Geoinformatica
DOI 10.1007/s10707-010-0121-4

Approximate and exact hybrid algorithms for private nearest-neighbor queries with database protection

Gabriel Ghinita · Panos Kalnis · Murat Kantarcioglu · Elisa Bertino

Received: 2 April 2010 / Revised: 16 August 2010 /
Accepted: 23 November 2010

Abstract Mobile devices with global positioning capabilities allow users to retrieve points of interest (POI) in their proximity. To protect user privacy, it is important not to disclose exact user coordinates to un-trusted entities that provide location-based services. Currently, there are two main approaches to protect the location privacy of users: (i) hiding locations inside cloaking regions (CRs) and (ii) encrypting location data using private information retrieval (PIR) protocols. Previous work focused on finding good trade-offs between privacy and performance of user protection techniques, but disregarded the important issue of protecting the POI dataset D. For instance, location cloaking requires large-sized CRs, leading to excessive disclosure of POIs ($O(|D|)$ in the worst case). PIR, on the other hand, reduces this bound to $O(\sqrt{|D|})$, but at the expense of high processing and communication overhead. We propose hybrid, two-step approaches for private location-based queries which provide protection for both the users and the database. In the first step, user locations are generalized to coarse-grained CRs which provide strong privacy. Next, a PIR protocol is applied with respect to the obtained query CR. To protect against excessive disclosure of POI locations, we devise two cryptographic protocols that privately evaluate whether a point is enclosed inside a rectangular region or a convex polygon. We also introduce algorithms to efficiently support PIR on dynamic POI

G. Ghinita (✉) · E. Bertino
Purdue University, West Lafayette, IN 47907, USA
e-mail: gghinita@cs.purdue.edu

E. Bertino
e-mail: bertino@cs.purdue.edu

P. Kalnis
King Abdullah University of Science and Technology, Jeddah, Saudi Arabia
e-mail: panos.kalnis@kaust.edu.sa

M. Kantarcioglu
University of Texas at Dallas, Richardson, TX 75080, USA
e-mail: muratk@utdallas.edu

Published online: 15 December 2010

sub-sets. We provide solutions for both approximate and exact NN queries. In the approximate case, our method discloses $O(1)$ POI, orders of magnitude fewer than CR- or PIR-based techniques. For the exact case, we obtain optimal disclosure of a single POI, although with slightly higher computational overhead. Experimental results show that the hybrid approaches are scalable in practice, and outperform the pure-PIR approach in terms of computational and communication overhead.

Keywords Location privacy · Private information retrieval · Homomorphic encryption

1 Introduction

Mobile devices with positioning capabilities (e.g., GPS) facilitate access to location-based services that provide information relevant to the users' geo-spatial context. Typically, users are interested in finding nearby *points of interest (POI)*, and send nearest-neighbor (NN) queries to *location servers (LS)* that own databases of POI. However, users are reluctant to disclose their exact locations to the un-trusted LS, since sensitive details about lifestyle, political or religious affiliation, etc., can be revealed by a person's whereabouts.

To address this threat, user locations are perturbed before being reported to the LS. On the other hand, replacing exact locations with coarse regions requires the disclosure of a large number of POIs to the user, such that result correctness is preserved. However, the LS wishes to protect its data against excessive disclosure, since the POI dataset represents a valuable asset to the service provider. For instance, consider that Bob asks the query "find the nearest restaurant to my current location". The LS may reward Bob with certain discounts, in the form of electronic coupons (e.g., digital gift card codes) that are associated with each POI. If the user is billed on a "per-retrieved-POI" basis, then a large number of results will increase the cost of using the service. On the other hand, if the LS offers the service with no charge to the user (e.g., advertisement-generated income), then users could abuse the system by redeeming a large number of coupons. This causes the LS to lose its competitive edge, and to cease providing the service.

Existing solutions for private location queries focus on user protection only, and can be broadly classified into two categories:

1. *Location Cloaking* techniques replace the exact location of a user with a *cloaking region (CR)*, typically of rectangular shape. To ensure result correctness, the CR must enclose the actual user location. Furthermore, CRs must satisfy certain constraints dictated by a privacy paradigm, which expresses the privacy requirements of the user (e.g., *spatial k-anonymity (SKA)* [1–4] requires each CR to contain at least k distinct users). Regardless of the method used to generate the CR, query processing at the LS side is performed with respect to a rectangular region, as opposed to an exact user location. In consequence, the result returned by the LS is a super-set of the actual query result.
2. *Private Information Retrieval (PIR)* techniques rely on a cryptographic protocol to achieve query privacy [5]. In a pre-processing phase, the LS organizes the

POI database into a data structure relevant to the supported type of query,[1] and maps it to an ordered array $D[1 \ldots n]$. At runtime, a query is transformed from a context-based (i.e., spatial) query to a query-by-index (i.e., return the i^{th} item), according to the pre-defined data organization which is known by the users. When a user wishes to retrieve $D[i]$, s/he creates an encrypted query object $q(i)$. Using a mathematical transformation, the LS computes privately (i.e., without learning the value of i) the result $r(D, q(i))$ and sends it back to the user. PIR protocols ensure that it is computationally hard for the LS to recover the value i from $q(i)$, but at the same time the user can easily re-construct $D[i]$ from r.

Previous work [3–5] evaluates location privacy techniques based on two criteria: *privacy* and *performance*. With respect to privacy, PIR offers strong guarantees for both one-time, as well as repetitive (i.e., continuous) queries. Furthermore, PIR does not require trusted components, such as anonymizer services or other trusted users. On the other hand, CR methods operate under a more restrictive set of trust assumptions, but are considerably more efficient in terms of computational and communication overhead. The cryptographic elements incorporated in PIR require powerful computational resources (e.g., parallel machines), and high-bandwidth communication channels.

However, there is a third, equally-important dimension in evaluating techniques for private location queries: the amount of protection provided to the database. To the best of our knowledge, this aspect has not been addressed before.[2] Nevertheless, as illustrated by the earlier customer-reward example, it is important to control tightly the amount of POI disclosure.

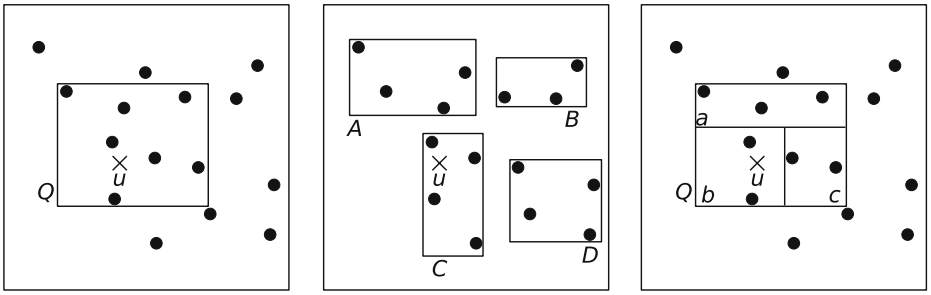
To illustrate the limitations of existing approaches, consider the example of Fig. 1, where the location server stores a database D of 15 POI (marked as full dots). User u asks a query for the nearest POI. If location cloaking is used (Fig. 1a), the user will retrieve all the seven POI enclosed[3] by query CR Q. As CRs grow large, location cloaking methods may disclose a large fraction of the database (possibly linear to $|D|$). On the other hand, the NN protocol from [5] does not use CRs (a detailed protocol description is given in Section 2). Instead, the dataset is partitioned into rectangular tiles $A \ldots D$, containing at most $\lceil \sqrt{15} \rceil = 4$ POI each (Fig. 1b). The boundaries of the tiles are sent in plain text to u, who determines that his/her location is enclosed by tile C. Only the POIs in tile C are revealed to u through a PIR request. This method discloses $O(\sqrt{|D|})$ exact POI locations. However, revealing the tile boundaries may result in additional disclosure of POI locations, especially if the tiles have small spatial extent.

We propose hybrid approaches to both approximate and exact NN queries. Figure 1c provides a brief illustration of how the hybrid two-step method works (we focus only on approximate queries in this example, the exact case is discussed in Section 6). The CR Q is sent to the LS, which determines a set of fine-grained tiles $\{a, b, c\}$ that cover the query area. We impose a constraint that each tile encloses at

[1] For instance, to answer NN queries, [5] uses a Voronoi diagram mapped to a regular grid.

[2] Previous work considered result set size only in the context of communication cost. However, this indirect approach is not effective due to other factors that influence bandwidth consumption (e.g., POI size may be negligible in comparison with other traffic components).

[3] The example considers an approximate query, where candidate NNs outside Q are ignored.



(a) Location Cloaking (b) PIR (c) Hybrid

Fig. 1 Benefit of the hybrid approach

most a constant number F of POI (a system parameter). The boundaries of the tiles are not sent to the user. Instead, the user and the LS engage in a novel cryptographic protocol that privately determines which one of the tiles encloses the location of u. At the end of the protocol, the LS learns nothing about the user location (except that u is inside Q), whereas the user only learns the identifier of the tile that encloses u (but not the boundaries of any of the tiles). Finally, the user requests through PIR the contents of the enclosing tile[4] (in this case, b). The hybrid approach has two benefits: first, it controls strictly the amount of POI disclosed, which is bounded by a constant. This improvement is clearly superior to location cloaking and pure-PIR approaches, which disclose $O(|D|)$ and $O(\sqrt{|D|})$ POI, respectively. Second, the hybrid approach incurs considerably less overhead than the pure PIR method, since the cryptographic protocol is applied only on a partition of the database.

Our specific contributions are:

(i) We propose a cryptographic protocol that relies on homomorphic encryption and allows private evaluation of point-in-rectangle enclosure. This protocol makes use of the homomorphic computation of the addition operation, and forms the basis for our private approximate NN queries solution.

(ii) We extend the point-in-rectangle enclosure protocol to a more general case that allows private evaluation of point-in-convex-polygon enclosure. The latter protocol relies on the homomorphic computation of both addition and multiplication-with-a-constant operations, and forms the basis for our private exact NN solution. Private evaluation of point-in-convex-polygon enclosure provides the means to determine to which Voronoi cell of a dataset of POI a user's location belongs to.

(iii) We develop a hybrid approach that efficiently supports PIR processing with respect to a user-generated cloaked region Q. The proposed method can handle CRs with large extents, and controls tightly the amount of disclosed POI. Furthermore, we show experimentally that it is considerably more efficient than its PIR-only counterpart.

[4]The indexing scheme we employ (Section 5) guarantees that the retrieved tile is not empty.

The rest of the paper is organized as follows: Section 2 surveys related work. Section 3 outlines the system architecture and the privacy assumptions. Section 4 introduces the proposed protocol for private evaluation of point-rectangle enclosure, whereas Section 5 presents the hybrid technique for processing private approximate NN queries based on dynamic cloaking regions.[5] Section 6 introduces the protocol for private evaluation of point-in-convex-polygon inclusion, and shows how exact NN queries can be answered with the help of Voronoi diagrams. We present the results of our experimental evaluation in Section 7, and conclude with directions for future research in Section 8.

2 Related work

Several approaches to private location queries have been proposed. In [6], the querying user sends to the server $k-1$ fake locations to reduce the likelihood of identifying the actual user position. *SpaceTwist* [7] performs a multiple-round incremental range query protocol, based on a fake *anchor* location that hides the user coordinates. In [8], a random cloaking region that encloses the user is generated. However, neither of these techniques is suitable if an adversary possesses background knowledge about user locations. Most CR-based solutions [1–4] implement the spatial k-anonymity (SKA) paradigm, and rely on a three-tier architecture: a trusted anonymizer service intermediates all interaction between users and LS, and generates CRs that contain at least k *real* user locations.

If the resulting CRs are *reciprocal* [4], SKA guarantees privacy for snapshots of user locations. However, supporting continuous queries [9] requires generating large-sized CRs. In [10, 11], the objective is to prevent the association between users and sensitive locations. Users define privacy profiles [11] that specify their sensitivity with respect to certain *feature types* (e.g., hospitals, bars, etc.), and every CR must cover a diverse set of sensitive and non-sensitive features.

A common limitation of CR-based techniques is that they disclose an excessive number of POIs.

In [12], the set of POI is first encoded according to a secret transformation by a trusted entity. A Hilbert-curve mapping (with secret parameters) transforms 2-D points to 1-D. Users (who know the transformation key) map their queries to 1D, and the processing is performed in the 1-D space. However, the mapping can decrease the result accuracy, and the transformation may be vulnerable to reverse-engineering.

Private Information Retrieval (PIR) protocols allow users to retrieve an object X_i from a set $X = \{X_1 \dots X_n\}$ stored by a server, without the server learning the value of i. The PIR concept was first formulated in [13], where it is shown that in the information theoretic setting, any single-server solution requires $\Theta(n)$ communication cost. In practice, this bound can be reduced by employing *Computational* PIR (cPIR), which offers protection against an attacker with polynomially-bounded

[5]Note that, we choose to present first the approximate NN solution, and then the exact one. While this may not seem intuitive at first glance, the rationale behind this presentation order is that the approximate NN method maps in a more natural manner to the PIR technique and associated data structures (i.e., PIR matrix). Therefore, presenting the approximate method first improves the readability of the paper.

computational capabilities. The PIR protocol in [14] relies on the *Quadratic Residuosity Assumption (QRA)*, which states that it is computationally hard to find the quadratic residues (in modulo arithmetic) of a large composite number $N = q_1 \cdot q_2$ (q_1, q_2 are large primes). Specifically, given a number $y \in \mathbb{Z}_N^{+1}$ ($\mathbb{Z}_N^{+1}$ is the sub-set of $\mathbb{Z}_N$ for which the Jacobi symbol [15] is $+1$) it is computationally hard (without knowing the factorisation of N) to determine whether y is a quadratic residue (QR) (i.e., $\exists x \in \mathbb{Z}_N | y = x^2 \mod N$) or a non-residue (QNR). Assume that all objects in X are bits. The client sends the server an ordered array of n numbers $Y = [y_1 \cdots y_n]$, such that y_i is QNR, whereas all the others are QR. The server performs a *masked* multiplication of values in Y, i.e., it multiples together only the y_j values for which $X_j = 1$. The client, who knows the factorisation of N, can determine that if the result of the multiplication is QNR, then $X_i = 1$, otherwise $X_i = 0$. The protocol can be applied bit-by-bit to support more complex objects.

The work in [5] extends the above-mentioned protocol for binary data to the LBS domain, and proposes approximate (*ApproxNN*) and exact (*ExactNN*) protocols for nearest-neighbor queries. Our work proposes hybrid alternatives to answer both approximative and exact queries. Since we use the solutions in [5] as a baseline in our experimental evaluation, we provide an overview of their functionality.

ApproxNN organizes the POI set such that spatial queries (e.g., NN) can be translated to queries "by-index", which are then answered using the QRA-based protocol. In an off-line phase, the server performs a partitioning of the POI set D using an R^*-tree index, which is constrained to have exactly two levels. Therefore, each leaf node holds at most $\sqrt{|D|}$ POI, and the root node contains at most $\sqrt{|D|}$ minimum bounding rectangles (MBR). Figure 2 shows the obtained index for the partitioned dataset in Fig. 1b. At query time, the user u first retrieves the root node in plaintext, and determines which leaf node encloses, or is nearest to, u's location. Next, u retrieves privately the contents of the selected leaf node. There are three limitations of this approach: *(i)* a large number ($O(\sqrt{|D|})$) of POI are directly disclosed, *(ii)* sending MBRs of leaf nodes to the user can indirectly disclose additional POI locations and *(iii)* the computational complexity of the PIR phase is $O(|D|)$, as all data elements are considered, and bandwidth consumption is high.

The functionality of ExactNN is illustrated in Fig. 3. The server determines the Voronoi tessellation of the POI dataset, and super-imposes a 2D grid on top of it. Both the client and the server know the granularity of the grid, and the PIR retrieval is performed with respect to grid cells. The PIR object associated to a grid cell contains all data points whose corresponding Voronoi cells intersect the grid cell. In the example, grid cell A_1 contains p_1 and p_3, whereas A_3 contains only p_2. However, to prevent the server from learning the query based on the size of the returned result, all PIR objects must have the same length, therefore each grid cell

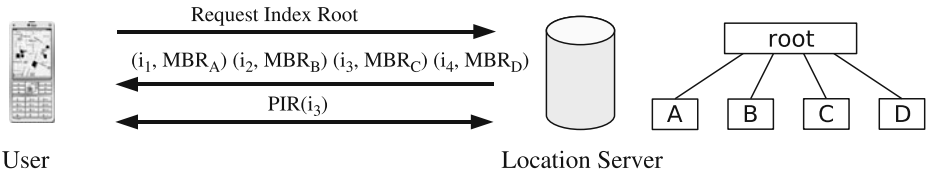


Fig. 2 Approximate NN PIR protocol from [5]

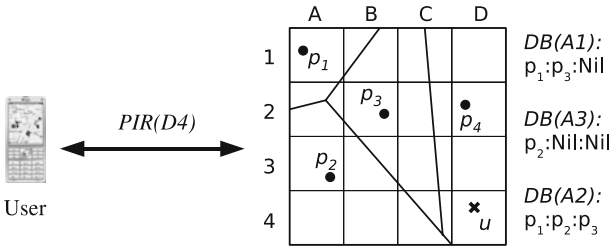


Fig. 3 Mapping Voronoi cells to grid in the exact NN PIR protocol from [5]

is padded with dummy POI (these are filtered out by the user). Therefore, each grid cell contains a number of POI equal to the maximum number of real POI hashed in an individual grid cell. For skewed datasets, this can be problematic: in addition to increasing computational and communication costs, collisions of a large number of POI in a single cell leads to high disclosure, since all POI in the grid cell that encloses the user's location are sent as results. In Section 6 we devise a hybrid technique that achieves optimal POI disclosure.

Several protocols that support secure multi-party computational geometry have been proposed. For instance, in [16] it is shown how to compute privately point-rectangle inclusion using secure scalar products, whereas [17] introduces a protocol for private point-circle inclusion evaluation. However, these protocols rely on SMC [18] primitives, and as a result they are very expensive and require multiple communication rounds. In contrast, our proposed point-in-rectangle and point-in-convex-polygon evaluation protocols use homomorhpic encryption, and only require a single communication round.

Closely related to our work are several methods [19–21] that use homomorphic encryption to solve the *millionaire's problem* [22]. In this setting, each of two parties owns as private input a number, and the parties want to determine who has the largest number without disclosing any of the inidividual inputs to the other party. The work in [21] provides the best performance to date, improving upon the methods in [19, 20]. However, it relies on bit-by-bit operations, which requires a number of homomorphic encryption operations linear to the bit length of the input numbers. Since in our setting we work with floating point numbers, with typical representations, a number of operations between 32 and 64 must be performed for each private comparison. In contrast, the method that we propose relaxes some of the security requirements but necessitates only two homomorphic operations per comparison.

3 System architecture and assumptions

3.1 Privacy Model

Many privacy models that rely on location cloaking have been proposed in literature [1–4, 10, 11]. The proposed hybrid approach can be used in conjunction with any of these methods. For instance, CRs can be built according to the spatial k-anonymity paradigm [1–4], which requires that at least k distinct user locations must be enclosed by the CR. Alternatively, CRs can be determined based on user-specified sensitivity

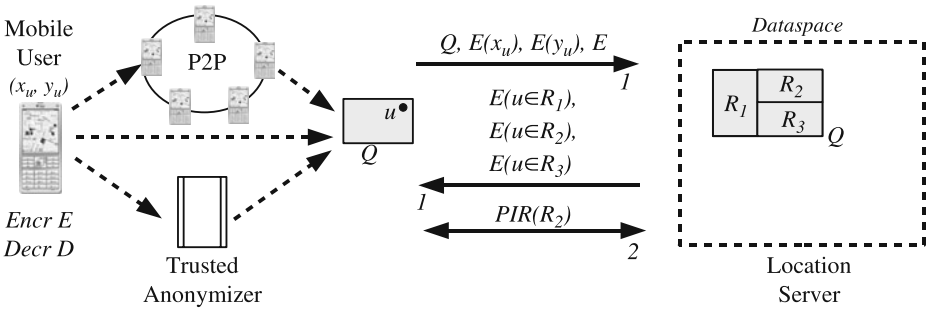


Fig. 4 System architecture

thresholds with respect to a set of sensitive feature types [10, 11]. The particular choice of privacy paradigm and CR generation technique is outside the scope of this work. We consider the CR as an input to our method, and we focus on two aspects: (i) how to efficiently perform PIR with respect to dynamically-generated CRs, and (ii) how to control tightly the amount of disclosed POIs. We do, however, factor in our system design provisions for CRs with *large* spatial extents, suitable to accommodate highly-demanding privacy requirements.

Note that, it has been discussed previously [5] that location cloaking may not be suitable for highly-mobile users issuing continuous queries. However, as shown in [9], cloaked regions can be generated in a manner that accommodates continuous queries. Furthermore, if the CR is large enough to cover an entire user trajectory, private continuous queries can be supported with strong privacy guarantees. To illustrate this claim, consider the example of user Jin, who often visits karaoke lounges. Jin wishes to keep her passion for karaoke secret, so she does not want a malicious attacker to learn that she was in the proximity of such an establishment. On the other hand, Jin may be comfortable with disclosing the fact that she is currently in Koreatown, which is a large area. In addition, while Jin remains within the perimeter of Koreatown, her privacy is protected even if she issues continuous queries. In Section 7, we experimentally evaluate our proposed method using CRs that cover large portions of the dataspace.

3.2 System Overview

The proposed system architecture is shown in Fig. 4. The system model is flexible, and can accommodate several distinct solutions for creating input CRs. For instance, users can cloak their locations by themselves, as considered in [10, 11]. Alternatively, users can send their queries to a trusted anonymizer service which creates the CRs [1–4]. Or, users can build CRs in a collaborative fashion [23–25].

Given the query CR Q, the LS returns the NN POI of the user by executing a two-round protocol, as shown in Fig. 4. In the first round (arrows labeled 1), the user[6] generates an encryption (E)/decryption (D) key pair, which are part of a

[6] Alternatively, the trusted anonymizer or a trusted peer can perform the described protocol on behalf of the user.

homomorphic encryption family, such as Paillier [26]. The user sends to the LS the query CR Q, together with the encryption (i.e., public) key E and the encrypted user coordinates $E(x_u)$ and $E(y_u)$.

The LS processes the range query with argument Q and partitions the result set into a set of disjoint regions. In the case of approximate NN queries, these regions are rectangular tiles. In the case of exact NN queries, the regions are convex polygons representing all Voronoi cells in the POI dataset tessellation that intersect query Q. For brevity, the example in Fig. 4 only shows the rectangular tiles for the approximate queries, but the concept used for exact queries is similar. Each rectangular region contains a number of POI bounded by constant F, which is a system parameter.[7] For the given query, the set of tiles $\{R_1, R_2, R_3\}$ is obtained. The LS evaluates privately, using the properties of homomorphic encryption,[8] the enclosure condition between point (x_u, y_u) and the resulting tiles. The encrypted evaluation outcome is returned to the user, who will decrypt and find which of the given rectangles encloses its location, in this case R_2. The private point-rectangle enclosure evaluation is necessary because the query result tiles can be arbitrarily small. Sending these tiles in plain text to the user (as it is done in [5], with the root of the two-level index) would give away excessive information about the distribution of POI. Finally, in the second round of the protocol, the user issues a private request for the contents of R_2, and determines which of the retrieved POI is closest to his/her location.

4 Private evaluation of point-rectangle enclosure

In this section, we introduce a two-party protocol between parties A and B, which determines privately whether a given point p owned by A is enclosed in a rectangle R owned by B. The protocol protects the privacy of both parties involved. Specifically, A learns only if the point p is enclosed by R, but does not find any additional information about the boundaries of R. In addition, B does not learn any information about the point p of A.

Our protocol relies on the Paillier public-key homomorphic encryption scheme introduced in [26]. Paillier encryption operates in the message space of integers $\mathbb{Z}_N$, where N is a large composite modulus. Similar to the PIR protocol in [14] (described in Section 2), the security of Paillier encryption relies on the QRA assumption with respect to modulus N. Denote by D and E the decryption and encryption functions, respectively. Given the ciphertexts $E(m_1)$ and $E(m_2)$ of plaintexts m_1 and m_2, the ciphertext of the sum $m_1 + m_2$ can be obtained by multiplying individual ciphertexts:

$$D(E(m_1) \cdot E(m_2)) = (m_1 + m_2) \mod N \tag{1}$$

In addition, given ciphertext $E(m)$ and plaintext $r \in \mathbb{Z}_N$, we can obtain the ciphertext of the product $r \cdot m$ by exponentiation with r, as follows:

$$D(E(m)^r) = r \cdot m \mod N \tag{2}$$

[7] In the case of exact NN queries, each convex polygon (i.e., Voronoi cell) has exactly one POI.

[8] Details about the private evaluation of point-rectangle and point-in-convex-polygon enclosure are given in Sections 4 and 6, respectively.

Furthermore, Paillier encryption provides semantic security, meaning that encrypting the same plaintext with the same public key E twice will result in distinct ciphertexts. Therefore, the scheme is secure against chosen plaintext attacks.

In our setting, the querying user wishes to find whether his/her location is enclosed inside some rectangular region R stored by the server. This can be achieved by privately evaluating the difference between the user coordinates and the boundary coordinates of rectangle R. Furthermore, to prevent leakage of POI locations, only the sign of the difference should be revealed to the user, and not the absolute value.

We introduce the protocol for private evaluation of point-rectangle enclosure in an incremental fashion. Assume that parties A and B hold two numbers a and b, respectively. In Section 4.1 we show how to privately evaluate $sign(b - a)$. Next, in Section 4.2 we give the complete protocol for point-rectangle inclusion.

4.1 Private evaluation of $sign(b - a)$

We show how to evaluate privately $sign(b - a)$ in two steps: first, we give an auxiliary protocol that privately evaluates the difference $(b - a)$. Then, we extend the auxiliary protocol to disclose only the sign of the difference, but not its absolute value. Note that, the difference protocol has no practical value by itself, since disclosing the value of $(b - a)$ to one of the parties (say A) automatically discloses the value held by the other party (since A can determine the value of b based on $b - a$ and a). However, the private difference protocol introduces a construction that is later used in the private evaluation of $sign(b - a)$.

Paillier encryption allows the computation of the ciphertext of sums based on the ciphertexts of individual terms. However, only the addition operation is supported, and not subtraction. Furthermore, the message space $\mathbb{Z}_N$ consists of positive integers only, hence the trivial solution of setting $m_1 = (-a)$, $m_2 = b$ and computing $E(m_1) \cdot E(m_2) = E(b - a)$ is not suitable. We overcome this limitation imposed on the message space by simulating complement arithmetic for N-bit integers.

Assume that $a, b \in \mathbb{Z}_{N'}$, where $N' < N$. Party A computes $m_1 = N - a$ and sends $E(m_1)$ to B, who in turn sets $m_2 = b$, and determines

$$E(m_3) = E(m_1) \cdot E(m_2) = E(m_1 + m_2) = E(N + (b - a)) \tag{3}$$

Party B returns $E(m_3)$ to A who decrypts the message and learns the value of $m_3 = N + (b - a)$. The difference $b - a$ can be computed from m_3 as shown in Fig. 5.

Let $I_1 = \{0, 1, \ldots, N'\}$ and $I_2 = \{N - N', \ldots, N - 1\}$. If $(b - a) \geq 0$, then $m_3 \in I_1$, otherwise $m_3 \in I_2$. To correctly interpret the result, it is necessary that $I_1 \cap I_2 = \emptyset$. A sufficient condition to ensure that the two intervals are disjoint is

$$[(N' - 0 + 1)] + [(N - 1) - (N - N') + 1] \leq N \Leftrightarrow N' \leq \left\lfloor \frac{N - 1}{2} \right\rfloor \tag{4}$$

Fig. 5 Determining the value of $b - a$

Private Evaluation (b-a)
Input: value a held by party A, b held by party B
Output: A learns $(b - a)$, B learns nothing

1.	A(Client):	$m_1 = N - a$
2.		Send E, $E(m_1)$ to B
3.	B(Server):	$m_2 = b$
4.		$E(m_3) = E(m_1) \cdot E(m_2)$
5.		Send $E(m_3)$ to A
6.	A(Client):	$m_3 = D(E(m_3))$
7.		**if** $(0 \leq m_3 \leq N')$
8.		$b - a = m_3$
9.		**else**
10.		$b - a = -(N - m_3)$

Fig. 6 Private evaluation of $(b - a)$

Party A determines that

$$b - a = \begin{cases} m_3, & 0 \leq m_3 \leq N' \\ -(N - m_3), & N - N' \leq m_3 \leq N - 1 \end{cases} \tag{5}$$

The pseudocode in Fig. 6 details the protocol for private computation of $(b - a)$. The protocol requires only one round of communication. Note that, A can immediately learn from $(b - a)$ the value of b. Next, we show how to protect against this inference.

We modify the protocol for evaluating $(b - a)$ to only disclose $sign(b - a)$, without revealing any additional information about b. The main idea is to multiply m_3 in the previous protocol by a random blinding factor,[9] such that the absolute value of $(b - a)$ can no longer be reconstructed by A. Consider random integer ρ uniformly distributed in the set $\{1, 2, \cdots, M\}$, such that

$$M \leq \left\lfloor \frac{N-1}{2N'} \right\rfloor \tag{6}$$

(we will give the rationale for this condition shortly). Steps 1–4 of the protocol in Fig. 6 remain unchanged. However, in step 5, instead of sending $E(m_3)$ back to A, B sends $E(m_4)$ obtained through exponentiation with plaintext ρ:

$$E(m_4) = E(m_3)^{\rho} = E(\rho \cdot m_3) = E(\rho \cdot (N + b - a)) \tag{7}$$

The value of $sign(b - a)$ can be computed from m_4 as shown in Fig. 7. In a similar manner to the protocol for difference, let $I'_1 = \{0, 1, \ldots, M \cdot N'\}$ and $I'_2 = \{N - M \cdot N', \ldots, N - 1\}$. If $(b - a) \geq 0$, then $m_4 \in I'_1$, otherwise $m_4 \in I'_2$. This time, the condition $I'_1 \cap I'_2 = \emptyset$ is equivalent to

$$[(M \cdot N' - 0 + 1)] + [(N - 1) - (N - M \cdot N') + 1] \leq N \Leftrightarrow N' \leq \left\lfloor \frac{N-1}{2M} \right\rfloor \tag{8}$$

[9] Random blinding is a frequently-used operation in cryptographic protocols [27].

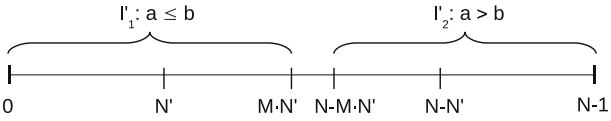


Fig. 7 Private evaluation of $sign(b - a)$

hence the requirement in Eq. 6. Party A determines that

$$sign(b - a) = \begin{cases} +1, & 0 \leq m_4 \leq M \cdot N' \\ -1, & N - M \cdot N' \leq m_4 \leq N - 1 \end{cases} \quad (9)$$

The proof of Eq. 9 is immediate: if $(a \leq b)$, then $0 \leq m_3 \leq N'$, and therefore $0 \leq \rho \cdot m_3 \leq M \cdot N'$. On the other hand, if $(a > b)$ we have $N - N' \leq m_3 < N$, therefore

$$M(N - N') \mod N \leq M \cdot m_3 < N \Leftrightarrow (N - M \cdot N') \mod N \leq M \cdot m_3 < N$$

Note that, in practice, the additional constraint imposed on the domain size N' by Eq. 8 does not represent a limitation. For security considerations, the magnitude of modulus N must be at least 768 bits large. Consider values of a and b that can be represented on 64 bits, for instance. Such values are sufficiently large for many applications. In this case, the random blinding factor domain will be bounded by $M = \frac{2^{768}}{2} \cdot \frac{1}{2^{64}}$, which is in the order of 2^{700}, sufficiently large to obtain a strong degree of protection through random blinding.

Security discussion The proposed private sign evaluation protocol (and consequently the point-rectangle enclosure evaluation protocol) inherits the security strength provided by the random blinding. Note that, this level of security is weaker than the information-theoretic security features offered by other security primitives, such as secure multi-party computation (SMC) [18], for instance. It is also weaker than computationally-secure solutions for the closely related solution in [21] to the Yao millionaire's problem. However, the above-mentioned protocols are prohibitively expensive, as discussed in Section 2.

On the other hand, random blinding offers good security features given that the blinding factors are large. As discussed above, the value of M is large. Denote by β the random blinding factor that hides the value of $(b - a)$. Within the space of 2^{700}, the random β value can have a large number of prime factors. An adversary that attempts to reconstruct the value of $(b - a)$ will factorize the product $\beta \times (b - a)$ and obtain a large number of factors. Any such factor, as well as a combination thereof, may represent a potential value for $(b - a)$. For instance, if we choose β to be the product of 10-bit numbers, (either primes or composite) then roughly 70 such numbers can be multiplied to obtain the blinding factor. Any of these numbers, or any combination of six of them (in order not to exceed the 64-bit limit, as $\lfloor 64/10 \rfloor = 6$) can represent a valid value for $(b - a)$. This results in roughly 130 million candidate combinations, equally likely to represent the value of $(b - a)$.

4.2 Private evaluation of point-rectangle enclosure

The protocol for private evaluation of point-rectangle enclosure builds upon the sign evaluation protocol of Section 4.1. Denote the user location by coordinates (x_u, y_u), and let the server-stored rectangle R be specified by its lowest-left (LL_x, LL_y) and

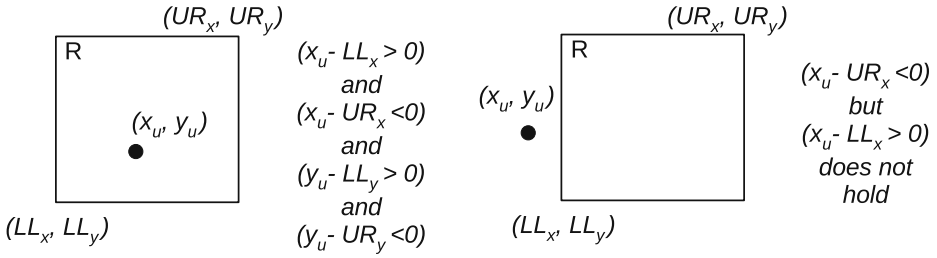


Fig. 8 Arithmetic conditions to determine point-rectangle enclosure

upper-right (UR_x, UR_y) coordinates. We maintain the notations from the previous sections, i.e., all coordinates $x, y \in \{0, 1, \ldots, N'\}$ and the random blinding factors in the set $\{0, 1, \ldots, M\}$, such that Eq. 8 is satisfied. Consider the example in Fig. 8a: the user location is situated inside the rectangle if and only if the four inequalities hold simultaneously. Conversely, if any of the inequalities does not hold (Fig. 8b), the user is outside the rectangle (or on the boundary of R).

The enclosure condition can be privately evaluated by running the $sign(b - a)$ protocol for each of the four inequalities, as shown in the pseudocode of Fig. 9. The user sends the server (lines 1 and 2) its public key E, as well as the encryption of messages m_x and m_y that encode the coordinates x_u and y_u as described in Section 4.1. The server will compute the ciphertext of the four subtraction operations (two for each of the x and y axes of coordinates), and blind them with random factors (lines 4 and 5). Note that, the protocol incurs only one round of communication. Furthermore, if the user wishes to evaluate enclosure with respect to more than one rectangle, the server can repeat the steps 4 and 5 for all rectangles, but the number of communication rounds does not increase (although the communication cost from the server to the user increases linearly to the number of rectangles).

Private Point-Rectangle Enclosure
Input: user location $p = (x_u, y_u)$, server rectangle $R = (LL_x, LL_y, UR_x, UR_y)$
Output: *true* if $p \in R$, *false* otherwise

1.	Client:	$m_x = N - x_u, m_y = N - y_u$
2.		Send E, $E(m_x)$, $E(m_y)$ to the server
3.	Server:	Generate random numbers r'_x, r'_y, r''_x, r''_y
4.		$E(m'_x) = (E(m_x) \cdot E(LL_x))^{r'_x}$, $E(m''_x) = (E(m_x) \cdot E(UR_x))^{r''_x}$
5.		$E(m'_y) = (E(m_y) \cdot E(LL_y))^{r'_y}$, $E(m''_y) = (E(m_y) \cdot E(UR_y))^{r''_y}$
6.		Send $E(m'_x)$, $E(m'_y)$, $E(m''_x)$, $E(m''_y)$ to the client
7.	Client:	**if** ($(0 \le m''_x \le M \cdot N')$ **and** $(N - M \cdot N' \le m'_x \le N - 1)$ **and**
8.		$(0 \le m''_y \le M \cdot N')$ **and** $(N - M \cdot N' \le m'_y \le N - 1)$)
9.		**return true**
10.		**else**
11.		**return false**

Fig. 9 Protocol for private evaluation of point-rectangle enclosure

In practice, spatial coordinates are represented as floating point numbers, either in single (32-bit) or double (64-bit) precision. On the other hand, Paillier encryption requires the use of positive integers alone. Nevertheless, the message space $\mathbb{Z}_N$ is large enough to accommodate even the most demanding application requirements with respect to coordinate precision. During the protocol execution, floating point values are converted to fixed precision. For instance, assume that the spatial data domain is $[0, 10^6]^2$ and 6 decimal points are required. Then, $2 \cdot \lfloor \log(10^6) \rfloor = 34$ bits are sufficient for this representation, much lower than the magnitude of N. This leaves a very large domain for the values of the random blinding factors.

5 Hybrid protocol for approximate nearest-neighbor query processing

We introduce a technique for processing PIR requests with respect to dynamically-generated query CRs. This method overcomes the drawbacks of [5] (discussed in Section 2), which performs PIR with respect to the entire POI dataset D. In the hybrid approach, the server knows that the user is located inside query CR Q, and therefore it can return a query result which discloses fewer POI and incurs less overhead.

A naive approach to restrict the set of POI included in the PIR protocol would work as follows: first, the server determines the set P_Q of POI that are located inside Q. Next, the points in P_Q are bulk-loaded into a two-level spatial index. Finally, the PIR retrieval is performed as in [5] with respect to the obtained index. There are several drawbacks of this approach: first, the index must be built on-line, which is time consuming. Second, although the number of disclosed POI is reduced from $\sqrt{|D|}$ to $\sqrt{|P_Q|}$, the resulting POI count can still be quite large, and it depends on the query Q (hence, it is not constant). Third, the root node of the index is sent in plain-text to the user. This discloses excessive information about the distribution of POI, since the minimum bounding rectangles (MBRs) of the leaf nodes may be small in size (especially if Q is not very large). The proposed hybrid technique addresses all these limitations.

The requirement of a two-level index restricts the flexibility in determining customized results for dynamic query CRs. We employ a multi-level index structure (computed off-line) that can efficiently find at run-time the leaf nodes that intersect query Q. Furthermore, we choose an index structure that strictly bounds the leaf node cardinality below a threshold F. Another important factor in developing the index structure is the fact that the cryptographic protocol of Section 4 allows private evaluation of point-rectangle inclusion, but not distance evaluation. This is a direct consequence of protecting the location of the POI. In order to ensure query correctness (i.e., that at least one of the leaf nodes includes the user location) we employ a space-partitioning index, rather than a data-partitioning one. We provide more details about the indexing structure used in Section 5.1.

Figure 10 gives an overview of the entire query processing protocol. In step *(a)*, the user sends to the server the CR Q, as well as the encrypted user coordinates $E(x_u)$ and $E(y_u)$. The server processes a range query with parameter Q (step *(b)*) and identifies all leaf nodes (in this case, R_1 and R_2) that intersect Q. The server also executes the private point-rectangle evaluation protocol (Section 4) and sends back to the user (step *(c)*) tuples $(id(R_i); E((x_u, y_u) \in R_i))$, i.e., a rectangle identifier and

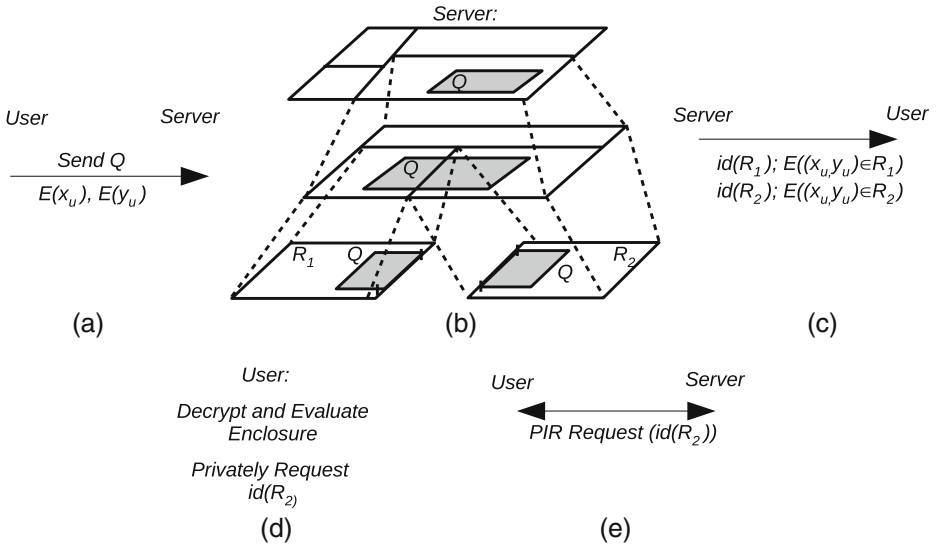


Fig. 10 Hybrid technique overview

the encrypted result of enclosure evaluation.[10] Next, in step *(d)*, the user decrypts the enclosure evaluation results and determines the identifier of the leaf node[11] that encloses (x_u, y_u), in this case R_2. Finally, the user and the server engage in a PIR round to retrieve the contents of R_2 (step *(e)*).

For clarity of presentation, we have highlighted each step individually. However, there are only two communication rounds, as in the case of [5].

5.1 Indexing structure

The choice of POI indexing structure is very important to the objectives of minimizing the POI disclosure and reducing query processing overhead. We consider a structure reminiscent of *k*-d-trees [28], which recursively cuts the space based on the number of data points in each partition. However, as opposed to *k*-d-trees, we do not require partition cuts to intersect data points. Furthermore, we do not restrict the axis of the cut at each step, and we use a more advanced split heuristic that factors criteria such as the perimeter of resulting partitions.

Consider the example of Fig. 11a, where the data is split according to median cut C_1, resulting in two sub-sets of equal cardinality (four points each). Assume that the node capacity is $F = 3$. Two additional splits are performed according to cuts C_2

[10] Note that, if disclosing the number of leaf nodes that intersect Q represents a privacy concern for the database, the server can include randomly generated rectangles (that do not intersect Q) in the enclosure evaluation phase, without affecting correctness.

[11] Due to the non-overlapping indexing of POI, exactly one rectangle will enclose the user.

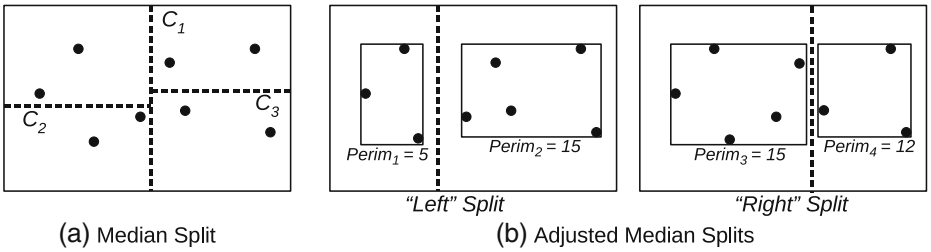


Fig. 11 Split heuristic

and C_3, resulting in four leaf nodes of two points each. The median split has two drawbacks: first, the number of POI retrieved by the user is less than the allowed value 3, which may decrease the result accuracy. Second, there are a total of four leaf nodes, although the original 8 points could be split into $\lfloor 8/3 \rfloor = 3$ nodes. A larger number of leaf nodes increases the cost of the point-rectangle enclosure evaluation.

We employ a variation of the median split that controls tightly the cardinality of leaf nodes. Given the cardinality c of the current partition, we ensure that at least one of the resulting partitions is a multiple of F. If this requirement is met at each cut, the amount of fragmentation (which is the reason why the median split underperformed) is considerably reduced. Consider Fig. 11b: there are two candidate splits across the x axis, $Left$ and $Right$. $Left$ places $\lfloor c/2/F \rfloor \cdot F$ points to the left of the cut axis and $c - \lfloor c/2/F \rfloor \cdot F$ to the right, whereas $Right$ places $(\lfloor c/2/F \rfloor + 1) \cdot F$ points to the left and $c - (\lfloor c/2/F \rfloor + 1) \cdot F$ to the right. For each of these candidates, a

NodeSplit
Input: Initial Node U, Leaf Cardinality Threshold F
Output: Two children nodes U_1 and U_2
/* $x - axis$ */
1. sort points in U increasingly according to x coordinate
   /* We use array notation to refer to the ponts in U */
   /* "$Left$" split* /
2. $Count_{left} = \lfloor |U|/2/F \rfloor \cdot F$
3. $Cost_{left} = perimeter(MBR(\{U[1], \ldots, U[Count_{left}]\}) +$
   $perimeter(MBR(\{U[Count_{left} + 1], \ldots, U[|U|]\})$
   /* "$Right$" split* /
4. $Count_{right} = (\lfloor |U|/2/F \rfloor + 1) \cdot F$
5. $Cost_{right} = perimeter(MBR(\{U[1], \ldots, U[Count_{right}]\}) +$
   $perimeter(MBR(\{U[Count_{right} + 1], \ldots, U[|U|]\})$
6. **if** $(Cost_{left} < Cost_{right})$
7. $U_1 = U[1 \ldots Count_{left}], U_2 = U[Count_{left} + 1 \ldots |U|]$
8. **else**
9. $U_1 = U[1 \ldots Count_{right}], U_2 = U[Count_{right} + 1 \ldots |U|]$
/* Repeat steps $1 - 9$ for $y - axis$ and choose the lowest cost*/

Fig. 12 Heuristic for index partitioning

benefit metric is evaluated, which measures the sum of perimeters[12] for the minimum bounding rectangles of points in each partition. The candidate that minimizes the sum of perimeters (in the example the $Left$ split) is chosen.[13] A similar evaluation of candidate splits is performed for the y axis. Figure 12 shows the pseudocode of the proposed *NodeSplit* technique for data partitioning. *NodeSplit* considers both the x and y axes, and chooses the split with the largest benefit (i.e., minimum sum of perimeters). Data points in the initial node U are sorted with respect to the selected axis (line 1). Next, the costs of the candidate splits $Cost_{left}$ and $Cost_{right}$ are evaluated as the sum of perimeters for the points in each region (lines 2–5). The split position that yields the lowest cost is chosen (lines 6–9). The computational complexity of the index creation is $O(|D| \log |D|)$, where $|D|$ is the dataset cardinality.

6 Hybrid protocol for exact nearest-neighbor query processing

So far, we considered approximate queries, for which an inherent trade-off exists between the amount of disclosed POI and accuracy of the results. The more disclosed POI, chances increase that one of them is the actual NN, or the distance from the user to the approximate NN is close to the distance to the actual NN. However, an ideal private query technique should return a single POI that is the *exact* NN. In this case, an optimal outcome is achieved from both the querying user's point of view, who receives his or her exact NN, as well as from the database point of view, since only a single data point is disclosed per query. In this section, we will present a method that achieves this optimal result.

The previous work in [5] has introduced a method (described in Section 2) for private exact NN queries that employs Voronoi tessellations and PIR. The idea is to use a regular 2D grid and to create a bucketing scheme where each cell in the 2D grid is assigned all data points whose Voronoi cells intersect the grid cell. At query time, the client performs a PIR query for the grid cell that encloses his or her location. Although the method guarantees that the exact NN is part of the result received by the user, the bucketization from Voronoi cells to grid cells involves an inherent loss of precision, due to the infeasibility of having a grid fine-grained enough such that only a single Voronoi cell is hashed in each grid cell. In fact, experimental results from [5] with a real-life dataset show that the average number of data points hashed in a grid cell (and hence disclosed in a single query) is 15. This number is far from the optimal 1. Using a more fine-grained 2D grid cell could potentially lower this number, but results into a rather large computational and communication overhead since the computational complexity of PIR is linear to the number of grid cells. Furthermore, if data are skewed, using a finer grid does not necessarily translates into a decrease in disclosed POI.

There is another important disadvantage of the Voronoi cell bucketization solution: in addition to the total number of grid cells, the maximum number of Voronoi

[12] A similar benefit metric has been used for R-trees [28].

[13] Although the MBRs are used in the benefit evaluation, the resulting partition is not pruned to the MBR, due to the requirement that the index must cover the entire data space.

cells hashed in any one grid cell is also a factor with linear influence on the PIR complexity, this time both in terms of computational and communication cost. Specifically, this collision factor dictates the "depth" of the PIR matrix, in other words the PIR protocol is executed separately for each such colliding point. Note that, even if there is a single grid cell with a large number of collisions, this affects all other grid cells as well, since the server must not be able to distinguish which grid cell is retrieved based on the depth of that cell, hence all cells must be padded to the depth of the grid cell with most collisions. To exacerbate the problem, this maximum depth parameter depends on the dataset, and no upper bound can be determined: in the worst case, the depth is linear to the database size, leading to excessive disclosure of data points and high processing overhead.

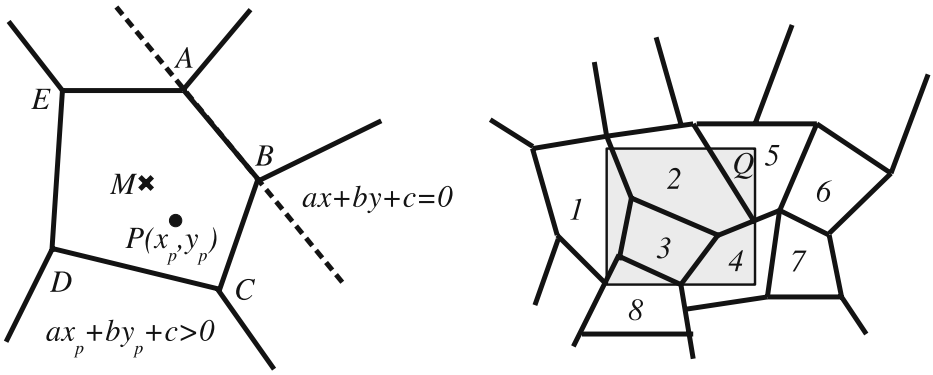
All the above-mentioned limitations can be tracked down to a single factor: within the PIR framework alone, the client and the server do not have the means to perform any form of interactive, privacy-preserving filtering of results without disclosure. Specifically, all that the client is able to do is to retrieve privately a fraction of the database, but the retrieval is done based on data item index, and not directly based on spatial information. The only time that spatial information is factored in is in the bucketization phase, but as we have discussed, the precision of the bucketization may be low. Furthermore the bucketization is done independent of the query. Therefore, it may not be possible to find a bucketization that optimally serves all queries. We propose a different approach, in which the query processing makes use of the techniques developed in Section 4.1 for interactive evaluation of arithmetic conditions using homomorphic encryption. In particular, we will show that the user can privately identify the index of the Voronoi cell that the user belongs to *without* need for bucketization.

Voronoi cells in two dimensions are convex polygons. Each side of the polygon belongs to a line with equation $ax + by + c = 0$. A well-known procedure from computational geometry [28] states that it can be determined whether a point $P(x_p, y_p)$ is included in a convex polygon by replacing the variables in the line equations of the polygon sides with the point coordinates. Specifically, denote by ℓ the number of polygon sides (which is also the number of vertices) and denote by

$$\begin{aligned} a_1 x + b_1 y + c_1 &= 0 \\ a_2 x + b_2 y + c_2 &= 0 \\ &\cdots \\ a_\ell x + b_\ell y + c_\ell &= 0 \end{aligned} \tag{10}$$

the set of equations corresponding to the polygon sides traversed in clockwise direction along the polygon perimeter. If the sign of all expressions obtained by replacing the coordinates of P in all equations in Eq. 10 is positive (or zero), then the point P is inside (or on the edge) of the polygon. Figure 13a illustrates this concept. For instance, by replacing the coordinates of the mass center M of the polygon in the equations of the individual sides, a non-negative value is always obtained for each expression.

Using this simple procedure in conjunction with homomorphic encryption allows clients to learn the index of the particular Voronoi cell they belong to, without



(a) Checking Inclusion of a Point in a Convex Polygon

(b) Filtering Voronoi Cells based on query CR

Fig. 13 Enclosure evaluation and filtering for Voronoi cells

learning neither the extent of that cell, nor the extent of any other Voronoi cell for that matter. In fact, the only information that the user learns is how many Voronoi cells are in the dataset (or in the window corresponding to the query CR). Note that, the expressions that need to be evaluated have linear form, hence the algorithms for private evaluation of the sign of sum/difference from Section 4 can be reused without change. Similar to the hybrid model used for approximate NN queries, where only intersecting rectangular tiles were included in the computation, the query CR Q is used to filter the Voronoi cells that are candidates for the exact NN result, as shown in Fig. 13b. Any existing spatial index structure can be used to efficiently determine matching cells. In our implementation, we employ R^*-trees to index Voronoi cells. In the remainder of this section, we focus on the privacy-preserving protocol for point-in-convex-polygon enclosure that is performed after filtering.

Note that, Paillier homomorphic encryption operations [26] are rather expensive in terms of computational cost. For the approximate NN algorithm of Section 4, only four evaluations (one for each side of a rectangle) were needed, and the number of rectangles was relatively small compared to the number of points. A Voronoi cell can have a number of sides considerably larger than four. However, we make the following observation, which indicates that determining polygon inclusion with Paillier encryption may be a feasible approach, at least as long as the size of the query range Q does not grow large: The most expensive Paillier operations are encryption and decryption. On the other hand, operations with ciphertexts are relatively inexpensive. We have measured the relative performance of these operations, and found that while addition and multiplication under the ciphertext for a 768-bit N amount to roughly 0.01 and 0.18 ms, respectively, encryption amounts to 15 ms, two orders of magnitude higher (experimental settings are described in Section 7). As we show next, the server only needs to perform a single encryption operation per polygon side,

for the free term c in the line equation. All other operations are ciphertext-ciphertext multiplications or ciphertext-plaintext exponentiations.

Similar to the approximate NN protocol, we assume that real-valued point coordinates are converted to integers (for a large-enough value of N the loss of precision is negligible). The protocol executed by the client and server to privately answer exact NN queries consists of the following steps:

1. The client, situated at location (x_p, y_p), generates a public key E and private key D for Paillier encryption with modulus N. The client sends the server public key E, the CR Q, as well as the ciphertexts $E(x_p)$, $E(y_p)$, $E(N - x_p)$, $E(N - y_p)$ (the need for the latter pair of ciphertexts will become evident in Step 2)
2. For each Voronoi cell v_j that intersects Q, denote by ℓ_j the number of sides of the polygon representing v_j. For every side i, $1 \leq i \leq \ell_j$, with corresponding line equation $a_i^j x + b_i^j y + c_i^j = 0$ the server computes the value

$$M_i^j = E(x_p)^{a_i^j} \times E(y_p)^{b_i^j} \times E\left(c_i^j\right)$$

   The value M_i^j corresponds to the ciphertext of the value that indicates if the user coordinates are inside the cell v_j with respect to side i. Only the sign of the value is required to evaluate enclosure, and to protect the spatial extent of the cell from the client, the server blinds the value with a multiplicative random constant $r > 0$:

$$M'^j_i = \left(M_i^j\right)^r$$

   All values M'^j_i are returned to the client. Note that, if any of the constants a_i^j or b_i^j are negative, then the above operations are done with respect to the absolute value of these constants, and the ciphertexts $E(N - x_p)$, $E(N - y_p)$ are used instead. This procedure solves the issue of Paillier encryption not supporting directly subtraction (a similar mechanism was explained in detail in Section 4 for approximate NN). In addition, to prevent the client from inferring any information about the geometry of adjacent cells based on the number of polygonal sides in consecutive cells, the M' values can be randomly permuted with respect to their j coordinate. In other words, the client receives the ciphertexts of blinded enclosure expressions of cells in random order (although the grouping of polygon sides with respect to each cell is preserved intact).
3. The client decrypts the received ciphertexts, and checks to see which cell j_0 has all values $D(M'^{j_0}_i)$ positive. Note that, as a performance optimization, if at least one M'^j_i value for the currently considered cell j is negative, the rest of the ciphertexts for that particular cell need not be decrypted, since it is clear that the enclosure condition does not hold for that cell. Finally, after identifying the enclosing Voronoi cell index, the client performs a PIR retrieval for the cell with index j_0. Note that, it is guaranteed that the PIR object associated to a Voronoi cell contains a single data point, hence the PIR phase is much more efficient than for either approximate NN or the exact NN protocol from [5], for which there are a large number of data points associated with each PIR matrix item.

At the end of the protocol, the client learns the value of a *single* data point which is the exact NN of the client's location.

7 Experiments

We evaluate experimentally the proposed hybrid methods for approximate and exact nearest-neighbor queries with respect to the effectiveness in controlling the disclosed POI and the incurred computational and communication overhead. We use a real database with points of interest: the Seqouia set[14] with 62,556 data points (Fig. 14). We consider values of F, the threshold for disclosed POI, in the range 20–80, and we randomly generate square-shaped cloaking regions Q with side between 1 and 10% of the dataspace side. Recall that, a larger CR provides stronger privacy for the user. For each experimental run, we randomly generate 1,000 user queries. The size of the modulus N used in the cryptographic protocols for PIR retrieval and private enclosure evaluation is 768 bits. The experiments were run on an Intel P4 3.0 GHz machine with 1GB of RAM. In Section 7.1 we evaluate the POI disclosure incurred by approximate NN methods. Section 7.2 compares head-to-head the performance of the pure-PIR versus hybrid methods for approximate NN queries, whereas Section 7.3 compares pure-PIR versus hybrid methods for exact NN queries.

7.1 POI disclosure

We evaluate the amount of protection offered to the database by the hybrid approximate NN method, in comparison with location cloaking (label *CR-only*) and the pure-PIR approximate technique from [5] (label *PIR-only*), for varying CR size. We consider only approximate NN queries, since the exact NN method presented in Section 6 is optimal with respect to disclosure (i.e., exactly one POI is returned to the user). For fairness of comparison, only candidate POI inside Q are returned by the CR-only method (this decreases the number of disclosed POI compared to the exact methods in [3, 4]). Figure 15 shows that the CR-only technique discloses an excessive amount of POI, especially as the CR size grows larger. Therefore, the privacy of the database is sacrificed for the sake of user privacy. The PIR-only method does not use CRs, and always discloses approximately 250 POI (square root of database cardinality). Note that, the hybrid method controls strictly the number of disclosed POI in the narrow band 20–80, up to one order of magnitude superior to PIR-only, and up to two orders of magnitude better than the CR-only method. This improvement is obtained for the same level of privacy offered to the user by the CR-only method (i.e., same CR sizes).

7.2 Performance comparison for approximate NN algorithms

In the following experiments, we compare the performance between the hybrid and the PIR-only approximate NN methods with respect to computational and communication overhead incurred by query processing. We do not include the CR-only method any further in the head-to-head comparison, since it offers virtually no amount of protection for the database. It is, however, well-understood [5] that CR-only techniques are more efficient in terms of overhead, because they do not make

[14] http://www.rtreeportal.org

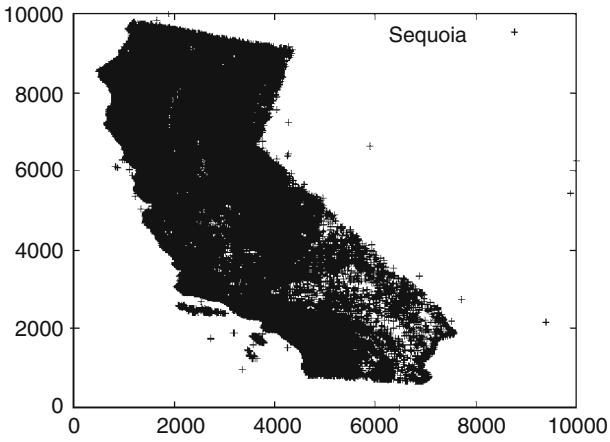


Fig. 14 Sequoia dataset

use of cryptographic elements. In general, the processing time is expected to take on average around one second [4].

Similar to previous work [4, 5], we consider that the set of POIs fits in memory, and that the processing time is dominated by CPU time. This is a reasonable assumption, especially since the compared methods use heavily cryptographic transformations, which are not I/O bound. Note that, in [5] optimizations based on parallel processing are proposed to improve execution time. Such optimizations are directly applicable for the hybrid methods as well. In our tests, we run both methods on a single-CPU machine, and we report the hybrid method execution time as the percentage of the time incurred by the PIR-only method.

Figure 16a shows the execution time when varying the POI disclosure bound F. In the worst case, the hybrid method is twice as fast as the PIR-only method. On the other hand, for all CR sizes with less than 10% of the dataspace side, the hybrid method is at least five times faster. The decreasing trend with F can be explained as follows: since the size of query Q is fixed, the number of POI included in the PIR step does not vary with F. On the other hand, a smaller F results into more rectangles for

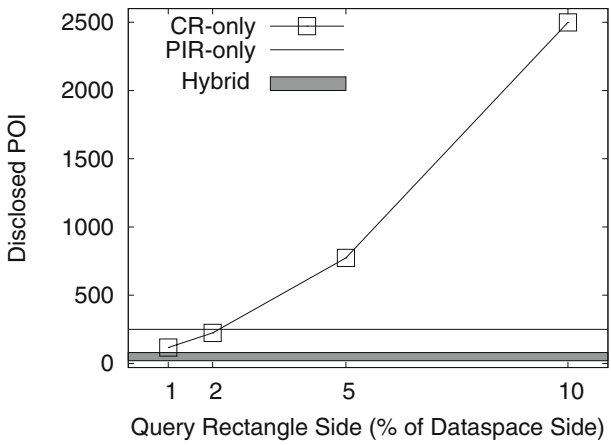
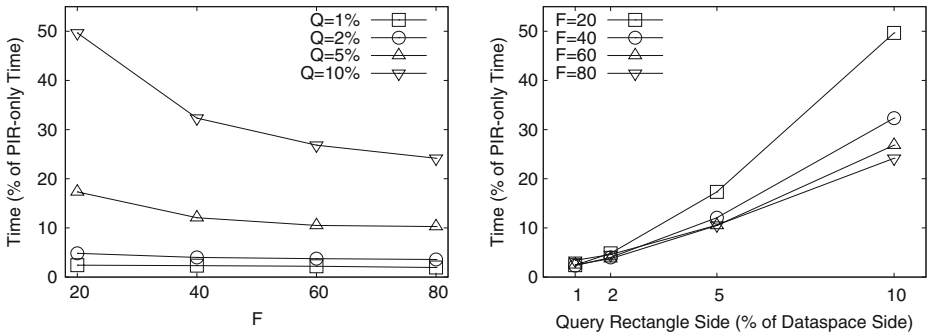


Fig. 15 Number of disclosed POI

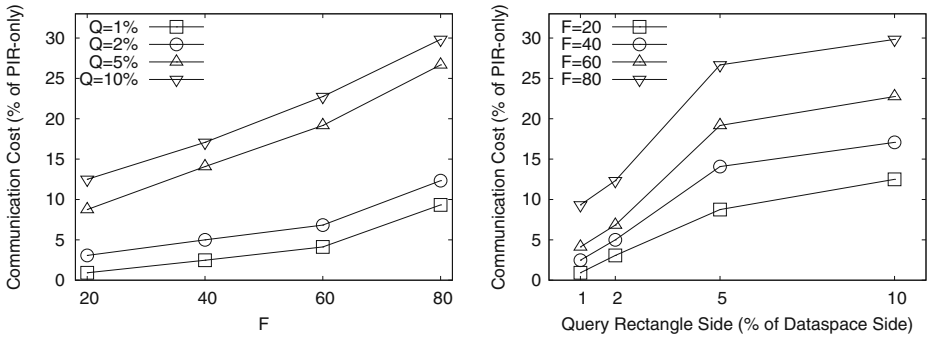


(a) Varying F (b) Varying CR Size

Fig. 16 Execution time for approximate NN methods

which the private point-rectangle enclosure evaluation protocol must be performed, leading to an increase in processing time. In absolute values, the execution time of the hybrid method on a single CPU requires roughly 0.5 s for queries spanning 2% of the dataspace, and between 1.2 and 1.9 s for queries spanning 5% of the dataspace. Figure 16b shows the variation of execution time with query CR size. A larger query window translates into more leaf nodes being included in the enclosure evaluation protocol. Furthermore, a larger number of data points are considered in the PIR retrieval phase. Hence the increase in processing time. In summary, the proposed method improves on its non-hybrid counterpart by a factor between 2 and 20, and in most cases the factor is larger than 5. Therefore, whereas the non-hybrid method takes about one second to execute on a 8-CPU machine [5], the hybrid method offers a competitive average of 0.2 s per query.

Figure 17 presents the result of communication overhead, also expressed as a percentage of the overhead incurred by the PIR-only method. In the worst case, the bandwidth consumption of the hybrid method is 30% that of PIR-only, whereas the overall improvement can be as high as 20 times. The cost increases with F (Fig. 17a) since more POI are retrieved from the server. For varying size of CR Q (Fig. 17b),



(a) Varying F (b) Varying CR Size

Fig. 17 Communication cost for approximate NN methods

the number of retrieved POIs remains unchanged as Q grows, but the number of leaf nodes considered in the point-rectangle enclosure protocol increases, hence the higher communication overhead. In absolute values, the communication cost of the hybrid method is in the range 40–140 KB for queries spanning 2% of the dataspace, and 100–280 KB for queries spanning 5% of the dataspace.

Finally, Table 1 shows the accuracy of NN results. Since both compared methods are approximative, the closest POI reported to the user may differ from the actual NN POI. Accuracy is measured as the average difference between the user-to-reported-NN distance and the user-to-actual-NN distance. The value is then normalized, and expressed as a percentage of dataspace side. Since the data points that are returned to the user depend only on the leaf node that encloses the user location, the accuracy of the hybrid method is independent of the query size. The only factor that influences accuracy is the POI disclosure threshold F. The accuracy of the PIR-only method is better, since it returns an excessive amount of POI to the user. On the other hand, in absolute values, the hybrid method achieves good precision. For instance, assume a city area of 50×50 km. An approximation error of 0.014% corresponds to a distance of 28 m. This is a reasonable error, considering that in practice, positioning devices report locations with accuracy of 10–20 m.

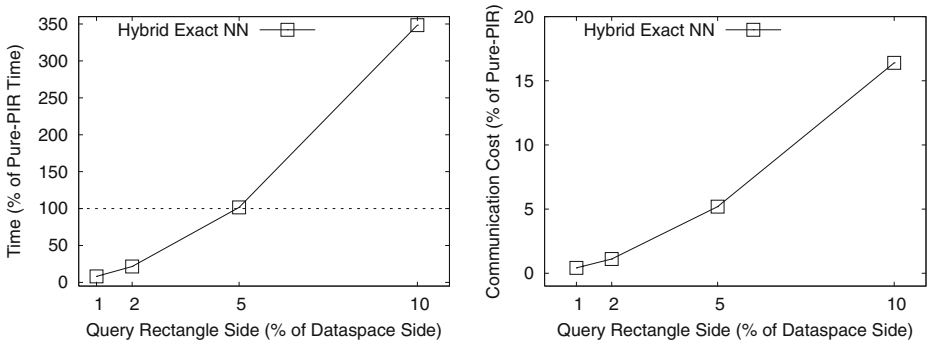
7.3 Performance comparison for exact NN algorithms

In this experiment, we compare head-to-head the computation time and communication cost of the hybrid exact NN method described in Section 6 with the pure-PIR exact NN algorithm from [5]. Recall that, the hybrid algorithm is optimal with respect to the POI disclosure, whereas the pure-PIR method discloses on average 15 POI per query, as measured in [5]. The results are shown for the hybrid method, expressed as a percentage of the values measured for the pure-PIR method (e.g., a 50% value means that the execution time required by the hybrid method is half that of the pure-PIR method).

Figure 18a shows the relative execution time of the hybrid method for variable query CR range Q. As expected, for smaller CRs, the performance of the hybrid method is net superior, between one and two orders of magnitude faster than the pure-PIR method. There are two reasons for this gain: first, fewer Voronoi cells are considered in the private convex polygon enclosure algorithm of Section 6, and second, the depth of the PIR matrix (i.e., the number of actual data points enclosed in a PIR item) is 1 for the hybrid method, whereas it is much larger for the pure-PIR method. However, as the size of the query range grows, the overhead required by the convex polygon enclosure computation increases, and for a query with side 5% of the dataspace, the advantage of the hybrid method diminishes. This is due solely to the expensive nature of the homomorhpic encryption operations. For the query

Table 1 Query result accuracy

Threshold F	Hybrid accuracy (%)	PIR-only accuracy (%)
20	0.014	0.003
40	0.011	
60	0.007	
80	0.005	



(a) Execution Time (b) Communication Overhead

Fig. 18 Exact NN algorithms: computational and communication overhead

with side 10% of the dataspace, the execution time is three and a half times worse than that of the pure-PIR method.

Figure 18b shows a similar trend with respect to query CR range. However, the communication cost is always lower, by a large margin, compared to the pure-PIR method. This can be explained by the fact that in the hybrid method, there is only one data point per PIR matrix object, whereas in the pure method there are a large number of "placeholder" points required to deal with the varying density of data across grid cells in different parts of the dataspace. In the worst case, the communication cost incurred by the hybrid method is 18% that of the pure-PIR technique.

In summary, despite the more complex procedure for query evaluation and the expensive nature of homomorphic encryption operations, the hybrid exact NN method outperforms its pure-PIR counterpart by a large margin for the lower end of the query CR range spectrum. On the other hand, as the query CR size increases, the performance of the hybrid method deteriorates quickly. Nevertheless, a query with side length that is 5% of the dataspace may be sufficient for many application scenarios. Furthermore, the hybrid exact NN method provides optimal POI disclosure.

8 Conclusions

This paper proposed hybrid techniques for approximate and exact private NN queries which provide protection for both the users and the service provider. Our solutions rely on cryptographic protocols for private evaluation of point-in-rectangle and point-in-convex-polygon enclosure. The hybrid techniques achieve far lower disclosure of POI compared to their CR-only and PIR-only counterparts. In fact, the hybrid exact NN method is optimal with respect to POI disclosure.

The proposed techniques are also efficient in practice, and outperform pure-PIR methods in most cases, with the sole exception of large-CR queries for the optimal-disclosure exact NN solution. In future work, we plan to extend our work to support private evaluation of more advanced spatial conditions and more complex types of queries, e.g., kNN queries for $k > 1$ and skyline queries.

References

1. Gruteser M, Grunwald D (2003) Anonymous usage of location-based services through spatial and temporal cloaking. In: Proc. of USENIX MobiSys
2. Gedik B, Liu L (2005) Location privacy in mobile systems: a personalized anonymization model. In: Proc. of ICDCS, pp 620–629
3. Mokbel MF, Chow CY, Aref WG (2006) The new Casper: query processing for location services without compromising privacy. In: Proc. of VLDB, pp 763–774
4. Kalnis P, Ghinita G, Mouratidis K, Papadias D (2007) Preserving location-based identity inference in anonymous spatial queries. IEEE TKDE 19(12):1719–1733
5. Ghinita G, Kalnis P, Khoshgozaran A, Shahabi C, Tan KL (2008) Private queries in location based services: anonymizers are not necessary. In: SIGMOD, pp 121–132
6. Kido H, Yanagisawa Y, Satoh T (2005) An anonymous communication technique using dummies for location-based services. In: International conference on pervasive services (ICPS), pp 88–97
7. Yiu ML, Jensen C, Huang X, Lu H (2008) SpaceTwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: International conference on data engineering (ICDE), pp 366–375
8. Cheng R, Zhang Y, Bertino E, Prahbakar S (2006) Preserving user location privacy in mobile data management infrastructures. In: Privacy enhancing technologies (PET), pp 393–412
9. Chow CY, Mokbel MF (2007) Enabling private continuous queries for revealed user locations. In: SSTD, pp 258–275
10. Gruteser M, Liu X (2004) Protecting privacy in continuous location-tracking applications. IEEE Secur Priv 2:28–34
11. Damiani M, Bertino E, Silvestri C (2008) PROBE: an obfuscation system for the protection of sensitive location information in LBS. Technical report 2001-145, CERIAS
12. Khoshgozaran A, Shahabi C (2007) Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: SSTD, pp 239–257
13. Chor B, Goldreich O, Kushilevitz E, Sudan M (1995) Private information retrieval. In: IEEE symposium on foundations of computer science, pp 41–50
14. Kushilevitz E, Ostrovsky R (1997) Replication is NOT needed: SINGLE database, computationally-private information retrieval. In: FOCS, pp 364–373
15. Flath DE (1998) Introduction to number theory. Wiley, New York
16. Atallah MJ, Du W (2001) Secure multi-party computational geometry. In: WADS '01: Proceedings of the 7th international workshop on algorithms and data structures, pp 165–179
17. Luo Y, Huang L, Zhong H (2007) Secure two-party point-circle inclusion problem. J Comput Sci Technol 22(1):88–91
18. Goldreich O, Micali S, Wigderson A (1987) How to play any mental game. In: Proceedings of ACM symposium on theory of computing (STOC), pp 218–229
19. Fischlin M (2001) A cost-effective pay-per-multiplication comparison method for millionaires. In: CT-RSA 2001: Proceedings of the 2001 conference on topics in cryptology, pp 457–472
20. Blake IF, Kolesnikov V (2004) Strong conditional oblivious transfer and computing on intervals. In: Advances in cryptology—ASIACRYPT 2004, pp 515–529
21. Lin HY, Tzeng WG (2005) An efficient solution to the millionaires' problem based on homomorphic encryption. In: Intl. conference on applied cryptography and network security, pp 456–466
22. Yao AC (1982) Protocols for secure computations. In: SFCS '82: Proceedings of the 23rd annual symposium on foundations of computer science, pp 160–164
23. Chow CY, Mokbel MF, Liu X (2006) A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: GIS, pp 171–178
24. Ghinita G, Kalnis P, Skiadopoulos S (2007) PRIVE: anonymous location-based queries in distributed mobile systems. In: WWW, pp 371–380
25. Ghinita G, Kalnis P, Skiadopoulos S (2007) MobiHide: a mobile peer-to-peer system for anonymous location-based queries. In: SSTD, pp 221–238
26. Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT, pp 223–238
27. Atallah MJ (1998) Algorithms and theory of computation handbook. CRC Press, Boca Raton
28. de Berg M, van Kreveld M, Overmars M, Schwarzkopf O (2000) Computational geometry: algorithms and applications, 2nd edn. Springer, Berlin

Gabriel Ghinita received a PhD degree in computer science from the National University of Singapore. He is a postdoctoral research associate in the Department of Computer Science, Purdue University, and a member of the Center for Education and Research in Information Assurance and Security (CERIAS). His research interests focus on access control for collaborative environments, privacy for spatial and relational data, and data management in highly distributed environments. In the past, he held visiting scientist appointments with the Hong Kong University and the Chinese University of Hong Kong. He served on the program committees of VLDB, ACM SIGSPATIAL GIS, and WWW, and as a reviewer for the IEEE Transactions on Knowledge and Data Engineering, VLDB Journal, and ACM Transactions on Information and System Security. He is a member of the IEEE and the IEEE Computer Society.

Panos Kalnis received the diploma in computer engineering from the Computer Engineering and Informatics Department, University of Patras, Greece, and the PhD degree from the Computer Science Department, Hong Kong UST. He is an associate professor of computer science at the King Abdullah University of Science and Technology (KAUST), Saudi Arabia. Prior to joining KAUST, he was a visiting scientist at Stanford University and an assistant professor at the National University of Singapore. His research interests include anonymity, peer-to-peer systems, mobile computing, OLAP, data warehouses, and spatial databases.

Murat Kantarcioglu is an Assistant Professor in the Computer Science Department and Director of the UTD Data Security and Privacy Lab at the University of Texas at Dallas. His research focuses on creating technologies that can efficiently extract useful information from any data without sacrificing privacy or security. Recently his work focuses on security and privacy issues raised by data mining, privacy issues in social networks, security issues in databases, privacy issues in health care, risk and incentive issues in assured information sharing, use of data mining for fraud detection and homeland security. He holds a B.S. in Computer Engineering from Middle East Technical University, and M.S. and Ph.D degrees in Computer Science from Purdue University. He has published over 60 papers including in premier journals such as VLDB, IEEE TKDE, IEEE TITB, and ACM TKDD and prestigious conferences such as ACM KDD, EDBT, IEEE ICDE, PKDD, WWW and ACM SACMAT. He is also a recipient of NSF Career award.

Elisa Bertino is professor of computer science at Purdue University and Research Director of the Center for Information and Research in Information Assurance and Security (CERIAS). Prior to joining Purdue, she was a professor and department head at the Department of Computer Science and Communication of the University of Milan. She has been a visiting researcher at the IBM Research Laboratory (now Almaden) in San Jose, at the Microelectronics and Computer Technology Corporation, at Rutgers University, at Telcordia Technologies. Her recent research focuses on database security, digital identity management, policy systems, and security for web services. She is a Fellow of ACM and of IEEE. She received the IEEE Computer Society 2002 Technical Achievement Award and the IEEE Computer Society 2005 Kanai Award. She a member of the editorial board of IEEE Transactions on Dependable and Secure Computing, and IEEE Security & Privacy. She is currently serving as chair of the ACM Special Interest Group on Security, Audit and Control (ACM SIGSAC).