is $V_k^t(H_q) = (\mathrm{Enc}_k(G), \mathscr{I}, \mathscr{Q}_1, \cdots, \mathscr{Q}_t)$, where $t \le q$

(iii) *Access Pattern*: existing the history H_q about the key k, the access pattern is defined as a tuple $R(H_q) = (R_{\mathscr{Q}_1}, \cdots, R_{\mathscr{Q}_q})$, where $R_{\mathscr{Q}_i}(1 \le i \le q)$ is the result set of optimal route finding matching to the query term $\mathscr{Q}_i$

(iv) *Search Pattern*: existing the history H_q about the key k, the search pattern is defined as a binary symmetric matrix $\prod_q$, such that $\prod_q[i, j] = 1$ if $\mathscr{Q}_i = \mathscr{Q}_j$ and $\prod_q[i, j] = 0$, otherwise, for $1 \le i, j \le q$

(v) *Trace*: existing the history H_q about the key k, the trace is defined as a tuple $T_r(H_q) = (|\mathrm{Enc}_k(G)|, R(H_q), \prod_q)$, where $|\mathrm{Enc}_k(G)|$ is the overall size of the outsourcing graph, and $R(H_q)$ and $\prod_q$ are the access pattern and the search pattern of the history H_q, respectively. The trace of partial history is defined as $T_r(H_q^t) = (|\mathrm{Enc}_k(G)|, R(H_q^t), \prod_t)$, where $t \le q$

The server on CCP will perform optimal route finding through the index and the query term and cannot get the contents of the query results and the query term. In our work, we prove our optimal route finding scheme meets the adaptive semantic security. About the adaptive attack model, the server on CCP can make a choice from the query request on account of the query term and the results about optimal

route finding of previous queries [10, 36]. For the consideration of security analysis, we follow the security idea adopted in the previous schemes [10, 36]. According to the security guarantee of our PORF scheme, the server on CCP cannot obtain the additional information apart from the trace, and hence our proposed scheme of optimal route finding is secure. The security theorem of our PORF scheme is stated below.

Theorem 1. *Our PORF scheme meets the adaptive semantic security of searchable symmetric encryption idea.*

Proof. To prove the security of the PORF scheme, we first describe a polynomial-size simulator ε. For all $q \in N$, in the case of existing the trace $T_r(H_q^t)$ of a partial history, the simulator ε can build a view $(V_q^t)^*$ which is used to simulate the view $V_k^t(H_q)$ of the adversary, and such that $(V_q^t)^*$ and $V_k^t(H_q)$ cannot be distinguished, where k is a symmetrical key and $0 \le t \le q$.

For $t = 0$, the simulator ε generates the simulative encrypted outsourcing graph with the same size as the real graph through random strings. In the meantime, the simulator ε constructs the index $\mathcal{I}^*$ through randomly generating strings on the $T_r(H_q^0)$ which is also used to simulate the real index $\mathcal{I}$ and has the same size as the real index. The index $\mathcal{I}^*$ will be used to simulate the real index in other partial views $(V_q^t)^*$, where $1 \le t \le q$. It is very obvious that the simulative encrypted graph is indistinguishable from the real outsourcing graph, and the index $\mathcal{I}^*$ is indistinguishable from the index $\mathcal{I}$. Otherwise, one can distinguish between the outputs of the semantically secure symmetric encryption and the random strings with the same size. Thus, $(V_q^0)^*$ is indistinguishable from $V_k^0(H_q)$.

For $1 \le t \le q$, the simulator ε can still use the index $\mathcal{I}^*$ that was built before. The search pattern matrix $\prod_t$ about t query terms belongs to the trace $T_r(H_q^t)$. The simulator ε will generate the query terms $(\mathcal{Q}_1^*, \cdots, \mathcal{Q}_t^*)$ that are contained in the view $(V_q^t)^*$. In the generation of these query terms, the query terms $(\mathcal{Q}_1^*, \cdots, \mathcal{Q}_{t-1}^*)$ contained in the view $(V_q^{t-1})^*$ may be reused. Or else, the simulator ε will regenerate these query terms from $T_r(H_q^{t-1})$.

To generate $\mathcal{Q}_t^*$, the simulator ε first needs to determine whether H_q^{t-1} contains $\mathcal{Q}_t$ through checking whether $\prod_t[t, j] = 1$, where $1 \le j \le t-1$. If H_q^{t-1} cannot contain $\mathcal{Q}_t$, the simulator ε utilizes the information of $T_r(H_q^t)$ about $R_{\mathcal{Q}_t}$, i.e, $R_{\mathcal{Q}_t} = (R(w_t\Delta 1), \cdots, R(w_t\Delta c))$. The simulator ε selects an address x_i at random from the simulative index $\mathcal{I}^*$ for $1 \le i \le c$, making sure that all addresses are different and generates the query term $\mathcal{Q}_t^* = (x_1, \cdots, x_c)$. The simulator ε will remember the correlation between $\mathcal{Q}_t^*$ and w_t. Otherwise, if H_q^{t-1} contains w_t, the simulator ε will retrieve the query contents in connection with w_t and assigns it to $\mathcal{Q}_t^*$. This is to ensure that if H_q^{t-1} contains repeated query terms, then the query contents that are involved in $(V_q^t)^*$ are identical.

It is very obvious that the query terms $(\mathcal{Q}_1^*, \cdots, \mathcal{Q}_t^*)$ in $(V_q^t)^*$ are indistinguishable from the query terms $(\mathcal{Q}_1, \cdots, \mathcal{Q}_t)$ in $V_k^t(H_q)$. Otherwise, one could distinguish between the outputs of the semantically secure symmetric encryption and the random strings with the same size. Therefore, for $0 \le t \le q$, there is no polynomial-size adversary that could distinguish between $(V_q^t)^*$ and $V_k^t(H_q)$. Thus, the security theorem of the PORF scheme has been proven.

5. Experimental Evaluations

In this section, we will carry out experimental analysis of our scheme on the Enron email network graph [40, 41] and then give the evaluation results. The content of the experiment is completed through using C language program coding over the server on CCP and the local machine. The server on CCP is configured with the Linux operating system of 6 CPU cores with 3.0 GHz and 16 GB of RAM, and the local machine runs on the Windows 10 operating system equipped with Intel Core 4 CPU of 2.6 GHz. In our experimental analysis and evaluation, the index generation, query term generation, and the decryption of query results are performed on the local machine. The operation of optimal route finding is implemented on the server of CCP.

To verify the efficiency of our scheme in the experimental analysis, we compare our PORF scheme with the optimal route finding scheme in plaintext, which is referred to as MORF. The MORF scheme is similar with the PORF scheme, and the index is constructed in a similar way. But in the MORF scheme, the data and index are not encrypted. Comparing our PORF scheme with the MORF scheme, it is intended to evaluate the time and memory overhead over encrypted graph. For the outsourcing graph of the same number of vertices, the difference of the number of edges can have a certain effect on the experimental analysis. Therefore, for the outsourcing graph used in our experiment, we adopt five graph data sets chose at random and think about two circumstances to compare and evaluate the performances. One circumstance is that the outsourcing graph includes more edges, and the number of edges in the graph sets is, respectively, 7958, 16934, 35819, 73586, and 119823. The other circumstance includes less edges which contains half of the number or so of the first circumstance. In the experimental evaluation, we use PORF1 and MORF1 to denote the experiments containing much more edges, and PORF2 and MORF2 to denote the experiments containing less edges. The comparative analysis of our experiment about these circumstances can assess overhead issues of optimal route finding and validate the efficiency of our PORF scheme.

5.1. Index Building. To perform secure optimal route finding on CCP, we first need to transform the graph vertices to complete the semantic search requirements and generate new word set. The chain tables are then built on top of the new word set to hold the optimal route information, and finally, the index is generated based on all the chain tables through

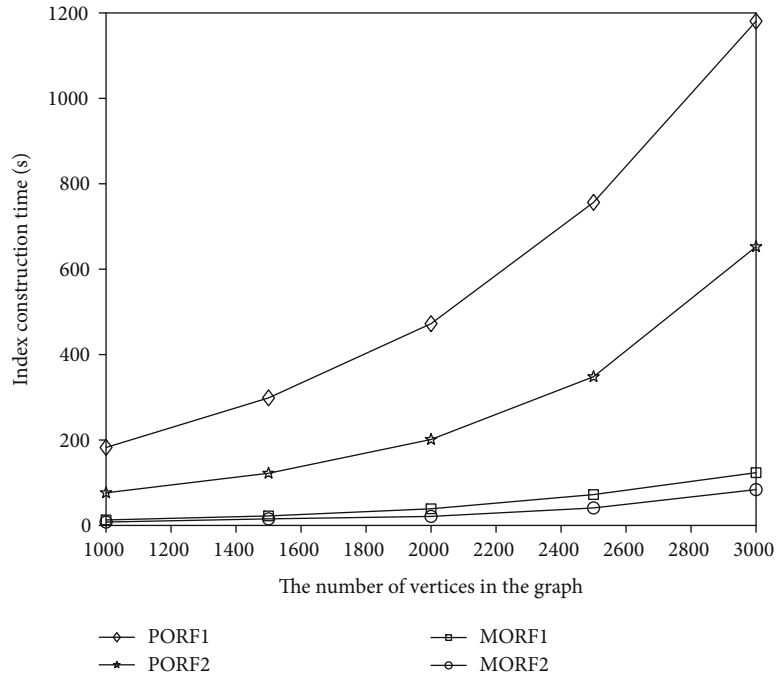

Figure 2: Index building time.

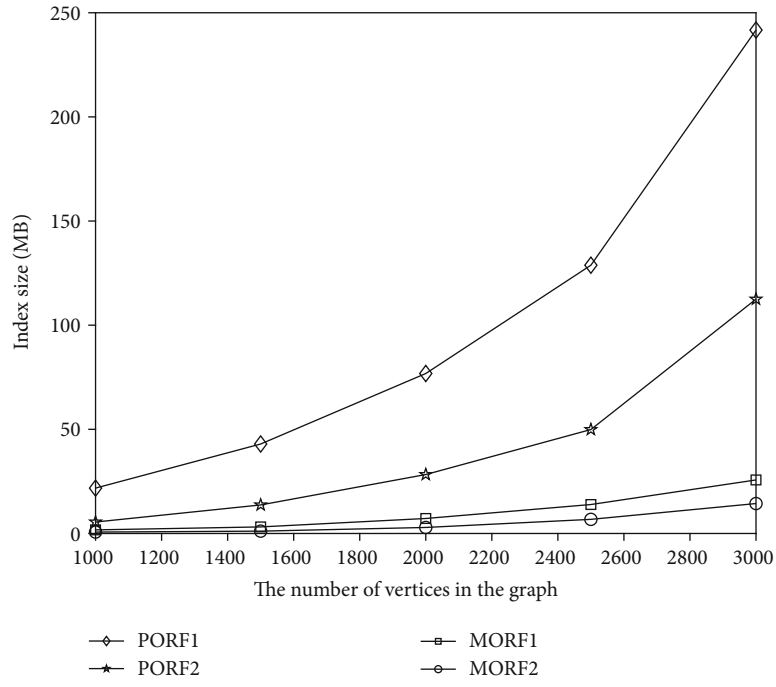

Figure 3: Index building size.

the *Indexbuilding* algorithm. The experiments evaluated the index building time and index size, respectively. Experimental analysis and evaluation on the experimental graph data are conducted under four conditions, and the experimental result figures of building index are given. The analysis results about the time of index generation are shown in Figure 2, where the abscissa represents the number of vertices in the graph data, and the ordinate shows the time of index generation.

From Figure 2, we can conclude that index building time and the number of vertices are closely related, and the time of index generation increases nearly linearly with the number of vertices under four conditions of PORF1, PORF2, MORF1, and MORF2. Generally, an outsourcing graph with more numbers of edges can have more optimal routes. Therefore, the time of index generation under PORF1 condition is more than that under PORF2 condition, and similarly the time of

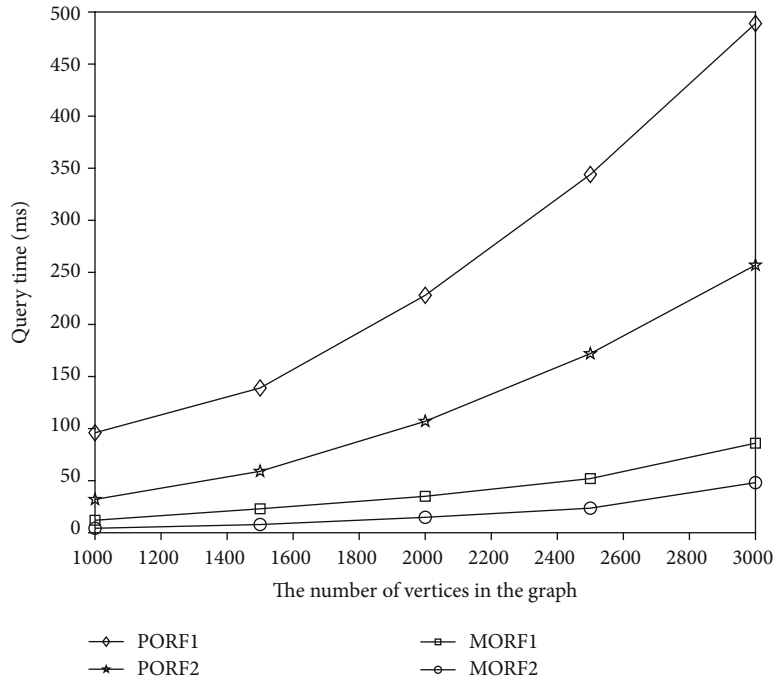

Figure 4: Query time.

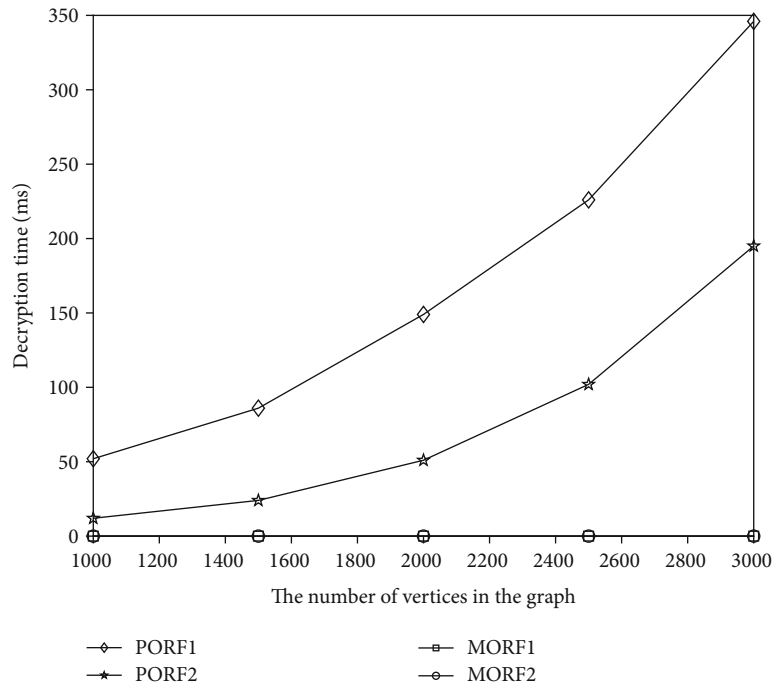

Figure 5: Decryption time.

index generation under MORF1 condition is more than that under MORF2 condition. For the encrypted graph query, we need to encrypt the graph data and build the encrypted index. As a result, the time of index generation is more than that on the plaintext graph. Under PORF condition, we get the security of private data with encryption time cost. After building the encrypted index, the queries on CCP can meet security requirements, and customers' privacy information cannot be compromised. Therefore, it is an effective way to increase index building time costs appropriately, and the process of encryption is done locally.

The experimental analysis of the size about index generation is plotted in Figure 3. The abscissa of the figure shows the number of vertices in the graph data, and the ordinate

represents the size of index generation. The curves of the size about index generation are changing approximately linearly with the vertex count increases under the four conditions. For outsourcing graphs of the same number of vertices, the more number of edges there are, the larger building size of the index is. Therefore, the index generation size of PORF1 is larger than that of PORF2, and the index generation size of MORF1 is larger than that of MORF2. The proposed PORF scheme can ensure the security of query process with additional storage overhead. The index generation size of PORF is a little larger than that in the plaintext query method, but the difference is not significant.

5.2. Performing Query. In the process of optimal route finding, the server on CCP makes use of the index and query terms to implement the query task by the *Queryperformin g* algorithm, and the experimental analysis includes query time evaluation and decryption time evaluation. The experimental results about query and decryption are, respectively, shown in Figures 4 and 5, where the horizontal axis represents the number of vertices in the graph data, and the vertical axis represents the time of query or decryption. From Figure 4, we can conclude that the time of performing query changes nearly linearly with the rise of vertex count. The query processes under PORF2 and MORF2 conditions, respectively, take less time than that under PORF1 and MORF1 conditions.

After the query is processed, the server on CCP sends the encrypted retrieval results to the query customer that completes the decryption locally. The results of experimental analysis about decryption are plotted in Figure 5. The time of the decryption process in our experiment is related to the decryption mechanism and the size of the query results. We adopt the same decryption mechanism under PORF1 and PORF2 two conditions. The decryption time of PORF1 and PORF2 increases almost linearly with the rise of vertex count. Time consumption of the decryption process under PORF2 condition is less than that under PORF2 condition.

In general, the index building of our PORF scheme is completed on local machine, and the time and size of the index are nearly linear to vertex count of the outsourcing graph. The query process is executed by the server on CCP, and the query time also increases with the increasing of vertex count. Meanwhile, the server on CCP does not get the privacy contents about the retrieval results and query terms. Our PORF scheme implements optimal route finding with support for semantic search on the encrypted graph and satisfies the privacy and efficiency of the query process.

6. Conclusion

In this paper, we propose a novel solution to address the problem of optimal route finding with support for semantic search, in which we adopt searchable encryption idea and porter stemmer mechanism. We first convert all graph vertices into a new word set by porter stemmer mechanism to satisfy semantic search. Then, we build the chain tables based on the new word set to place optimal route information and build an index based on the chain tables which is used to execute optimal route finding. Secondly, we prove the security of our scheme through formal analysis. Finally, we give experimental analysis and evaluation, and the results show that our scheme has good performance.

For our future work, we intend to build dynamic optimal route finding scheme to meet the needs of dynamic graphs. In addition, our other research direction is to combine encryption graph query with secret key management and update to meet a wider range of query requirements.

Data Availability

All relevant data to support the findings in this study belong to all authors and will be used for our future research. Requests for access to the data should be made to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank the editor and the reviewers' comments and helpful suggestions. This work was supported in part by the Nature Science Foundation of China under Grant 61762055, Grant 61962029, Grant 61662039, and Grant 61741111, in part by the Jiangxi Provincial Natural Science Foundation of China under Grant 20181BAB202014, Grant 20202ACBL202005, Grant 20181BAB202011, and Grant 20202BAB212006, in part by the Key Scientific and Technological Research Project of Jiangxi Provincial Education Department of China under Grant GJJ190899, in part by the Jiangxi Key Natural Science Foundation under Grant 20192ACBL20031, in part by the Science and Technology Research Project of Jiangxi Education Department under Grant GJJ180904, in part by the Humanities and Social Sciences Foundation of Colleges and Universities in Jiangxi Province under Grant TQ18111, and in part by the Key Program of Zhejiang Provincial Natural Science Foundation of China under Grant LZ18F020001.

References

[1] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Computational Science and Its Applications - ICCSA 2008, International Conference*, pp. 1249–1259, Perugia, Italy, 2008.

[2] F. Berger, P. Gritzmann, and S. de Vries, "Computing cyclic invariants for molecular graphs," *Networks*, vol. 70, no. 2, pp. 116–131, 2017.

[3] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[4] N. Cao, Z. Yang, C. Wang, K. Ren, and W. Lou, "Privacy-preserving query over encrypted graph-structured data in cloud computing," in *2011 International Conference on Distributed Computing Systems (ICDCS)*, pp. 393–402, Minneapolis, MN, USA, 2011.

[5] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Third International Conference on Applied Cryptography and Network Security (ACNS 2015)*, pp. 442–455, Berlin, Heidelberg, 2005.

[6] M. Chase and S. Kamara, "Structured encryption and controlled disclosure," in *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 577–594, Singapore, 2010.

[7] R. Ciucanu and P. Lafourcade, "GOOSE: a secure framework for graph outsourcing and SPARQL evaluation," in *Data and Applications Security and Privacy -34th Annual IFIP WG 11.3 Conference, DBSec 2020*, pp. 347–366, Regensburg, Germany, 2020.

[8] Z. Cui, Z. Lu, H. Yang, Y. Zhang, and S. Zhang, "A novel range search scheme based on frequent computing for edge-cloud collaborative computing in CPSS," *IEEE Access*, vol. 8, pp. 80599–80609, 2020.

[9] Z. Cui, Z. Wu, C. Zhou et al., "An efficient subscription index for publication matching in the cloud," *Knowledge-Based Systems*, vol. 110, pp. 110–120, 2016.

[10] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006)*, pp. 79–88, Alexandria, VA, USA, 2006.

[11] N. M. Donnell, E. Howley, and J. Duggan, "Dynamic virtual machine consolidation using a multi-agent system to optimise energy efficiency in cloud computing," *Future Generation Computer Systems*, vol. 108, pp. 288–301, 2020.

[12] D. Leilei, K. Li, Q. Liu, Z. Wu, and S. Zhang, "Dynamic multi-client searchable symmetric encryption with support for boolean queries," *Information Sciences*, vol. 506, pp. 234–257, 2020.

[13] Z. Fan, B. Choi, J. Xu, and S. S. Bhowmick, "Asymmetric structure-preserving subgraph queries for large graphs," in *31st IEEE International Conference on Data Engineering, ICDE 2015*, pp. 339–350, Seoul, South Korea, 2015.

[14] G. Gao, Z. Cui, and C. Zhou, "Blind reversible authentication based on PEE and CS reconstruction," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 1099–1103, 2018.

[15] G. Gao and G. Jiang, "Bessel-fourier moment-based robust image zero-watermarking," *Multimedia Tools and Applications*, vol. 74, no. 3, pp. 841–858, 2015.

[16] G. Gao and Y.-Q. Shi, "Reversible data hiding using controlled contrast enhancement and integer wavelet transform," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 2078–2082, 2015.

[17] E.-J. Goh, "Secure indexes," *IACR Cryptology ePrint Archive, 2003*, 2003.

[18] O. Goldreich, *The Foundations of Cryptography - Volume 2: Basic Applications*, Cambridge University Press, 2010.

[19] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of computer and system sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[20] R. Handa, C. R. Krishna, and N. Aggarwal, "Searchable encryption: a survey on privacy-preserving search schemes on encrypted outsourced data," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 17, 2019.

[21] L. Jiang, C. Xu, X. Wang, B. Luo, and H. Wang, "Secure outsourcing SIFT: efficient and privacy-preserving image feature extraction in the encrypted domain," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 1, pp. 179–193, 2020.

[22] M. E. Kabir, A. N. Mahmood, H. Wang, and A. K. Mustafa, "Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 408–417, 2020.

[23] B. Kay, P. Date, and C. D. Schuman, "Neuromorphic graph algorithms: extracting longest shortest paths and minimum spanning trees," in *NICE '20: Neuro-inspired Computational Elements Workshop*, pp. 1–6, Heidelberg, Germany, 2020.

[24] B. Klimt and Y. Yang, "Introducing the enron corpus," in *CEAS 2004 - First Conference on Email and Anti-Spam*, Mountain View, California, USA, 2004.

[25] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

[26] W. Liao, C. Luo, S. Salinas, and L. Pan, "Efficient secure outsourcing of largescale convex separable programming for big data," *IEEE Transactions on Big Data*, vol. 5, no. 3, pp. 368–378, 2019.

[27] Z. Liu, P. Zhou, Z. Li, and M. Li, "Think like a graph: real-time traffic estimation at city-scale," *IEEE Transactions on Mobile Computing*, vol. 18, no. 10, pp. 2446–2459, 2019.

[28] H. Li, Y. Yang, Y. Dai, S. Yu, and Y. Xiang, "Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data," *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 484–494, 2020.

[29] L. Li, M. Zhang, W. Hua, and X. Zhou, "Fast query decomposition for batch shortest path processing in road networks," in *36th IEEE International Conference on Data Engineering, ICDE 2020*, pp. 1189–1200, Dallas, TX, USA, 2020.

[30] C. Moral, A. de Antonio, R. Imbert, and J. Ramrez, "A survey of stemming algorithms in information retrieval," *Information Research*, vol. 19, no. 1, 2014.

[31] M. A. Rogov and A. Sedakov, "Coordinated influence on the opinions of social network members," *Automation and Remote Control*, vol. 81, no. 3, pp. 528–547, 2020.

[32] M. Shen, B. Ma, L. Zhu, R. Mijumbi, D. Xiaojiang, and H. Jiankun, "Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 940–953, 2018.

[33] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *2000 IEEE Symposium on Security and Privacy*, pp. 44–55, Berkeley, CA, USA, 2000.

[34] S. Tahir, L. Steponkus, S. Ruj, M. Rajarajan, and A. Sajjad, "A parallelized disjunctive query based searchable encryption scheme for big data," *Future Generation Computer System*, vol. 109, pp. 583–592, 2020.

[35] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *IEEE Transactions on parallel and distributed systems*, vol. 23, no. 8, pp. 1467–1479, 2012.

[36] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *Proceedings of the IEEE INFOCOM*, pp. 451–459, Orlando, FL, USA, 2012.

[37] Q. Wang, M. He, D. Minxin, S. S. M. Chow, R. W. F. Lai, and Q. Zou, "Searchable encryption over feature-rich data," *IEEE*

Transactions on Dependable and Secure Computing, vol. 15, no. 3, pp. 496–510, 2018.

[38] Y. Xiao and G. Gao, "Digital watermark-based independent individual certification scheme in wsns," *IEEE Access*, vol. 7, pp. 145516–145523, 2019.

[39] Z. Zhou, M. Yan, and Q. M. J. Wu, "Coverless image steganography using partial-duplicate image retrieval," *Soft Computing*, vol. 23, no. 13, pp. 4927–4938, 2019.

[40] Z. Zhou, Q. J. Wu, Y. Yang, and X. Sun, "Region-level visual consistency verification for large-scale partial-duplicate image search," *ACM Transactions on Multimedia Computing, Communications, and Application*, vol. 16, no. 2, pp. 1–25, 2020.

[41] C. Zygowski and A. Jaekel, "Optimal path planning strategies for monitoring coverage holes in wireless sensor networks," *Ad Hoc Networks*, vol. 96, p. 101990, 2020.