Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 1685054, 10 pages
http://dx.doi.org/10.1155/2016/1685054

Research Article

Efficient and Secure Top-k Query Processing on Hybrid Sensed Data

Haiqin Wu and Liangmin Wang

Department of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China

Correspondence should be addressed to Liangmin Wang; wanglm.ujs@gmail.com

Received 15 September 2016; Accepted 6 November 2016

Academic Editor: Christophe Guyeux

The ubiquity of mobile devices equipped with various sensors has promoted the advent of a novel data sensing paradigm. Based on the traditional static sensing mode, the mobile sensing (sensor) nodes collaboratively collect data with the static sensor nodes. This large volume of hybrid sensed data is then sent to the storage nodes for flexible management and top-k query services. One crucial security issue is that the compromised storage node may falsify or drop some data during the query processing, which returns fake or incorrect result to the query users. In this paper, we propose an efficient and verifiable scheme (EVTopk) for secure top-k query processing on hybrid sensed data, which is suitable for the tiered hybrid sensing network where mobile nodes exist. The basic idea is to bind each data record, generated by static or mobile sensing nodes, with the corresponding location where it is sensed. Then some verification information is created sequentially, which is submitted along with the encrypted locations and hybrid sensed data for user's verification. The security and efficiency of EVTopk are thoroughly analyzed in theory and evaluated in our experiments, respectively.

1. Introduction

Nowadays, the proliferation of mobile devices (e.g., smartphones), embedded with powerful sensors, has witnessed a revolution in the data sensing paradigm of tiered sensor networks. Traditionally, the two-tiered sensor network [1–3] consists of many energy constrained static (or fixed) sensor nodes and some resource abundant storage nodes at the lower and upper level, respectively. By contrast, the novel sensing paradigm relies on both static sensor nodes and mobile devices (regarded as mobile sensing or sensor nodes) to jointly monitor data about surroundings. It brings great benefits especially to some application-specific scenarios, such as traffic monitoring in smart city. The mobile phones or the vehicles, as the mobile sensing nodes, can monitor traffic conditions in the area where the fixed traffic sensors are not covered. Apparently, the introduction of mobile sensing nodes provides more comprehensive information and reduces the cost for deploying static sensors. Moreover, the energy consumption of static sensors is well balanced.

The storage nodes, acting as the data center, are responsible for storing the hybrid sensed data and providing assorted query services for network user (query user). In particular, one important type of queries is represented by top-k query, which asks for k highest (or lowest) data records in a specific region during a specified time slot. Continue with the above example of traffic monitoring, the traffic department can retrieve the 10 most crowded roads (according to the traffic flow) in a city from 8 AM to 9 AM by issuing top-10 query.

However, the security issue still remains as a critical concern for pragmatic top-k query, since the storage nodes manage large volume of data and are vulnerable to various attacks. For instance, the compromised storage node may falsify or drop some data during the query processing, which returns fake or incorrect result to the query users. To solve this problem, some verification schemes [4–8] for secure top-k query in tiered sensor networks were proposed in previous researches, which aims to enable the query user to verify the authenticity and correctness of query results. Specifically, Dai et al. [4, 5] generated verification information by chaining the ordered and adjacent data records. Similarly, Zhang et al. [7] bound adjacent data records and some auxiliary ID lists with message authentication code (MAC). Unfortunately, these solutions are only suitable for the static tiered sensor network,

as it is assumed that the query users and storage node know the mapping between the nodes' locations and their corresponding IDs. Clearly, this assumption does not hold in our application scenario, where mobile sensing nodes exist at the lower sensing level. More specifically, different from the static sensor nodes, the mobile sensing nodes generate data at different locations and the compromised storage node may replace the data records sensed in the query region with others sensed outside the region. This attack is difficult to be detected with existing verification methods for static sensor nodes.

In this paper, we propose an efficient and verifiable scheme for secure top-k query processing (EVTopk) on hybrid sensed data, which enables the query user to efficiently verify the authenticity and correctness of the query result in our novel tiered sensing network. Our specific contributions are summarized as follows:

(i) We formulate a novel tiered sensing network model (tiered hybrid sensing network) on the basis of the traditional tiered sensor network, where mobile sensing nodes jointly collect data records with static sensor nodes at the lower level.

(ii) To achieve EVTopk, we give a concrete sequence relationship based method by binding each data record with its corresponding sensing location. Moreover, the data records generated by mobile nodes are returned in different formations according to their sensing locations, which enables the query user to detect any inauthentic or incorrect result.

(iii) We present the theoretical analysis and conduct extensive experiments to show the security and efficiency of our proposed scheme.

The rest of the paper is organized as follows. In Section 2, we review related work, followed by a preliminary of system model and problem formulation in Section 3. Section 4 presents our efficient and verifiable top-k query scheme EVTopk and the theoretical analysis in terms of security and communication overhead. In Section 5, extensive experiments are conducted and the simulation results are analyzed in detail. Finally, we conclude this paper with directions for future work in Section 6.

2. Related Work

Recently, due to the advantages of better scalability and capacity, the two-tiered wireless sensor network (TWSN) has drawn increasing attention among researchers. In TWSN, the storage nodes are more vulnerable to be compromised by attackers for their important role in data storage and query management, which brings two challenging security issues: data privacy and data integrity (query integrity).

Many privacy preserving methods [9–12] were proposed to prevent the compromised storage nodes from knowing the sensitive information of data collected by sensor nodes and/or the queries issued by users. Peng et al. [10] proposed a Bloom filter based scheme for secure top-k query; moreover, the author adopted an obfuscation coding method to hide the real data distribution and query preference. In [12], a secure top-k query method, called PCTopk, was proposed to preserve the data privacy and query correctness simultaneously. However, this line of work is orthogonal to ours, as in this paper, we only focus on the security property of the query results instead of the privacy concerns, because the hybrid sensed data are not sensitive and are accessible to the public in our scenario.

For another line of research, considerable attentions have been paid to ensuring the query integrity (both the authenticity and correctness of the query result), such as verifiable range query [13, 14], secure kNN query [15, 16], and top-k query in different network environments. Three schemes were first proposed by Zhang et al. [2] to verify the fine-grained top-k query result in two-tiered sensor network, which were further improved in [7] to deal with various attacks launched by compromised storage nodes and/or sensor nodes. However, in addition to returning one verification message for each unqualified sensor node, this scheme requires the sensor nodes to exchange their highest scores with other nodes, which results in a large communication overhead. The same problem also exists in [17], for each sensor node needs to send the node IDs and hashes of their neighbors. Additionally, more than k data records are returned to the query user due to the existence of dummy readings. In addition, Dai et al. [4, 5] proposed an efficient verifiable top-k query (EVTQ) by chaining ordered and adjacent data records. Although the scheme is feasible to verify the query result, the storage node needs to send multiple verification messages for those unqualified sensor nodes. As an improvement, He et al. [8] proposed an efficient top-k query processing with integrity verification (ETQ-RIV), in which sensor nodes are not required to submit information about neighbors and only one verification message is returned for all unqualified sensor nodes. Therefore, the query communication cost is significantly reduced. However, all the schemes above only consider the static sensor nodes and are not suitable for our network model where mobile sensing nodes exist. Recently, Liu et al. [18] first proposed a novel verifiable scheme named VTMSN for top-k query in tiered mobile sensor networks, which is the most relevant work to ours. Besides the location information of each data record, both the static and mobile nodes also require to send the encrypted chaining locations to the storage node, some of which are then sent to the query users for verification. Unfortunately, large extra communication cost is incurred since the static nodes send many redundant locations. Moreover, it is assumed that all the sensor nodes switch their state between static and mobile state periodically, which is a little different from our scenario.

3. Preliminary

In this section, we introduce our system, attack model, and design goals and briefly state the problems investigated in this paper.

3.1. System Model. In this paper, our system model is based on the two-tiered sensor network, which consists of sensor nodes and storage nodes at the lower level and upper level,

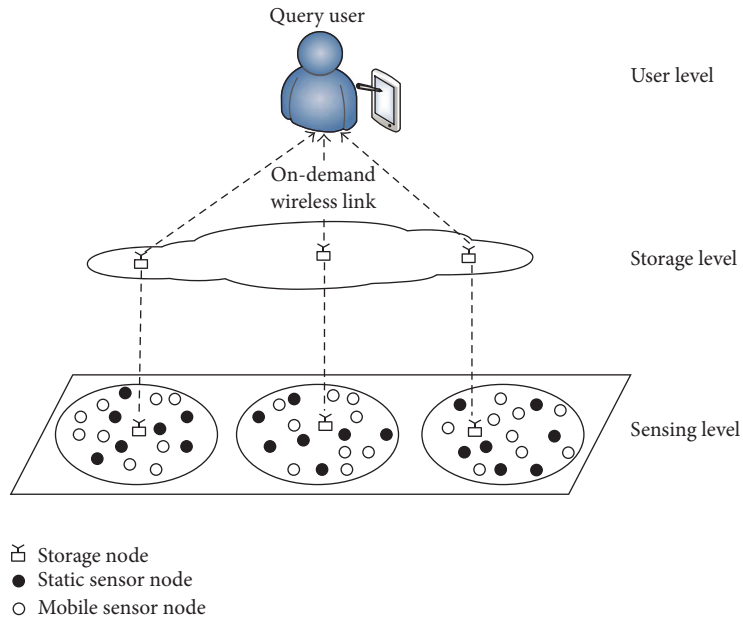

Figure 1: System architecture.

respectively. Different from the traditional two-tiered sensor networks, we introduce mobile sensing (sensor) nodes in the lower level. As shown in Figure 1, in each sensing cell at the sensing level, some mobile sensor nodes (*e.g.*, mobile vehicles in the smart city) also exist for cooperative sensing with the resource-constrained static sensor nodes. Obviously, this novel sensing paradigm can relieve the energy consumption of the static nodes. In addition to the hybrid sensor nodes, each sensing cell contains a storage node, which is responsible for collecting the hybrid sensed data and answering various query requests from the external query users. Specifically, the query users at the user level can issue a top-k query and obtain the query results via an on-demand wireless link.

For ease of presentation, we assume that the whole network is partitioned into many sensing cells according to the real geographic location. As introduced above, each cell consists of many fixed (static) sensor nodes, mobile sensor nodes, and a storage node. In particular, similar to [18], we assume that the mobile nodes only move in its affiliated sensing cell. During the sensing period, the mobile nodes move and stop regularly to save their energy, and we suppose that the sensed data are only generated when the mobile nodes stop moving. In other words, a data record generated by the mobile node corresponds to the specific location where it is sensed. In particular, some localization techniques [19–21] are available to estimate the location of static or mobile nodes. At the end of each sensing time slot, the static and mobile sensor nodes send their hybrid sensed data to their corresponding storage node.

3.2. Attack Model. Compared with the sensor node, the storage node with large volume of sensed data is more vulnerable to attacks. Hence, in this paper, we mainly solve the security problem of compromised storage node that may return fake or incorrect query results to the query users. In particular, the following attacks are mainly considered in our model.

(i) *Replacement attack*: the compromised storage node may replace some qualified top-k query results with other unqualified data records or those even not generated by the sensor nodes. More specifically, since the mobile nodes generate data records at different locations, the compromised storage node may replace the data records sensed in the query region with those sensed by the same mobile node at other locations.

(ii) *Deletion attack*: the compromised storage node may drop some qualified data records generated in the query region and return the incomplete results to query users.

3.3. Design Goals. In terms of the aforementioned attacks, our design goal is to enable the query users to verify the authenticity and correctness of the query results returned by the storage node. In addition, the query and verification efficiency should also be guaranteed. In general, our method is aimed to achieve the following security goals and performance guarantee.

(i) *Authenticity*: all the k data records in the result set, returned by the storage node, are generated by the static sensor nodes or mobile sensor nodes which have ever moved in the query region.

(ii) *Correctness*: all the k data records in the result set are in the query region, and they have the highest k scores among the data records sensed in the query region.

(iii) *Efficiency*: since the communication cost plays a significant role in the energy consumption of the whole network, the two security goals above should

be achieved with as little communication overhead as possible.

3.4. Problem Statement. Without loss of generality, we only focus on top-k query processing that covers one sensing cell for simplicity, which consists of N sensor nodes $\{S_i\}_{i=1}^{N}$ (including static and mobile sensor nodes) and a storage node. Moreover, we assume that the query user can distinguish the static nodes from mobile nodes by their IDs. Suppose that each sensor node $S_i \in \{S_i\}_{i=1}^{N}$ senses n_i data records, denoted by $D_i = \{D_{i,j}\}_{j=1}^{n_i}$, during each time slot t. In particular, the location of generating each record $D_{i,j}$ is denoted by $l_{i,j}$. Obviously, if S_i is a static sensor node, $l_{i,j}$ is the same for each $j \in \{1, n_i\}$. At the end of slot t, the storage node can receive $\sum_{i=1}^{N} n_i$ data records ($D = \bigcup_{i=1}^{N} D_i$). Note that the sensed data records may have multiple attributes (*e.g.*, temperature, humidity), since the sensor nodes are usually embedded with multiple sensors.

As presented in [7], we assume that each record $D_{i,j} \in D_i$ can be scored by some public scoring functions [22] according to the specific query application. Let $f(\cdot)$ denote a specified increasing scoring function, and then the score of data record $D_{i,j}$ can be denoted by $d_{i,j} = f(D_{i,j})$. For simplicity, we only consider the scalar sensed data and the situation where $D_{i,j}$ and $D_{i,j'}$ show partial order is not considered in this paper. Therefore, if $D_{i,j} \leq D_{i,j'}$, which means each attribute value in $D_{i,j}$ is smaller than or equal to that in $D_{i,j'}$, then we have $d_{i,j} \leq d_{i,j'}$. In other words, if $D_{i,j}$ and $D_{i,j'}$ are both generated in the query region and $D_{i,j}$ is in the top-k query result, so does $D_{i,j'}$.

Furthermore, we formulate the top-k query as $Q_t = \langle C_q, R_q, t, k\rangle$, where C_q and R_q denote the ID of the query sensing cell and geographic query region, respectively; t and k are the number of query time slot and desired data records. Given the query request Q_t, the issue investigated in this paper is how to enable the query user to efficiently verify the authenticity and correctness of the query result, as we described in Section 3.3.

To make our description clear, the major notations used in this paper is summarized in Notation Settings.

4. Secure Top-k Query Processing

In this section, we propose a novel verifiable top-k query scheme EVTopk to efficiently verify the authenticity and correctness of top-k query result over the hybrid sensed data. We mainly describe our detailed scheme in Section 4.1, and the security analysis against the compromised storage node is presented in Section 4.2.

4.1. Manipulations at Three Levels. In this section, we describe the data preprocessing, top-k query processing, and query verification at the sensing, storage, and user level, respectively. The details are as follows.

4.1.1. Data Preprocessing at Sensing Level. In addition to submitting the sensed data, the sensor nodes require to send some additional information to the storage node, and some of them are then returned to the query users for query verification. Specifically, we generate the verification information by chaining ordered hybrid data records of each sensor node via a cryptographic hash function. Note that, to perform effective query verification for mobile sensor nodes, we bind each data record with its corresponding sensing location.

Consider the sensor node S_i as an example; first, S_i sorts its sensed data records D_i during each time slot in the descending order according to records' scores. We assume that the scores of data records in D_i differ from each other, which indicates that a unique correct query result exists for top-k query. Suppose that an ordered list $\langle D_{i,1}, D_{i,2}, \ldots, D_{i,n_i}\rangle$ is generated such that $d_{i,1} > d_{i,2} > \cdots > d_{i,n_i}$. In addition, similar to [7], we assume that S_i shares a distinct key $K_{i,t}$ with the query user during time slot t, which is used to encrypt the location information and compute the hash message authentication code (HMAC) as verification information. Initially, S_i is preloaded with $K_{i,0}$. At the end of time $t \geq 1$, the time slot key $K_{i,t}$ is updated with $H(K_{i,t-1})$ [7], where $H(\cdot)$ denotes a hash function. Note that, to provide stronger security, it is necessary to update the encryption key in each time slot. According to the sequence of the ordered data records, S_i then chains adjacent data scores by recursively computing

$$v_{i,j} = \begin{cases} \mathrm{HMAC}_{K_{i,t}}\left(v_{i,j+1} \parallel l_{i,j} \parallel d_{i,j}\right) & 1 \leq j \leq n_i, \\ \mathrm{HMAC}_{K_{i,t}}\left(t \parallel i\right) & j = n_i + 1, \end{cases} \tag{1}$$

where $\mathrm{HMAC}_{K_{i,t}}(\cdot)$ denotes the hash message authentication code keyed with $K_{i,t}$ and $\parallel$ refers to the concatenation. Note that, if S_i is a static node, we have $l_{i,j} = l_{i,1}$ for $j \in [2, n_i]$. We hereafter use $l_{i,1}$ to denote the location of static node S_i for simplicity. Moreover, if $n_i = 0$, we set $v_{i,1} = \mathrm{HMAC}_{K_{i,t}}(t \parallel i)$.

Let $E_{K_{i,t}}(\cdot)$ denote the symmetric encryption function (in this paper, we use AES to encrypt the location information for its better performance especially in dealing with large volume of data) with key $K_{i,t}$; subsequently, each sensor node $S_i \in \{S_i\}_{i=1}^{N}$ submits the following information Ψ_i to the storage node. Specifically, if S_i is a static node, the information is as follows:

$$\Psi_i = \begin{cases} \langle i, t, v_{i,1}\rangle & n_i = 0, \\ \left\langle i, t, E_{K_{i,t}}\left(l_{i,1}\right), D_{i,1}, \ldots, D_{i,n_i}, v_{i,1}, \ldots, v_{i,n_i}\right\rangle & n_i > 0. \end{cases} \tag{2}$$

While if S_i is a mobile node, the information is

$$\Psi_i = \begin{cases} \langle i, t, v_{i,1}\rangle & n_i = 0, \\ \left\langle i, t, E_{K_{i,t}}\left(l_{i,1} \parallel \cdots \parallel l_{i,n_i}\right), D_{i,1}, \ldots, D_{i,n_i}, v_{i,1}, \ldots, v_{i,n_i}\right\rangle & n_i > 0. \end{cases} \tag{3}$$

Note that, different from [2, 7], we assume that the location information of each sensor node is unknown to the storage node and query users in advance. Therefore, it is necessary to send some location information to the storage node in this phase. After receiving the message Ψ_i, the storage node will store the corresponding information of S_i, including its data records $D_i = \{D_{i,j}\}_{j=1}^{n_i}$, the encrypted sensing locations $E_{K_{i,t}}(l_{i,1} \parallel \cdots \parallel l_{i,n_i})$ and the verification information $v_{i,1}, \ldots, v_{i,n_i}$.

4.1.2. Top-k Query Processing at Storage Level. The storage node, after receiving a top-k query $Q_t = \langle C_q, R_q, t, k\rangle$ from the query user, will first locate the query cell C_q and its corresponding query region R_q. It is worth noting that R_q may completely cover multiple static sensor nodes or partially cover multiple mobile sensor nodes during time slot t, as mobile nodes may move out of R_q during this time slot. Let I_t denote the set of static or mobile sensor nodes in R_q, and we call the set of its corresponding data records a candidate set, which is denoted by $\mathscr{C}_t$. Note that some data records in $\mathscr{C}_t$ may be not in the query region.

Subsequently, the storage node retrieves the highest k data records RS_t in $\mathscr{C}_t$ with their sensing locations in R_q, among which the lowest record score is denoted by τ. In particular, for each $S_i \in I_t$, we assume that there are γ_i data records with their scores not lower than τ. Accordingly, we have $0 \le \gamma_i \le n_i$ and $k \le \sum_{S_i \in I_t} \gamma_i \le |\mathscr{C}_t|$, as some mobile sensor nodes may generate data records outside R_q but with their scores higher than τ. Moreover, all these data records are in the candidate set $\mathscr{C}_t$. For each $D_{i,j} \in \mathscr{C}_t$, we make the following definition.

$$T_i = \begin{cases} D_{i,j} & l_{i,j} \in R_q, \\ d_{i,j} & \text{o.w,} \end{cases} \tag{4}$$

which means if the data record $D_{i,j}$ is generated in R_q, $T_{i,j}$ equals $D_{i,j}$. Otherwise, it equals the score of $D_{i,j}$. Then the query user can simply identify if the data records are in R_q according to their data format (here referring to the data dimensions). For each candidate $S_i \in I_t$, the storage node returns the following information $\mathfrak{R}_i$ to the query user.

If S_i is a static sensor node, the information is

$$\mathfrak{R}_i = \begin{cases} \langle i, v_{i,1}\rangle & n_i = \gamma_i = 0, \\ \langle i, E_{K_{i,t}}(l_{i,1}), d_{i,1}, v_{i,1}\rangle & n_i = 1,\ \gamma_i = 0, \\ \langle i, E_{K_{i,t}}(l_{i,1}), d_{i,1}, v_{i,2}, v_{i,1}\rangle & n_i \ge 2,\ \gamma_i = 0, \\ \langle i, E_{K_{i,t}}(l_{i,1}), D_{i,1}, \ldots, D_{i,\gamma_i}, d_{i,\gamma_i+1}, v_{i,\gamma_i+2}, v_{i,1}\rangle & n_i \ge 2,\ 0 < \gamma_i < n_i, \\ \langle i, E_{K_{i,t}}(l_{i,1}), D_{i,1}, \ldots, D_{i,\gamma_i}, v_{i,1}\rangle & n_i = \gamma_i \ge 1. \end{cases} \tag{5}$$

If S_i is a mobile sensor node, the information is

$$\mathfrak{R}_i = \begin{cases} \langle i, v_{i,1}\rangle & n_i = \gamma_i = 0, \\ \langle i, E_{K_{i,t}}(l_{i,1}), d_{i,1}, v_{i,1}\rangle & n_i = 1,\ \gamma_i = 0, \\ \langle i, E_{K_{i,t}}(l_{i,1} \parallel \cdots \parallel l_{i,n_i}), d_{i,1}, v_{i,2}, v_{i,1}\rangle & n_i \ge 2,\ \gamma_i = 0, \\ \langle i, E_{K_{i,t}}(l_{i,1} \parallel \cdots \parallel l_{i,n_i}), T_{i,1}, \ldots, T_{i,\gamma_i}, d_{i,\gamma_i+1}, v_{i,\gamma_i+2}, v_{i,1}\rangle & n_i \ge 2,\ 0 < \gamma_i < n_i, \\ \langle i, E_{K_{i,t}}(l_{i,1} \parallel \cdots \parallel l_{i,n_i}), T_{i,1}, \ldots, T_{i,\gamma_i}, v_{i,1}\rangle & n_i = \gamma_i \ge 1. \end{cases} \tag{6}$$

While for each $S_i \notin I_t$, the information is

$$\mathfrak{R}_i = \begin{cases} \langle i, E_{K_{i,t}}(l_{i,1})\rangle & S_i \text{ is static,} \\ \langle i, E_{K_{i,t}}(l_{i,1} \parallel \cdots \parallel l_{i,n_i})\rangle & \text{o.w.} \end{cases} \tag{7}$$

As we can see, for the sensor nodes in I_t, besides the qualified top-k data records in RS_t, the storage node requires to return the data scores that are higher than τ but are generated outside R_q, so that the query user can recompute $v_{i,1}$ with (1). While for the sensor nodes outside the query region, they only need to return the node IDs and their encrypted location information (7) to the query user for verification.

4.1.3. Result Verification at User Level. Now we discuss how the user verifies the authenticity and correctness of the query result, as we mentioned in Section 3.3. For each information $\mathfrak{R}_i$ received from the storage node, the query user will defer to the following verification steps.

(1) First, the user determines which of the above cases $\mathfrak{R}_i$ belongs to according to its message format. More specifically, the user can distinguish (5) from (6) according to the first part (*i.e.*, node's ID), since it is assumed that users can identify the static nodes by IDs in Section 3.4. In addition, as we can see, the information in (7) only contains two parts (ID

and at least 20-bit encrypted location), while the information in (5) or (6) contains at least two parts (ID and 160-bit verification information, as well as other data). Accordingly, the user can determine if $\mathfrak{R}_i$ belongs to (7) easily. Ultimately, if $\mathfrak{R}_i$ satisfies none of cases, the query result will be declined, or the verification is continued.

(2) Authenticity verification. If $\mathfrak{R}_i$ satisfies any case in (5) or (6), which means $S_i \in I_t$. The query user first computes the scores of all the returned data records according to the public function $f(\cdot)$ and then decrypts the encrypted locations; finally, he can derive $v_{i,1}$ using (1). For example, if the user receives $\mathfrak{R}_i = \langle i, E_{K_{i,t}}(l_{i,1}), D_{i,1}, \ldots, D_{i,\gamma_i}, v_{i,1}\rangle$ in (5), which indicates $n_i = \gamma_i$, he can first compute $d_{i,1}, \ldots, d_{i,\gamma_i}$ with $f(\cdot)$ and then get the location $l_{i,1}$ after decryption. To derive $v_{i,1}$, $v_{i,\gamma_i+1} = \mathrm{HMAC}_{K_{i,t}}(t \parallel i)$ is first computed by (1) (the second case). Then similarly, the user can compute $v_{i,\gamma_i} = \mathrm{HMAC}_{K_{i,t}}(v_{i,\gamma_i+1} \parallel l_{i,\gamma_i} \parallel d_{i,\gamma_i})$ (the first case in (1)). Note that here $l_{i,\gamma_i} = l_{i,\gamma_i-1} = \cdots = l_{i,1}$ for static node S_i. Sequentially, $v_{i,\gamma_i-1}, \ldots, v_{i,2}$ can be computed iteratively. At last, the user can compute $v_{i,1} = \mathrm{HMAC}_{K_{i,t}}(v_{i,2} \parallel l_{i,1} \parallel d_{i,1})$. If $v_{i,1}$ is equal to the last field of $\mathfrak{R}_i$ (i.e., returned $v_{i,1}$), the user considers the data records and scores in $\mathfrak{R}_i$ authentic. Moreover, the query result RS_t is regarded as authentic if $\mathfrak{R}_i$ passes the authenticity verification for each $S_i \in I_t$.

(3) Correctness verification. For each $\mathfrak{R}_i$ satisfying the format in (7), which means S_i is not covered by R_q, the query user first decrypts the encrypted location information with $K_{i,t}$. Subsequently, he checks if all the locations $(l_{i,1}, \ldots, l_{i,n_i})$ are outside R_q. Similarly, for each $\mathfrak{R}_i$ satisfying the format in (5) or (6), the user decrypts the location information to check if all the locations of returned data records are within R_q. If so, he continues the following verification.

(i) The user further identifies there are total k POIs records in RS_t with their locations in R_q and determines the lowest data scores τ.

(ii) For each data score $d_{i,j}$ ($j \in [1, \gamma_i]$) returned in $\mathfrak{R}_i$, the user checks if $d_{i,j} > \tau$ and $l_{i,j}$ is outside R_q. Moreover, he checks whether $d_{i,r_i+1} < \tau$ is satisfied.

If the above three steps above are all passed, the user considers the query result RS_t authentic and correct. Consider an example where the query user asks for top-4 data records in cell 1. Suppose that the query region in cell 1 covers one static sensor node S_1 and two mobile nodes S_2 and S_3. For simplicity, we assume that $n_1 = n_2 = n_3 = 4$, and $\tau = 30$ in top-4 records. Moreover, γ_1, γ_2, and γ_3 are assumed to be 2, 3, and 0, respectively. Specially, suppose that $D_{2,2}$, sensed by S_2, is outside the query region but its score is higher than τ. Since $\gamma_2 < n_2$, the fourth case in (6) should be returned. We have $T_{2,1} = D_{2,1}$, $T_{2,2} = d_{2,2}$, and $T_{2,3} = D_{2,3}$; hence $\mathfrak{R}_2 = \langle 2, E_{K_{2,t}}(l_{2,1} \parallel \cdots \parallel l_{2,4}), D_{2,1}, d_{2,2}, D_{2,3}, d_{2,4}, v_{2,5}, v_{2,1}\rangle$. Similarly, S_1 follows the fourth case in (5) and the storage node returns $\mathfrak{R}_1 = \langle 1, E_{K_{1,t}}(l_{1,1}), D_{1,1}, D_{1,2}, d_{1,3}, v_{1,4}, v_{1,1}\rangle$. S_3 satisfies the third case in (6) and $\mathfrak{R}_3 = \langle 3, E_{K_{3,t}}(l_{3,1} \parallel \cdots \parallel l_{3,4}), d_{3,1}, v_{3,2}, v_{3,1}\rangle$ is returned to the user. Obviously, we have $\mathrm{RS}_t = \{D_{1,1}, D_{1,2}, D_{2,1}, D_{2,3}\}$. Based on the returned information, the user can deduce $\{v_{i,1}\}_{i=1}^{3}$ with (1). If they are consistent with the last field of $\mathfrak{R}_1$, $\mathfrak{R}_2$, and $\mathfrak{R}_3$ respectively, the query result is considered authentic. Subsequently, the user decrypts the location information and checks if all the data records in RS_t are within R_q. In addition, he checks whether $d_{1,3} < 30$ and $d_{2,2} > 30$ but $l_{2,2}$ is outside R_q, $d_{2,4} < 30$, and $d_{3,1} < 30$. If so, the query result is considered correct.

4.2. Performance Analysis

4.2.1. Detection of the Replacement Attack

Theorem 1. *EVTopk can detect any replacement attacks launched only by a compromised storage node.*

Proof. Recall that, in Section 3.2, the compromised storage node may perform replacement attack in two ways. First, it may replace some qualified data records sensed by some node (*e.g.*, S_i) with other unqualified data records sensed by S_i or some other sensor nodes (*e.g.*, S_j). Second, the compromised storage node may replace some qualified data records with others not generated by the sensor nodes at all, which means some data records with high scores are forged.

Assume that S_i contributes μ_i data records to RS_t during time slot t. If S_i is a static sensor node, we have $\mu_i = \gamma_i$; otherwise, $\mu_i \leq \gamma_i$ is satisfied for mobile nodes, as some data records with scores higher than τ but not in R_q are not qualified for the query result. For the first attack method, if a qualified data record $D_{i,m}$ of static node S_i is replaced by an unqualified record $D_{i,m+1}$, it is easy to be detected in step (2) since the ordered data scores are chained together with HMAC, while for the mobile node S_i, if a qualified data record $D_{i,m}$ is replaced by $d_{i,m}$, which implies $D_{i,m}$ is not in R_q. Although it can pass the authentication verification in step (2), the correctness check in step (3) can detect this attack by decrypting the encrypted location. Observe that $l_{i,m}$ is in R_q, leading to a contradiction with the above implication. On the other hand, if a qualified data record $D_{i,m}$ is replaced by an unqualified record $D_{j,m+1}$ sensed by S_j, it is impossible to deduce the same $v_{i,1}$ as computed by S_i. Hence, this attack can be detected in step (2).

As for the second attack method, if some data records of either static or mobile nodes are forged, it will fail the authentication verification since the compromised storage node cannot generate corresponding verification information without knowing $K_{i,t}$. □

4.2.2. Detection of the Deletion Attack

Theorem 2. *EVTopk can detect any deletion attacks launched only by a compromised storage node.*

Proof. If the compromised storage node drops part of the qualified data records sensed by a static node (*e.g.*, S_i), it will fail the authentication verification as the adjacent data

scores are chained with HMAC. Similarly, if all the qualified data records of S_i are dropped, it is easy to be detected during the authentication verification because the storage node still requires to return the maximum data score d_{i,γ_i+1} and verification information v_{i,γ_i+2}, which obviously cannot derive the same $v_{i,1}$ as computed by S_i.

While for a mobile node S_j, if part of its qualified data records is dropped, it may pass the authentication verification but will fail the correctness check in step (3). Assume that $\mathfrak{R}_j = \langle D_{j,1}, d_{j,2}, D_{j,3}, d_{j,4}, v_{j,5}, v_{j,1}\rangle$ is supposed to be returned to the user, which indicates that $D_{j,1}$ and $D_{j,3}$ are two qualified data records, and $d_{j,2}$ is higher than τ but outside R_q. If the compromised storage node drops $D_{j,3}$ and returns $D_{j,1}$, $d_{j,2}$, $v_{j,3}$, and $v_{j,1}$ to the user, the user can derive the correct $v_{j,1}$ based on the first three item $D_{j,1}$, $d_{j,2}$, and $v_{j,3}$. However, the format indicates that $D_{j,1}$ is a qualified record and $d_{j,2}$ is lower than τ, which contradicts with the fact that $d_{j,2}$ is higher than τ. Similar to the static node, if all the qualified data records of S_j are dropped, it is easy to be detected during the authentication verification. □

4.2.3. Communication Cost among Three Levels. Assuming that the length of encrypted data is the same as that of its corresponding plaintext. Let L_{id}, L_{loc}, L_{score}, and L_h denote the bit-lengths of a sensor node's ID, location, score, and each HMAC, respectively. Moreover, let L be the average number of hops between a sensor and a storage node. Additionally, suppose that there are $N_1(S_1, S_2, \ldots, S_{N_1})$ static sensor nodes and $N_2 = N - N_1(S_{N_1+1}, S_{N_1+2}, \ldots, S_N)$ mobile nodes in a query cell. For ease of presentation, we assume that each sensor node S_i, $i \in [1, N]$ generate $n_i = n$ data records during slot t. Then we have the following theorems.

Theorem 3. *Without considering the transmission of some fundamental information, the extra communication cost between the sensing and the storage level in EVTopk is given by*

$$C_{ss} = \left((N_2 n + N_1) \cdot L_{loc} + Nn \cdot L_h\right) \cdot L. \tag{8}$$

Proof. As shown in Section 4.1.1, since the node IDs, time slot, and sensed data records are the fundamental information to be sent to the storage node, we omit the cost for transmitting them for clarity.

For each static node S_i, $i \in [1, N_1]$, it requires to send encrypted location $E_{K_{i,t}}(l_i, 1)$ and n verification information $v_{i,1}, \ldots, v_{i,n}$ to the storage node. Note that the length of encrypted data is the same as that of its corresponding plaintext. Thus, the additional communication cost incurred by the static node is

$$C_{\mathrm{static}} = \left(L_{\mathrm{loc}} + nL_h\right) \cdot N_1 \cdot L. \tag{9}$$

While for each mobile node S_i, $i \in [N_1 + 1, N]$, it requires to send encrypted location $E_{K_{i,t}}(l_{i,1} \parallel \cdots \parallel l_{i,n})$ and n verification information $v_{i,1}, \ldots, v_{i,n}$ to the storage node. Thus, the additional communication cost incurred by the mobile node is

$$C_{\mathrm{mobile}} = n\left(L_{\mathrm{loc}} + L_h\right) \cdot N_2 \cdot L. \tag{10}$$

Hence, by integrating (9) and (10), we have

$$\begin{aligned} C_{\mathrm{ss}} &= C_{\mathrm{static}} + C_{\mathrm{mobile}} \\ &= \left(L_{\mathrm{loc}} + nL_h\right) \cdot N_1 \cdot L + n\left(L_{\mathrm{loc}} + L_h\right) \cdot N_2 \cdot L \\ &= \left((N_2 n + N_1) \cdot L_{\mathrm{loc}} + Nn \cdot L_h\right) \cdot L. \end{aligned} \tag{11}$$

□

Let δ_q denote the number of qualified sensor nodes that contribute their records to RS_t; specifically, δ_q contains δ_s and δ_m static and mobile nodes, respectively. Then the number of the unqualified sensor nodes is $N - \delta_q$, which contains $N_1 - \delta_s$ static nodes and $N_2 - \delta_m$ mobile nodes, respectively.

Theorem 4. *Without considering the transmission of qualified data records and its corresponding node IDs, the maximum extra communication cost between storage node and query user in EVTopk is given by*

$$\begin{aligned} C_{SU} \le{}& NL_{id} + \left(N_2 n + N_1\right) L_{loc} \\ &+ \left(N_2 n + N - \delta_m\right) L_{score} + 2NL_h. \end{aligned} \tag{12}$$

Proof. As shown in (5), for δ_s qualified static nodes, it will additionally incur $C_{\mathrm{qs}} = (L_{\mathrm{id}} + L_{\mathrm{loc}} + L_{\mathrm{score}} + 2L_h) \cdot \delta_s$ communication overhead at most (the forth case), while for δ_m qualified mobile nodes, at most $C_{\mathrm{qm}} = (L_{\mathrm{id}} + nL_{\mathrm{loc}} + (n-1) \cdot L_{\mathrm{score}} + 2L_h) \cdot \delta_m$ communication cost will be additionally incurred (in the fourth case, $n - 2$ data records with scores higher than τ are outside R_q; i.e., only one qualified data record is returned for each qualified mobile node). Similarly, for $N_1 - \delta_s$ unqualified static nodes, it will additionally incur $C_{\mathrm{us}} = (L_{\mathrm{id}} + L_{\mathrm{loc}} + L_{\mathrm{score}} + 2L_h) \cdot (N_1 - \delta_s)$ communication cost at most (the third case), while for $N_2 - \delta_m$ unqualified mobile nodes, at most $C_{\mathrm{us}} = (L_{\mathrm{id}} + nL_{\mathrm{loc}} + nL_{\mathrm{score}} + 2L_h) \cdot (N_2 - \delta_m)$ communication overhead is additionally incurred (in the fourth case, all the data records with scores higher than τ are outside R_q). As a result, the communication overhead between storage node and the query user follows that

$$\begin{aligned} C_{\mathrm{SU}} &\le C_{\mathrm{qs}} + C_{\mathrm{qm}} + C_{\mathrm{us}} + C_{\mathrm{um}} \\ &= \left(L_{\mathrm{id}} + L_{\mathrm{loc}} + L_{\mathrm{score}} + 2L_h\right) \cdot \delta_s \\ &\quad + \left(L_{\mathrm{id}} + nL_{\mathrm{loc}} + (n-1) \cdot L_{\mathrm{score}} + 2L_h\right) \cdot \delta_m \\ &\quad + \left(L_{\mathrm{id}} + L_{\mathrm{loc}} + L_{\mathrm{score}} + 2L_h\right) \cdot \left(N_1 - \delta_s\right) \\ &\quad + \left(L_{\mathrm{id}} + nL_{\mathrm{loc}} + nL_{\mathrm{score}} + 2L_h\right) \cdot \left(N_2 - \delta_m\right) \\ &= NL_{\mathrm{id}} + \left(N_2 n + N_1\right) L_{\mathrm{loc}} \\ &\quad + \left(N_2 n + N - \delta_m\right) L_{\mathrm{score}} + 2NL_h. \end{aligned} \tag{13}$$

□

5. Simulation Results

In this section, we mainly evaluate the efficiency of our proposed scheme EVTopk and validate the theoretical results

Table 1: Default simulation settings.

Para.	Val.
N	500
n	10
k	50
r	100
L_{id}	16
L_{loc}	20
L_h	160
L_{score}	20

we discussed in Section 4 using experimental simulations on a synthetic dataset. Our simulations are implemented on Matlab. More specifically, we assume a sensing cell of 1000 m × 1000 m with 500 randomly distributed sensor nodes and a storage node at the center, in which static and mobile nodes account for 50%, respectively, and the mobile nodes move in a random waypoint model. Without loss of generality, we suppose that each static or mobile sensor node generates 10 data records during the time slot, and the packets are transmitted without collision and error. Table 1 presents the default parameter settings in our experiments, unless stated otherwise.

Specifically, we use 160-bit HMAC-SHA1 to compute the verification information for each sensor node, and the default query region in the cell is 400 m × 400 m. The main performance metrics used to evaluate our proposed scheme include the aforementioned two aspects: the communication cost between the sensing and storage level, as well as that between the storage and user level. It is worth noting that these metrics are not compared with those in other schemes as none of them are fit for our network model where both static and mobile nodes exist. To avoid the accidental error, in all sets of experiments, our scheme is measured on an average of 100 random simulations.

5.1. Communication Cost between Sensing and Storage Level. Figure 2 shows the simulation and theoretical results of the communication cost between the sensing and storage level with varying N, which is the total number of sensor nodes in the query cell. It is clear that our simulation result matches the analytical result closely. More specifically, the communication cost increases linearly as N goes from 200 to 800. The reason is that more static or mobile sensor nodes require to send the location and verification information to the storage node with increasing N. Specially, due to the mobility of mobile nodes, they need to send encrypted concatenate locations, as shown in (3), which results in more communication cost than static nodes. Moreover, the actual hops between a mobile sensor and a storage node may be more than the average hops which we considered in Theorem 3. Therefore, the simulation result incurs slightly more communication cost than that in theory.

Figure 3 shows the impact of n, the number of data records generated by each node per slot, on the sensing-storage communication cost in both our simulation and theoretical analysis. Similarly, we can observe that the simulation and theoretical results closely match as n increases, and the sensing-storage communication cost of them both grows linearly with n. The reason is that more location and verification information need be sent to the storage node for each static or mobile node. Moreover, our simulation exhibits slightly more expensive cost due to the same reason as we analyzed above.

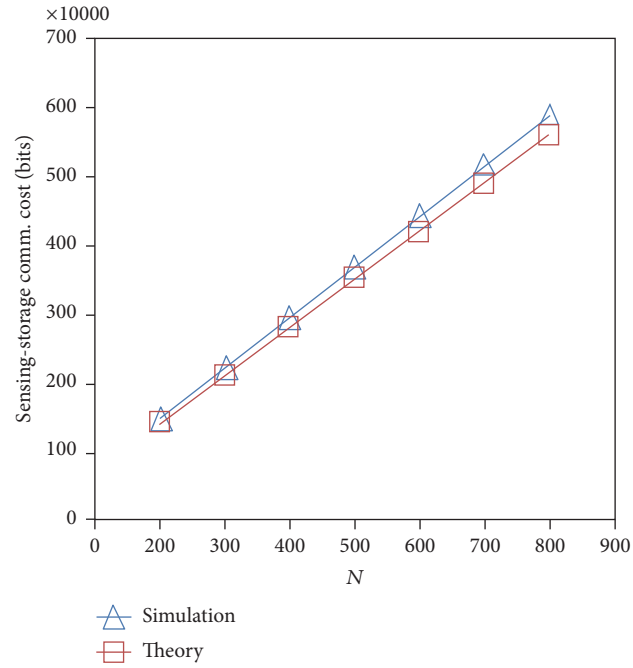

Figure 2: Impact of N on the sensing-storage communication cost.

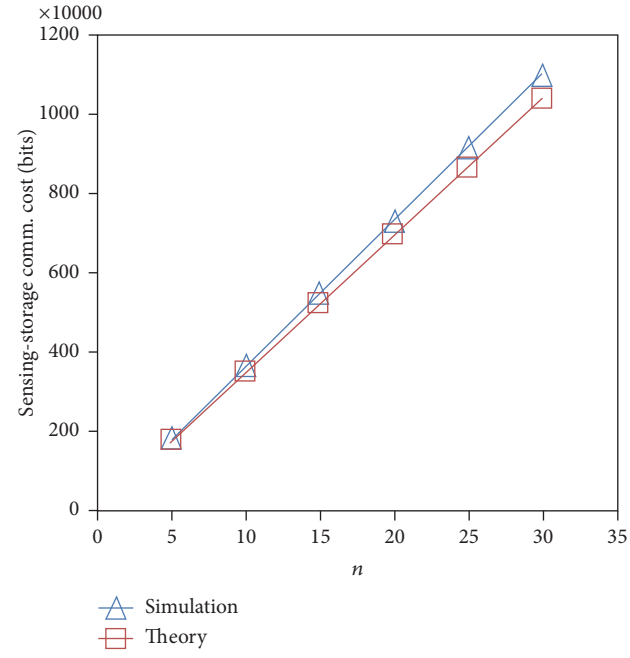

Figure 3: Impact of n on the sensing-storage communication cost.

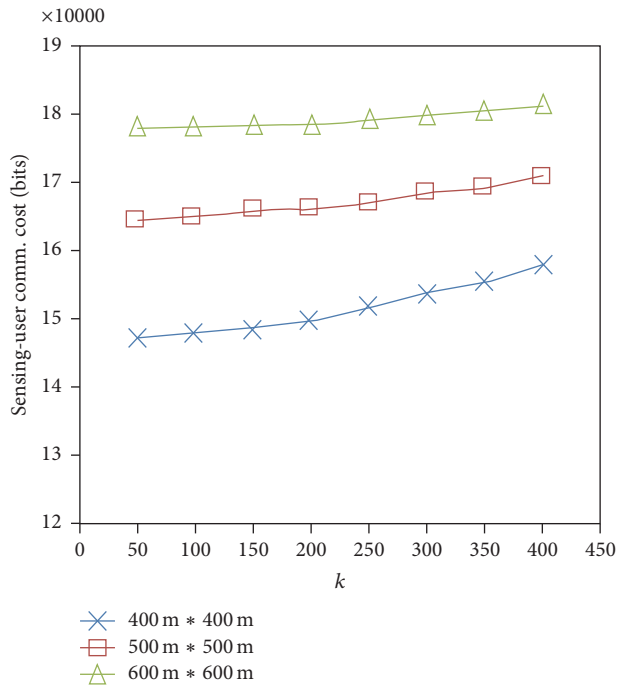

Figure 4: Impact of k and R_q on the storage-user communication cost.

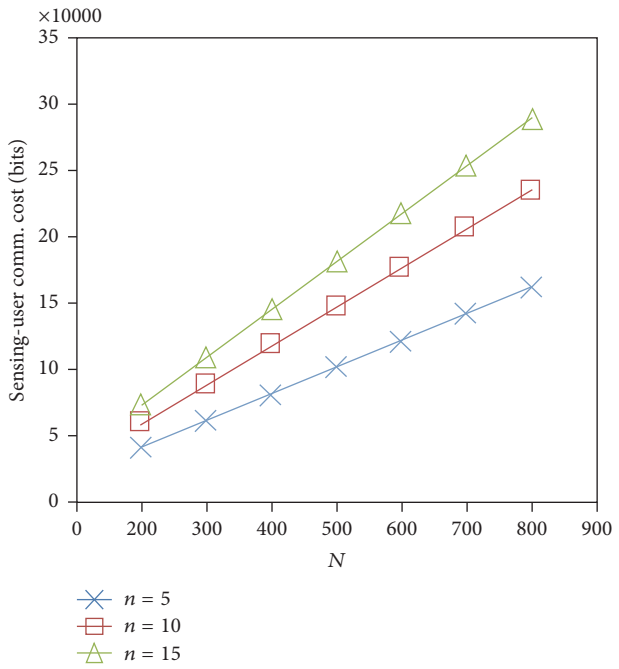

Figure 5: Impact of N and n on the storage-user communication cost.

5.2. Communication Cost between Storage and User Level. As for the communication cost between storage and user level, since we only give the theoretical bound instead of the detailed cost in Theorem 4, we do not compare the theoretical and simulation results of our scheme in this section. Figure 4 illustrates the impact of k on the storage-user communication cost where the query region R_q = 400 m × 400 m, 500 m × 500 m, and 600 m × 600 m, respectively.

As we can see, on one hand, with the increasing number of k, the communication cost grows slowly for a fixed query region. On the other hand, if k keeps unchanged, the larger query region is, the more expensive cost will be incurred. The reason is that more verification information about qualified or unqualified sensor nodes is transmitted to the query user as k or R_q increases. More specifically, if k keeps unchanged, more static or mobile nodes are unqualified with the extension of R_q and hence incurs more verification information about unqualified nodes. In contrast, if k increases, there may be more qualified sensor nodes in a given query region and hence incurs more verification information about qualified nodes. Note that the impact of k is much smaller than that of R_q on the storage-user communication cost, due to the fact that unqualified nodes incur more cost than the qualified nodes. Moreover, it is not required to send the verification information for each qualified data record.

In addition, Figure 5 depicts the impact of N and n on the storage-user communication cost. Similarly, the communication cost between the storage and user level increases linearly with N, which is consistent with the theoretical bound given in Theorem 4. This is because the query region will get denser as N grows in a given query cell, and more verification information about unqualified sensor nodes needs to be sent to the query user. In contrast, with the increasing number of n, more location information about mobile nodes need be transmitted and thus leads to the larger communication cost.

6. Conclusion

In this paper, we study the secure top-k query processing in a novel sensing paradigm, where hybrid sensed data are generated by both static and mobile sensor nodes. To tackle the security threats posed by the compromised storage node, we propose an efficient and verifiable scheme that enables the query user to efficiently verify the authenticity and correctness of the query result. A novel data binding method is designed to generate the verification information, with which the storage node is not required to return verification information for each qualified data record. Theoretical analysis and simulation results demonstrate the security and efficiency of our scheme. Our future work is to investigate privacy preserving top-k query processing with integrity verification on the hybrid sensed data in tiered hybrid sensing network.

Notation Settings

N: The number of sensor nodes in a sensing cell
S_i: The ith sensor node in a sensing cell
r: The communication radius of sensor nodes
D_i: The set of data records sensed by S_i
$D_{i,j}$: The jth data record in D_i
$d_{i,j}$: The score of $D_{i,j}$

$l_{i,j}$: The geographic location where $D_{i,j}$ is generated
n_i: The number of data records in D_i
C_q: The ID of query sensing cell
R_q: The query region
$K_{i,t}$: The key shared between S_i and user in time slot t
I_t: The set of sensor nodes in R_q
$\mathscr{C}_t$: The set of data records sensed by nodes in I_t
RS_t: The set of k query results in time slot t
τ: The lowest data score in query results
γ_i: The number of data records sensed by S_i in RS_t.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant no. 61272074 and no. U1405255 and the Industrial Science and Technology Foundation of Zhenjiang City under Grant no. GY2013030.

References

[1] R. K. Tripathi, Y. N. Singh, and N. K. Verma, "Two-tiered wireless sensor networks—base station optimal positioning case study," *IET Wireless Sensor Systems*, vol. 2, no. 4, pp. 351–360, 2012.

[2] R. Zhang, J. Shi, Y. Liu, and Y. Zhang, "Verifiable fine-grained top-k queries in tiered sensor networks," in *Proceedings of the 29th Conference on Information Communications (INFOCOM '10)*, pp. 1–9, San Diego, Calif, USA, March 2010.

[3] H. Dai, T. Wei, Y. Huang, J. Xu, and G. Yang, "Random secure comparator selection based privacy-preserving MAX/MIN query processing in two-tiered sensor networks," *Journal of Sensors*, vol. 2016, Article ID 6301404, 13 pages, 2016.

[4] H. Dai, G. Yang, F. Xiao, and Q. Zhou, "EVTQ: an efficient verifiable top-k query processing in two-tiered wireless sensor networks," in *Proceedings of the 9th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN '13)*, pp. 206–211, IEEE, Dalian, China, December 2013.

[5] H. Dai, G. Yang, H. Huang, and F. Xiao, "Efficient verifiable Top-k queries in two-tiered wireless sensor networks," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 6, pp. 2111–2131, 2015.

[6] X. Ma, H. Song, J. Wang, J. Gao, and G. Min, "A novel verification scheme for fine-grained top-k queries in two-tiered sensor networks," *Wireless Personal Communications*, vol. 75, no. 3, pp. 1809–1826, 2014.

[7] R. Zhang, J. Shi, Y. Zhang, and X. Huang, "Secure top-k query processing in unattended tiered sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4681–4693, 2014.

[8] R. He, H. Dai, G. Yang, T. Wang, and J. Bao, "An efficient top-k query processing with result integrity verification in twotiered wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2015, Article ID 538482, 8 pages, 2015.

[9] S. Ye, J. Liu, and J. Zhang, "Privacy preservation in a two-tiered sensor network through correlation tracking," in *Proceedings of the 3rd Global Congress on Intelligent Systems (GCIS '12)*, pp. 294–297, IEEE, Wuhan, China, November 2012.

[10] H. Peng, X. Zhang, H. Chen, Y. Wu, J. Zeng, and D. Li, "Enable privacy preservation for k-NN query in two-tiered wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '15)*, pp. 6289–6294, IEEE, London, UK, June 2015.

[11] Y. Yao, L. Ma, and J. Liu, "Privacy-preserving top-k query in twotiered wireless sensor networks," *International Journal of Advancements in Computing Technology*, vol. 4, article 6, 2012.

[12] Y.-T. Tsou, Y.-L. Hu, Y. Huang, and S.-Y. Kuo, "PCTopk: privacy- and correctness-preserving functional top-k query on untrusted data storage in two-tiered sensor networks," in *Proceedings of the 33rd IEEE International Symposium on Reliable Distributed Systems (SRDS '14)*, pp. 191–200, IEEE, Nara, Japan, October 2014.

[13] J. Shi, R. Zhang, and Y. Zhang, "Secure range queries in tiered sensor networks," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 945–953, April 2009.

[14] S. Ling and Q. Dong-Yang, "Secure and low energy consumption range query in tiered sensor networks," *International Journal of Online Engineering*, vol. 12, no. 7, 2016.

[15] D. Yung, Y. Li, E. Lo, and M. L. Yiu, "Efficient authentication of continuously moving k NN queries," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1806–1819, 2015.

[16] Y. Jing, L. Hu, W.-S. Ku, and C. Shahabi, "Authentication of k nearest neighbor query on road networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 6, pp. 1494–1506, 2014.

[17] C.-M. Yu, G.-K. Ni, I.-Y. Chen, E. Gelenbe, and S.-Y. Kuo, "Top-k query result completeness verification in tiered sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 1, pp. 109–124, 2014.

[18] F. Liu, X. Ma, J. Liang, and M. Lin, "Verifiable top-k query processing in tiered mobile sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 437678, 14 pages, 2015.

[19] F.-J. Wu, H.-C. Hsu, Y.-C. Tseng, and C.-F. Huang, "Non-location-based mobile sensor relocation in a hybrid static-mobile wireless sensor network," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications (SENSORCOMM '09)*, pp. 643–649, IEEE, June 2009.

[20] H. Chenji and R. Stoleru, "Toward accurate mobile sensor network localization in noisy environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 6, pp. 1094–1106, 2013.

[21] J.-P. Sheu, W.-K. Hu, and J.-C. Lin, "Distributed localization scheme for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 516–526, 2010.

[22] G. Das, D. Gunopulos, N. Koudas, and D. Tsirogiannis, "Answering top-k queries using views," in *Proceedings of the 32nd international Conference on Very Large Data Bases*, pp. 451–462, VLDB Endowment, Seoul, Republic of Korea, 2006.