Computers, Materials & Continua
DOI:10.32604/cmc.2021.019462
Article

Tech Science Press

Road Distance Computation Using Homomorphic Encryption in Road Networks

Haining Yu[1], Lailai Yin[1,*], Hongli Zhang[1], Dongyang Zhan[1,2], Jiaxing Qu[3] and Guangyao Zhang[4]

[1]School of Cyberspace Science, Harbin Institute of Technology, Harbin, 150001, China
[2]The Ohio State University, Columbus, 43202, USA
[3]Heilongjiang Province Cyberspace Research Center, Harbin, 150001, China
[4]National Internet Emergency Response Center, Harbin, 150001, China
[*]Corresponding Author: Lailai Yin. Email: yinlailai163@163.com
Received: 14 April 2021; Accepted: 26 May 2021

Abstract: Road networks have been used in a wide range of applications to reduces the cost of transportation and improve the quality of related services. The shortest road distance computation has been considered as one of the most fundamental operations of road networks computation. To alleviate privacy concerns about location privacy leaks during road distance computation, it is desirable to have a secure and efficient road distance computation approach. In this paper, we propose two secure road distance computation approaches, which can compute road distance over encrypted data efficiently. An approximate road distance computation approach is designed by using Partially Homomorphic Encryption and road network set embedding. An exact road distance computation is built by using Somewhat Homomorphic Encryption and road network hypercube embedding. We implement our two road distance computation approaches, and evaluate them on the real city-scale road network. Evaluation results show that our approaches are accurate and efficient.

Keywords: Road network; road distance; homomorphic encryption

1 Introduction

Nowadays, road networks have been widely used in many application domains for sciences and engineering, such as closeness testing in spatial social networks, route planning, ride hailing or navigation in road networks, etc. For example, online ride hailing services such as Uber and DiDi employ large road networks with millions or even billions of vertices and edges in their operation. A well-maintained road network computation system plays a significant role. It not only reduces the cost of transportation, both in terms of money and time, but also improves the quality of upper services.

The shortest road distance computation has been considered as one of the most fundamental operations of road networks computation and has a wide range of applications. There are many efficient shortest distance (path) algorithms, such as Dijkstra's algorithm and Bellman Ford's Algorithm. In some application scenarios, the shortest road distance must be computed

in encryption form to avoid privacy leaks. As an example, an online ride hailing service enables a rider to find the closest driver to offer ride service. To enjoy this service, both riders and drivers have to update their locations to the online ride hailing service provider, while the service provider computes the shortest road distances from the rider to all drivers, and select the closest driver. But the service providers are not always honest, they may track users or infer their profiles for economic advantage. To alleviate this privacy concerns, the riders and the driver submit their encrypted locations, and the service provider can compute the encrypted road distance over received ciphertexts. However, it is not a trivial problem to compute shortest road distance in ciphertext domain. Some schemes have been presented in the literature to compute shortest road distance in a secure manner, which make use of cryptographic primitives to encrypt the road network itself or the corresponding pre-generated index, e.g., Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE), Yao's garbled circuits (GCs). Shen et al. [1] proposed a graph encryption scheme based on symmetric-key primitives and SHE, which enables approximate constrained shortest distance queries. Meng et al. [2] presented three schemes based on distance oracle and structured encryption for approximate shortest distance queries. Wang et al. [3] proposed a secure Graph DataBase (SecGDB) encryption scheme based on PHE and Yao's GCs, which supports exact shortest distance/path queries. Wu et al. [4] proposed an efficient cryptographic protocol for fully-private navigation based on compressing the next-hop routing matrices, symmetric Private Information Retrieval (PIR) and Yao's GCs. However, existing schemes are not efficient enough to compute large-scale shortest distances in real time.

To tackle the practical limitations of the state-of-the-art, we propose two secure road distance computation approaches, which can compute road distance over encrypted data efficiently. We summarize main contributions as:

- We propose an efficient approximate road distance computation approach over encrypted data, by using PHE and road network set embedding. Our approach only needs several additive homomorphic operations to compute an encrypted approximate road distance.
- We propose an efficient exact road distance computation approach over encrypted data, by using SHE and road network hypercube embedding. Our approach only needs several additive and multiplicative homomorphic operations over packed ciphertexts to compute an encrypted exact road distance.
- We implement our approximate road distance computation approach using Paillier Cryptosystem and exact road distance computation approach using FV scheme. Their performance is evaluated on the real city-scale road network. Evaluation results show that they achieve high accuracy, and keep efficient.

The remainder of this paper is structured as follows. In Section 2, we briefly introduce necessary preliminaries. In Section 3, we propose two road distance computation approaches over encrypted data. In Section 4, we evaluate their performance. Finally, we review the related literature and summarize the paper.

2 Preliminaries

2.1 Paillier Cryptosystem

Partially homomorphic encryption (PHE) allows to carry out operations over ciphertexts. Paillier cryptosystem [5] is a popular PHE scheme, which relies on the decisional composite residuosity assumption. We briefly summarize it as follows for better understanding and description of our plan.

- *Key Generation*: $(pk_{\text{PHE}}, sk_{\text{PHE}}) \leftarrow \text{KeyGen}_{\text{PHE}}(1^{\lambda})$. Select two primes p, q and calculate $N = p \times q$ and $\lambda = \text{lcm}(p-1, q-1)$, where lcm means to take the least common multiple of $p-1$ and $q-1$. Then, choose a random $g \in \mathbb{Z}_{N^2}^*$ so that $\gcd\left(\text{L}\left(g^{\lambda} \bmod N^2\right), N\right) = 1$, where $\text{L}(x) = (x-1)/N$. The public key is $pk_{\text{PHE}} = (N, g)$ and the private key is $sk_{\text{PHE}} = \lambda$.
- *Encryption*: $\hat{c} \leftarrow \text{Enc}_{\text{PHE}}(m, pk_{\text{PHE}})$. Given a plaintext $m \in \mathbb{Z}_N$ and a random number $r \in \mathbb{Z}_N$, the ciphertext can be derived as $\hat{c} = \text{Enc}_{\text{PHE}}(m \bmod N; r \bmod N) = g^m r^N \bmod N^2$.
- *Decryption*: $m \leftarrow \text{Dec}_{\text{PHE}}(\hat{c}, sk_{\text{PHE}})$. Let $\hat{c} \in \mathbb{Z}_{N^2}$ be a ciphertext, the plaintext of it is given by

$$m = \frac{\text{L}\left(\hat{c}^{\lambda} \bmod N^2\right)}{\text{L}\left(g^{\lambda} \bmod N^2\right)} \bmod N.$$

Paillier cryptosystem has properties of additive homomorphism and the mixed multiplication homomorphism: for any m_1, m_2, r_1, $r_2 \in \mathbb{Z}_N$, we obtain $\text{Enc}_{\text{PHE}}(m_1, r_1) \cdot \text{Enc}_{\text{PHE}}(m_2, r_2) = \text{Enc}_{\text{PHE}}(m_1 + m_2, r_1 r_2) \bmod N^2$, $\text{Enc}_{\text{PHE}}{}^{m_2}(m_1, r_1) = \text{Enc}_{\text{PHE}}\left(m_1 m_2, r_1^{m_2}\right) \bmod N^2$.

2.2 FV Scheme

FV scheme [6] is a widely used SHE scheme which can support a finite number of both multiplications and additions on data in the cipher domain. Mathematically, FV scheme depends on a hard computation problem named as Ring Learning with Errors (RLWE) problem. Set $R_t = \mathbb{Z}_t[x]/(x^n + 1)$, and in the ring structure R_t, x^n will be converted to –1. The plain text space in FV scheme is R_t, and the cipher text is the polynomial array in R_q. Given w being a base, $\ell + 1 = \lfloor \log_2 q \rfloor + 1$ represents the number of terms when the integer in the base q is decomposed into the base w. The $\ell + 1$ polynomials is obtained by decomposing the polynomials in R_q into base-w components coefficient-wise. With $a \xleftarrow{\$} \boldsymbol{\mathcal{S}}$ we uniformly sample a from the finite set $\boldsymbol{\mathcal{S}}$, and $[\cdot]_q$ represent reduction modulo q into the interval $(-q/2, q/2]$. The FV scheme is briefly introduced as follows.

- *Key Generation*: $\text{KeyGen}_{\text{SHE}}(n, q, t, \chi, w)$. Sample $a \xleftarrow{\$} R_2$ and output $sk_{\text{SHE}} = s$. Sample $a \xleftarrow{\$} R_q$, and $e \leftarrow \chi$. Output $pk_{\text{SHE}} = ([-(as + e)]_q, a)$. For $i \in \{0, \ldots, \ell\}$, sample $a_i \xleftarrow{\$} R_q$, $e_i \leftarrow \chi$. Output $\mathbf{evk} = ([-(a_i s + e_i) + w^i s^2]_q, a_i)$.
- *Encryption*: $\hat{c} \leftarrow \text{Enc}_{\text{SHE}}(m, pk_{\text{SHE}})$. For $p \in R_t$, let $pk_{\text{SHE}} = (b, a)$. Sample $u \xleftarrow{\$} R_2$, e_1, $e_2 \leftarrow \chi$ and output $\hat{c} = ([\lfloor q/t \rfloor m + bu + e_1]_q, [au + e_2]_q) \in R_q^2$.
- Decryption: $m \leftarrow \text{Dec}_{\text{SHE}}(\hat{c}, sk_{\text{SHE}})$. Set $s = sk_{\text{SHE}}$, $c_0 = \hat{c}[0]$, $c_1 = \hat{c}[1]$. Output $m = [\lfloor t/q \cdot [c_0 + c_1 s]_q \rceil]_t \in R_t$.
- *Homomorphic Addition*: $([\hat{c}_0[0] + \hat{c}_1[0]]_q, [\hat{c}_0[1] + \hat{c}_1[1]]_q) \leftarrow \hat{c}_0 \boxplus \hat{c}_1$. We have $\text{Dec}_{\text{SHE}}(\hat{c}_0 \boxplus \hat{c}_1, sk_{\text{SHE}}) \equiv \text{Dec}_{\text{SHE}}(\hat{c}_0, sk_{\text{SHE}}) + \text{Dec}_{\text{SHE}}(\hat{c}_1, sk_{\text{SHE}})$.
- *Homomorphic Multiplication*: $(c_0', c_1') \leftarrow \hat{c}_0 \boxtimes \hat{c}_1$. Calculate $c_0 = [\lfloor \frac{t}{q} \hat{c}_0[0] \hat{c}_1[0] \rceil]_q$, $c_1 = [\lfloor \frac{t}{q} (\hat{c}_0[0] \hat{c}_1[1] + \hat{c}_0[1] \hat{c}_1[0]) \rceil]_q$ and $c_2 = [\lfloor \frac{t}{q} \hat{c}_0[1] \hat{c}_1[1] \rceil]_q$.

Express c_2 in base w as $c_2 = \sum_{i=0}^{\ell} c_2^{(i)} w^i$. Set $c_0' = c_0 + \sum_{i=0}^{\ell} \mathbf{evk}[i][0] c_2^{(i)}$, $c_1' = c_1 + \sum_{i=0}^{\ell} \mathbf{evk}[i][1] c_2^{(i)}$, and output (c_0', c_1'). Note that we have $\text{Dec}_{\text{SHE}}(\hat{c}_0 \boxtimes \hat{c}_1, sk_{\text{SHE}}) \equiv \text{Dec}_{\text{SHE}}(\hat{c}_0, sk_{\text{SHE}}) \cdot \text{Dec}_{\text{SHE}}(\hat{c}_1, sk_{\text{SHE}})$.

3 Secure Distance Computation

In this section, we propose two different methods to compute road distance in ciphertext domain.

3.1 Approximate Road Distance Computation with PHE

3.1.1 Road Network Set Embedding

By using Road Network Set Embedding (RNSE) technique [7], the planar road network can be converted into a high-dimensional space, in which we can convert the complex calculation of the shortest road distance into a simple calculation supported through existing encryption primitives.

We model the road network as a weighted planar graph $\mathcal{G} = (V, E, W)$. Define V as the set of vertices in $\mathcal{G}$ (i.e., road junctions) and E as the set of edges in $\mathcal{G}$ (i.e., road sections). Let $\mathbf{R}$ be a set of subsets of V and it describes a high-dimensional embedding space:

$$\mathbf{R} = \left\{ V_{1,1}, \ldots, V_{1,\alpha}, \ldots, V_{\beta,1}, \ldots, V_{\alpha,\beta} \right\}, \tag{1}$$

where α and β are equal to $O(\log |V|)$. Subset $V_{i,j}$ is composed of 2^i nodes randomly selected from V. The shortest road distance between node $v \in V$ and subset $V_{i,j}$ can be calculated by

$$dist_R\left(v, V_{i,j}\right) = \min_{v' \in V_{i,j}} dist_R\left(v, v'\right). \tag{2}$$

Based on the above definition, the coordinate of a node v, which is a vector with $\mathrm{O}(\log^2 |V|)$ dimensions, is defined as the distance from the node v to each subset:

$$\mathbf{c}_v = \langle dist_R(v, V_{1,1}), \ldots, dist_R(v, V_{1,\alpha}), \ldots, dist_R(v, V_{\beta,1}), \ldots, dist_R(v, V_{\alpha,\beta}) \rangle. \tag{3}$$

Then we can use $\Omega = \{\mathbf{c}_v \mid v \in V\}$ to represent the embedded road network of $\mathcal{G}$.

For the coordinate of a position l on the road section $(v_s, v_d) \in E$, it can be denoted by

$$\mathbf{c}_l = \langle dist_R(l, V_{1,1}), \ldots, dist_R(l, V_{1,\alpha}), \ldots, dist_R(l, V_{\beta,1}), \ldots, dist_R(l, V_{\alpha,\beta}) \rangle, \tag{4}$$

where $dist_R(l, V_{i,j}) = \min\{dist_R(l, v_s) + dist_R(v_s, V_{i,j}),\ dist_R(l, v_d) + dist_R(v_d, V_{i,j})\}$.

Without losing generality, let the embedded road network have ω dimensions, s.t. $\omega \leq \lceil \log^2 |V| \rceil$. By calculating the chessboard distance from $\mathbf{c}_s$ to $\mathbf{c}_d$, the shortest distance from location l_s to l_d can be approximately represented as

$$\begin{aligned} dist_A(l_s, l_d) &\approx dist_C(\mathbf{c}_s, \mathbf{c}_d) \\ &= \max_{V_{i,j} \in \mathbf{R}} |dist_R(l_s, V_{i,j}) - dist_R(l_d, V_{i,j})| \\ &= \max_{1 \leq i \leq \omega} |\mathbf{c}_s[i] - \mathbf{c}_d[i]|, \end{aligned} \tag{5}$$

where $dist_C(\cdot, \cdot)$ denotes the chessboard distance amid two coordinates.

3.1.2 Encrypted Approximate Road Distance Computation Using Paillier Cryptosystem

Suppose that the road network is represented by $\mathcal{G} = (V, E, W)$ and the dimension of the embedded road network is ω, we can use the RNE technique described in Section 3.1.1 to calculate the coordinates of each point in $\mathcal{G}$, where the coordinate is a vector of ω dimension. Then, we will obtain the embedded road network denoted by $\Omega_{\mathrm{A}} = \{\mathbf{c}_v \mid v \in V\}$. Given a pair of

points on the road networks (l_s, l_d), let $\mathbf{c}_s = (\mathbf{c}_s[1], \ldots, \mathbf{c}_s[\omega])$ and $\mathbf{c}_d = (\mathbf{c}_d[1], \ldots, \mathbf{c}_d[\omega])$ denote the coordinates of l_s and l_d in the embedded road network Ω_{A}.

The coordinates $\mathbf{c}_{l_s}$ and $\mathbf{c}_{l_d}$ can be encrypted element-by-element using the public key pk_{PHE} generated by Paillier cryptosystem, respectively. The encrypted coordinates are represented as

$$[[\mathbf{c}_s]] = (\mathrm{Enc}_{\mathrm{PHE}}(\mathbf{c}_u[1], pk_{\mathrm{PHE}}), \ldots, \mathrm{Enc}_{\mathrm{PHE}}(\mathbf{c}_u[\omega], pk_{\mathrm{PHE}})), \tag{6}$$

$$[[\mathbf{c}_d]] = (\mathrm{Enc}_{\mathrm{PHE}}(\mathbf{c}_d[1], pk_{\mathrm{PHE}}), \ldots, \mathrm{Enc}_{\mathrm{PHE}}(\mathbf{c}_d[\omega], pk_{\mathrm{PHE}})). \tag{7}$$

The encrypted approximate road distance between l_s and l_d can be computed as follows:

1) Compute the ciphertext vector $[[\mathbf{dist}(l_s, l_d)]]$ over $[[\mathbf{c}_s]]$ and $[[\mathbf{c}_d]]$ based upon homomorphism operations of the Paillier cryptosystem:

$$\begin{aligned}&[[\mathbf{dist}(l_s, l_d)]]\\ &= (\mathrm{Enc}_{\mathrm{PHE}}(\mathbf{c}_s[1] - \mathbf{c}_d[1] + 2^{\ell}), \ldots, \mathrm{Enc}_{\mathrm{PHE}}(\mathbf{c}_s[\omega] - \mathbf{c}_d[\omega] + 2^{\ell}))\\ &= (\mathrm{Enc}_{\mathrm{PHE}}(\mathbf{c}_s[1])\mathrm{Enc}_{\mathrm{PHE}}{}^{-1}(\mathbf{c}_d[1])\mathrm{Enc}_{\mathrm{PHE}}(2^{\ell}), \ldots, \mathrm{Enc}_{\mathrm{PHE}}(\mathbf{c}_s[\omega])\mathrm{Enc}_{\mathrm{PHE}}{}^{-1}(\mathbf{c}_d[\omega])\mathrm{Enc}_{\mathrm{PHE}}(2^{\ell})).\end{aligned} \tag{8}$$

Note that $\mathrm{Enc}_{\mathrm{PHE}}(2^{\ell})$ is used to make sure that every element in $[[\mathbf{dist}(l_s, l_d)]]$ is positive.

2) Because Paillier cryptosystem has a plaintext space much larger than the upper limit of the road distance, several ciphertexts can be packed into one ciphertext using ciphertext packing technology, which can improve the efficiency. In the Paillier cryptosystem, assuming that $p = \lfloor N/(\ell + 1) \rfloor$ is the number of slots in a single packed ciphertext, p ciphertexts can be packed into one ciphertext. The main idea of the ciphertexts packing technique is described as follows. Assuming $a_1, \ldots, a_l$ are integers of ℓ bits $(1 \le l \le p)$, we use $[[a_1]], \ldots, [[a_l]]$ to express their ciphertexts. The packed ciphertext is constructed by $[[[a_1]]| \cdots |[[a_l]]] = \prod_{i=1}^{l} [[a_i]]^{2^{\ell(l-i)}}$.

After performing a decryption operation, we can get the packed plaintext $[a_1| \cdots |a_l] = \sum_{i=0}^{l} a_i 2^{\ell(l-i)}$, and then recover $a_1, \ldots, a_l$. Using the above ciphertext packing technique, every encrypted element in $[[\mathbf{dist}(l_s, l_d)]]$ can be packed into a same packed ciphertext:

$$[[\mathrm{dist}(l_s, l_d)]] = [[[\mathbf{dist}_1(l_s, l_d)]]| \cdots |[[\mathbf{dist}_{\omega}(l_s, l_d)]]] = \prod_{i=1}^{\omega} [[\mathbf{dist}_i(l_s, l_d)]]^{2^{\ell(\omega-i)}}, \text{ s.t., } p \ge \omega.$$

3) The approximate road distance between l_s and l_d , i.e. $dist_A(l_s, l_d)$, is the maximum element in $\mathbf{dist}(l_s, l_d)$, which is hidden in $[[\mathrm{dist}(l_s, l_d)]]$. Further post-process is required to extract the maximum from $\mathbf{dist}(l_s, l_d)$. In the simplest way, $[[\mathrm{dist}(l_s, l_d)]]$ can be decrypted with the secret key sk_{PHE}, and then unpacked to recover $\mathbf{dist}(l_s, l_d)$. There are, of course, other more complex scenes, where the maximum needs to be selected in an oblivious manner. For these scenes, some well-established cryptographic tools can be integrated, such as Yao's garbled circuit and secret sharing scheme. More details about secure comparison over encrypted integers can be referred to [8–11].

3.2 Exact Road Distance Computation with SHE

3.2.1 Road Network Hypercube Embedding

The m-dimensional hypercube $\mathcal{H}^m$, is a graph whose node set V consists of 2^m m-dimensional boolean vectors, i.e., vectors with binary coordinates 0 or 1, where two nodes are adjacent whenever they are different in exactly one coordinate. Moreover, the size of $\mathcal{H}^m$ is $m2^{m-1}$ and its older is 2^m. Road Network Hypercube Embedding (RNHE) technique [7] is concerned with finding mappings between a road network and a higher-dimensional hypercube that preserve

certain topological properties. Let the weighted graph $\mathcal{G} = (V, E, W)$ represent the road network. The vertices set is V and the edge set is E. Every edge (v_i, v_j) in E is related to a weight $W(v_i, v_j)$, which denotes the road distance of the edge. For a vertex $v_i \in V$, its coordinate can be expressed by a boolean vector $\mathbf{v}_i$ with m dimensions, and we use $\Omega_{\mathrm{H}} = \{\mathbf{v}_i \mid v_i \in V\}$ represents the embedded road network. We can obtain the shortest road distance between arbitrary two vertices by computing Hamming distance between their coordinates. Note that the location in the road network and its position in corresponding planar graph can typically be converted from one to the other, and hereafter they are used interchangeably.

Related definitions are as follows.

- The *interior face* $\mathcal{F}$ indicates a cycle of $\mathcal{G}$ surrounding a connected domain, and the *outer face* of $\mathcal{G}$ indicates an unbounded face.
- An *even*(*odd*) face means a face with *even*(*odd*) edges. Let $dia_{\mathcal{F}}$ be the diameter of $\mathcal{F}$, and if $d(v_i, v_i') = d(v_j, v_j') = dia_{\mathcal{F}}$ holds, edges $e = (v_i, v_j)$ and $\$e' = (v_i', v_j')$ in face $\mathcal{F}$ are *opposite*. For each edge $e \in \mathcal{F}$, it will have a unique opposite edge when $\mathcal{F}$ is an even face and have two opposite edges when $\mathcal{F}$ is an odd face.
- A cut $\mathcal{L}$ means a series of edges $\{e_1, e_2, e_3, \ldots, e_k\}$ satisfying the following three properties: 1) Either $e_1 = e_k$ or e_k and e_1 are all the edge of a outer face; 2) $\exists \mathcal{F}\,(e_i, e_{i+1} \in \mathcal{F})$; 3) in face $\mathcal{F}$, edges e_i and e_{i+1} are opposite. If graph $\mathcal{G}$ removes the edges of a cut $\mathcal{L}$, then $\mathcal{G}$ is divided into two subgraphs $\{\mathcal{G}/\mathcal{L}\}_0$ and $\{\mathcal{G}/\mathcal{L}\}_1$.
- An *alternating cut* refers to a cut which alternates on the odd faces. In other words, if the cut turns left (resp. right) on an odd face, then it turns right (resp. left) on the following odd face. We can view an alternating cut as a line which passes through the graph and intersects only the selected edges.

The embedded road network Ω_{H} can be constructed by $\mathcal{G}$ as follows. Each alternating cut $\mathcal{L}$ corresponds to two connected components $\{\mathcal{G}/\mathcal{L}\}_0$ and $\{\mathcal{G}/\mathcal{L}\}_1$. The coordinate of every vertex in $\{\mathcal{G}/\mathcal{L}\}_0$ will be appended with 0 and the coordinate of every vertex in $\{\mathcal{G}/\mathcal{L}\}_1$ will be appended with 1. We can find all alternating cuts which contain e as below.

- Starting with e, we move in both directions, take opposite edge on even face and end when we meet the first odd face in both directions.
- Next, we turn right on one odd face and left on the other (we can obtain more alternating cuts by changing the selection of odd face).
- Proceeding in both directions, we alternate at all odd faces and end up with reaching the outer face. Clearly, the coordinate is an m-dimensional boolean vector, where m is the total number of alternating cuts. At last, $\mathcal{G}$ can be embedded into an m-dimensional hypercube $\mathcal{H}^m$.

The computational complexity of hypercube embedding is $\mathrm{O}(|V|\sqrt{|V|})$. In Fig. 1, the road network corresponds to the hypercube $\mathcal{H}^{14}$, and its embedded road network is shown in Tab. 1. Note that above hypercube embedding is not affected by different road network topology.

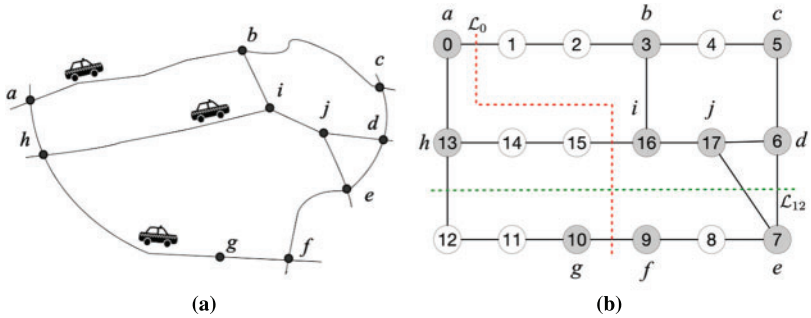

Figure 1: An example of alternating cuts of a road network (a) A simple road network. (b) Alternating cuts (e.g., $\mathcal{L}_0$ and $\mathcal{L}_{12}$)

Table 1: The embedded road network of the road network in Fig. 1

Vertex	Label
$\mathbf{v}_a$	00000000000000
$\mathbf{v}_b$	11111100000000
$\mathbf{v}_c$	11111111001100
$\mathbf{v}_d$	11111111111100
$\mathbf{v}_e$	11111101111110
$\mathbf{v}_f$	11010001111111
$\mathbf{v}_g$	01000001111111
$\mathbf{v}_h$	00000000110000
$\mathbf{v}_i$	11111100110000
$\mathbf{v}_j$	11111100111100

Let Ω_{H} be the embedded road network of the road network $\mathcal{G}$. The coordinates of two nodes $v_s, v_d \in V$ in Ω_{H} are expressed as $\mathbf{v}_s = (\mathbf{v}_s[0], \ldots, \mathbf{v}_s[m-1])$ and $\mathbf{v}_d = (\mathbf{v}_d[0], \ldots, \mathbf{v}_d[m-1])$. We can calculate the shortest road distance from v_s to v_d as below.

$$\begin{aligned} dist_E(v_s, v_d) &= \frac{1}{2} dist_H(\mathbf{v}_s, \mathbf{v}_d) = \frac{1}{2} \sum_{i=0}^{m-1} (\mathbf{v}_s[i] \oplus \mathbf{v}_d[i]) \\ &= \frac{1}{2} \left(\sum_{i=0}^{m-1} \mathbf{v}_s[i] + \sum_{i=0}^{m-1} \mathbf{v}_d[i] - 2 \sum_{i=0}^{m-1} \mathbf{v}_s[i] \mathbf{v}_d[i] \right), \end{aligned} \tag{9}$$

where $dist_E(v_s, v_d)$ means the exact shortest road distance, $dist_H(\cdot, \cdot)$ means the Hamming distance. In Fig. 1, the shortest road distance from v_a to v_e is calculated by $dist_E(v_a, v_e) = \frac{1}{2} d_H(\mathbf{v}_a, \mathbf{v}_e) = 6$.

Given $l = (v, \Delta)$ denoting a location in the road network $\mathcal{G}$, v means the nearest node to l and Δ means the shortest road distance between v and l. Let two locations be $l_s = (v_s, \Delta_s)$ and $l_d = (v_d, \Delta_d)$ respectively, and the shortest road distance from l_s to l_d is computed by

$$\begin{aligned} dist_E(l_s, l_d) &= dist_E(v_s, v_d) + \Delta_s + \Delta_d \\ &= \frac{1}{2} dist_H(\mathbf{v}_s, \mathbf{v}_d) + \Delta_s + \Delta_d. \end{aligned} \tag{10}$$

3.2.2 Encrypted Exact Road Distance Computation Using FV Scheme

For locations $l_s = (v_s, \Delta_s)$ and $l_d = (v_d, \Delta_d)$, the road distance between them can be computed in ciphertext domain by using FV Scheme. In a basic way, we can encrypt the respective coordinate $\mathbf{v}_s = \langle \mathbf{v}_s[0], \ldots, \mathbf{v}_s[m-1] \rangle$ and $\mathbf{v}_d = \langle \mathbf{v}_d[0], \ldots, \mathbf{v}_d[m-1] \rangle$ of v_s and v_d bit-by-bit with the public key pk_{SHE} to obtain two ciphertext sequences, denoted as:

$$[[\mathbf{v}_s]] = \langle \mathrm{Enc}_{\mathrm{SHE}}(\mathbf{v}_s[0], pk_{\mathrm{SHE}}), \ldots, \mathrm{Enc}_{\mathrm{SHE}}(\mathbf{v}_s[m-1], pk_{\mathrm{SHE}}) \rangle, \tag{11}$$

$$[[\mathbf{v}_d]] = \langle \mathrm{Enc}_{\mathrm{SHE}}(\mathbf{v}_d[0], pk_{\mathrm{SHE}}), \ldots, \mathrm{Enc}_{\mathrm{SHE}}(\mathbf{v}_d[m-1], pk_{\mathrm{SHE}}) \rangle. \tag{12}$$

Using the homomorphic property of FV Scheme, the encrypted $dist_E(v_s, v_d)$ can be calculated over $[[\mathbf{v}_s]]$ and $[[\mathbf{v}_d]]$ by:

$$\begin{aligned}[[dist_E(v_s, v_d)]] = \frac{1}{2}\Bigg(& \sum_{i=0}^{m-1} \mathrm{Enc}_{\mathrm{SHE}}(\mathbf{v}_s[i], pk_{\mathrm{SHE}}) \boxplus \sum_{i=0}^{m-1} \mathrm{Enc}_{\mathrm{SHE}}(\mathbf{v}_d[i], pk_{\mathrm{SHE}}) \\ & \boxminus 2 \sum_{i=0}^{m-1} \mathrm{Enc}_{\mathrm{SHE}}(\mathbf{v}_s[i], pk_{\mathrm{SHE}}) \boxtimes \mathrm{Enc}_{\mathrm{SHE}}(\mathbf{v}_d[i], pk_{\mathrm{SHE}}) \Bigg).\end{aligned} \tag{13}$$

Then, the encrypted $dist_E(l_s, l_d)$ can be computed by $[[dist_E(l_s, l_d)]] = [[dist_E(v_s, v_d)]] \boxplus [[\Delta_s]] \boxplus [[\Delta_d]]$.

When the length of m is long, the computation overhead of above basic distance computation method is heavy, since it is inefficient to encrypt/decrypt coordinate bit-by-bit. To reduce computation overhead, we propose an optimized approach with ciphertext packing to compute the shortest road distance efficiently. We now describe the details of two packed ciphertext constructions for the coordinates $\mathbf{v}_s$ and $\mathbf{v}_d$ as follows.

For the coordinate $\mathbf{v}_s = \langle \mathbf{v}_s[0], \ldots, \mathbf{v}_s[m-1] \rangle$, let $f_s(\mathbf{v}_s)$ represent the packed plaintext:

$$f_s(\mathbf{v}_s) = \sum_{i=0}^{m-1} \mathbf{v}_s[i] x^i \in R_t, \quad (m \le n), \tag{14}$$

where $\mathbf{v}_s$ can be converted into the polynomial coefficients of $f_s(\mathbf{v}_s)$ by packing. The packed ciphertext of $\mathbf{v}_s$ is calculated by encrypting $f_s(\mathbf{v}_s)$ as follows:

$$\hat{\mathbf{v}}_s = \mathrm{Enc}_{\mathrm{SHE}}(f_s(\mathbf{v}_s), pk_{\mathrm{SHE}}). \tag{15}$$

Given the coordinate $\mathbf{v}_d = \langle \mathbf{v}_d[0], \ldots, \mathbf{v}_d[m-1] \rangle$, let $\mathbf{v}_d$ represent the packed plaintext:

$$f_d(\mathbf{v}_d) = -\sum_{j=0}^{m-1} \mathbf{v}_d[j] x^{n-j} \in R_t, \quad (m \le n), \tag{16}$$

where $\mathbf{v}_d$ is converted into the polynomial coefficients of $f_d(\mathbf{v}_d)$ by packing. The packed ciphertext of $\mathbf{v}_d$ is calculated by encrypting $f_d(\mathbf{v}_d)$ as follows:

$$\hat{\mathbf{v}}_d = \mathrm{Enc}_{\mathrm{SHE}}(f_d(\mathbf{v}_d), pk_{\mathrm{SHE}}). \tag{17}$$

Encrypted $dist_E(v_s, v_d)$ can be computed by two packed ciphertexts denoted by $\hat{\mathbf{v}}_s$ and $\hat{\mathbf{v}}_d$. Using multiplicative homomorphism, the ciphertext of Hamming weight of $\mathbf{v}_s$ is calculated by the plaintext polynomial and the packed ciphertext, denoted as:

$$\hat{\mathbf{v}}_s \boxtimes \left(-\sum_{j=0}^{m-1} x^{n-j} \right), \tag{18}$$

then we have

$$\begin{aligned} f_s(\mathbf{v}_s) \times \left(-\sum_{j=0}^{m-1} x^{n-j} \right) &= -\sum_{i=0}^{m-1} \mathbf{v}_s[i]\, x^n + (\text{the other terms}) \,(\text{mod } t) \\ &= \sum_{i=0}^{m-1} \mathbf{v}_s[i] + (\text{non-constant terms}) \,(\text{mod } t), \end{aligned} \tag{19}$$

where $x^n = -1 (\text{mod} x^n + 1)$. For plaintext modulus t that is large enough, the constant term in Eq. (19) equals the Hamming weight of $\mathbf{v}_s$. Likewise, the ciphertext of Hamming weight of $\mathbf{v}_d$ is calculated by the plaintext polynomial and the packed ciphertext, denoted as:

$$\hat{\mathbf{v}}_d \boxtimes \sum_{i=0}^{m-1} x^i, \tag{20}$$

then we have

$$\begin{aligned} f_d(\mathbf{v}_d) \times \sum_{i=0}^{m-1} x^i &= -\sum_{i=0}^{m-1} \mathbf{v}_d[i]\, x^n + (\text{the other terms}) (\text{mod } t) \\ &= \sum_{i=0}^{m-1} \mathbf{v}_d[i] + (\text{non-constant terms}) \,(\text{mod } t). \end{aligned} \tag{21}$$

The constant term in Eq. (21) equals the Hamming weight of $\mathbf{v}_d$. Using multiplicative homomorphism, the ciphertext of the inner product of coordinates $\mathbf{v}_s$ and $\mathbf{v}_d$ is calculated by two packed ciphertexts as follows:

$$\hat{\mathbf{v}}_s \boxtimes \hat{\mathbf{v}}_d, \tag{22}$$

then we have

$$\begin{aligned} f_s(\mathbf{v}_s) \times f_d(\mathbf{v}_d) &= \sum_{i=0}^{m-1} \mathbf{v}_s[i] x^i \times \left(-\sum_{j=0}^{m-1} \mathbf{v}_d[j] x^{n-j} \right) \\ &= -\sum_{i=0}^{m-1} \mathbf{v}_s[i]\, \mathbf{v}_d[i]\, x^n + (\text{the other terms}) (\text{mod } t) \\ &= \sum_{i=0}^{m-1} \mathbf{v}_s[i] \mathbf{v}_d[i] + (\text{non-constant terms}) \,(\text{mod } t). \end{aligned} \tag{23}$$

The inner product of two coordinates $\mathbf{v}_s$ and $\mathbf{v}_d$ is equal to the constant term in Eq. (23).

Based on Eq. (9), we can use three packed ciphertexts (18), (20) and (22) to calculate the encrypted $dist_E(l_s, l_d)$ as follows:

$$\widehat{dist}_E(v_s, v_d) = \hat{\mathbf{v}}_s \boxtimes \left(-\frac{1}{2} \sum_{j=0}^{m-1} x^{n-j} \right) \boxplus \hat{\mathbf{v}}_d \boxtimes \frac{1}{2} \sum_{i=0}^{m-1} x^i \boxminus \hat{\mathbf{v}}_s \boxtimes \hat{\mathbf{v}}_d. \tag{24}$$

Based on Eq. (10) and (24), the ciphertext of the distance between location $l_s = (v_s, \Delta_s)$ and $l_d = (v_d, \Delta_d)$ as follows:

$$\hat{d}_R(l_s, l_d) = \hat{d}_R(v_s, v_d) \boxplus \hat{\Delta}_s \boxplus \hat{\Delta}_d, \tag{25}$$

where we have $\hat{\Delta}_s = \mathrm{Enc}_{\mathrm{SHE}}(\Delta_s, pk_{\mathrm{SHE}})$ and $\hat{\Delta}_d = \mathrm{Enc}_{\mathrm{SHE}}(\Delta_d, pk_{\mathrm{SHE}})$.

It needs only three multiplicative homomorphisms operations and four subtractive/additive homomorphisms operations for the calculation of the shortest road distance over two locations.

4 Experiment Evaluation

Our experiments are performed on the real road network of California, which consists of 21048 vertices and 21693 edges (www.cs.utah.edu/~lifeifei/SpatialDataset.htm). Following the assumptions made in [7], we need to delete some trivial edges and insert virtual vertices on edges with fixing the unit distance. For PHE, we use the Paillier cryptosystem library (acsc.cs.utexas.edu/libpaillier). For the modified Paillier cryptosystem, we set N and g to 1024 bits and 160 bits, respectively. For SHE, we use FV scheme built on FV-NFLlib (github.com/CryptoExperts/FV-NFLlib), and the degree of polynomials in FV scheme n is set to 2048. All our experiments are conducted and executed on a PC running Ubuntu 18.04 LTS, with an Intel i7 processor at 3.4GHZ and 16GB RAM.

We evaluate the accuracy and the efficiency of our proposed road distance approaches in a k-Nearest Neighbors (kNN) query application [12,13]. We first generate some locations on the edges of the road networks in a random fashion. Then, one location is randomly picked as the starting location, from which all road distances to other locations are computed by using our approaches. Finally, the closest location is selected as the nearest neighbor. Fig. 2 depicts the accuracy of kNN query by using Euclidean distance, approximate road distance (dimension $\omega = 8, 16, 24, 32$) and exact road distance under different location scales. Euclidean distance is considered as the lower bound of accuracy, which always stays from 85% to 90%. We can see the accuracy of approximate road distance raises steadily as the dimension of the embedded road network increases. It is roughly 95% when the dimension is higher than 24. That is because higher dimension indicates higher accurate approximation. When we vary the location scale from 1000 to 4000, the accuracy of Euclidean distance gradually increases as the location scale increases, because larger location scale means there may exist closer destination locations located around the starting location. But the accuracy of Euclidean distance is still less than 90%. The accuracy of approximate road distance is always high under any driver scale, which is roughly 95% if the dimension is higher than 24. As expected, the accuracy of exact road distance keeps almost 100% under any location scale. Above experimental results demonstrate that both approximate road distance computation approach and exact road distance computation approach can reach a higher accuracy due to the choice of road distance. We use average online computation cost for per kNN query to evaluate the efficiency. As shown in Fig. 3, the computation cost of the approximate road distance computation approach raises with the dimension increases. The

reason is that higher dimension requires more encryption operations over a location coordinate. Meanwhile, the computation cost of both the approximate road distance computation approach and the exact road distance computation approach increase almost linearly as the location scale increases. That is because more distance computation is required with larger location scale. From above evaluation, we can see that the two approaches achieve high accuracy and efficiency.

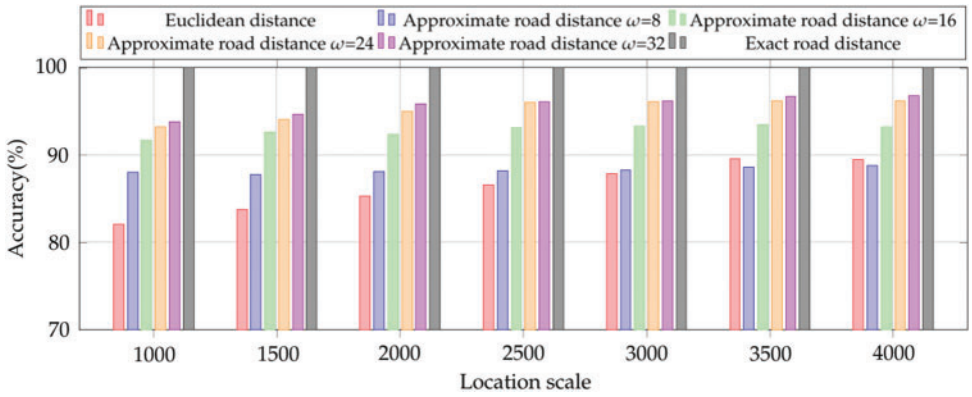

Figure 2: The accuracy of Euclidean distance, approximate road distance and exact road distance

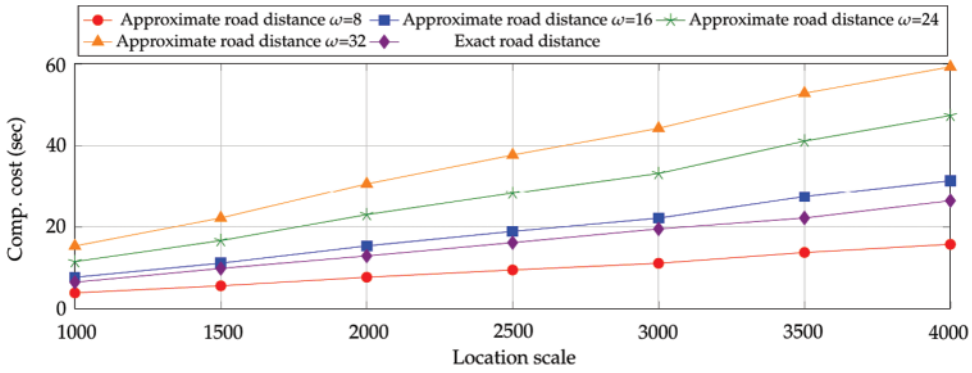

Figure 3: The accuracy of Euclidean distance, approximate road distance and exact road distance

5 Related Works

Numerous protocols have been proposed for private shortest road distance computation in different applications, such as kNN query and navigation. For (yet related) privacy issues of distance computation, some approaches utilize structural anonymization [14], differential privacy [15–17] or Private Information Retrieval (PIR) [18] to guarantee privacy for the client or the server. However, these approaches suffer from limited privacy, performance or scalability. There is also a vast literature on privacy-preserving shortest distance computation in structured encryption [19] or graph encryption, which focuses on protecting graph data when outsourced to third-party servers or on the cloud [20–22]. The most famous class of structured encryption schemes are searchable symmetric encryption (SSE) schemes [23]. Generally speaking, SSE schemes usually encrypt indexes or search trees for the purpose of efficiently searching on encrypted data. Another

line of work executing graph algorithms over encrypted graphs is to develop data-oblivious algorithms [24] or data structures [25]. In these solutions, the graph data is stored in an Oblivious RAM (ORAM) [26] or an oblivious data structure on the server. The client can compute the shortest distances on the server without leaking its access patterns. Also relevant are the works based on SMC, such as Yao's GCs and ORAM. The generic solution is to construct a GC that contains the entire graph structure for a shortest-path algorithm and apply Yao's protocol. However, above approaches are often prohibitively expensive and impractical for city-scale road networks [27,28]. For instance, the GC-based approach by Carter et al. [29,30] requires several minutes to compute a single shortest path in a road network with just 100 vertices. Another generic approach combining GCs and ORAM requires communication overhead on the order of GB and run-times ranging from tens of minutes to several hours for a single computation on a network with 1024 vertices. Recently, some schemes are proposed to support computation over large-scale encrypted graphs. Shen et al. [1] proposed a graph encryption scheme based on symmetric-key primitives and SHE, which enables approximate constrained shortest distance queries. Meng et al. [2] presented three schemes based on distance oracle and structured encryption for approximate shortest distance queries. Above two schemes provide an estimate on the shortest distance, along with sacrificing accuracy. Wang et al. [3] proposed a secure Graph Data Base (SecGDB) encryption scheme based on PHE and Yao's GCs, which supports exact shortest distance/path queries. Wu et al. [4] proposed an efficient cryptographic protocol for fully-private navigation based on compressing the next-hop routing matrices, symmetric Private Information Retrieval (PIR) and Yao's GCs, which requires about 1.5 s and less than 100 KB of bandwidth for each hop in city-scale road network. Compared with existing schemes, our two road distance computation approaches are more efficient to compute large-scale shortest distances in real time.

6 Conclusions

In this paper, we proposed two secure road distance computation approaches, which can compute road distance over encrypted data efficiently. An approximate road distance computation approach is designed by using Partially Homomorphic Encryption and road network set embedding. An exact road distance computation is built by using Somewhat Homomorphic Encryption and road network hypercube embedding. According to the evaluation over a real city-scale road network, we have verified that our approaches are accurate and efficient.

Funding Statement: This work was partially supported by National Natural Science Foundation of China (Grant Nos. 61601146, 61732022), National Key R&D Program of China (Grant No. 2016QY05X1000).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Shen, B. Ma, L. Zhu, R. Mijumbi and X. Du, "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.

[2] X. Meng, S. Kamara, K. Nissim and G. Kollios, "GRECS: Graph encryption for approximate shortest distance queries," in *22nd ACM SIGSAC Conf. on Computer and Communications Security*, Denver, Colorado, USA, pp. 504–517, 2015.

[3] Q. Wang, K. Ren, M. Du, Q. Li and A. Mohaisen, "SecGDB: Graph encryption for exact shortest distance queries with efficient updates," in *21st Int. Conf. on Financial Cryptography and Data Security*, Sliema, Malta, pp. 79–97, 2017.

[4] D. J. Wu, J. Zimmerman, J. Planul and J. C. Mitchell, "Privacy-preserving shortest path computation," in *23rd Annual Network and Distributed System Security Sym., NDSS 2016*, San Diego, California, USA, 2016.

[5] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Prague, Czech Republic, pp. 223–238, 1999.

[6] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *IACR Cryptology ePrint Archive*, vol. 2012, pp. 144, 2012.

[7] S. Gupta, S. Kopparty and C. Ravishankar, "Roads, codes, and spatiotemporal queries," in *23rd ACM SIGMOD-SIGACT-SIGART Sym. on Principles of Database Systems*, Paris, France, pp. 115–124, 2004.

[8] H. Yu, X. Jia, H. Zhang, X. Yu and J. Shu, "PSRide: Privacy-preserving shared ride matching for online ride hailing systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1425–1440, 2019.

[9] H. Yu, J. Shu, X. Jia, H. Zhang, X. Yu *et al.,* "Lightweight and privacy-preserving ride matching over road networks in online ride hailing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10418–10428, 2019.

[10] H. Yu, H. Zhang, X. Yu, X. Du and M. Guizani, "PGRide: Privacy-preserving group ridesharing matching in online ride hailing services," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5722–5735, 2020.

[11] H. Yu, X. Jia, H. Zhang and J. Shu, "Efficient and privacy-preserving ride matching using exact road distance in online ride hailing services," *IEEE Transactions on Services Computing*, 2020. https://doi.org/10.1109/TSC.2020.3022875.

[12] C. Shahabi, M. R. Kolahdouzan and M. Sharifzadeh, "A road network embedding technique for k-nearest neighbor search in moving object databases," *GeoInformatica*, vol. 7, no. 3, pp. 255–273, 2003.

[13] A. Bachir, I. M. Almanjahie and M. K. Attouch, "The k nearest neighbors estimator of the m-regression in functional statistics," *Computers, Materials & Continua*, vol. 65, no. 3, pp. 2049–2064, 2020.

[14] J. Gao, J. X. Yu, R. Jin, J. Zhou, T. Wang *et al.,* "Neighborhood privacy protected shortest distance computing in cloud," in *2011 ACM SIGMOD Int. Conf. on Management of Data*, Athens, Greece, pp. 409–420, 2011.

[15] A. Sealfon, "Shortest paths and distances with differential privacy," in *35th ACM SIGMOD-SIGACT-SIGAI Sym. on Principles of Database Systems*, San Francisco, California, USA, pp. 29–41, 2016.

[16] H. Chen, S. Li and Z. Zhang, "A differential privacy based (k-ψ)-anonymity method for trajectory data publishing," *Computers, Materials & Continua*, vol. 65, no. 3, pp. 2665–2685, 2020.

[17] W. Han, M. Cheng, M. Lei, H. Xu, Y. Yang *et al.,* "Privacy protection algorithm for the internet of vehicles based on local differential privacy and game model," *Computers, Materials & Continua*, vol. 64, no. 2, pp. 1025–1038, 2020.

[18] Y. Xi, L. Schwiebert and W. Shi, "Privacy preserving shortest path routing with an application to navigation," *Pervasive and Mobile Computing*, vol. 13, pp. 142–149, 2014.

[19] M. Chase and S. Kamara, "Structured encryption and controlled disclosure," in *Int. Conf. on the Theory and Application of Cryptology and Information Security*, Singapore, pp. 577–594, 2010.

[20] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su *et al.,* "A survey on access control in the age of internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, 2020.

[21] L. Jiang and Z. Fu, "Privacy-preserving genetic algorithm outsourcing in cloud computing," *Journal of Cyber Security*, vol. 2, no. 1, pp. 49–61, 2020.

[22] X. Liu, J. Zhang, X. Li, S. Zhou, S. Zhou *et al.,* "A block compressed sensing for images selective encryption in cloud," *Journal of Cyber Security*, vol. 1, no. 1, pp. 29–41, 2019.

[23] F. Hahn and F. Kerschbaum, "Searchable encryption with secure and efficient updates," in *2014 ACM SIGSAC Conf. on Computer and Communications Security*, Scottsdale, Arizona, USA, 310–320, 2014.

[24] M. Blanton, A. Steele and M. Alisagari, "Data-oblivious graph algorithms for secure computation and outsourcing," in *8th ACM SIGSAC Sym. on Information, Computer and Communications Security*, Hangzhou, China, pp. 207–218, 2013.

[25] M. Keller and P. Scholl, "Efficient, oblivious data structures for MPC," in *Proc. of Int. Conf. on the Theory and Application of Cryptology and Information Security*, Berlin, Germany, pp. 506–525, 2014.

[26] E. Stefanov, M. Van Dijk, E. Shi, C. Fletcher, L. Ren *et al.,* "Path ORAM: An extremely simple oblivious ram protocol," in *2013 ACM SIGSAC Conf. on Computer & Communications Security*, Berlin, Germany, pp. 299–310, 2013.

[27] S. Su, Z. Tian, S. Liang, S. Li, S. Du *et al.,* "A reputation management scheme for efficient malicious vehicle identification over 5G networks," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 46–52, 2020.

[28] Z. Tian, X. Gao, S. Su and J. Qiu, "Vcash: A novel reputation framework for identifying denial of traffic service in internet of connected vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3901–3909, 2020.

[29] H. Carter, B. Mood, P. Traynor and K. Butler, "Secure outsourced garbled circuit evaluation for mobile devices," in *22nd USENIX Conf. on Security*, Washington D.C., USA, pp. 289–304, 2013.

[30] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du *et al.,* "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.