Authenticating Top-k Queries in Location-based Services with Confidentiality

Qian Chen, Haibo Hu, Jianliang Xu

Department of Computer Science, Hong Kong Baptist University
Kowloon Tong, Hong Kong
{qchen, haibo, xujl}@comp.hkbu.edu.hk

ABSTRACT

State-of-the-art location-based services (LBSs) involve data owners, requesting clients, and service providers. As LBSs become new business opportunities, there is an increasing necessity to verify the genuineness of service results. Unfortunately, while traditional query authentication techniques can address this issue, they fail to protect the confidentiality of data, which is sensitive location information when LBSs are concerned. Recent work has studied how to preserve such location privacy in query authentication. However, the prior work is limited to range queries, where private values only appear on one side of the range comparison. In this paper, we address the more challenging authentication problem on top-k queries, where private values appear on both sides of a comparison. To start with, we propose two novel cryptographic building blocks, followed by a comprehensive design of authentication schemes for top-k queries based on *R*-tree and *Power Diagram* indexes. Optimizations, security analysis, and experimental results consistently show the effectiveness and robustness of the proposed schemes under various system settings and query workloads.

1. INTRODUCTION

The boom of smartphones brings prosperity to location-based services (LBSs) in almost all social and business sectors, such as geo-social networks, merchandizing, marketing, and logistics. While these LBSs drive new business opportunities, there is a rising necessity from the mobile users to verify the genuineness of service results, such as a list of recommended local restaurants sorted by location and user rating. This issue is even more critical in an outsourced model where businesses (or data owners) publish their data to a third-party service provider (SP), who handles LBS queries based on these data. As the SP is alleged to manipulate query results in favor of their "sponsors", to sustain growth amid fierce competition, it will soon be obliged to provide users not only the results, but also the proof of correctness.

In the spatial database literature, there are a lot of works on query authentication [25, 26, 28]. In these works, the data owner publishes not only data (e.g., spatial objects) to the SP, but also the endorsements of the data being published. These endorsements are signed by the data owner against tampering with by the SP. Given a query, the SP returns both the query results and a proof, called *verification object* (VO). In the verification phase, the querying client uses this VO, together with the query results, to reconstruct the endorsements and thus verify the correctness of the results.

However, one key limitation of all these works is that during the verification phase, the client is assumed to be completely trusted and entitled to receive any data values, even if they are not part of the results. Unfortunately, this assumption is flawed in LBSs whose data is often sensitive locations and should remain confidential against the client [12, 5, 23, 9]. For example, in online real-estate sites, the address of a property is often suppressed as business confidentiality. As another more recent example, due to tremendous concerns about privacy [24], Facebook reportedly pulled back the newly-launched "Find Friends Nearby" feature, which sends the user a list of recommended users according to their proximity. All these call for *privacy-preserving* query authentication techniques in LBSs that ensure the confidentiality of location data against the client.

In [7], we proposed privacy-preserving authentication for location-based range queries. Being the first work to address location privacy in authentication, the techniques cannot be applied to other queries. As location-based advertisement and recommendation are often recognized as one of the most profitable LBS businesses and thus provoke the greatest controversy with their ranking results [29], in this paper we study privacy-preserving authentication for *location-based top-k queries*, where the *rank* value of an object is a linear combination of distance penalty and non-spatial score (e.g., user average rating). This query definition is similar to [14] and is a generalization of various location-based top-k queries defined in [11, 29] and even the k-nearest neighbor (kNN) queries (by setting all non-spatial scores to 0).

The first challenge of privacy-preserving location-based top-k queries is its security model. Unlike a range query, the results of a top-k query imply the relative ranking of various objects. To address this, we introduce a formal security model based on the computational indistinguishability of relative *rank* values. Second, the major cryptographic challenge of this problem is comparing the rank values of two objects without disclosing their locations or scores, or in its primitive form, the distances of two private points from a query point. To this end, we design two new cryptographic building blocks, one with optimized online com-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Articles from this volume were invited to present their results at The 40th International Conference on Very Large Data Bases, September 1st - 5th 2014, Hangzhou, China.
Proceedings of the VLDB Endowment, Vol. 7, No. 1
Copyright 2013 VLDB Endowment 2150-8097/13/09... $ 10.00.

putation (the private-Paillier based method, or PPB) and the other with optimized offline computation (the pre-signed line based method, or PLB). Third, to address the performance challenge from a database perspective, we propose two authentication schemes for the R-tree and Power Diagram (a weighted form of Voronoi Diagram) based indexes, respectively. Each scheme consists of a complete set of authentication data structures, VO construction and client verification algorithms. In addition, we design strategies to optimize the offline computation and storage overhead for the PLB method. To summarize, our main contributions made in this paper are as follows:

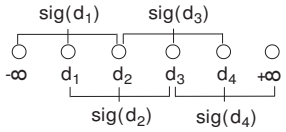
- To the best of our knowledge, this is the first work that addresses privacy-preserving top-k query authentication. Our query definition encompasses most existing location-based top-k and kNN queries.
- We introduce a formal security model and design two cryptographic building blocks (namely PPB and PLB) that can prove to the client the relation of rank values of two objects w.r.t. a query point without disclosing the locations and non-spatial scores.
- We develop a complete set of authentication schemes for both the R-tree and Power Diagram based indexes.
- We propose strategies for both the data owner and the SP to optimize the storage cost of the PLB method.
- We conduct extensive experiments and security analysis to evaluate the performance and robustness of the proposed authentication schemes.

The rest of this paper is organized as follows. Section 2 introduces the research background and related works in query authentication. Section 3 formally defines the problem and security model. Section 4 presents the two private ranking comparison methods, namely PPB and PLB. Section 5 presents the two authentication schemes based on the R-tree and Power Diagram indexes, followed by Section 6 where their security is analyzed. Section 7 studies the optimization strategies for the PLB comparison method. Section 8 shows the experimental results, followed by a conclusion.

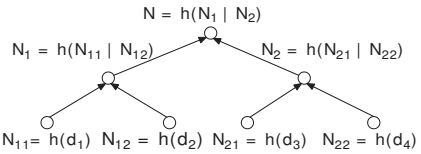
2. BACKGROUND AND RELATED WORKS

There is a large body of research works on query authentication for indexed data. These works originate from either digital signature chaining or Merkle hash tree.

Based on asymmetric cryptography, digital signature is produced by the message owner using encryption with its own private key. Then the verifier can verify the authenticity of a received message by the owner's public key and the signature. Based on this scheme, early works on query authentication impose a signature for every data value. The VB-tree [22] augments a conventional B^+-tree with a signature in each leaf entry. By verifying the signatures of all returned values, the client can guarantee the soundness of these results. However, the simple signature-based approach cannot guarantee the completeness, as the server can deliberately miss some results without being noticed. In [21], Pang et al. further proposed signature chaining, which connects a signature with adjacent data values to guarantee no result can be left out. Figure. 1(a) illustrates signature chaining for four sorted values d_1, d_2, d_3, d_4. The signature of each value depends not only on its own value but also on the immediate left and right values. Consider a range query which covers d_2 and d_3. When the server returns d_2 and d_3 to the client, it will also send a verification object (VO)



(a) Signature Chaining



(b) Merkle Hash Tree

Figure 1: Basic Authentication Tools

that contains: (1) the signatures of d_2 and d_3, and (2) the boundary values d_1 and d_4. Given the VO, the client can verify the query results through the facts that: (1) the two boundary values fall outside the query range, and (2) all signatures are valid. The first condition ensures no results are missing and the second guarantees no values are tampered with. Signature aggregation and chaining were adapted to multi-dimensional R-tree indexes by Cheng and Tan [3].

The Merkle hash tree (MHT) was introduced to authenticate a large set of data values [18]. Figure 1(b) shows an MHT for the same data values in Figure 1(a). It is a binary tree. Each leaf node with data value d_i is assigned a *digest* $h(d_i)$, where $h()$ is a one-way hash function. Each internal node N_i is assigned a digest which is derived from its child nodes, e.g., $N_1 = h(N_{11}|N_{12})$, where "|" denotes concatenation. In MHT, only the digest value of the root is signed by the data owner, and therefore it is more efficient than signature chaining schemes. An MHT can be used to authenticate any subset of data values. For example in Figure 1(b), the server sends d_1 and d_2 to the client; and to prove their authenticity, the server also sends the client a VO, which includes the digest of N_2 and the signed root digest N. The client computes $h(d_1)$ and $h(d_2)$, then $N_1 = h(h(d_1)|h(d_2))$, and finally $N = h(N_1|N_2)$. This computed root digest is then compared with the signed root digest in the VO. If they are the same, the client can verify that d_1 and d_2 are not tampered with by the server and no results are omitted.

The notion of MHT has been generalized to an f-way tree and widely adapted to various index structures. Typical examples include Merkle B-tree and its variant Embedded Merkle B-tree (EMB-tree) [15]. The latter reduces the VO size by embedding a tiny EMB-tree in each node. For multi-dimensional datasets and queries, similar techniques were proposed by Yang et al., who integrated an R-tree with the MHT (which is called Merkle R-tree or MR-tree) for authenticating multi-dimensional range queries [25, 26]. Besides selection and range queries, recent studies focus on the authentication of more complex query types, including top-k queries [4], kNN queries [28, 10], shortest paths [27], skyline queries [17], join queries [26], and aggregation queries [16].

Our work differs from all these works by preserving the data privacy against the verifier while accomplishing the same authentication task. In [7], we presented a solution for range queries, which is based on a cryptographic construct that can prove to the client that a private number is larger than a public number. However, this construct works for range queries only, and cannot be applied to any query

type that requires the comparison of two private numbers. Therefore, the design of new cryptographic constructs to prove the relation of two private numbers is a prerequisite for this work.

3. PROBLEM FORMULATION

Without loss of generality, we model a d-dimensional dataset $\mathbb{D}$ in an *integer*-domain space. For ease of presentation, we assume each object $p_i \in \mathbb{D}$ is a pair $\langle \lambda, \omega \rangle$, where λ is p_i's location vector and ω is its non-spatial score. The results of a top-k query $Q = \langle q, k \rangle$ (where q is the query point) are $R = \{r_1, r_2, \cdots, r_k\}$,[1] where r_i is the *id* of the i^{th} ranked object in $\mathbb{D}$ with respect to the following ranking function (known as *Euclidean scoring function* [1]):

$$rank(r_i, q) = \|r_i.\lambda - q.\lambda\|^2 + r_i.\omega^2,$$

where $\|r_i.\lambda - q.\lambda\|$ is the *Euclidean* distance between r_i and q, and an object with a lower value is ranked higher.[2] For ease of presentation, we omit explicit weights of the spatial distance and non-spatial score in the above linear combination. Nonetheless, the non-spatial score ω can be normalized in preprocessing to reflect its relative weighting. Figure 2 illustrates a top-3 query, whose results are $R = \{p_1, p_3, p_4\}$. The query Q is executed by the service provider (SP) on the dataset $\mathbb{D}$, which is authorized and signed by the data owner (DO). The authentication problem is for the querying client to verify that the SP executes Q faithfully in terms of two authenticity conditions: (1) *soundness* condition: the returned objects are all genuine top-k results and no returned *id*s are tampered with; (2) *completeness* condition: no genuine top-k results are missing. It is noteworthy that due to the nature of top-k queries, the completeness is implied by the soundness. The *privacy-preserving authentication problem* in this paper is to authenticate the top-k query results while guarding objects' location and score information against the client. That is, the client cannot infer any more information about the rank value of any object, beyond what is implied from the results.

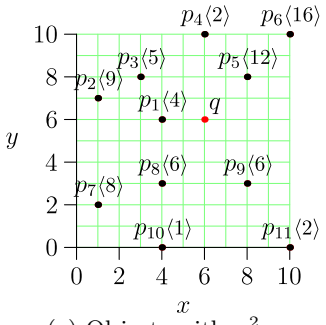
If privacy were not a concern, authenticating a top-k query would follow the following procedures. The SP returns a verification object (VO) to the client, along with the query results R. As a bottomline solution, the VO may include the location points and scores of all objects in the dataset $\mathbb{D}$ and a signature of $\mathbb{D}$. The querying client uses the VO to verify the soundness (and completeness) of the results by testing the following four conditions:

- None of the locations, scores, and *id*s of the result objects in R are tampered with;
- No locations and scores of the objects in $\mathbb{D} - R$ are missing and none of them are tampered with;
- All result objects are ranked no lower than r_k, i.e., $\forall r_i \in R$, $rank(r_i, q) \leq rank(r_k, q)$;[3]

[1]In a real location-based service, Q may return specific contents to the querying client, such as the users' names or their Facebook pages. We assume these contents can be retrieved faithfully using the returned *id*s.

[2]This definition slightly differs from a sum ranking function: $rank(r_i, q) = ||r_i.\lambda - q.\lambda|| + r_i.\omega$. Nonetheless, we show in [2] that the top-k results of such a ranking function can be derived from top-k' results of our ranking function.

[3]According to our security model, the client cannot learn the order of top-k results.



(a) Objects with ω^2

object	ω^2	ranking value
p_1	4	8
p_2	9	35
p_3	5	18
p_4	2	18
p_5	12	20
p_6	16	48
p_7	8	49
p_8	6	19
p_9	6	19
p_{10}	1	41
p_{11}	2	54

(b) Ranking Table

Figure 2: Top-k Query Example

- All non-result objects are ranked no higher than r_k, i.e., $\forall r_i \in \mathbb{D} - R$, $rank(r_i, q) \geq rank(r_k, q)$.

Verifying the latter two conditions without disclosing object locations or scores requires a private ranking comparison of two objects, which will be studied in Section 4. Furthermore, to avoid enumerating all objects in $\mathbb{D} - R$ when verifying the second and fourth conditions and thus minimize the VO size, we will propose authentication schemes on two common spatial indexes in Section 5.

3.1 Security Model

As with the previous works [7, 15, 25], we assume that: (1) the DO is trusted by (but not colluding with) the querying client or SP; (2) the SP has read-only access to the object locations, scores, and the query point; (3) all parties follow a semi-honest model [13]. The security model to achieve in this problem is two-folded: (1) the location and score information of every object is *semantic secure* against the client; and (2) the results of a query are both complete and sound. The former can be derived from the semantic security of rank value as follows [13]:

DEFINITION 3.1. **Semantic Security of Rank Values.** *Given a query $Q = \langle q, k \rangle$, a scheme is semantic secure for rank values against a probabilistic polynomial-time client, if given any two result objects (resp. non-result objects) $s, t \neq r_k$, where $rank(s,q) \leq rank(t,q) \leq rank(r_k,q)$ (resp. $rank(r_k,q) \leq rank(s,q) \leq rank(t,q)$), the client can succeed in deriving $rank(s,q) \leq rank(t,q)$ with probability at most negligibly greater than $1/2$.*

For a single query, the semantic security of rank values immediately lends us that of their ingredients — the locations and non-spatial scores. However, when the client issues queries continuously, this no longer holds, as the query points that dominate the change of a rank value are known. The semantic security model for continuous queries will be introduced in Section 6.

4. PRIVATE RANKING COMPARISON

In this section, we propose two primitive methods for the client to privately compare $rank(s, q)$ and $rank(t, q)$ without knowing the locations and scores of objects s, t. These two methods form the basic cryptographic constructs of privacy-preserving top-k query authentication schemes in Section 5.

4.1 Private-Paillier based (PPB) Method

The main idea of this method is to apply encryption on objects. To enable the ranking function on cipher-texts, we adopt the Paillier homomorphic cryptosystem [20].

4.1.1 Introduction to Paillier Cryptosystem

Paillier is a public-key homomorphic cryptosystem that satisfies additive homomorphism, which means that one can compute a cipher-text of m_1+m_2 by only having the public key and cipher-texts of m_1 and m_2.

Paillier has the following properties [20]:

$$\forall m_1, m_2 \in \mathbb{Z}_n, \forall z_1, z_2 \in \mathbb{Z}_n^*,$$

$$E(m_1, z_1) \cdot E(m_2, z_2) \equiv E(m_1 + m_2, z_1 \cdot z_2) \mod n^2 \quad (1)$$

$$E(m_1, z_1)^{m_2} \equiv E(m_1 \cdot m_2, z_1^{m_2}) \mod n^2 \quad (2)$$

$$E(m_1, z_1) \equiv E(m_1 + k \cdot \phi(n^2), z_1) \mod n^2, k = 1, \cdots \quad (3)$$

where $\mathbb{Z}_n^*$ is the subset of $\mathbb{Z}_n$ (i.e., 0, 1, $\cdots$, $n-1$) whose elements have multiplicative inverses (modulo n), z_1, z_2 are random values, $E()$ is the encryption function of the Paillier cryptosystem, and $\phi()$ is the Euler totient function. Eqn. 3 implies that it is hard to find a collision for any plain-text m, due to the difficulty of computing $\phi(n^2)$.

4.1.2 Private Ranking Comparison with PPB Method

Proving $rank(s,q) \geq rank(t,q)$ is equivalent to proving

$$rank(s,q) - rank(t,q) = \delta, \text{ where } \delta \geq 0. \quad (4)$$

Let x_p, y_p, ω_p denote the x, y coordinate and score of an object p, respectively. By expanding $rank(s,q)$ and $rank(t,q)$, we rewrite Eqn. 4 as follows:

$$2x_tx_q+2y_ty_q+x_s^2+y_s^2+\omega_s^2 = 2x_sx_q+2y_sy_q+x_t^2+y_t^2+\omega_t^2+\delta.$$

If both sides of this equation are encrypted by Paillier, according to Eqn. 1 and 2, it is equivalent to proving the following equation instead:[4]

$$E(2x_t)^{x_q}E(2y_t)^{y_q}E(x_s^2)E(y_s^2)E(\omega_s^2) = E(2x_s)^{x_q}E(2y_s)^{y_q}E(x_t^2)E(y_t^2)E(\omega_t^2)E(\delta) \mod n^2. \quad (5)$$

In Eqn. 5, except for x_q, y_q, $E(\delta)$ (which is computed by the SP as shown below), all items can be precomputed and signed by the DO offline. And since only the DO possesses the private key of Paillier, these items cannot be decrypted by the client. Thus, the client can verify Eqn. 5 without knowing the coordinates or scores of s, t.

Verifying Eqn. 5 only proves that $E(\delta)$ is the true encrypted difference of rank values, and the client is yet to verify (without knowing δ itself) that $\delta \geq 0$. We propose a novel method called *encrypted seeds decomposition*. The key observation is that proving $\delta \geq 0$ is equivalent to showing that δ has a *canonical* decomposition of base B:[5]

$$\delta = \sum_{i=1}^{m} \delta_i \cdot B^i, \quad (6)$$

where $\delta_i \in [0, 1, ..., B-1]$, $m = log_B(U)$, U is the upper bound of δ. Applying Eqn. 1 to Eqn. 6, we get:

$$E(\delta) = \prod_{i=1}^{m} E(\delta_i \cdot B^i) \mod n^2. \quad (7)$$

[4]By the definition of Paillier encryption, Eqn. 5 holds when the following two conditions are satisfied. First, the random values of z for x_s and y_s (resp. x_t and y_t) are the same. Second, the client knows the random value of z for δ and can thus multiply some constants to balance Eqn. 5.

[5]If $\delta < 0$, according to Eqn. 3, the server cannot find another $\delta' \geq 0$ such that $E(\delta') = E(\delta)$.

We call these $E(\delta_i \cdot B^i)$ "encrypted seeds" and let the DO pre-sign them. Once the client receives the corresponding encrypted seeds for δ from the SP, it can verify that $\delta \geq 0$ by assembling a verified $E(\delta)$ using Eqn. 7.

The following is the whole Private-Paillier based ranking comparison procedure. During the service initialization, the DO sends the following to the SP: (1) all signed encrypted seeds $E(\delta_i \cdot B^i)$ $(i = 1, 2, \cdots, m, \delta_i \in [0, B-1])$, and (2) $E(2x)$, $E(2y)$, $E(x^2)$, $E(y^2)$, and $E(\omega^2)$ for every object. Upon a comparison request $rank(s,q) \geq rank(t,q)$, besides sending $E(2x)$, $E(2y)$, $E(x^2)$, $E(y^2)$ and $E(\omega^2)$ of s and t, the SP also sends $E(\delta_i \cdot B^i)$ with their signatures to the client. By assembling $E(\delta)$ using Eqn. 7 and testing if Eqn. 5 holds, the client can verify that $rank(s,q) \geq rank(t,q)$. A rigorous proof of its security is given in Section 6.1.

As for cost analysis, the communication cost M_{PPB} and the total client CPU cost C_{PPB} are:

$$M_{PPB} = (10+m) \cdot M_{enc} + M_{sign},$$
$$C_{PPB} = (3m+9) \cdot C_{mul} + C_{sign},$$

where M_{enc} and M_{sign} are the lengths of a Paillier cipher-text and a signature, respectively, C_{mul} and C_{sign} are the CPU costs of a modular multiplication and a signature verification, respectively (see [2] for detailed analysis).

4.2 Pre-signed Lines based (PLB) Method

While the PPB method can compare the ranking privately for any arbitrary pair of objects, the extensive use of homomorphic functions results in costly computation. In this subsection, we propose an alternative method where the DO pre-computes and pre-signs the ranking comparison result for a selected pair of objects.

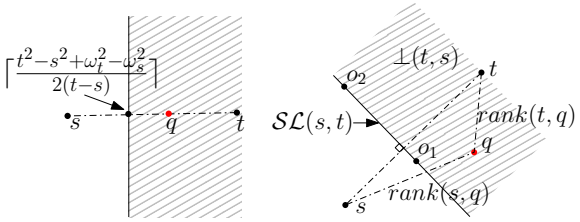
4.2.1 Preliminary — 1D Case

Here we assume s, t and q are all 1D points. As shown in Figure 3(a), $rank(s,q) \geq rank(t,q)$ if and only if q is to the right side of $\frac{t^2-s^2+\omega_t^2-\omega_s^2}{2(t-s)}$. In other words, the ranking value comparison is reduced to comparing q with a private value $\frac{t^2-s^2+\omega_t^2-\omega_s^2}{2(t-s)}$, or in the integer form $q \geq \lceil \frac{t^2-s^2+\omega_t^2-\omega_s^2}{2(t-s)} \rceil$. To verify $q \geq \alpha$ without the client knowing the value of α, we adopt the method in [21]. The idea is to let the client and SP jointly compute the digest g of value $U-\alpha$, where U is the upper bound of the domain of α. The SP first computes $g(q-\alpha)$ and sends it to the client, who then computes $g(U-\alpha) = g(U-q) \otimes g(q-\alpha)$, where $\otimes$ is a well-defined operation on the digest. Note this equation is guaranteed by the homomorphic property of the digest function $g()$, which accepts only non-negative numbers. As such, by sending $g(q-\alpha)$, the server claims $q \geq \alpha$. The client verifies $q \geq \alpha$ by comparing the jointly computed $g(U-\alpha)$ value with the $g(U-\alpha)$ value signed by the DO.

4.2.2 Private Ranking Comparison with PLB method

When s, t, and q are 2D points, we propose a geometric approach that can reduce the 2D ranking comparison to a 1D value comparison as outlined above. First, we introduce the notion of score-shifted half-plane.

DEFINITION 4.1. ***Score-Shifted Half-Plane***. *Given objects s and t, the score-shifted half-plane $\perp(t,s)$ (the shaded*



(a) 1 Dimension (b) 2 Dimensions

Figure 3: Ranking Comparison in Different Dimensions

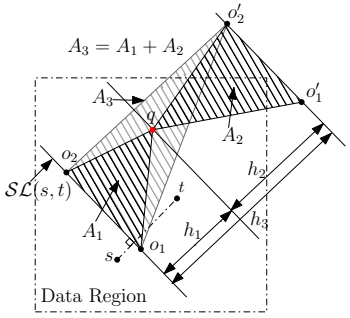


Figure 4: Pre-signed Line Based Method

part in Figure 3(b)) is defined as the set of points that have a lower rank value to t than to s:

$$\perp(t,s) = \{p \in \mathbb{R}^2 \mid rank(t,p) \leq rank(s,p)\}.$$

The line to which s and t have equal rank values is called the **score-shifted line** *of s and t, denoted as $\mathcal{SL}(s,t)$. As its name suggests, we can find it by shifting the 2D perpendicular bisector of s and t by their score difference:*

$$\mathcal{SL}(s,t) = \{p \mid (t-s)^T p = \frac{1}{2}(\|t\|^2 - \|s\|^2 + \omega_t^2 - \omega_s^2)\}.$$

From Figure 3(b), verifying $rank(s,q) \geq rank(t,q)$ is equivalent to verifying $q \in \perp(s,t)$. And from Figure 4, verifying the latter is equivalent to verifying the "directed" area of $\triangle qo_2o_1$ is non-negative, where o_1, o_2 are two arbitrary points on $\mathcal{SL}(s,t)$. However, sending o_1 and o_2 to the client will disclose $\mathcal{SL}(s,t)$ and thus the coordinates of s and t. To avoid this, the SP takes the following approach. First, it finds another line $l(o'_1, o'_2)$ parallel to $l(o_1, o_2)$ and so far away that it does not intersect with the data region. Further, let $\|o'_1o'_2\| = \|o_1o_2\|$. Let A_1 denote the directed area of $\triangle qo_2o_1$, A_2 the directed area of $\triangle qo'_1o'_2$, and A_3 the directed area of $\triangle o'_2o_2o_1$, we have

$$\begin{aligned} A_3 &= \tfrac{1}{2}\|o_1o_2\| \cdot h_3 = \tfrac{1}{2}\|o_1o_2\| \cdot (h_1 + h_2) \\ &= \tfrac{1}{2}\|o_1o_2\| \cdot h_1 + \tfrac{1}{2}\|o'_1o'_2\| \cdot h_2 = A_1 + A_2. \end{aligned} \quad (8)$$

The above equation resembles the 1D case where the digest value $g(U-\alpha)$ is jointly computed by the SP (for digest $g(q-\alpha)$) and the client (for digest $g(U-q)$). Here by analogy, $g(A_1)$ is computed by the SP (because it involves private points o_1 and o_2), $g(A_2)$ is computed by the client based on o'_1, o'_2 and q,[6] and $g(A_3)$ does not involve q and can thus be pre-computed and signed by the DO. The client verifies $A_1 \geq 0$ (and thus $rank(s,q) \geq rank(t,q)$) by comparing the jointly computed $g(A_3)$ value with the $g(A_3)$ value signed by the DO. If they are the same, the client can verify $A_1 \geq 0$. More rigorous security analysis is given in Section 6.1.

As for cost analysis, the communication cost M_{PLB} and the total client CPU cost C_{PLB} are:

$$M_{PLB} = m/2 + (m + 4 + \lceil log_2 m \rceil) \cdot M_{digest} + M_{sign},$$
$$C_{PLB} = (B(m+1) + \lceil log_2 m \rceil + 2) \cdot C_{hash} + C_{sign},$$

where M_{sign}, and M_{digest} are the lengths of a signature and a digest, respectively, C_{sign} and C_{hash} are the CPU costs of a signature verification and a hash operation, respectively (see [2] for detailed analysis).

5. AUTHENTICATING TOP-K QUERIES WITHOUT COMPROMISING PRIVACY

Equipped with the PPB and PLB methods on private ranking comparison, in this section we study privacy-preserving authentication of top-k queries on a dataset $\mathbb{D}$. Recall that $R = \{r_1, r_2, \cdots, r_k\}$ are the results, and the authentication verifies the following conditions: (1) $\forall r_i \in R$, $rank(r_i, q) \leq rank(r_k, q)$, and (2) $\forall r_i \in \mathbb{D} - R, rank(r_i, q) \geq rank(r_k, q)$. As with all existing authentication techniques, we assume that the authentication is carried out on a spatial index. In this paper, we focus on R-tree and Power Weighted Voronoi Diagram based authentication schemes. Each scheme consists of the offline construction of the authenticated data structure (ADS), the online construction of the VO for a query, and the client verification procedure.

5.1 Authentication on MR-tree

If we upscale a 2D object $p(\lambda, \omega)$ into a 3D point $p'(x_p, y_p, \omega)$ and the query point q into $q'(x_q, y_q, 0)$, the rank value of p w.r.t. q is the (squared) Euclidean distance of p', q':

$$\|p' - q'\|^2 = (x_p - x_q)^2 + (y_p - y_q)^2 + (\omega - 0)^2 = rank(p,q).$$

Let these 3D points be indexed by an R-tree, and **the original top-k query on objects p_i's with scores is reduced to a kNN query on all p'_i points**. For ease of presentation, we omit the upscale sign $'$ when the context is clear. In the rest of this subsection, we first introduce the general framework on Merkle R-tree based kNN authentication without privacy-preserving requirements, and then present our privacy-preserving scheme.

5.1.1 Preliminary — Merkle R-tree and kNN Query Authentication in 3D

Merkle R-tree (MR-tree) is an integration of R*-tree and Merkle Hash tree (MHT) [25, 26]. Figure 6(a) shows an MR-tree for the data objects in Figure 5. Every entry N_i in a non-leaf node has a minimum bounding box (MBB) (denoted by $N_i.mbb$) and a digest for its child entries (denoted by H_i), while every leaf entry p_i has a corresponding data object (denoted by $p_i.p$) and a digest of its *id* (denoted by h_i). Inspired by MHT, the digest of a non-leaf entry is the hash value of the concatenation of all its child entries' MBBs (or objects) and their digests, and the digest of a leaf entry is simply the hash value of its object *id*. For example, in Figure 6(a), for non-leaf entry N_1, its digest $H_1 = h(p_1.p|h_1|p_2.p|h_2|p_3.p|h_3)$; for leaf entry p_1, its digest $h_1 = h(p_1.id)$. The digests of all entries in the MR-tree are recursively computed in a bottom-up fashion, and the digest of the root entry is signed by the DO using its private key.

The kNN query processing can be conducted by existing algorithms such as the *best-first search* [9]. This algorithm

[6] In order not to disclose o'_1 and o'_2 to the client, A_2 is further encrypted. More details about this and the selection of points (o_1, o_2) and (o'_1, o'_2) are in the full technical report [2].

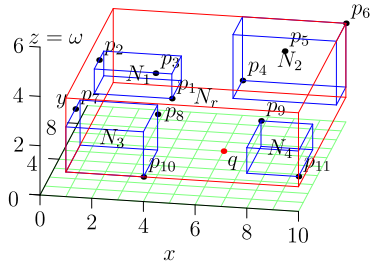
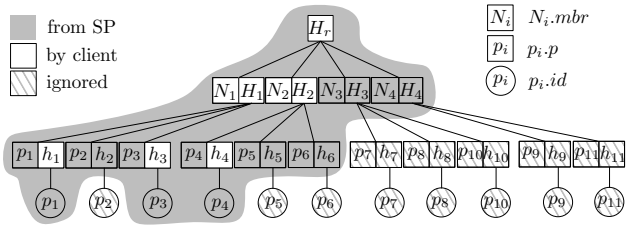
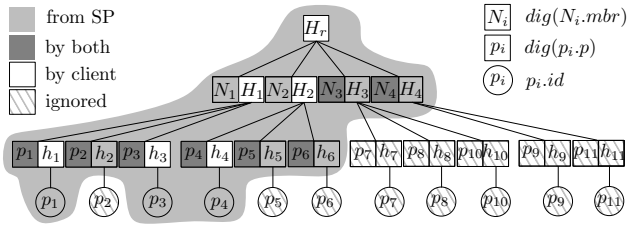


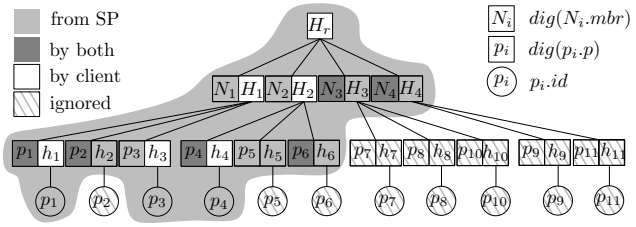
Figure 5: Nodes, Objects, and Query



(a) MR-tree Index and VO (without privacy-preserving requirement)



(b) MR-tree Index and VO (applying PPB method only)



(c) MR-tree Index and VO (accelerated with PLB method)

Figure 6: Query Authentication on MR-tree

maintains a priority queue H of to-be-explored nodes, sorted by their minimum rank value w.r.t. q, and repeatedly pops up the top entry e in H, accesses e for its child entries, and enqueues them into H. This procedure terminates when k leaf entries (i.e., result data objects $r_1, r_2, \cdots, r_k$) have been popped up from H. We denote the remaining entries in the heap H as H_{rm}.

To authenticate the query results, the client needs to verify: (1) $\forall r_i \in R, rank(r_i, q) \leq rank(r_k, q)$; (2) $\forall r_i \in H_{rm}$, $rank(r_i, q) \geq rank(r_k, q)$; and (3) no r_i is omitted or tampered with. (1) can be verified directly from the result objects, and (2) can be verified if H_{rm} is included in the VO. (3) can be verified by restoring the root digest of the MR-tree, because any missing or wrong r_i will result in an incorrect restoration of the root digest. Therefore, without privacy requirements, the VO should include: (1) the result objects in R and the MBBs (or objects) of the entries in H_{rm}; (2) the signed root digest; and (3) the digest components necessary for the client to restore the root digest.

Consider a 3NN query example in Figure 5, where $N_1, N_2, \cdots$ are non-leaf entries and $p_1, p_2, \cdots$ are leaf entries. The query results are $R = \{p_1, p_3, p_4\}$ and $H_{rm} = \{p_5, p_2, p_6, N_3, N_4\}$. Figure 6(a) shows the VO for this example, which includes:

- the objects and MBBs in $R \cup H_{rm}$, including: (1) p_1, p_2, p_3, p_4, p_5, p_6 and (2) the MBBs of N_3, N_4;
- the signed digest of the root node;
- all the digest components, necessary for the client to compute the root digest, including (1) the digests h_2, h_5, h_6 for leaf entries p_2, p_5, p_6; and (2) the digests H_3, H_4 for non-leaf entries N_3, N_4.

In Figure 6(a), all items returned by the SP are shown in light-grey color (e.g., h_2, H_3), and all the digests, computed by the client itself after receiving the VO (e.g., h_1, H_1), are shown in white color.

5.1.2 Private Ranking Comparison between an MBB and an Object

As outlined above, the authentication of kNN queries involves ranking comparisons not only between objects but also between objects in an MBB and an individual object. Specifically, if the minimum rank value of objects in an MBB w.r.t. the query point q is larger than the rank value of the top-k^{th} object r_k, the corresponding MR-tree node can be verified as a whole, without accessing its child entries. Since the PPB or PLB method only compares between objects, we present the following private ranking comparing method between an MBB and an object.

Since q is always 0 in the z-axis and any object has a positive ω value, the minimum projected rank value of an MBB on z-axis is simply ω^2_{min}, the minimum ω^2 in this MBB. In what follows, we only need to obtain the minimum projected rank value on x- and y-axis. Let p_1, p_2, p_3, p_4 denote the four corner points, and l_1, l_2, l_3, l_4 the four boundary lines of the projected MBB M. Obviously, if q locates inside M, the minimum projected rank value on x- and y-axis is 0. If q locates outside of M, there will be two cases. The first case is when q locates in Partitions I, III, VII, and IX (see Case A in Figure 7), and the minimum rank value w.r.t. q occurs on one of the corner points p. In this figure, $p = p_4$. So the proof of $rank(M, q) \geq rank(r_k, q)$ can be reduced to proofs of: (1) $x_q \geq x_{p_4}$ and $y_q \leq y_{p_4}$ using the 1D private comparison method [21], and (2) $rank(p, q) \geq rank(r_k, q)$ using the PPB method.

The second case is when q locates in Partitions II, IV, VI, and VIII (see Case B in Figure 7), and the minimum rank value w.r.t. q occurs on one of the boundary lines. In this figure, the minimum rank value w.r.t. q occurs on line l_4 (in between lines l_1 and l_2). Let p denote this closest point; then the proof of $rank(M, q) \geq rank(r_k, q)$ can be reduced to proofs of: (1) $x_q \geq x_{p_4}$ and $y_{p_4} \leq y_q \leq y_{p_2}$ using the 1D private comparison method [21], and (2) $rank(p, q) \geq rank(r_k, q)$ using the PPB method.[7]

5.1.3 Authenticated Data Structure

Now we present the privacy-preserving authentication scheme on the MR-tree, starting with the authenticated data structure. First, we define the digest for a leaf entry p_i, as $h_i = h(p_i.id)$. Since we cannot disclose any coordinates or scores of $p_i.p$ to the client, we define its digest $dig(p_i.p)$ based on the PPB method as follows:

$$dig(p_i.p) = h(E(2x_{p_i})|E(2y_{p_i})|E(x^2_{p_i})|E(y^2_{p_i})|E(\omega^2_{p_i})). \quad (9)$$

[7] In this case, the y coordinate of p equals that of q, whose digest can be computed by the client locally.

The digest of a non-leaf entry N_i follows the same definition as in [7] as:

$$H_i = h^2(dig(N_{c_1}.mbb)) \cdot h^2(dig(N_{c_1})) \cdots \\ \cdot h^2(dig(N_{c_m}.mbb)) \cdot h^2(dig(N_{c_m})) \mod n, \quad (10)$$

where N_{c_j} is N_i's j^{th} child entry, $n = uv$ and u, v are two large primes. The digest of an MBB is defined as:

$$dig(mbb) = h^2(dig(mbb.\mathbf{l})) \cdot h^2(dig_g(mbb.\mathbf{l})) \\ \cdot h^2(dig(mbb.\mathbf{u})) \cdot h^2(dig_g(mbb.\mathbf{u})) \mod n, \quad (11)$$

where $mbb.\mathbf{l}$ and $mbb.\mathbf{u}$ are bottom-left and top-right corner points and $dig_g()$ is the digest of a corner point's $g()$ values:

$$dig_g(p) = h(g(x_p - L)|g(U - x_p)|g(y_p - L)|g(U - y_p)), \quad (12)$$

which is used for boundary verification during ranking value comparison between an MBB and an object. Since q always locates under the MBBs, the z-axis value (i.e., ω) is not included in the digest $dig_g()$.

5.1.4 VO Construction and Verification

Recall that R denotes the set of query results and H_{rm} denotes the remaining entries in the priority queue H. Similar to Section 5.1.1, to authenticate the results, the client needs to verify: (1) $\forall r_i \in R, rank(r_i, q) \leq rank(r_k, q)$; (2) $\forall r_i \in H_{rm}, rank(r_i, q) \geq rank(r_k, q)$; and (3) no $r_i \in R \cup H_{rm}$ is omitted or tampered with. While (3) can still be verified by restoring the root digest of the MR-tree as in Section 5.1.1, since neither the points nor their rank values can be disclosed to the client, verifying (1) and (2) is no longer trivial and requires the PPB comparison on two objects or on an MBB and an object. Therefore, the VO includes: (1) the digest or digest components of each $r_i \in R$ to privately compare with $rank(r_k, q)$; (2) the digest or digest components of each $r_i \in H_{rm}$ to privately compare with $rank(r_k, q)$; (3) the signed root digest; and (4) all the digest components necessary for the client to restore the root digest.

Figure 6(b) shows the VO of the same 3NN query as in Figure 5, which includes:

- the digest components for each $r_i \in R$ to privately compare with $rank(r_k, q)$, including: $E()$ values for objects p_1, p_3, p_4;[8]
- the digest components for each $r_i \in H_{rm}$ to privately compare with $rank(r_k, q)$, including: (1) $E()$ values for objects p_2, p_5, p_6; (2) $E()$ values, $g()$ values and components for the corner points of MBBs N_3, N_4;
- the signed digest of the root node;
- the digest components, necessary for the client to compute the root digest, including: (1) the digests h_2, h_5, h_6 for leaf entries p_2, p_5, p_6; (2) the digests H_3, H_4 for non-leaf entries N_3, N_4; and (3) the digests $dig(N_1.mbb)$, $dig(N_2.mbb)$.

In Figure 6(b), the light-grey and white colors mean the same in Figure 6(a), and the dark-grey color represents those digest components that are jointly computed by the SP and the client. For example, $dig(p_1.p)$ is computed by the client based on the $E()$ values of $p_1.p$ returned from the SP.

[8] $E()$ values consist of $E(2x)$, $E(2y)$, $E(x^2)$, $E(y^2)$, $E(\omega^2)$ and $\{S(E(\delta_i \cdot B^i))\}$.

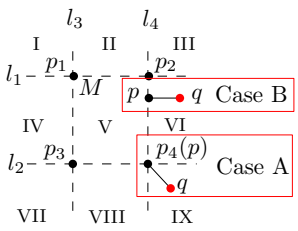


Figure 7: Ranking Comparison between an MBB and an Object

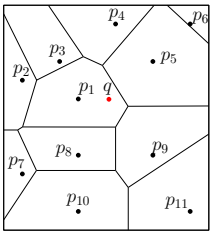


Figure 8: Power Diagram and Query

5.1.5 Acceleration using PLB Method

If the DO pre-signs some pairs of objects in advance, the SP can replace some PPB calls with PLB calls to accelerate the authentication. In the example of Figure 5, we assume that $\mathcal{SL}(p_3, p_4)$ and $\mathcal{SL}(p_4, p_5)$ are pre-signed by the DO, comparisons of $rank(p_4, q) \geq rank(p_3, q)$ and $rank(p_5, q) \geq rank(p_4, q)$ can then be verified using the PLB method. Specifically, the SP no longer returns $E()$ values for p_3 and p_5. Instead, it returns digest components of o'_1, o'_2, $g(\lfloor A_1 \rfloor)$ and signatures of those pre-signed lines. Figure 6(c) shows the updated VO with this PLB acceleration, where the digests of entries p_3 and p_5, jointly computed by the SP and the client in Figure 6(b), are now returned directly.

5.2 Authentication on the Power Diagram

In the MR-tree based scheme, since only the root digest is signed, the verification of any query must go all the way up to the root. This incurs a significant number of necessary digests or digest components to be included in the VO, particularly unfavorable to queries of small k. As an extreme example, when $k = 1$, even though the result comprises only one data object, the VO still includes the digests of all neighbor objects in the same leaf node, of all neighbor entries in the parent node, and so on. In this subsection, we propose an alternative scheme that is based on the Power Diagram [19], a weighted form of Voronoi Diagram.

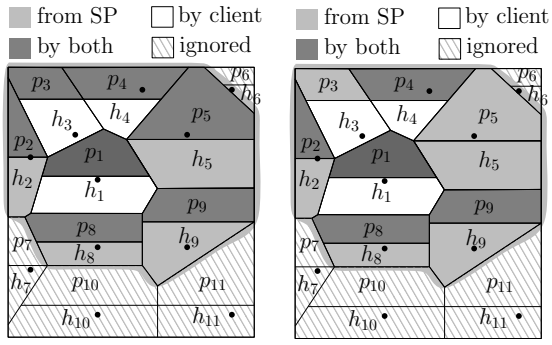
5.2.1 Properties of Power Diagram

As shown in Figure 8, given the set of objects $\{p_1, p_2, \cdots\}$ in $\mathbb{D}$, the *Power Diagram* of $\mathbb{D}$, denoted by $\mathcal{PD}(\mathbb{D})$, partitions the *Euclidean* space $\mathbb{R}^2$ into disjoint *Power Voronoi Cells*. Each cell corresponds to one object p_i, denoted by $\mathcal{VC}(p_i)$. If a query point q locates in this cell, p_i is its top-1 result. Similar to Voronoi Diagram, a 2D Power Diagram can be constructed in $O(n \log n)$ time [19]. The difference, however, is that p_i itself may not locate in its own cell, and some objects even have no corresponding cells.

To get the remaining top-k results, we augment the non-spatial score ω to a third dimension and get a 3D Voronoi Diagram. If p_i and p_j share a common face in it, we say p_j is a *Voronoi neighbor* of p_i, denoted as $\mathcal{VN}(p_i)$. Voronoi Diagram has the following property on finding the top-k^{th} object based on the top-$(k-1)$ objects [19]:

PROPERTY 5.1. *If $P = \{r_1, r_2, \cdots, r_{k-1}\} \subset \mathbb{D}$ are the top-$(k-1)$ objects of a query point q, the top-k^{th} object (i.e., r_k) must be in the set $\bigcup \mathcal{VN}(r_i \in P)$.*

5.2.2 Authenticated Data Structure



(a) Basic (b) PLB Accelerated

Figure 9: VO and Authentication on Power Diagram

In the Power Diagram based scheme, the DO signs the digest of each data object p_i, which is defined as:

$$dig(p_i) = dig(p_i.p) \cdot h^2(p_i.id) \cdot h^2(p_i.neigh_1) \\ h^2(p_i.neigh_2) \cdots \mod n, \quad (13)$$

where $p_i.neigh_j$ is the id of p_i's j^{th} neighbor, and $dig(p_i.p)$ is the digest of $\langle \lambda_{p_i}, \omega_{p_i} \rangle$, which shares the same definition as in Eqn. 9 in the MR-tree based scheme.

5.2.3 Top-k Query Processing, VO Construction, and Verification

We assume the Power Diagram, together with its authenticated data structure, has been materialized on external storage. Any data object, including its Power Voronoi cell, digest and signature, can be efficiently accessed using the corresponding *id*. As such, a top-k query can be incrementally processed according to Property 5.1 as follows. First, the SP finds r_1 as the object whose cell corresponds to the query point q. Next, the SP finds r_2 from the Voronoi neighbors of r_1 (i.e., $\mathcal{VN}(r_1)$). In general, the SP finds the r_k from the Voronoi neighbors of all $k-1$ objects found so far.

Let P_{neigh} denote the set of objects in $\bigcup \mathcal{VN}(r_i \in R) - R$, which are all the Voronoi neighbors of results R (except the objects already in R). To authenticate the query results, the client needs to verify: (1) $q \in \mathcal{VC}(r_1)$; (2) $\forall r_i \in R, rank(r_i, q) \leq rank(r_k, q)$; (3) $\forall r_i \in P_{neigh}, rank(r_i, q) \geq rank(r_k, q)$; and (4) no $r_i \in R \cup P_{neigh}$ is omitted or tampered with. While (4) can be verified by restoring the digests of all objects in $R \cup P_{neigh}$, (1)(2)(3) can be verified in the same manner by the PPB method or PLB method (if the DO has pre-signed the lines to be compared). In particular, (1) is equivalent to verifying $\forall r_i \in \mathcal{VN}(r_1), rank(r_i, q) \geq rank(r_1, q)$. Therefore, the VO includes: (1) the digest or digest components of each $r_i \in \mathcal{VN}(r_1)$ to privately compare with $rank(r_1, q)$; (2) the digest or digest components of each $r_i \in R$ to privately compare with $rank(r_k, q)$; (3) the digest or digest components of each $r_i \in P_{neigh}$ to privately compare with $rank(r_k, q)$; (4) the signed digests for all objects in $R \cup P_{neigh}$; and (5) the digest components necessary for the client to compute the digests of objects in $R \cup P_{neigh}$.

Figure 8 illustrates the same top-3 example as in the MR-tree based scheme. In this figure, $p_1, p_2, \cdots$ are data objects, the top-3 result $R = \{p_1, p_3, p_4\}$ and $P_{neigh} = \{p_2, p_5, p_8, p_9\}$. Figure 9(a) shows the VO, which includes:

- the digest components of each $r_i \in \mathcal{VN}(r_1)$ to privately compare with $rank(r_1, q)$, including: $E()$ values for objects p_2, p_3, p_4, p_5, p_8, p_9;
- the digest components of each $r_i \in R$ to privately compare with $rank(r_k, q)$, including: $E()$ values of objects p_1, p_3, p_4 (some in duplicate);
- the digest components of each $r_i \in P_{neigh}$ to privately compare with $rank(r_k, q)$, including: $E()$ values for objects p_2, p_5, p_8;
- the signed digests of all objects in $R \cup P_{neigh}$, including signatures for p_1, p_2, p_3, p_4, p_5, p_8, p_9;
- the digest components necessary for the client to compute the digests of objects in $R \cup P_{neigh}$, including: the digest components h_2, h_5, h_6, h_7, h_8, h_9, h_{10}, h_{11}.

As with the MR-tree based scheme, the Power Diagram based scheme can be accelerated using the PLB method. Figure 9(b) illustrates the VO when $\mathcal{SL}(p_3, p_4)$ and $\mathcal{SL}(p_4, p_5)$ are pre-signed. Since these pairs of objects are pre-compared, ranking comparisons on them no longer go through the PPB method. As such, in the figure the SP returns light-grey parts instead of dark-grey parts for p_3 and p_5. That is, instead of $E()$ values of p_3 and p_5, the SP only returns digest components o'_1, o'_2, $g(\lfloor A_1 \rfloor)$ and the corresponding signatures.

6. SECURITY ANALYSIS

In this section, we analyze how the proposed PPB/PLB methods and the authentication schemes achieve our security model. Since the completeness and soundness of the result, which is the second objective in this model, has been solved by the authentication schemes, this section focuses on the first objective, i.e., the semantic security of object location and score information. We will prove this for single/snapshot top-k queries based on Definition 3.1, and then elaborate the security model for continuous top-k queries.

6.1 Security of PPB and PLB methods

THEOREM 6.1. *PPB satisfies semantic security of object location and score information in the presence of an eavesdropper.*

PROOF SKETCH. In PPB, the adversary (eavesdropper) receives $E(\delta_i \cdot B^i)$ of all i's. According to Diffie-Hellman assumption, it is hard to distinguish δ_i from a random group member in $[0, B-1]$. On the other hand, $\delta_i \cdot B^i$ is protected by the Paillier encryption E, which is semantic secure against chosen-plaintext attacks [20]. □

THEOREM 6.2. *PLB satisfies semantic security of object location and score information in the presence of an eavesdropper (adversary).*

PROOF SKETCH. In PLB, the adversary receives: (1) $g(A_1)$ and $g(A_3)$, and (2) o'_1 and o'_2. (1) is created by the one-way function g, and thus the adversary cannot distinguish A_1 or A_3 from a random $A \in \mathbb{Z}_n$. On the other hand, given any random o'_1, there exists o'_2 that satisfies $A_1 + A_2 = A_3$, and vice versa. As such, the adversary cannot distinguish o'_1 or o'_2 from any other random point. □

6.2 *MR*-tree Based Authentication Scheme

To prove an authentication scheme (especially its VO) achieves semantic security of object location and score information, we adopt *security proof by simulation* [6]. By "simulating the view" of the client, we prove that if the client has a-priori knowledge of object u of score b being at

position a with probability $P(u = a, \omega = b)$, after receiving the VO, its posterior probability $P(u = a, \omega = b \mid VO)$ remains the same. For ease of presentation, we adopt the PPB method when private ranking comparisons are used.

According to Section 5.1.4, depending on whether u is a result, the information disclosed by the VO to the client is in one of three cases: (1) if $u = r_k$, i.e., u is the top-k^{th} object, then the client knows $rank(u, q) \leq rank(A, q)$, where A is any MBB in the heap H_{rm}; (2) if $u \in R$ and $u \neq r_k$, then the client knows $rank(u, q) \leq rank(r_k, q)$; (3) if $u \notin R$, then the client knows $rank(u, q) \geq rank(r_k, q)$. In the following lemmas, we show all these cases have the posterior probability equal to the a-priori probability.

LEMMA 6.3. *Let $u = r_k$, $\forall A \in H_{rm}$, $P(u = a, \omega = b) = P(u = a, \omega = b \mid rank(u, q) \leq rank(A, q))$.*

PROOF.

$$P(u = a, \omega = b \mid rank(u, q) \leq rank(A, q))$$
$$= \frac{P(rank(u, q) \leq rank(A, q) \mid u = a, \omega = b) \cdot P(u = a, \omega = b)}{P(rank(u, q) \leq rank(A, q))}$$
$$= \frac{P(rank(u, q) \leq rank(A, q) \bigwedge u = a \bigwedge \omega = b)}{P(rank(u, q) \leq rank(A, q))}$$
$$= P(u = a, \omega = b)$$

The first equality is due to Bayes' Theorem and the third equality is due to the fact that $rank(u, q) \leq rank(A, q)$ is independent of $u = a$ and $\omega = b$ as the rank value of A is unknown to the client. In fact, knowing $rank(u, q) \leq rank(A, q)$ does not limit the placement and score of u. □

Similarly, we can obtain the following lemmas:

LEMMA 6.4. *Let $u \in R$ and $u \neq r_k$, $P(u = a, \omega = b) = P(u = a, \omega = b \mid rank(u, q) \leq rank(r_k, q))$.*

LEMMA 6.5. *Let $u \notin R$, $P(u = a, \omega = b) = P(u = a, \omega = b \mid rank(u, q) \geq rank(r_k, q))$.*

Based on these lemmas, we present the following theorem on the security of the scheme.

THEOREM 6.6. *The MR-tree based scheme does not leak the location or score of any object u to the client, given any VO.*

PROOF. Equivalently, we show there is a polynomial-time simulator SIM that can simulate the view of the client without knowing the data of SP. Specifically, it reproduces the VO of the client with the same probability distribution as if it were sent from the real SP.

According to Lemmas 6.3, 6.4 and 6.5, without changing the distribution $P(u = a, \omega = b)$, SIM is allowed to know (1) if $A \subseteq Q$ and (2) if $A \bigcap Q \neq \emptyset$, for any MBB A. As such, SIM can reproduce the VO from the heap H_{rm} according to Section 5.1.4 as follows. For leaf entry u (whether $u \in R$ or $\notin R$), SIM adds to the VO u's digest components for the private ranking comparisons on objects; if $u \in R$, SIM further adds its digest component for id authentication; else for MBB A, SIM adds to the VO the digest components for the private ranking comparisons on an MBB and an object. This VO has the same probability distribution as generated by the real SP. Also SIM runs in polynomial time. □

6.3 Power Diagram based Scheme

According to Section 5.2.3, the information disclosed by the VO to the client is in one of three cases: (1) if $u = r_k$, then the client knows $\forall r_i \in P_{neigh}, rank(r_i, q) \geq rank(r_k, q)$; (2) if $u \in R$ and $u \neq r_k$, then the client knows $rank(u, q) \leq rank(r_k, q)$; (3) if $u \notin R$, then the client knows $rank(u, q) \geq rank(r_k, q)$ and $u \in P_{neigh}$, that is, u is a Voronoi neighbor of some r_i. While the first two cases are the same as in the MR-tree based scheme, we show in the following lemma that the third case also has the posterior probability equal to the a-priori probability.

LEMMA 6.7. *Let $u \notin R$, $P(u = a, \omega = b) = P(u = a, \omega = b \mid rank(u, q) \geq rank(r_k, q) \bigwedge u \in \mathcal{VN}(r_i))$.*

PROOF. Proof is similar to that of Lemma 6.3, and more details are given in the full technical report [2]. □

Now we reach the following theorem on the security of Power Diagram based scheme.

THEOREM 6.8. *The Power Diagram based scheme does not leak the location or score of any object u to the client, given any VO.*

PROOF. Proof follows that of Theorem 6.6. □

6.4 Security Model for Continuous Top-k Queries

So far we have shown that our authentication scheme is semantic secure against a single top-k query, which offers sound privacy protection for snapshot queries. Nevertheless, when the client issues queries continuously, the moving query point causes the change of query results (i.e., the relative rank values), which are known to the client. As such, the client may gradually learn an object is more probable to be in one location than in another, which undermines the semantic security of object location and score information. To remedy this, we propose the following security model for continuous top-k queries, by restraining the scope of semantic security.

DEFINITION 6.9. **Continuous Semantic Security with γ Radius.** *Given a continuous set of queries $Q = \{\langle q_i, k_i \rangle \mid i = 1, 2, \cdots\}$, an authentication scheme is semantic secure against the probabilistic polynomial-time client, if the latter cannot distinguish an object p from a pseudo object in a sphere with a radius of γ.*

This sphere is thus called "privacy sphere". Initially, each object is assigned with a large enough privacy sphere that at least covers one other object beyond distance γ. To achieve continuous semantic security, the server monitors and maintains the *current* privacy sphere of each object. Upon receiving a new query, it checks whether this incoming query will cause some spheres to shrink, as the client gets to know the rank comparison results between each object and r_k w.r.t the new query point. If the radius of some privacy sphere drops below threshold γ, the server will reject this query. Figure 10 illustrates this procedure for a top-3 query q, where for simplicity we assume the non-spatial score $\omega = 0$ and thus each object has a privacy circle (instead of a sphere) before and after accepting q, shown by dashed and solid circles, respectively. The new privacy circle of a result object (except r_k), e.g., p_2, is the inscribed circle of its old circle and $C_{int}(q)$, the interior of all query results (i.e., the circumscribed circle

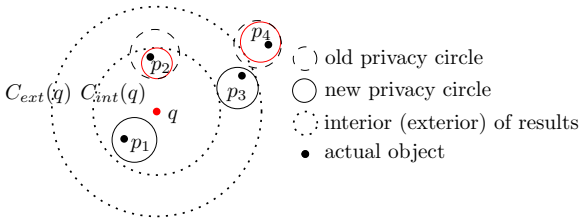


Figure 10: Privacy Circles before and after Query q

of r_k's (p_3's) circle centered at q, inside which any object must be a result). On the other hand, the new privacy circle of a non-result object, e.g., p_4, is the inscribed circle of its old circle that circumscribes $C_{ext}(q)$, the exterior of all query results (i.e, the inscribed circle of r_k's circle centered at q, outside which any object cannot be a result).

7. OFFLINE AND ONLINE STRATEGY ON PRE-SIGNED LINES

As shown in Sections 5.1.5 and 5.2.3, pre-signed lines can accelerate the authentication on both MR-tree and Power Diagram based schemes. Ideally the authentication cost is minimum if all pairs of MBB corner points (for the MR-tree) or data points (for the Power Diagram) are pre-signed and thus the costly PPB method can be replaced by the PLB method in all comparisons. However, this requires a prohibitively huge amount of DO computation time and storage cost at the SP. In this section, we assume that the DO has a limited *budget* of pre-signing a number of pairs of objects. The problem is two-fold: (1) to decide offline which objects to be pre-signed by the DO; and (2) to decide online for the SP which pre-signed lines to choose for the construction of VO, so that the number of PPB calls is minimum.

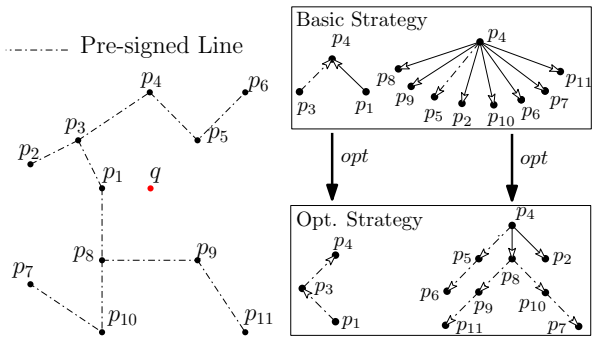
7.1 DO Offline Strategy on Pre-signed Lines

Without a-priori knowledge of the queries, the DO should pre-sign those pairs with the highest probabilities of comparisons. In the Power Diagram based scheme, a private ranking comparison is between two objects close in the diagram. As such, the DO's strategy can be as follows. For every object p_i, the DO first signs it with all its Voronoi neighbors. If budget allowed, the DO continues to sign all 2-hop, 3-hop, $\cdots$ neighbors in the Power Diagram.

In the MR-tree based scheme, a private ranking comparison is often between the top-k^{th} object r_k and an MBB M. To improve the utility of pre-signed pairs, we choose points from close-by MBB pairs to pre-sign, in the hope that if for two MBBs M_1, M_2, the pre-signed pairs prove $Dist(M_2, q) \geq Dist(M_1, q)$, then proving $Dist(M_1, q) \geq Dist(r_k, q)$ will also prove $Dist(M_2, q) \geq Dist(r_k, q)$. To this end, each MBB chooses four sibling MBBs in the same R-tree node which are the closest to each of its corner points, and signs these corner pairs. If budget allowed, the DO continues to sign every corner point with the second, third closest corner points, and so on. We call them 2-hop, 3-hop points for consistency with the Power Diagram based scheme.

7.2 SP Online Strategy on Pre-signed Lines

Figure 11(a) illustrates some pre-signed lines (plotted by dashed lines) in the top-3 query running example, where the results $R = \{p_1, p_3, p_4\}$. When constructing the VO for a specific query, the SP should form the two ranking comparison chains for the objects in R and $\mathbb{D} - R$, respectively, with the maximum use of the above pre-signed lines.



(a) Pre-signed Lines (b) Comparison Chains and Spanning Trees

Figure 11: Online Strategy on Pre-signed Lines. (a) the pre-signed lines in the top-3 query running example. (b) the spanning trees and comparison chains for both strategies.

We show two strategies for the SP in Figure 11(b), where the creation of the comparison chain follows the creation of a spanning tree. The basic one applies the PLB method (shown in dashed arrowed lines) only if a pre-signed line with the top-k^{th} object (i.e., p_4) exists, and applies the PPB method otherwise (shown in solid arrowed lines). Using this strategy, only 2 pre-signed lines are used. By contrast, an optimized strategy uses 8 pre-signed lines. The idea is to replace the comparison on the top-k^{th} object with some other pivot objects which are pre-signed.[9] Starting from the top-k^{th} object (i.e., p_4), the SP visits the objects in $\mathbb{D} - R$ in ascending order of rank values. When p_j is visited, the SP checks whether some p_i in the existing tree has a pre-signed line with p_j. If such p_i (e.g., p_8) exists, the SP spans p_j (e.g., p_9) from p_i and uses this pre-signed line; otherwise, the SP spans p_j (e.g., p_2) from the root and uses the PPB method. A similar spanning tree (and thus the comparison chain) is created for the objects in R.

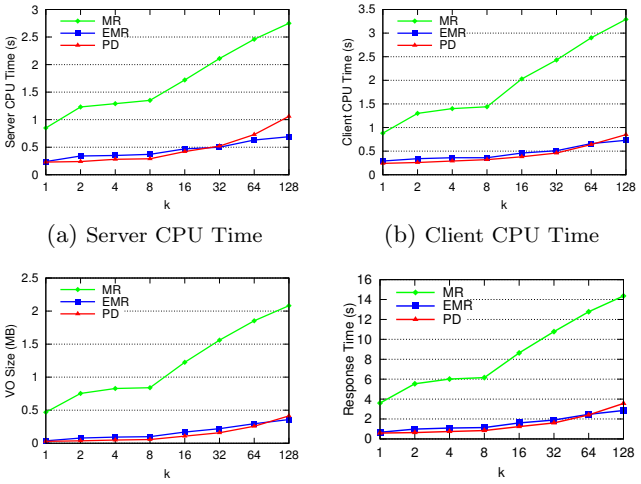
8. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the MR-tree (MR) and Power Diagram (PD) based privacy-preserving authentication schemes on location-based top-k queries. We also implement EMR, an enhanced MR scheme with an embedded kd-tree in each internal MR-tree node [15]. The experiments use the Gowalla dataset in Stanford Large Network Dataset Collection, which records $6,442,890$ user check-ins at $1,280,969$ unique locations. These locations serve as the objects in the experiments and their non-spatial scores are the deviations of their check-in counts from the maximum count of all objects, designating the unpopularity or "distance penalty" as in [14]. To balance the distance and non-spatial score in the ranking function, we normalize the latter so that its average score (ω) equals the average Euclidean distance between neighboring objects, which is approximately 5,000 m. All location coordinates and scores are rounded to their closest integers. We simulate a moving client using the random waypoint mobility model [8], with an average speed of 10 m/s. The client issues a top-k query every 10 seconds, with k randomly selected from 1 to 128. We adopt the continuous semantic security model and test with a variety of radius γ of the privacy sphere. Table 1 shows the query acceptance rates after the system becomes

[9]This also changes the relative rank value to be disclosed, from r_k to the pivot objects.

γ (m)	500	1000	2000	5000
Acceptance Rate	100%	98.7%	71.0%	38.0%

Table 1: Query Acceptance Rate v.s. Privacy Sphere Radius γ



(a) Server CPU Time (b) Client CPU Time

(c) VO Size (d) Query Response Time

Figure 12: Basic Query Authentication Performance

stable (after 1,000 queries). We observe that γ does not significantly affect the acceptance rate until it approaches 5,000 m (when an average privacy sphere contains neighboring objects). This verifies the effectiveness of the proposed security model for continuous top-k queries. We use γ = 1,000 m in the subsequent experiments.

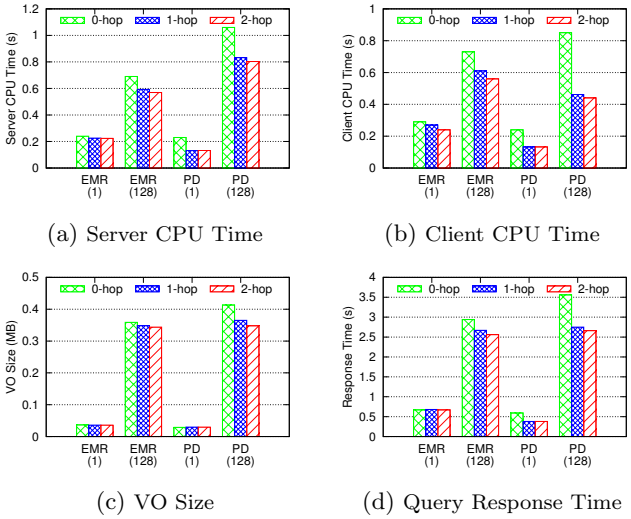
For performance evaluation, the client side is set up on a laptop computer, with entry-level Intel Core i3 processor and 4GB RAM, running Windows XP x64 SP3. The SP is set up on a HP Proliant DL360 G7, with Dual 6-Core Intel Xeon X5650 2.66GHz CPU and 32GB RAM, running GNU/Linux. The code is written in Java and executed in 64-bit OpenJDK 1.6. The hash function used in digest computation is 160-bit SHA-1; the homomorphic encryption function is 1024-bit Paillier; and the signature function is 2048-bit condensed RSA. We use the same optimized digest function $g()$ as in [21] with the base of canonical representation set to 16. The performance metrics are: the SP CPU time (for query processing and VO construction), the client CPU time (for verification), the communication overhead (in terms of the size of VO), and the overall query response time (as the total CPU time plus the communication time over a typical 3G network at 2Mbps download rate).

8.1 Basic Query Authentication Performance

In this subsection, we evaluate the authentication performance of the three schemes without DO pre-signing any lines. In other words, the results here apply the PPB method only. We vary k from 1 to 128 and plot the performance in Figure 12. We observe that EMR consistently outperforms MR, thanks to its small fanout during VO construction and verification. The performance gap further enlarges as k increases. On the other hand, although EMR is comparable to PD, the latter outperforms EMR in small and medium-sized queries ($k \leq 32$). This can be explained by the fact that MR-tree clusters objects effectively and its efficiency is better exploited when querying for more results. We also evaluate the overhead of confidential top-k authentication by showing the multiples of each cost against non-confidential authentication in Table 2 for the EMR and PD schemes.

	Server CPU		Client CPU		VO Size		Server Memory	
k	**EMR**	**PD**	**EMR**	**PD**	**EMR**	**PD**	**EMR**	**PD**
1	44.4	1928	7.20	320	33.2	37.4	5.12	1.26
64	39.5	890.3	14.3	97.6	63.1	59.5	4.99	1.22
128	38.3	754.6	16.7	82.1	67.7	63.0	4.93	1.18

Table 2: Cost Increase as Multiples of Non-Confidential Authentication (base)



(a) Server CPU Time (b) Client CPU Time

(c) VO Size (d) Query Response Time

Figure 13: Performance with Pre-signed Lines (k = 1 or 128)

8.2 Performance with Pre-signed Line Optimization

	EMR	**PD**
	CPU Time (mins)	
0-hop	23.6	25.3
1-hop	50.5	38.0
2-hop	108.8	69.0
	Pre-signed Storage (GB)	
0-hop	6.7	6.8
1-hop	13.2	9.8
2-hop	28.6	17.4

Table 3: Construction Cost

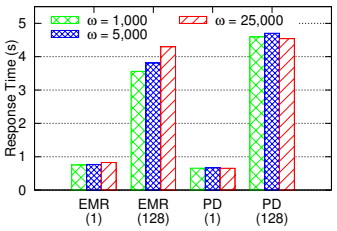


Figure 14: Impact of Scores

In this subsection, we evaluate the performance of DO and SP's optimization strategies on PLB method (Section 7). We focus only on the EMR and PD schemes because they are more comparable while the MR scheme has heavy overhead in all metrics. In the experiment, we vary the number of pre-signed lines, by letting the DO sign 0-hop (i.e., the basic PPB method), 1-hop, and 2-hop, respectively. The results are shown in Figure 13. We observe that by introducing the PLB method, both of the schemes have reduced cost in terms of all performance metrics. It is also noteworthy that PD is most sensitive to pre-signed lines, which gains significant performance boost by introducing 1-hop PLB.

While it is clear that the more lines the DO pre-signs, the more efficient the authentication will be, the pre-signing cost can be prohibitively high when the hops increase. For example, pre-signing 2-hop for EMR takes about 2 hours and requires more than 25GB storage at the SP. Table 3 summarizes the construction time and storage cost for different schemes and pre-sign settings. Based on these results, we can find a balanced scheme between the pre-sign cost and query response time for different k settings as follows. When k is as large as 128 or even larger, EMR with 0-hop and 1-hop are the best schemes for no pre-signing and with pre-signing budget, respectively. When k is small or medium

sized, PD with 0-hop and 1-hop are the best schemes for no pre-signing and with pre-signing budget, respectively.

8.3 Impact of Non-spatial Scores

In this subsection, we evaluate the impact of non-spatial scores on the overall query response time in Figure 14. We vary the average score ω, from 1,000 m to 25,000 m. It is observed that as the non-spatial score has a higher weight in the ranking function, the response time of EMR becomes poorer due to a less effective clustering of objects in a 3D MR-tree with both spatial and non-spatial attributes. Nonetheless, the performance of PD remains good and robust for various score weights, as the construction of PD is less vulnerable to them. This shows that PD is a suitable scheme for location-based top-k queries where location is not always the dominating factor in the ranking function.

9. CONCLUSIONS

In this paper, we studied the problem of privacy-preserving authentication for top-k queries in LBSs. By designing cryptographic building blocks of private ranking-value comparisons, we have presented two authentication schemes based on multi-dimensional R-tree and Power Diagram indexes. The performance and security of our proposed schemes have been verified and analyzed under various system settings.

As for future work, we plan to extend the proposed authentication schemes to a general framework for the queries based on private value comparisons. This covers classic queries such as skylines and distance joins. Furthermore, as location-based services usually adopt distance metrics other than Euclidean distance, a more general version of private value comparisons in metric space is yet to be designed.

Acknowledgments

We would like to thank all reviewers for their valuable suggestions. This work was supported by RGC/GRF HKBU 211512, 210811 and 210612.

10. REFERENCES

[1] S. Chaudhuri and L. Gravano. Evaluating top-k selection queries. VLDB, pages 397–410, 1999.

[2] Q. Chen, H. Hu, and J. Xu. Privacy-preserving authentication of knn queries in location-based services. Technical report, HKBU, 2013, `http://www.comp.hkbu.edu.hk/~qchen/auth-topk.pdf`.

[3] W. Cheng and K. Tan. Query assurance verification for outsourced multi-dimensional databases. *Journal of Computer Security*, pages 101–126, 2009.

[4] S. Choi, H. Lim, and E. Bertino. Authenticated top-k aggregation in distributed and outsourced databases. SOCIALCOM-PASSAT12, pages 779–788, 2012.

[5] C. Chow, M. Mokbel, and W. Aref. Casper*: Query processing for location services without compromising privacy. *ACM TODS*, pages 24:1–24:48, 2009.

[6] O. Goldreich. *The Foundations of Cryptography – Volume 2*. Cambridge University Press, 2004.

[7] H. Hu, J. Xu, Q. Chen, and Z. Yang. Authenticating location-based services without compromising location privacy. In *Proc. SIGMOD*, pages 301–312, 2012.

[8] H. Hu, J. Xu, and D. L. Lee. A generic framework for monitoring continuous spatial queries over moving objects. In *Proc. SIGMOD*, pages 479–490, 2005.

[9] H. Hu, J. Xu, C. Ren, and B. Choi. Processing private queries over untrusted data cloud through privacy homomorphism. In *Proc. ICDE*, pages 601–612, 2011.

[10] L. Hu, W.-S. Ku, S. Bakiras, and C. Shahabi. Spatial query integrity with voronoi neighbors. *IEEE TKDE*, 25(4):863 –876, 2013.

[11] H. Jung, B. K. Cho, Y. D. Chung, and L. Liu. On processing location based top-k queries in the wireless broadcasting system. In *Proc. SAC*, 2010.

[12] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *TKDE*, 19(12):1719–1733, 2007.

[13] J. Katz and Y. Lindell. *Introduction to modern cryptography*. Chapman&Hall/CRC Press, 2008.

[14] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. Lars: A location-aware recommender system. In *Proc. ICDE*, pages 450–461, 2012.

[15] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin. Dynamic authenticated index structures for outsourced databases. In *Proc. SIGMOD*, 2006.

[16] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin. Authenticated index structures for aggregation queries. *ACM TISSEC*, 13(32):1–35, 2010.

[17] X. Lin, J. Xu, and H. Hu. Authentication of location-based skyline queries. In *Proc. CIKM*, 2011.

[18] R. C. Merkle. A certified digital signature. In *Proc. Crypto*, pages 218–238, 1989.

[19] A. Okabe, B. Boots, K. Sugihara, S. N. Chiu, and D. G. Kendall. *Spatial Tessellations, Concepts and Applications of Voronoi Diagrams, Second Edition*. John Wiley and Sons, Inc., 2008.

[20] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proc. EUROCRYPT*, pages 223–238, 1999.

[21] H. Pang, A. Jain, K. Ramamritham, and K. lee Tan. Verifying completeness of relational query results in data publishing. In *Proc. SIGMOD*, 2005.

[22] H. Pang and K.-L. Tan. Authenticating query results in edge computing. In *Proc. ICDE*, 2004.

[23] S. Papadopoulos, S. Bakiras, and D. Papadias. Nearest neighbor search with strong location privacy. In *VLDB*, pages 619–629, 2010.

[24] J. D. Sutter. After outcry, Facebook pulls 'find friends nearby' feature from social network web site. *ABC News*, June 2012.

[25] Y. Yang, S. Papadopoulos, D. Papadias, and G. Kollios. Spatial outsoucing for location-based services. In *Proc. ICDE*, pages 1082–1091, 2008.

[26] Y. Yang, S. Papadopoulos, D. Papadias, and G. Kollios. Authenticated indexing for outsourced spatial databases. *VLDBJ*, 18(3):631–648, 2009.

[27] M. L. Yiu, Y. Lin, and K. Mouratidis. Effcient verification of shortest path search via authenticated hints. In *Proc. ICDE*, pages 237–248, 2010.

[28] M. L. Yiu, E. Lo, and D. Yung. Authentication of moving knn queries. In *Proc. ICDE*, 2011.

[29] R. Zhang, Y. Zhang, and C. Zhang. Secure top-k query processing via untrusted location-based service providers. In *Proc. INFOCOM*, pages 1170–1178, 2012.