Authentication of Skyline Query over Road Networks

Xiaoyu Zhu[1], Jie Wu[2], Wei Chang[3], Guojun Wang[4(✉)], and Qin Liu[5]

[1] School of Information Science and Engineering, Central South University, Changsha 410083, China
[2] Center for Networked Computing, Temple University, Philadelphia, PA 19122, USA
[3] Department of Computer Science, Saint Joseph's University, Philadelphia, PA 19131, USA
[4] School of Computer Science and Technology, Guangzhou University, Guangzhou 510006, China
csgjwang@gmail.com
[5] School of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

Abstract. With the increase of location-aware and Internet-capable mobile handset devices, location-based services (LBSs) have experienced an explosive growth in recent years. To scale up services, location-based service providers (LBSPs) outsource data management to third-party cloud service providers (CSPs), which in turn provide data query services to users on behalf of LBSPs. However, the CSPs cannot be trusted, which may return incorrect or incomplete query results to users, intentionally or not. Skyline query is an important kind of query, which asks for the data that is not spatially dominated by any other data. Therefore, enabling users to authenticate skyline query results is essential for outsourced LBSs. In this paper, we propose an authentication solution to support location-based skyline query. By embedding each data with its skyline neighbors in the data's signature, our solution allows users to efficiently verify the soundness and completeness of location-based skyline query results. Through theoretical analysis, we demonstrate the effectiveness of our proposed solution.

Keywords: Data outsourcing · Query authentication
Skyline query · LBSP · Road network

1 Introduction

With the explosive growth of mobile handset devices, such as smartphones and tablet computers, location-based services (LBSs) attract increasing attention from both research and industry communities. Mobile users carrying location-aware and Internet-capable mobile devices are able to perform queries to learn about points of interests (POIs) anywhere and at any time. As the adoption of cloud computing increases, which provides LBSs an efficient way to outsource

G. Wang et al. (Eds.): SpaCCS 2018, LNCS 11342, pp. 72–83, 2018.
https://doi.org/10.1007/978-3-030-05345-1_6

POI datasets and various data queries to third-party cloud service providers (CSPs). Outsourcing POI searching to third-party CSPs provides a cost-effective way to support large scale data storage and query processing.

As one important class among various types of location-based queries, location-based skyline queries (LBSQs) [1–3] ask for the POIs that are not dominated by any other POI with respect to a given query position, and we say one POI dominates another if the former is both closer to the query position and preferable in the numeric attribute of interest. In Fig. 1, a POI is characterized by a location and a price, such as o_1's location and price are 1 and 2, respectively. We say one POI dominates another if the former is both cheaper. For example, o_4 dominates o_3 because o_4 is both closer to q and cheaper than o_3. We can observe that the LBSQ results are unpredictable, because as the query position moves, the POIs' distance to the query position changes.

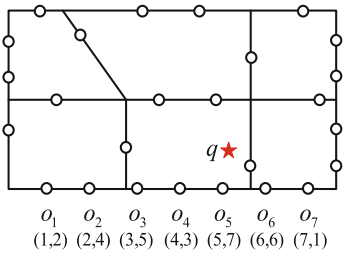


Fig. 1. An example of road network.

Due to security concerns, query-result integrity needs be protected against possibly dishonest CSPs. The CSPs may return incorrect results for a variety of reasons. For example, the CSPs may manipulate LBSP's POI dataset or return biased results in favor of POIs willing to pay. In addition, the CSPs may opt to return incomplete results in order to save computation resources, or they may return overly large results so that the CSPs can charge fees for the communication bandwidth. Hence, it is vital to provide users a capability to authenticate query results to ensure the authenticity and completeness. Results are authentic if every result appears in the original POI datasets and are complete if all the skyline POIs are included in the query results.

Most recently, Chen et al. [4] proposed a LBSQ authentication method in which the POIs are modeled as distributed over a road network. However, Chen's solutions still have several limitations. Firstly, the data preprocess is complex. The LBSP needs to preprocess the dataset, generate the skyline neighbor set and skyline neighbor range, and then generate MHT for query verification. Secondly, the query process has a high computation overhead. The dataset is first divided into two subsets, then CSP do skyline query three times on the dataset and its two subsets respectively. Finally, the size of the verification object is large, each skyline result contains an auxiliary set for verification.

In this paper, we propose a novel method to solve the LBSQ authentication problem: our method enables simple data preprocess, efficient skyline query and lower communication overhead.

The contributions of this paper are summarized as follows:

- We propose a novel authentication solution to verify skyline query in road network. Our method supports efficient skyline neighbor generation process and each record is chained with its distance neighbors and skyline neighbors.
- Our method supports efficient data query process and small communication overhead from LBSP to CSP and from CSP to user.
- We give the performance analysis, which shows the effectiveness and efficiency of our method.

The remains of the paper is organized as follows: Sect. 2 summarizes the related work. Section 3 presents the problem formulation. Section 4 describes the details of our proposed solution. Section 5 presents the security analysis and overhead analysis. Finally, Sect. 6 concludes our paper.

2 Related Work

Query authentication has been studied extensively. Most studies [5–8] are based on either Merkle Hash Tree (MH-tree) [5] or signature chain [6]. In signature chain, each data in the dataset is signed by the data owner, while the signatures of results and non-result boundaries are returned to the client. Various types of queries have been studied, including range queries [9,10], spatial top-k queries [11–13], multi-dimensional top-k queries [14,15], kNN queries [8,16], shortest-path queries [17], skyline queries [4,18,19] etc. It is common to let the data owner outsource both its dataset and its signatures of the dataset to the service provider, which returns both the query result and a VO computed from the signatures for the querying user to verify query integrity.

The skyline query can be widely adopted in information retrieval [20–22], searchable encryption [23–25], system monitoring [26], resource allocation [27], and etc. References [4,18,19,28] are the most related work targeting verifiable outsourced skyline query processing via untrusted CSPs. In [18,19], Lin et al. presented several schemes based on a data structure called Merkle Skyline R-tree assumed that the POIs are distributed in a general 2D plane. Recently, Chen et al. [4] proposed a LBSQ authentication method which is more practical in real situations, the POIs are modeled as distributed over a road network rather than a 2D plane. However, their work returns large verification objects, incurs high communication overhead. Our work aims at decreasing the communication cost in LBSQ authentication methods.

3 Models and Problem Formulation

In this section, we introduce our system model, the definition of LBSQ, and problem formulation.

3.1 System Model

Our system model involves three types of parties: LBSP, CSP, and data user. The general setting works as follows: First, the LBSP makes some pre-computation on the POI dataset and computes the dataset's signatures S. Second, the LBSP uploads the POI dataset and their signatures to the CSP. Third, a user sends a skyline query to the CSP, and the CSP computes the results, a verification object and sends both of them back to the user. Finally, the user verifies the soundness and completeness.

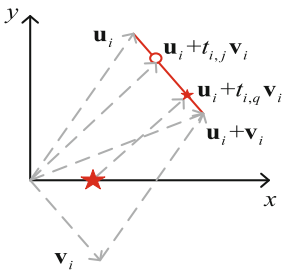


Fig. 2. Representation of a road segment. (Color figure online)

The dataset $\mathcal{O}$ contains a set of POIs of the same category, e.g., hotel, and each POI is characterized by its location and one numeric attribute (e.g., price). We adopt the similar settings with Chen's [4]. As shown in Fig. 1, the POIs reside in a road network is represented by a planar graph $G = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V}$ is the set of vertices, and $E = \{e_1, \cdots, e_m\}$ is the set of road segments. The representative red road segment and POI are e_i and $o_{i,j}$, and the query position is denoted as q. As shown in Fig. 2, we use $\{e_i = \boldsymbol{u}_i + t\boldsymbol{v}_i\}$ to denote the segment, where $\boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{R}^2$ are two reference vectors, $\boldsymbol{u}_i$ and $\boldsymbol{u}_i + \boldsymbol{v}_i$ are two end points.

The dataset is denoted as $\mathcal{O} = \bigcup_{i=1}^{m} \mathcal{O}_i$, where $\mathcal{O}_i$ is the set of POIs in road segment e_i. Assume there are n_i POIs in road e_i, and we use $o_{i,j}$ to denote the jth POI in road e_i. Each POI can be represented as $o_{i,j} = \{t_{i,j}, \lambda_{i,j}\}$, where $t_{i,j}$ is $o_{i,j}$'s relative position with respect to road e_i, $\lambda_{i,j}$ is the numeric attribute of interest. The query q can be projected on segment e_i and its relative position is denoted as $t_{i,q}$. We assume that a POI only belongs to one road segment, and the POIs are at different positions and have different numerical values.

3.2 Location-Based Skyline Query

Assuming that a lower numeric attribute (e.g., price) is preferable, we now give the definitions for spatial dominance and location-based skyline query.

Definition 1 (Distance). For any two POIs $o_{i,j}$ and $o_{i',j'}$ in one road segment e_i, the distance between $o_{i,j}$ and $o_{i',j'}$ is denoted as $d(o_{i,j}, o_{i',j'}) = |t_{i,j} - t_{i',j'}|$.

Definition 2 (Query distance). For a POI $o_{i,j}$ in road segment e_i, the query distance between query position q and POI $o_{i,j}$ is denoted as $d(q, o_{i,j}) = |t_{i,j} - t_{i,q}|$.

Definition 3 (Dominance). For any two POIs $o_{i,j}$ and $o_{i',j'}$ in one road segment e_i, we say $o_{i,j}$ spatially dominates $o_{i',j'}$ with respect to query position q if and only if $d(q, o_{i,j}) \leq d(q, o_{i',j'})$ and $\lambda_{i,j} \leq \lambda_{i',j'}$ but the two equalities do not both hold.

Definition 4 (Location-based skyline query). A location-based skyline query $sky(O|q)$ asks for the POIs that are not spatially dominated by any other POI in O with respect to q.

3.3 Problem Formulation

Assume a user submits a LBSQ query $\langle q, I \rangle$ to the CSP where $q \in \mathbb{R}^2$ is the query position and $I \subseteq \{1, \cdots, m\}$ is a set of indexes of road segments $E = \{e_1, \cdots, e_m\}$. After receiving $\langle q, I \rangle$ from the user, the CSP returns the results $sky(O|q)$, where $O = \bigcup_{i \in I} \mathcal{O}_i$.

The LBSP is assumed trusted; however, the CSP is considered untrusted due to a variety of reasons. For example, the CSP may modify LBSP's POI dataset, forge non-existent POI records, return some results that are not skyline records, or omit some skyline records.

Our security goal is to offer approaches for authenticating LBSQ queries. In our setting, we consider the CSP is dishonest and may present to the user a tampered result. Our proposed solutions can allow the user to verify the soundness and completeness of the query results.

Soundness: The user can verify that all qualifying data records returned are correct. They have not been tampered with nor have spurious data records been introduced.

Completeness: The user can verify that the results covers all the qualifying skyline POI records.

4 Basic Solution

In this section, we introduce the basic solution for verifiable LBSQ processing via an untrusted CSP.

4.1 Properties of LBSQ

We adopt Proposition 1 from [4].

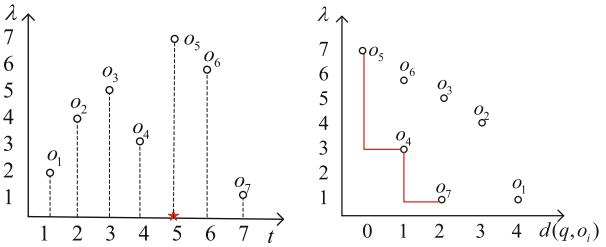
Proposition 1. Let $\mathcal{O}$ be the set of POIs distributed along road segment $e = \{\boldsymbol{u} + t\boldsymbol{v}\}$. For any query position $q \in \mathbb{R}^2$, we have

$$sky(\mathcal{O}|q) = sky(\mathcal{O}|t_q). \tag{1}$$

In other words, $sky(\mathcal{O}|q)$ is determined by the query q's relative position t_q, where t_q is defined as

$$t_q = \frac{\boldsymbol{v}^T(q - \boldsymbol{u})}{||\boldsymbol{v}||_2^2} \tag{2}$$

Figure 3(a) shows 7 POIs on a road segment, where all POIs are distributed along a road segment, and the x- and y-coordinates represent each POI's relative position and numeric attribute, respectively. Each POI o_i can be represented by its numerical value λ_i and relative position t_i. For example, $o_1 = (1, 2)$, where 1 and 2 are the o_1's relative position and numerical value, respectively. The query position's relative position is $t_q = 5$. Figure 3(b) shows the LBSQ results for q, where $d(q, o_i) = |t_i - t_q|$ represents the query distance, and the LBSQ results are $sky(\mathcal{O}|q) = \{o_5, o_4, o_7\}$. The other POIs are dominated by the POI in $sky(\mathcal{O}|q)$, for example, o_6 is dominated by o_4, o_3 is dominated by o_7.



(a) Sample POIs (b) LBSQ for query q

Fig. 3. LBSQ results.

We give a formal description of the properties of LBSQ. Without loss of generality, for a given query q, we assume that there are u results $\{o_1, o_2, \cdots, o_u\}$, where $o_i = (t_i, \lambda_i)$ is the ith result. Which are ordered by query distance $d(q, o_1) < d(q, o_2) < \cdots < d(q, o_u)$. The LBSQ results have the following properties:

- The LBSQ results can also be ordered based on the numerical attribute, $\lambda_1 > \lambda_2 > \cdots > \lambda_u$.
- o_1 should be a POI with the smallest query distance for all $o_i \in \mathcal{O}$.
- o_u should be the POI with the smallest numerical value for all $o_i \in \mathcal{O}$, and it has no skyline neighbors.
- For every adjacent pair (o_x, o_{x+1}), $x \in [1, u-1]$, o_{x+1} is the closest POI towards the query position in the subset under o_x, represented as $d(q, o_{x+1}) \leq d(q, o_j)$ for all o_j in $\mathcal{O}_x^- = \{o_i | \lambda_i < \lambda_x, o_i \in \mathcal{O}\}$.

Based on these properties, for each POI o_i, our method finds the first skyline result by finding the POI having the smallest query distance, and then continue

to find the next result, the next result's numerical value is smaller and its query distance is smallest in all candidate POIs.

In the following subsection, we define distance neighbor to find the closest POI to the query position, which has smallest query distance. We define skyline neighbor to find the next skyline result, which has smallest query distance in all POIs whose numerical value is smaller than the previous result.

4.2 Distance and Skyline Neighbor

Definition 5 (Distance neighbor). For any $o_i \in \mathcal{O}$, we define that its left (or right) distance neighbor with respect to relative position denoted by $N_l(o_i)$ (or $N_r(o_i)$), as the closest POI $o_j \in \mathcal{O}$ with $t_j < t_i$ (or $t_j > t_i$).

For each POI o_i, its distance neighbors are two POIs having smallest distance with o_i on its left and right sides. For each POI $o_i \in \mathcal{O}$, o_i's query distance is computed as $d(q, o_i) = |t_i - t_q|$.

We define the closest POI towards the query position on this segment as o_{min} if $d(q, o_{min}) = min\{d(q, o_i)|o_i \in \mathcal{O}\}$. The left and right distance neighbor of o_{min} are denoted as $N_l(o_{min})$ and $N_r(o_{min})$, respectively.

Note that only the closest POI o_{min}'s query distance is not larger than its both left and right distance neighbors' query distance for all POIs in the same road segment, which is described as $d(q, o_{min}) \leq d(q, N_l(o_{min}))$ and $d(q, o_{min}) \leq d(q, N_r(o_{min}))$.

For each POI o_i, we divide $\mathcal{O}$ into two subsets according to t_i as

$$\begin{aligned} \mathcal{O}_l^- &= \{o_j|o_j \in \mathcal{O}, \lambda_j < \lambda_i, t_j < t_i\} \\ \mathcal{O}_r^- &= \{o_j|o_j \in \mathcal{O}, \lambda_j < \lambda_i, t_j > t_i\} \end{aligned} \tag{3}$$

Definition 6 (Skyline neighbor). For any $o_i \in \mathcal{O}$, we define its left (or right) skyline neighbor in subset $\mathcal{O}_l^-$ (or $\mathcal{O}_r^-$), denoted by $N_l^-(o_i)$ (or $N_r^-(o_i)$), as the closest POI $o_j \in \mathcal{O}_l^-$ (or $o_j \in \mathcal{O}_r^-$) according to relative position.

For each POI o_i, its skyline neighbors are two POIs chosen from POIs with a numeric attribute smaller than o_i's numerical attribute, meanwhile the skyline neighbors have smallest distance with o_i on its left and right sides.

4.3 Data Preprocessing

The LBSP preprocesses its POI dataset $\mathcal{O} = \{o_i|1 \leq i \leq n\}$ before outsourcing it to the CSP, where $o_i = (t_i, \lambda_i)$. Without loss of generality, we assume that $t_1 < t_2 < \cdots < t_n$.

For every POI record $o_i, i \in [1, n]$, the LBSP computes the distance neighbors $N_l(o_i)$, $N_r(o_i)$ and skyline neighbors $N_l^-(o_i)$, $N_r^-(o_i)$. If the left or right neighbor does not exist, then assign null as its neighbor.

LBSP creates a signature for $o_i, i \in [1, n]$ by chaining o_i with its four neighbors:

$$\begin{aligned} s(o_i) = &Sig(H(H(o_i)|H(N_l(o_i))|H(N_r(o_i)) \\ &|H(N_l^-(o_i))|H(N_r^-(o_i)))) \end{aligned} \tag{4}$$

Here, $H(\cdot)$ is a hash function, and Sig is a signature generation algorithm. The total number of signatures is n. Then LBSP sends the POI dataset $\mathcal{O}$ and signatures S to the CSP.

4.4 Query Processing

Assume that the user issues an LBSQ $sky(\mathcal{O}|t_q)$. The CSP constructs the query result as follow.

- Compute t_q from q as in Eq. (1).
- For every $o_i \in \mathcal{O}$, find the closest POI point o_{min} with query position, which has the minimum distance $d(q, o_{min})$; Select o_{min} as the first skyline result, put o_{min} into $sky(\mathcal{O}|t_q)$, set the skyline result o_j equal to o_{min}. If there are two POIs having the minimum distance, then choose the one with a smaller numerical value.
- Put the skyline result o_j into $sky(\mathcal{O}|t_q)$, find the next skyline result from o_j's two candidate skyline neighbors; Select the POI with a smaller query distance from $N_l^-(o_j)$ and $N_r^-(o_j)$ and set it as the next result o_j. If two skyline neighbors have the same distances, then choose the one with a smaller numerical value, as the skyline neighbor with smaller numerical value dominates the other one.
- Repeat the previous step until o_j has no skyline neighbor.
- For each $o_i \in sky(\mathcal{O}|t_q)$, the CSP returns its neighbors $N_l^-(o_i)$, $N_r^-(o_i)$, $N_l(o_i)$ and $N_r(o_i)$ and its signature $s(o_i)$.

4.5 Query Result Verification

On receiving the query results from the CSP, the user verifies the results' authenticity and completeness. Without loss of generality, assume that the query results are $\{o_1, \cdots, o_u\}$, where $d(q, o_1) < d(q, o_2) < \cdots < d(q, o_u)$. The verification object contains all the signatures of results, $\{s(o_1), \cdots, s(o_u)\}$, and the distance and skyline neighbors of results, $\{N_l(o_1), N_r(o_1), N_l^-(o_1), N_r^-(o_1), \cdots\}$.

During authenticity verification, for each $x \in [1, u]$; since its neighbors are in the query result, the user uses them to compute its signature $s(o_x)$. If the query result is authentic, the user proceeds to check the completeness of the query result in the following three steps.

First, user checks whether o_1 is the closest POI to query position by checking if $d(q, o_1) \leq d(q, N_l(o_1))$ and $d(q, o_1) \leq d(q, N_r(o_1))$, as only the closest POI's distance to query position is not greater than its both neighbors' distance. If there are two same minimum distances, then check the numerical value. If $d(q, o_1) = d(q, N_l(o_1))$, then $\lambda_1 < \lambda(N_l(o_1))$. If $d(q, o_1) = d(q, N_r(o_1))$, then $\lambda_1 < \lambda(N_r(o_1))$. Second, user verifies that o_u is the last POI by checking whether o_u's numeric value λ_u is equal to λ_{min}.

Third, user checks every pair of adjacent POIs in $\{o_1, \cdots, o_u\}$ are indeed skyline neighbors of each other with respect to query position t_q using its neighbors. Specifically, for every $o_x, x \in [1, u-1]$, the user checks its next

neighbor o_{x+1} with its skyline neighbors $N_l^-(o_x)$ and $N_r^-(o_x)$. If $N_l^-(o_x)$ is equal to o_{x+1}, then check if $d(q, N_l^-(o_x)) < d(q, N_r^-(o_x))$, or if $d(q, N_l^-(o_x)) = d(q, N_r^-(o_x))$ and $\lambda(N_l^-(o_1)) < \lambda(N_r^-(o_1))$. If $N_r^-(o_x)$ is equal to o_{x+1}, then check if $d(q, N_r^-(o_x)) < d(q, N_l^-(o_x))$, or if $d(q, N_l^-(o_x)) = d(q, N_r^-(o_x))$ and $\lambda(N_r^-(o_1)) < \lambda(N_l^-(o_1))$. If any POI does not pass the verification, the query result is considered incomplete. If all the verifications succeed, the user considers the query result as complete and incomplete otherwise.

5 Performance Analysis

In this section, we study the performance of the proposed solutions through security, comparison with Chen's and overhead analysis.

5.1 Security Analysis

We prove that the proposed skyline query authentication scheme can achieve the security goals as follows. Let $sky(\mathcal{O}|t_q) = \{o_1, \cdots, o_u\}$ be the query results, where $d(q, o_1) < d(q, o_2) < \cdots < d(q, o_u)$.

We first discuss the case in which $sky(\mathcal{O}|t_q)$ is not sound: As for an adversary, in order to change the value of a record, he must be able to generate the corresponding signature. However, it is computationally infeasible without knowing the private key of LBSP.

Then we discuss three cases in which $sky(\mathcal{O}|t_q)$ is not complete:

Case 1: If the initial result o_1 is forged, then the adversary must forge a fake POI o_1', its distance neighbors $N_l(o_1')$ and $N_r(o_1')$ and its signature $s(o_1')$, which satisfy $d(q, o_1') < d(q, N_l(o_1'))$ and $d(q, o_1') < d(q, N_r(o_1'))$. However, there is only one record o_1 in the road segment which satisfies this requirement; it is computationally infeasible to compute $s(o_1')$ without knowing the private key of LBSP.

Case 2: The end result o_u is forged. The adversary must forge a fake POI o_u' whose numerical attribute is λ_{min} and its signature $s(o_u')$. It is computationally infeasible to compute $s(o_u')$ without knowing the private key of LBSP.

Case 3: Two contiguous records o_x and o_{x+1} in $sky(\mathcal{O}|t_q)$ are not skyline neighbors. Since every o_x, $x \in [1, u-1]$, its signature $s(o(x))$ contains its candidate skyline neighbors $\{N_l^-(o_x), N_r^-(o_x)\}$, the adversary cannot forge a fake POI $o_{x+1}' \notin \{N_l^-(o_x), N_r^-(o_x)\}$. Suppose $o_{x+1} = N_l^-(o_x)$, the user can further check if $d(q, o_{x+1}) < d(q, N_r^-(o_x))$. Any fake o_{x+1}' will be detected by the user.

5.2 Comparison with Chen's

We illustrate the processes of the benchmark method in [4] (denoted by Chen's) using Fig. 3(a) as an example.

In data preprocessing process, for each POI, the LBSP needs to issue a LBSQ to obtain the POI's skyline neighbor set, and each POI binds with its candidate skyline neighbors.

In query process, if $t_q = t_5$, the CSP issues three LBSQs and obtains results $sky(\mathcal{O}^-|t_q) = \{o_1, o_4, o_5\}$, $sky(\mathcal{O}^+|t_q) = \{o_6, o_7\}$ and $sky(\mathcal{O}|t_q) = \{o_4, o_5, o_7\}$. Then it returns $\bigcup_{o_i \in sky(\mathcal{O}^-|t_q) \cup sky(\mathcal{O}^+|t_q)} \mathcal{T}_i$, where $\mathcal{T}_i$ is the set of non-leaf nodes required along with the leaf node o_i to compute the Merkle root hash.

In query result verification process, for each $o_i \in \{o_1, o_4, o_5, o_6, o_7\}$, the user uses $\mathcal{T}_i$ to compute the Merkle root hash. The user also needs to check whether all the results are indeed skyline neighbors, in addition, it checks that every $o_i \in \{o_1, o_6\}$ is indeed dominated by some other returned POI, and every $o_i \in \{o_4, o_5, o_7\}$ is indeed not dominated by any other returned POI.

We can observe from the examples that Chen's methods have some limitations. Each POI record is bound with its candidate skyline neighbors which can up to $n-1$, while our method's neighbors is 4. Meanwhile, Chen's needs 3 skyline query in query process, while our method is much simpler, meanwhile our results and verification object are smaller than theirs.

5.3 Overhead

We analyze the overhead introduced by the proposed technique on the LBSP, the CSP, and the user side, respectively, and we compare our methods with Chen's methods.

(1) LBSP Overhead: In the data preprocess, Chen's methods need to generate skyline neighbor set and skyline neighbor range for each POI, which takes $O(n)$ skyline query operations, and the skyline query incurs high computation overhead. While our methods do not need to do the skyline query in data preprocess and our skyline neighbor generation process is simpler than Chen's methods.

In addition, in Chen's methods, the LBSP generates a MHT as the authentication structure, which needs 1 signature generation and $O(n)$ hash computation, then the LBSP outsources 1 signature and $O(n)$ digests to CSP. Our solution's main cost is related to the number of the signatures, which is proportional to the cardinality of POI dataset n. The computation and communication cost of our solution is $O(n)$. Since MHT and signature chain are two different authentication structures.

(2) CSP Overhead: In our methods, the CSP compares n POI's distance to the query position and chooses the minimum one as the first result. Then it finds the next skyline neighbor subsequently, which takes $o(n)$ comparisons. The remaining cost comes from constructing the verification object and sending it to users. The verification object contains $O(k)$ signatures and $4k$ neighbors, where k is the number of query results. In Chen's, the CSP needs to do 3 skyline query for n POI, which takes $o(n^2)$ comparisons. The verification object contains 1 signature and $k'logn$ digests, where k' is the number of query results, which is larger than k. Thus, Chen's query cost and VO size are larger than our solution.

(3) User Overhead: In our methods, user takes k signature verify operation in verification process, while Chen's takes 1 signature verify operation and $k'logn$ hash computations.

6 Conclusion

In this paper, we consider the problem of authenticating location-based skyline queries. We propose novel solutions that allow users to verify if the query results are sound and complete. By embedding skyline neighbors with each POI to its signature, our solutions achieve better performance in query process and communication overhead compared with the existing scheme. We prove that without knowing the private key of the data owner, it is computationally infeasible for an adversary to forge query results without being detected. Our extensive performance evaluation shows the proposed solutions are practical and can be used in real-world applications.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Grants 61632009 & 61472451, in part by the Guangdong Provincial Natural Science Foundation under Grant 2017A030308006 and High-Level Talents Program of Higher Education in Guangdong Province under Grant 2016ZJ01, in part by NSF and CSC grants CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, in part by the China Scholarship Council under Grant 201606370141, in part by Hunan Provincial Natural Science Foundation 2017JJ2333.

References

1. Zheng, B., Lee, K.C.K., Lee, W.-C.: Location-dependent skyline query. In: Proceedings of the 9th International Conference on Mobile Data Management, pp. 148–155 (2008)
2. Lee, K.C.K.: Efficient evaluation of location-dependent skyline queries using non-dominance scopes. In: Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications, p. 14 (2011)
3. Goncalves, M., Torres, D., Perera, G.: Making recommendations using location-based skyline queries. In: Proceedings of the 23rd International Workshop on Database and Expert Systems Applications (DEXA), pp. 111–115 (2012)
4. Chen, W., Liu, M., Zhang, R., Zhang, Y., Liu, S.: Secure outsourced skyline query processing via untrusted cloud service providers. In INFOCOM, pp. 1–9 (2016)
5. Devanbu, P., Gertz, M., Martel, C., Stubblebine, S.G.: Authentic data publication over the internet. J. Comput. Secur. **11**(3), 291–314 (2003)
6. Pang, H.H., Jain, A., Ramamritham, K., Tan, K.-L.: Verifying completeness of relational query results in data publishing. In: ACM SIGMOD, pp. 407–418 (2005)
7. Ku, W.-S., Hu, L., Shahabi, C., Wang, H.: Query integrity assurance of location-based services accessing outsourced spatial databases. In: Mamoulis, N., Seidl, T., Pedersen, T.B., Torp, K., Assent, I. (eds.) SSTD 2009. LNCS, vol. 5644, pp. 80–97. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02982-0_8
8. Yiu, M.L., Lo, E., Yung, D.: Authentication of moving KNN queries. In ICDE, pp. 565–576 (2011)
9. Yang, Y., Papadopoulos, S., Papadias, D.: Authenticated indexing for outsourced spatial databases. VLDB J. **18**(3), 631–648 (2009)
10. Hu, H., Xu, J., Chen, Q., Yang, Z.: Authenticating location-based services without compromising location privacy. In: ACM SIGMOD, pp. 301–312 (2012)

11. Zhang, R., Zhang, Y., Zhang, C.: Secure top-k query processing via untrusted location-based service providers. In: INFOCOM, pp. 1170–1178 (2012)
12. Chen, Q., Hu, H., Xu, J.: Authenticating top-k queries in location-based services with confidentiality. Proc. VLDB Endow. **7**(1), 49–60 (2013)
13. Zhang, R., Sun, J., Zhang, Y., Zhang, C.: Secure spatial top-k query processing via untrusted location-based service providers. IEEE Trans. Dependable Secure Comput. **12**(1), 111–124 (2015)
14. Yang, G., Cai, Y., Hu, Z.: Authentication of function queries. In: ICDE, pp. 337–348 (2016)
15. Zhu, X., Wu, J., Chang, W., Wang, G., Liu, Q.: Authentication of multi-dimensional top-k query on untrusted server. In: Proceedings of the IEEE/ACM International Symposium on Quality of Service (IWQoS) (2018)
16. Hu, L., Ku, W.-S., Bakiras, S., Shahabi, C.: Spatial query integrity with Voronoi neighbors. IEEE Trans. Knowl. Data Eng. **25**(4), 863–876 (2013)
17. Yiu, M.L., Lin, Y., Mouratidis, K.: Efficient verification of shortest path search via authenticated hints. In: ICDE, pp. 237–248 (2010)
18. Lin, X., Xu, J., Hu, H.: Authentication of location-based skyline queries. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 1583–1588 (2011)
19. Lin, X., Xu, J., Hu, H., Lee, W.-C.: Authenticating location-based skyline queries in arbitrary subspaces. IEEE Trans. Knowl. Data Eng. **26**(6), 1479–1493 (2014)
20. Liu, Q., Wu, S., Pei, S., Wu, J., Peng, T., Wang, G.: Secure and efficient multi-attribute range queries based on comparable inner product encoding. In: 2018 IEEE Conference on Communications and Network Security (CNS), pp. 1–9. IEEE (2018)
21. Zhang, S., Wang, G., Bhuiyan, Md.Z.A., Liu, Q.: A dual privacy preserving scheme in continuous location-based services. IEEE Internet Things J. **5**(5), 4191–4200 (2017)
22. Liu, Q., Wang, G., Li, F., Yang, S., Jie, W.: Preserving privacy with probabilistic indistinguishability in weighted social networks. IEEE Trans. Parallel Distrib. Syst. **28**(5), 1417–1429 (2017)
23. Zhang, Q., Liu, Q., Wang, G.: PRMS: a personalized mobile search over encrypted outsourced data. IEEE Access **6**, 31541–31552 (2018)
24. Zhu, X., Liu, Q., Wang, G.: A novel verifiable and dynamic fuzzy keyword search scheme over encrypted data in cloud computing. In: Trustcom/BigdataSE/ISPA, pp. 845–851 (2017)
25. Liu, Q., Wang, G., Liu, X., Peng, T., Wu, J.: Achieving reliable and secure services in cloud computing environments. Comput. Electric. Eng. **59**, 153–164 (2016)
26. Zheng, H., Chang, W., Wu, J.: Coverage and distinguishability requirements for traffic flow monitoring systems. In: IEEE/ACM 24th International Symposium on Quality of Service, pp. 1–10 (2016)
27. Chang, W., Wu, J.: Progressive or conservative: rationally allocate cooperative work in mobile social networks. IEEE Trans. Parallel Distrib. Syst. **26**(7), 2020–2035 (2015)
28. Lo, H., Ghinita, G.: Authenticating spatial skyline queries with low communication overhead. In: Proceedings of the third ACM Conference on Data and Application Security and Privacy, pp. 177–180 (2013)