Hindawi
Security and Communication Networks
Volume 2021, Article ID 9939796, 16 pages
https://doi.org/10.1155/2021/9939796

Research Article

STQ-SCS: An Efficient and Secure Scheme for Fine-Grained Spatial-Temporal Top-k Query in Fog-Based Mobile Sensor-Cloud Systems

Jie Min,[1] **Junbin Liang**,[2] **Xingpo Ma**,[3] **and Hongling Chen**[4]

[1]*School of Information Engineering, Xinyang Agriculture and Forestry University, Xinyang 464000, Henan, China*
[2]*School of Computer and Electronics Information, Guangxi University, Nanning 530004, Guangxi, China*
[3]*School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, Henan, China*
[4]*Guangdong Polytechnic of Science and Technology, Zhuhai 519090, Guangdong, China*

Correspondence should be addressed to Xingpo Ma; maxingpo@xynu.edu.cn

Received 11 March 2021; Revised 5 May 2021; Accepted 19 May 2021; Published 29 May 2021

Academic Editor: Lu Liu

With the emergence of the fog computing and the sensor-cloud computing paradigms, end users can retrieve the desired sensory data generated by any wireless sensor network (WSN) in a fog-based sensor-cloud system transparently. However, the fog nodes and the cloud servers may suffer from many kinds of attacks on the Internet and become semitrusted, which threatens the security of query processing in the system. In this paper, we investigated the problem of secure, fine-grained spatial-temporal Top-k query in fog-based mobile sensor-cloud systems (FMSCSs) and proposed a novel scheme named STQ-SCS to tackle the problem based on the virtual grid construction and the size-order encryption-binding techniques. STQ-SCS can preserve the privacy of the sensed data items and their scores and make end users verify the completeness of the query results of fine-grained spatial-temporal Top-k queries with a 100% successful rate even if the fog nodes and the cloud servers are not totally trustworthy. Besides the good security performance, simulation results indicate that STQ-SCS is also an efficient scheme that incurs a much lower communication cost than the state-of-the-art schemes on securing fine-grained spatial-temporal Top-k query in FMSCSs.

1. Introduction

As one important component of Internet of Things (IoT) [1], wireless sensor networks (WSNs) [2] can be used in many application scenarios and are still being studied [3] by many researchers even though extensive research has been carried out on WSNs for the past two decades. However, traditional WSNs are usually *single-user centric* [4], where a user deploys and owns its own WSN and another party is not able to access the sensed data generated by such a WSN. To remedy this shortcoming, researchers have conceived a new paradigm, namely, the *sensor-cloud* paradigm [5–7], in recent years. A typical sensor-cloud model is shown in Figure 1(a), where the sensor-cloud architecture serves as the intermediate stratum between the end users and the physical sensor nodes [4]. However, early sensor-cloud architectures are still not perfect, and they encounter many new challenges, such as providing real-times services and efficiently managing the physical sensor nodes. In [8], a new sensor-cloud architecture, namely, the fog-based sensor-cloud framework, was proposed, and the basic model of the fog-based sensor-cloud framework is shown in Figure 1(b). The main difference between early sensor-cloud architectures and the fog-based sensor-cloud framework is that the latter has a fog layer while the former does not have. The fog layer is mainly composed of fog nodes, which can fuse and store the collected sensed data, respond to real-time applications, and efficiently manage the physical sensor nodes [8]. In the fog-based sensor-cloud framework, end users can not only retrieve the sensed data items, which they are interested in directly from the nearby fog nodes, but also obtain the shared sensed data from the cloud by sending queries to the

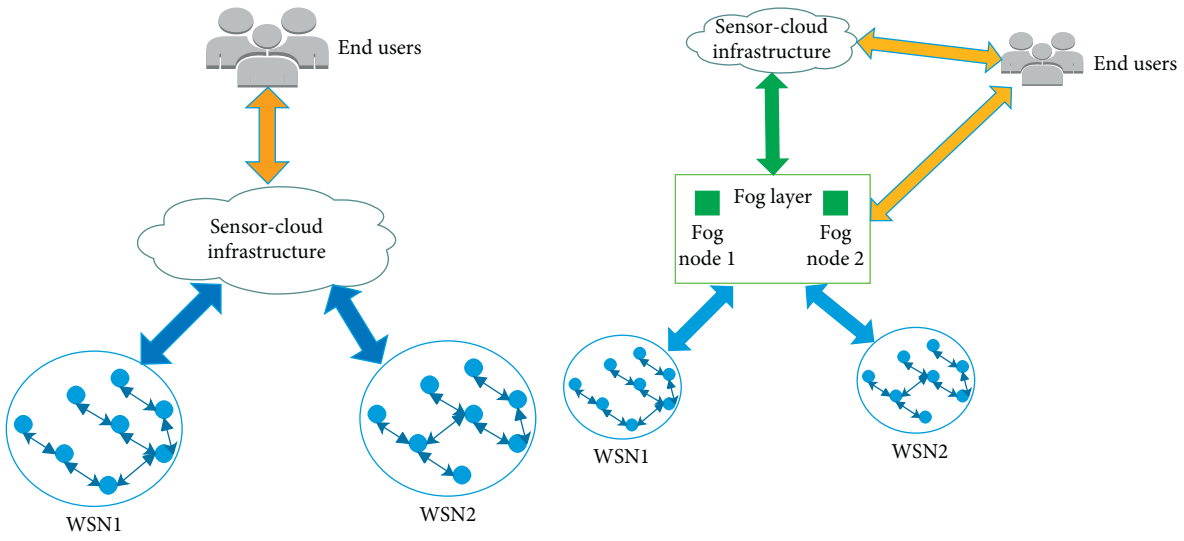

Figure 1: The sensor-cloud architecture (generally, there are more than 2 WSNs in real applications, and we just use 2 WSNs as representatives). (a) Common sensor-cloud architecture. (b) Fog-based sensor-cloud architecture.

cloud if there are no data which they want in the near fog nodes.

Although the fog-based sensor-cloud framework brings a lot of benefits as described in [8], it encounters many potential security threats. The fog nodes may be captured by the nearby attackers or may suffer from the attacks arising from the cloud. In other words, the fog nodes may become untrusted [9, 10] under such attacks. Meanwhile, the application servers in the cloud are facing many kinds of attacks, and some of the cloud servers may also not be trustworthy [11–13]. Under this background, how to ensure the integrity and the confidentiality of the sensed data items retrieved by the end users in the fog-based sensor-cloud systems is a thorny-and-burning problem. Such a problem is much more challenging in fog-based mobile sensor-cloud systems (FMSCSs), where the sensor nodes are mobile, considering that the sensed data retrieved by end users must satisfy the spatial-temporal requirements of the queries launched by end users.

In this paper, we focus on fine-grained spatial-temporal Top-k queries and make efforts to tackle the above-mentioned problem. The concept of fine-grained spatial-temporal Top-k queries is defined in Definition 1 in Section 3. In a word, a fine-grained spatial-temporal Top-k query refers to a query that tries to find out the top k sensed data items generated in a specific time interval and a specific region of a specific WSN deployment field. To our best knowledge, there is no work studying the problem of secure fine-grained spatial-temporal Top-k query in fog-based sensor-cloud systems at present. In brief, the main contributions of this paper are twofold:

(i) It studies the problem of secure fine-grained spatial-temporal Top-k query in FMSCSs and proposes a novel scheme named STQ-SCS to ensure the integrity and confidentiality of the sensed data items retrieved by end users. It provides sound theoretical analysis on the security of STQ-SCS. According to the analysis, STQ-SCS is not only able to preserve the privacy of the sensed data items retrieved by end users but also detect the incomplete query results successfully for fine-grained spatial-temporal Top-k query under the security model presented in this paper.

(ii) Extensive simulations were conducted in the paper, and the results show that STQ-SCS is much more efficient than the related state-of-the-art schemes.

The remainder of this paper is organized as follows. Section 2 summaries the related schemes; Section 3 describes the system model, the security model, the definitions of some terminologies, and the problem statement; Section 4 presents the proposed scheme STQ-SCS in detail; Section 5 analyzes the security of STQ-SCS; In Section 6, STQ-SCS is compared with the related state-of-the-art schemes through extensive simulations; Section 7 provides performance evaluation. Section 8 concludes this study.

2. Related Works

Since there is no work about secure fine-grained spatial-temporal Top-k query in FMSCSs at present, we mainly investigate the related works in Cloud Computing, Two-tiered Wireless Sensor Networks (TWSNs), and Two-tiered Mobile Wireless Sensor Networks (TMWSNs) in this section.

2.1. Securing Top-k Queries in Cloud Computing. Top-k queries in the cloud are generally securely processed based on the data that are outsourced on cloud servers by the same data owner. In Cloud Computing, the data owner knows all its outsourced data and thus can construct the encrypted

data structure, such as EHL [14], the binary heap [15], or other tree-like structures [16–18], based on the whole data set to facilitate Top-k query without losing data privacy, while in FMSCSs, expect for the fog nodes that are considered as not fully trusted, each sensor node just knows only a small part of the whole data generated by the WSN where it is located, and it thus cannot construct the encrypted data structure of the whole data before outsourcing its data to a fog node or the cloud.

Moreover, existing schemes proposed for secure Top-k query in Cloud Computing are based on the strong processing ability and rich resources of the cloud servers, and they never consider the resource-limited sensor nodes which are also weak in computing. Thus, they are not fit for FMSCSs.

2.2. Securing Top-k Queries in TWSNs. The study of securing Top-k queries in TWSNs was originally launched by the authors in [19], where three schemes are proposed to preserve the completeness of the Top-k query results in TWSNs. The three schemes were proposed based on the MAC (Message Authentication Code) technique, which requires each sensed data item to be attached with an MAC as its proof data. Then, many other schemes that use a similar technique appeared, such as those in [19–24]. However, the MAC-based technique is relatively less efficient because attaching an MAC to each sensed data item brings large quantity of extra data since a MAC takes almost 40% of the volume of a sensed data item according to [19].

Besides the MAC-based technique, some other methods were also proposed to ensure the privacy of the sensed data and the completeness of the Top-k query results in TWSNs, such as inserting digital watermarks or dummy readings into the normal ones [25] and constructing data aggregation trees [26, 27]. However, inserting digital watermarks or dummy readings into the measure data makes it hard and complicated for the users to extract the normal readings from the hybrid ones, and it also brings a lot of redundant data, which further leads to the increase of the communication cost of both the sensor nodes and fog nodes.

What is more, one of the most important common points of these schemes is that they are all proposed for TWSNs where nodes are static [28], and they cannot perfectly treat the security threats faced by spatial-temporal Top-k query in FMSCSs, where attackers can launch much more covert attacks. When a mobile sensor node travels from the queried region to other regions or vice versa in the queried time interval, some sensed data generated by the sensor node may be in the queried region, and others may not. Obviously, the sensed data generated out of the queried region by the traveling sensor node are not the qualified ones that satisfy the requirements of the spatial-temporal Top-k query. However, few securing Top-k query schemes proposed in TWSNs consider this, which leaves leaks for the attackers to launch new kinds of covert attacks. For example, the attackers may replace the data items that are generated in the queried region by a sensor node with those produced out of the queried region by the same sensor node.

2.3. Securing Top-k Queries in TMWSNs. The first work on securing Top-k queries in TMWSNs was done by Liu et al. in 2015 [29], when they presented a novel network architecture, namely, TMWSNs, and proposed a scheme VTMSN to ensure the completeness of spatial-temporal Top-k query in TMWSNs. The main techniques used in VTMSN are symmetric encryption and information binding. Specifically, it binds the score of each sensed data item with its corresponding generation time, location, and value ranking order by concatenating and encrypting them with the kept symmetric key. Although VTMSN increases the difficulty for the attackers to undermine the completeness of the query results because of the binding relationships, it still has shortcomings. One is that it cannot preserve the privacy of the sensed data items since it leaves the data items disclosed to the fog nodes for ease of Top-k query processing on them; another one is that there should be a large volume of location data transported together with the sensed readings, which greatly increases the communication cost of the sensor nodes and fog nodes.

To overcome the latter shortcoming of VTMSN, Wu et al. proposed a scheme named EVTopk [30] in 2016. EVTopk achieves completeness preservation of the Top-k query results by using the HMAC (Hash Message Authentication Code), which is formed by making hashing and encryption operations on the concatenated items including the score, the location, and the neighboring HMAC. However, since each sensed data item should be attached with an HMAC in EVTopk, the HMACs account for a large proportion of the data reports of the sensor nodes and the query results. Moreover, EVTopk is not able to achieve data privacy preservation either. In [31], a comparative study was made on the two schemes, EVTopk and VTMSN. To further decrease the volume of the proof data in the data reports and the query results, in 2018, a scheme named VIP-TQ was proposed to preserve the integrity of the query results for spatial-temporal Top-k query in TMWSNs. In VIP-TQ, sensed data are bound together with their location as well as their neighboring data score using pairwise-key-based encryption. Although the binding can effectively prevent the compromised fog nodes from undermining the integrity of the Top-k query results, it leaves the scores of the sensed data disclosed to the storage nodes, which increases the risk of divulging the privacy of the sensed data. In the same year, Ma et al. proposed two other schemes, namely, SSSTQ1 and SSSTQ2 [32], for securing spatial-temporal Top-k in TMWSNs. However, a large number of original locations associated with the sensed data items are added into the data reports and the query results for integrity verification, which heavily increases the communication cost of the systems.

In summary, although there are many schemes related to secure Top-k query in existing works, they either have obvious shortcomings or cannot be used in FMSCS, which motivates us to do further work in this paper.

3. Models, Notations, and Problem Statement

3.1. System Model. The system model of FMSCSs is shown in Figure 2. In the model, TA is short for trusted authority [33],

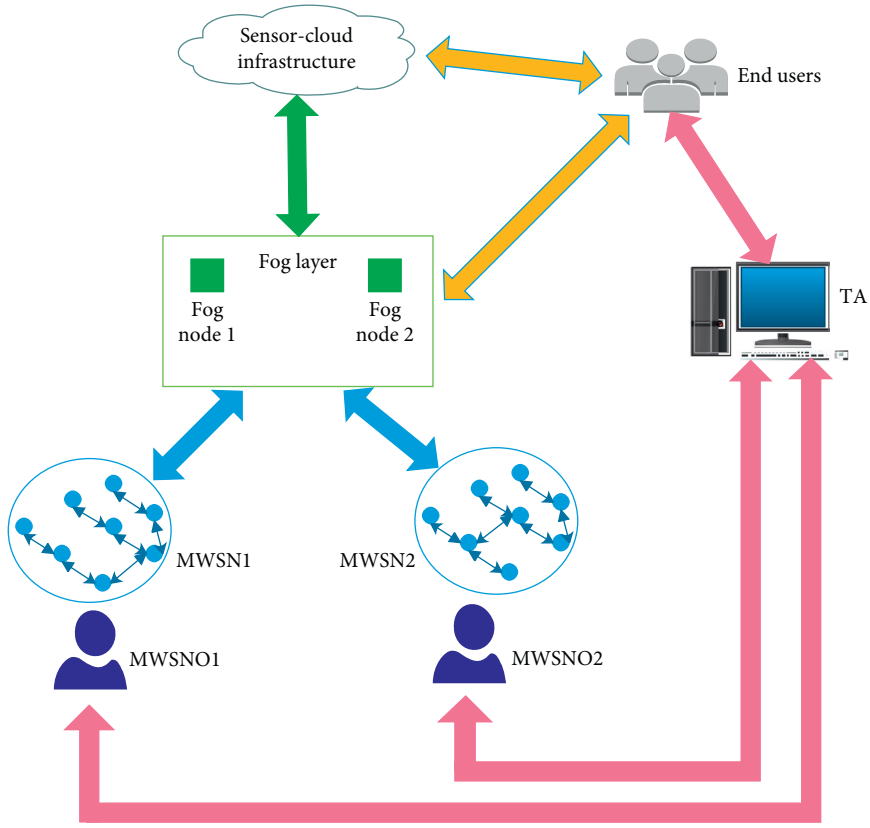

Figure 2: System model of FMSCSs.

which is a trustworthy party. TA is used to authenticate the identity of end users and MWSNOs (Mobile Wireless Sensor Network Owners) and distribute the secret keys to them. Each fog node in the fog layer connects and manages one MWSN (Mobile Wireless Sensor Network), and each MWSN is assumed to be composed of N mobile sensor nodes and is owned by a MWSNO. Specifically, the main responsibility of each fog node is as follows: (1) Collecting, processing, and storing the sensed data items updated by the sensor nodes in its corresponding WSN; (2) managing the mobile sensor nodes in its corresponding MWSN; and (3) responding to the queries that may be sent from the Cloud or the end users directly. End users can retrieve the desired data by launching and sending queries to the cloud or the fog nodes directly if they are not far from the fog nodes. If a cloud server receives a query from some end user, it first determines the fog node, which satisfies the region requirement of the query, and then sends the query to the fog node; if a fog node receives a query, it processes the query locally and sends the query result to the party (the cloud or the end user) who has sent the query.

The mobile sensor nodes in WSNs periodically upload their sensed data to the corresponding fog nodes in the fog layer. We divide time into epochs, and take the time length of each epoch as the period for each sensor node to upload its sensed data items. We assume that mobile sensor nodes in each WSN do not move all the time. They stay at some target locations for certain time intervals when they reach the positions, and go on moving to other target locations if it is necessary. Moreover, we assume that the mobile sensor nodes only generate sensed data items when they are staying at their target locations. Besides, it is assumed that each mobile sensor node just moves within the WSN field where it is located, since it will cost a lot of energy for the sensor nodes to move among different WSN-deployed fields.

In this paper, we use the set $\{D^t_{i,j,1}, D^t_{i,j,2}, \ldots, D^t_{i,j,\mu^t_{i,j}-1}, D_{i,j,\mu_{i,}}\ j^t t\}$ to denote the sensed data items generated by sensor node S_i at its j^{th} target location in the t^{th} epoch T^t, where $\mu^t_{i,j}$ is the total number of the sensed data items generated by S_i at its j^{th} target location in T^t. For any sensed data item $D^t_{i,j,x}$, its corresponding data score $d^t_{i,j,x}$ can be worked out using a public scoring function $f(*)$ [19], namely, $d^t_{i,j,x} = f(D^t_{i,j,x})$. Without loss of generality, we assume different sensed data items have distinct scores [19]. Moreover, in order to facilitate presentation, we assume that the ranking orders of the sensed data items generated by any sensor node at a target location are consistent with their subscript digital numbers. For example, there is $D^t_{i,j,1} < D^t_{i,j,2} < \cdots < D^t_{i,j,\mu^t_{i,j}-1} < D^t_{i,j,\mu^t_{i,j}}$, where i and j are the node ID and the target location ID of S_i, respectively. The specific meanings of the notations used in this paper are listed in Table 1.

Table 1: Notations and their meanings.

Notations	Meanings
S_i	The sensor node whose ID is $i\,(0 < i \le N)$
N	Total number of sensor nodes in one MWSN
T^t	The t^{th} epoch
λ_i^t	Total number of target locations of S_i in T^t
$\text{Loc}_{i,j}^t$	The j^{th} target location of S_i during T^t
$\mu_{i,j}^t$	Total data item numbers of S_i generated at $\text{Loc}_{i,j}^t$ in T^t
$n_{i,j}^t$	Total number of qualified Top-k data items generated by S_i at $\text{Loc}_{i,j}^t$ in T^t
Q^t	A spatial-temporal Top-k query
R^t	The query result of Q^t
I_{Q^t}	The ID of Q^t
I_{MWSN}	The ID of an MWSN
$\text{QR}_{I_{\text{MWSN}}}$	The queried region in an MWSN whose ID is I_{MWSN}
Key_i	The pairwise key which is distributed to sensor node S_i
$RT_{S_i}^t$	The data report generated by S_i in T^t
$E_{\text{Key}_i}\{*\}$	Symmetric encrypting operation with Key_i based on [34]
$E_{\text{OPE}}\{*\}$	Encrypting operation based on the OPE encryption scheme [35]
$\text{RST}_{S_i}^t$	The processed result of $\text{RT}_{S_i}^t$
Ω_i	Total number of the queried locations encrypted in $\text{RST}_{S_i}^t$
$\gamma_{i,j}^t$	Total number of the sensed data items encrypted in $\text{DPP}_{i,j}^t$
R_{tpk}	Set of the qualified Top-k data items extracted from R^t

3.2. Definitions. In this section, we introduce the definitions of some terminologies used in this paper. Specifically, we define the terminologies used in this paper as follows:

(i) Fine-grained spatial-temporal Top-k query: it is the query which tries to find out the top k sensed data items that have the biggest (or the smallest) scores among all the sensed data items generated in $\text{QR}_{I_{\text{MWSN}}}$ in T^t, where $\text{QR}_{I_{\text{MWSN}}}$ is a subregion of the deployment field of the MWSN whose ID is I_{MWSN}. The meta-language of a fine-grained spatial-temporal Top-k query Q^t in FMSCSs is shown in the following equation:

$$Q^t = \left\{I_{Q^t}, T^t, k, I_{\text{MWSN}}, \text{QR}_{I_{\text{MWSN}}}\right\}. \tag{1}$$

(ii) Queried node and queried location: given a spatial-temporal Top-k query $Q^t = \{I_{Q^t}, T^t, k, I$ MWSN, $\text{QR}_{I_{\text{MWSN}}}\}$, if a target location of any mobile sensor node falls in $\text{QR}_{I_{\text{MWSN}}}$ in T^t, the target location is one of the queried locations of Q^t; if at least one of the target locations of a mobile sensor node is one of the queried locations of Q^t, the sensor node is called a queried node of Q^t.

(iii) Qualified Top-k data items: given a spatial-temporal Top-k query $Q^t = \left\{I_{Q^t}, T^t, k, I_{\text{MWSN}}, \text{QR}_{I_{\text{MWSN}}}\right\}$, if a sensed data item $D_{\text{qualified}}^t$ satisfies the following two conditions, it is called the qualified Top-k data item of Q^t: (1) $D_{\text{qualified}}^t$ was generated in $\text{QR}_{I_{\text{MWSN}}}$ and T^t; (2) among all the sensed data items generated in T^t and T^t, there are at least $N_{Q^t} - k$ data items whose scores are smaller (or bigger) than the score of $D_{\text{qualified}}^t$, where N_{Q^t} refers to the total number of the sensed data items generated in $\text{QR}_{I_{\text{MWSN}}}$ and T^t.

(iv) Data-proof Packet $\text{DPP}_{i,j}^t$: for any target location $\text{Loc}_{i,j}^t\,(0 < j \le \lambda_i^t)$ of any mobile sensor node S_i $(1 \le i \le N)$, Data-proof Packet $\text{DPP}_{i,j}^t$ refers to the subreport produced by S_i for the sensed data generated at $\text{Loc}_{i,j}^t$ during T^t. Specifically, $\text{DPP}_{i,j}^t$ consists of the pairwise-key-encrypted sensed data items and the OPE-encrypted scores ("OPE" is short for "order-preserving encryption" [35]) as well as some proof information generated by S_i at $\text{Loc}_{i,j}^t$ during T^t. More specific contents of $\text{DPP}_{i,j}^t$ will be described in Algorithm 1 in Section 4.

3.3. Security Model. In FMSCSs, fog nodes and the cloud servers are assumed to be untrusted, while most of the mobile sensor nodes and TA are trustworthy. We assume that the untrusted fog nodes and cloud servers are not only curious but also malicious. Specifically, a curious fog node or cloud server will try to disclose the sensed data items as well as the data scores computed based on the public scoring function, and a malicious fog node or cloud server will do its best to undermine the completeness of the results of the fine-grained spatial-temporal Top-k queries. To execute a malicious attack, an untrusted fog node may put none or only part of the qualified top k data items into the Top-k query result, and it may also put some fabricated data items and/or the unqualified-but-real ones into the query result when processing a spatial-temporal Top-k query. For example, suppose the complete query result should be $\{D_1^t, D_2^t, D_3^t\}$. Then, an incomplete query result may be $\{D_1^t\}$ or $\{D_1^t, D_4^t, D_{\text{fabricated}}^t\}$, where D_4^t is a real but unqualified sensed data item and $D_{\text{fabricated}}^t$ is a fabricated data item. An untrusted cloud server may also make some wrong deletions or replacements to undermine the integrity of the query results before it transmits the query results to end users.

In our security model, the privacy of the sensed data items, which are generated by the mobile sensor nodes in FMSCSs, and their corresponding scores should be

Ensure: target location set $\left\{\text{Loc}_{i,1}^t, \text{Loc}_{i,2}^t, \ldots, \text{Loc}_{i,\lambda_i^t-1}^t, \text{Loc}_{i,\lambda_i^t}^t\right\}$; all the sensed data items generated by S_i in T^t; the pairwise key Key_i; the master key used for OPE;
Require: $\text{RT}_{S_i}^t$;
(1) Compute the score of each sensed data item using the public scoring function;
(2) **for** $j = 1$ to λ_i^t **do**
(3) **if** $\mu_{i,j}^t = 0$ **then**
(4) Set $\text{DPP}_{i,j}^t$ to $\left\{\text{Loc}_{i,j}^t, E_{\text{Key}_i}\left\{0, \text{Loc}_{i,j}^t\right\}\right\}$;
(5) **end if**
(6) **if** $\mu_{i,j}^t = 1$ **then**
(7) Set $\text{DPP}_{i,j}^t$ to $\left\{\text{Loc}_{i,j}^t, E_{\text{Key}_i}\left\{1, \text{Loc}_{i,j}^t\right\}, E_{\text{Key}_i}\left\{d_{i,j,1}^t, \text{Loc}_{i,j}^t\right\}, E_{\text{OPE}}\left\{d_{i,j,1}^t\right\}, E_{\text{Key}_i}\left\{D_{i,j,1}^t, \text{Loc}_{i,j}^t\right\}\right.$;
(8) **end if**
(9) **if** $\mu_{i,j}^t > 1$ **then**
(10) Sort the sensed data items generated by S_i at $\text{Loc}_{i,j}^t$ in T^t according to their scores;
(11) Set $\text{DPP}_{i,j}^t$ to $\left\{\text{Loc}_{i,j}^t, E_{\text{Key}_i}\left\{\mu_{i,j}^t, \text{Loc}_{i,j}^t\right\}, E_{\text{Key}_i}\left\{d_{i,j,1}^t, \text{Loc}_{i,j}^t, E_{\text{OPE}}\left\{d_{i,j,1}^t\right\}, E_{\text{Key}_i}\left\{1, D_{i,j,1}^t, \text{Loc}_{i,j}^t, \ldots, E_{\text{OPE}}\left\{d_{i,j,\mu_{i,j}^t-1}^t\right\}, E_{\text{Key}_i}\left\{\mu_{i,j}^t - 1,\right.\right.\right.\right.$
$\left.\left.\left.\left.D_{i,j,\mu_{i,j}^t-1}^t, \text{Loc}_{i,j}^t, E_{\text{OPE}}\left\{d_{i,j,\mu_{i,j}^t}^t\right\}, E_{\text{Key}_i}\left\{D_{i,j,\mu_{i,j}^t}^t, \text{Loc}_{i,j}^t\right\}\right.\right.\right.\right.$;
(12) **end if**
(13) **end for**
(14) Set $\text{RT}_{S_i}^t$ to $\left\{i, t, E_{\text{key}_{i,t}}\left\{\text{Loc}_{i,1}^t, \text{Loc}_{i,2}^t, \ldots, \text{Loc}_{i,\lambda_i^t-1}^t, \text{Loc}_{i,\lambda_i^t}^t, \text{DPP}_{i,1}^t, \text{DPP}_{i,2}^t, \ldots, \text{DPP}_{i,\lambda_i^t-1}^t, \text{DPP}_{i,\lambda_i^t}^t\right.\right.$
(15) Return $\text{RT}_{S_i}^t$.

Algorithm 1: Secure data preprocessing on $S_i\,(0 < i \le N)$.

protected. Other information, such as spatial-temporal Top-k query and the generation locations of the sensed data items, will be leaked to fog nodes. It is hard to enable fog nodes to process spatial-temporal Top-k query smoothly and successfully without such leaks. Fortunately, the leaked information brings little threat to the safety of the systems. Moreover, we assume each mobile sensor node is assumed to be equipped with the tamper-proof hardware, with the help of which the adversaries cannot disclose the encryption materials stored in the hardware even if they capture the sensor nodes [24].

3.4. Problem Statement and Design Goal. Under the system and the security models described above, the problem tackled in this paper can be presented as follows: how to make the end users in FMSCSs obtain the query results of the fine-grained spatial-temporal Top-k queries launched by them without disclosing the sensor data items and their corresponding scores to the fog nodes and the cloud servers and verify the completeness of the corresponding query result correctly and efficiently. Our design goal is to propose a novel scheme that enables efficient privacy-preservation and integrity-verifiable query processing for fine-grained spatial-temporal Top-k query in FMSCSs. Specifically, three objects as follows should be achieved:

(i) The privacy preservation goal: our proposed scheme should preserve the privacy of the sensed data items and their scores collected from the mobile sensor nodes.

(ii) The integrity verification goal: our proposed scheme should enable end users to verify the completeness of spatial-temporal Top-k query results, no matter what attacking means introduced in the security model are adopted.

(iii) The efficiency goal: our proposed scheme should be effective in communication and computation. It should greatly decrease the additional communication cost of the sensor nodes, since the sensor nodes are energy-limited. Here, the additional communication cost mainly refers to the cost of transmitting the proof data that are used to verify the completeness of the query results.

4. Our Scheme STQ-SCS

This section presents our scheme STQ-SCS. We first make a high-level description of the scheme as follows. At first, each MWSNO obtains the secret keys from TA and preload the keys to its own MWSN. Then, using the secret keys, each sensor node encrypts its own sensed data items and the scores, and uploads the encrypted data items and their scores to the corresponding fog node. If an end user wants to retrieve the query result of a fine-grained spatial-temporal Top-k query, it sends the query to the cloud server or to the fog node directly if it is near the fog node of the target MWSN. If a cloud server receives the query, it first determines which fog node should be the target node of the query, and then sends the query to the target fog node. If the target fog node receives the query, it will work out all the qualified Top-k data items, put them into the query result packet, and send them to the cloud server or to the end user directly if the query is received by the fog node from the end user. If a cloud server receives the query result from the fog node, it will transmit the query result to the end user who is the launcher of the query.

As a whole, STQ-SCS can be mainly divided into five parts: (1) secret key distribution; (2) virtual-location construction; (3) secure data preprocessing; (4) secure spatial-temporal Top-k query processing; (5) completeness verification of the query results. In the following sections, the five parts of STQ-SCS are described in great detail.

4.1. Secret Key Distribution. In STQ-SCS, all secret keys used in FMSCSs are distributed by TA. To obtain the secret keys, each MWSNO sends a key-request message, which contains its own public key, the ID of its own MWSN, the IDs of the mobile sensor nodes in the MWSN, and some authentication information, to TA. After authenticating the identity of the MWSNO using some existing authentication method such as UAP-BCIoT [36], TA knows whether the MWSNO has the authority to obtain the secret keys or not. If TA determines to send the keys to the MWSNO, TA distributes a master key for the MWSN and a pairwise key for each mobile sensor node in the MWSN, encrypts them using the public key of the MWSNO, and then sends them to the MWSNO. The pairwise keys are generated based on the method in [34], while the master key is generated according to the scheme in [35]. Using the similar way, legal end users can also obtain the keys of each mobile sensor node in any MWSN from TA.

In our scheme, two encryption methods are leveraged to encrypt the sensed data items and their scores: one is the latest order preservation encryption (OPE) scheme [35] and the other one is the pairwise-key-based encryption [34]. The former is used to encrypt the scores of the sensed data items using the master keys, while the latter is used to encrypt the sensed data items and the proof data, such as the target locations of the sensor nodes and the ranking orders of the sensed data items, using the pairwise keys. Section 4.3 will describe this in detail.

4.2. Construction of the Virtual Grids. In STQ-SCS, the sensor deployment field is divided into many virtual grids. Each virtual grid should be as small as possible so that the central location of the grid can be approximately taken as the location of every point in the grid in real applications. Then, we design an ID distribution law for the virtual grids. Based on the law, the real locations of each mobile sensor node can be worked out easily if the IDs of the virtual grids where it has moved to are known.

Specifically, the ID distribution law is described as follows. Suppose the FMSCSs-deployed field is a $L * L$ square rectangle. STQ-SCS divides the rectangle into $\eta = (L/\zeta)^2$ small virtual grids, where ζ is a small digital number that can divide the length L with no remainder. Clearly, the smaller ζ is, the larger η is. Then, each virtual grid is given an ID, which is a sequence number ranging from 1 to η. The virtual grids in the first row at the upper side of the rectangle are given the IDs 1, 2, 3, ..., $L/(\zeta - 1)$, and L/ζ, respectively, from the left to the right in order; the IDs $L/(\zeta + 1)$, $L/(\zeta + 2)$, ..., $2 * (L/\zeta) - 1$, and $2 * (L/\zeta)$ are assigned to those in the second row orderly; ...; those in the last row have the IDs $\eta - L/(\zeta + 1)$, $\eta - L/(\zeta + 2)$, ..., $\eta - 1$, and η, respectively.

Using such an ID distribution law, each sensor node first works out the IDs of the virtual grid where it has moved to, and then takes the IDs as the coordinate values of its target locations.

4.3. Secure Data Preprocessing. This section describes how each sensor node generates its data report, which will be uploaded to the corresponding fog node at the end of each epoch, based on its own sensed data items under the privacy-and-integrity preservation requirements. Specifically, for any sensor node S_i $(0 < i \leq N)$, the procedure of data report generation in STQ-SCS is shown in Algorithm 1.

In the protocol, S_i firstly computes the score of each sensed data item generated by itself based on the public scoring function; then, it works out $\mathrm{DPP}_{i,j}^t$ $(0 < j \leq \lambda_i^t)$ for each of its target locations which it has been moved to during epoch T^t. To do this, three cases are considered: $\mu_{i,j}^t = 0$, $\mu_{i,j}^t = 1$, and $\mu_{i,j}^t > 1$. If $\mu_{i,j}^t = 0$, $\mathrm{DPP}_{i,j}^t$ should include $E_{\mathrm{Key}_i}\{0, \mathrm{Loc}_{i,j}^t\}$ to show that no sensed data were generated by S_i at $\mathrm{Loc}_{i,j}^t$ in epoch T^t, where $E_{\mathrm{Key}_i}\{*\}$ is a symmetric encrypting operation with Key_i based on [34]; if $\mu_{i,j}^t = 1$, $\mathrm{DPP}_{i,j}^t$ should contain $E_{\mathrm{Key}_i}\{0, \mathrm{Loc}_{i,j}^t\}$ to indicate that only one sensed data item was generated by S_i at $\mathrm{Loc}_{i,j}^t$ in epoch T^t, and it also needs to include both the pairwise-key-encrypted score and the OPE-encrypted score of the only data item. The former will be used as part of the proof information for integrity verification, and the latter will be used by fog nodes to process spatial-temporal Top-k query smoothly. The only sensed data item should also be encoded using the pairwise key and included in $\mathrm{DPP}_{i,j}^t$. If $\mu_{i,j}^t > 1$, the contents of $\mathrm{DPP}_{i,j}^t$ are a little complex. Specifically, it contains not only the OPE-encrypted scores and the pairwise-key-encrypted data items and scores but also the chaining relationships of the ranked sensed data items. The chaining relationships, which are used to prevent the adversaries from destroying the integrity of the Top-k query results by dropping part of the qualified Top-k data items, are achieved by encrypting each sensed data item together with its ranking order number, which is called the sequence number in the following of this paper, using the pairwise key Key_i. Moreover, each sensed data item is bond together with its corresponding target location to further strengthen the integrity preservation of the Top-k query results. The final output $\mathrm{RT}_{S_i}^t$ in Algorithm 1 is the very data report which will be uploaded to the corresponding fog node of S_i.

4.4. Secure Spatial-Temporal Top-k Query Processing. This section presents how a fine-grained spatial-temporal Top-k query is processed in FMSCSs in our proposed scheme STQ-SCS. When a cloud server receives a fine-grained spatial-temporal Top-k query from an end user, it first finds out the destination of the query according to the mapping relationships between the MWSN IDs and the fog nodes (Information about the mapping relationships is assumed to be stored in the cloud server). Then, the cloud server sends the query to the target fog node. When the target fog node receives the query, it processes the query according to Algorithm 2. After that, it sends the processing result back to the cloud server. If the query is sent from an end user, the fog node will send the query result back to the end user directly.

In Algorithm 2, the fog node first processes every data report uploaded by the sensor nodes in MWSN I_{MWSN} and then packets all the processing results of the data reports collected in the queried MWSN to form the final query result of the spatial-temporal Top-k query. Specifically, lines 1–9

Ensure: $\{\mathrm{RT}^t_{S_1}, \mathrm{RT}^t_{S_2}, \ldots, \mathrm{RT}^t_{S_{N-1}}, \mathrm{RT}^t_{S_N}\}$; $Q^t = \{I_{Q^t}, T^t, k, I_{\mathrm{MWSN}}, \mathrm{QR}_{I_{\mathrm{MWSN}}}\}$;
Require: R^t;
(1) **for** $i = 1$ to N **do**
(2) $n[i] = 0$;
(3) **for** $j = 1$ to λ^t_i **do**
(4) **if** $\mathrm{Loc}^t_{i,j}$ is in $\mathrm{QR}_{I_{\mathrm{MWSN}}}$ **then**
(5) put $\mathrm{DPP}^t_{i,j}$ into set Θ;
(6) $n[i] = n[i] + 1$;
(7) **end if**
(8) **end for**
(9) **end for**
(10) Find out the pairwise-key-encrypted qualified Top-k data items among all the pairwise-key-encrypted data items in set Θ according to their corresponding OPE-encrypted scores;
(11) Calculate $n^t_{i,j}$ for each $i \in [1, N]$ and $j \in [1, \lambda^t_i]$;
(12) **for** $i = 1$ to N **do**
(13) **if** $n[i] = 0$ **then**
(14) Set RST_{S_i} to $\{i, t, E_{\mathrm{key}_{i,t}}\{\mathrm{Loc}^t_{i,1}, \mathrm{Loc}^t_{i,2}, \ldots, \mathrm{Loc}^t_{i,\lambda^t-1}, \mathrm{Loc}^t_{i,\lambda^t}\}\}$
(15) **else**
(16) **for** $j = 1$ to $n[i]$ **do**
(17) **if** $\mu^t_{i,x_j} = 0$ **then**
(18) Set DPP^t_{i,x_j} to $\{E_{\mathrm{Key}_i}\{0, \mathrm{Loc}^t_{i,x_j}\}\}$;
(19) **end if**
(20) **if** $n^t_{i,x_j} = 0, \mu^t_{i,x_j} > 0$ **then**
(21) set DPP^t_{i,x_j} to $\{E_{\mathrm{Key}_i}\{d^t_{i,x_j,1}, \mathrm{Loc}^t_{i,x_j}\}\}$;
(22) **end if**
(23) **if** $0 < n^t_{i,x_j} = \mu^t_{i,x_j} \le k$ **then**
(24) **if** $n^t_{i,x_j} = 1$ **then**
(25) Set DPP^t_{i,x_j} to $\{E_{\mathrm{Key}_i}\{1, \mathrm{Loc}^t_{i,x_j}\}, E_{\mathrm{Key}_i}\{D^t_{i,x_j,1}, \mathrm{Loc}^t_{i,x_j}\}\}$;
(26) **end if**
(27) **if** $n^t_{i,x_j} > 1$ **then**
(28) set DPP^t_{i,x_j} to $\{n^t_{i,x_j}, E_{\mathrm{Key}_i}\{\mu^t_{i,x_j}, \mathrm{Loc}^t_{i,x_j}\}, E_{\mathrm{Key}_i}\{1, D^t_{i,x_j,1}, \mathrm{Loc}^t_{i,x_j}\}, \ldots, E_{\mathrm{Key}_i}\{\mu^t_{i,x_j} - 1, D^t_{i,x_j,\mu^t_{i,x_j}-1}, \mathrm{Loc}^t_{i,x_j}\}, E_{\mathrm{Key}_i}\{D^t_{i,x_j,\mu^t_{i,x_j}}, \mathrm{Loc}^t_{i,x_j}\}$;
(29) **end if**
(30) **end if**
(31) **if** $0 < n^t_{i,x_j} \le k, \mu^t_{i,x_j} > n^t_{i,x_j}$ **then**
(32) **if** $\mu^t_{i,x_j} = n_{x_i,t} + 1$ **then**
(33) Set DPP^t_{i,x_j} to $\{n^t_{i,x_j}, E_{\mathrm{Key}_i}\{\mu^t_{i,x_j}, \mathrm{Loc}^t_{i,x_j}\}, E_{\mathrm{Key}_i}\{1, D^t_{i,x_j,1}, \mathrm{Loc}^t_{i,x_j}\}, \ldots, E_{\mathrm{Key}_i}\{n^t_{i,x_j}, D^t_{i,x_j,n^t_{i,x_j}}, \mathrm{Loc}^t_{i,x_j}\}, E_{\mathrm{Key}_i}\{D^t_{i,x_j,\mu^t_{i,x_j}}, \mathrm{Loc}^t_{i,x_j}\}$;
(34) **end if**
(35) **if** $\mu^t_{i,x_j} > n^t_{i,x_j} + 1$ **then**
(36) set DPP^t_{i,x_j} to $\{n^t_{i,x_j}, E_{\mathrm{Key}_i}\{\mu^t_{i,x_j}, \mathrm{Loc}^t_{i,x_j}\}, E_{\mathrm{Key}_i}\{1, D^t_{i,x_j,1}, \mathrm{Loc}^t_{i,x_j}\}, \ldots, E_{\mathrm{Key}_i}\{n^t_{i,x_j}, D^t_{i,x_j,n^t_{i,x_j}}, \mathrm{Loc}^t_{i,x_j}\}, E_{\mathrm{Key}_i}\{n^t_{i,x_j} + 1, D^t_{i,x_j,n^t_{i,x_j}+1}, \mathrm{Loc}^t_{i,x_j}\}$;
(37) **end if**
(38) **end if**
(39) **end for**
(40) Set $\mathrm{RST}^t_{S_i}$ to $\{i, t, E_{\mathrm{Key}_i}\{\mathrm{Loc}^t_{i,1}, \mathrm{Loc}^t_{i,2}, \ldots, \mathrm{Loc}^t_{i,\lambda^t_i-1}, \mathrm{Loc}^t_{i,\lambda^t_i}, \mathrm{DPP}^t_{i,x_1}, \mathrm{DPP}^t_{i,x_2}, \ldots, \mathrm{DPP}^t_{i,x_{n[i]-1}}, \mathrm{DPP}^t_{i,x_{n[i]}}\}$;
(41) **end if**
(42) **end for**
(43) Return set $\{I_{Q^t}, \mathrm{RST}^t_{S_1}, \mathrm{RST}^t_{S_2}, \ldots, \mathrm{RST}^t_{S_{N-1}}, \mathrm{RST}^t_{S_N}\}$.

Algorithm 2: Secure spatial-temporal Top-k query processing on the target fog node.

aim to find out the number of locations that fall in $\mathrm{QR}_{I_{\mathrm{MWSN}}}$ of each sensor node in MWSN I_{MWSN} and the corresponding Data – proofPackets generated at those locations; from lines 12 to 42, there is a big "for" loop, which is used to process every report generated in MWSN I_{MWSN} in T^t. Line 14 shows the processing result of $RT^t_{S_i}$ considering the case that no target location of S_i falls in $\mathrm{QR}_{I_{\mathrm{MWSN}}}$ in T^t; lines 16–39 describe the procedure of processing $\mathrm{RT}^t_{S_i}$ considering the case that there is at least one location of S_i that falls in $\mathrm{QR}_{I_{\mathrm{MWSN}}}$ in T^t. In the abovementioned latter case, all the Data – proofPackets

that correspond to the target locations located in $QR_{I_{MWSN}}$ are processed based on the exact values of μ^t_{i,x_j} and/or n^t_{i,x_j}, where μ^t_{i,x_j} and n^t_{i,x_j} denote the total data number and the qualified data number, respectively, corresponding to the location Loc^t_{i,x_j}, which is supposed to be in the queried region $QR_{I_{MWSN}}$. During the procedure of processing the Data − proofPackets, the OPE-encrypted items are all removed from the original Data − proofPackets since the only use of them is to make fog nodes find out the qualified Top-k data items encrypted with the pairwise keys. Moreover, all the unqualified data items except for the one which follows the last qualified Top-k data item in each Data − proofPackets are also removed from each original Data − proofPackets, and the reserved one will be used for completeness verification of the spatial-temporal Top-k query results.

4.5. Completeness Verification of the Query Results. The procedure for an end user to verify the completeness of the Top-k query result R^t is presented in Algorithm 3, the output of which is the value of the Boolean variable completeness. If completeness is false, R^t is considered as incomplete; otherwise, R^t is complete and the final R_{tpk} in Algorithm 3 is composed of all the qualified Top-k data items corresponding to the fine-grained spatial-temporal Top-k query Q^t.

The main idea of Algorithm 3 to verify the completeness of R^t is to find out the minimal data score of the qualified Top-k data items and the maximal score of the unqualified ones generated in the queried region from R^t, and compare them with each other. Normally, the former should be bigger than the latter if the query aims to find out the biggest top k data items. If this condition does not hold in R^t, R^t is considered incomplete. However, it is not correct yet to declare that R^t is complete even if such a condition holds in R^t. Before doing such a comparison, it is necessary to check whether each sensor report was processed properly by the compromised fog node (lines 2–53 in Algorithm 3) based on the proof information included in R^t. To achieve this, each Data − proofPacket in R^t should be checked. When checking the Data − proofPackets, three cases need to be considered, namely, $\gamma^t_{i,x_j} = 0$ (lines 16–25), $\gamma^t_{i,x_j} = 1$ (lines 26–32), and $\gamma^t_{i,x_j} > 1$ (lines 33–51). If $\gamma^t_{i,x_j} = 0$, either S_i did not generate any data items at Loc^t_{i,x_j} in T^t or no data item generated by S_i at Loc^t_{i,x_j} in T^t is the qualified Top-k data item. Thus, in such a case, either $E_{\text{Key}_i}\left\{d^t_{i,x_j,1}, \text{Loc}^t_{i,x_j}\right\}$ or $E_{\text{Key}_i}\left\{0, \text{Loc}^t_{i,x_j}\right\}$ should be originally included in DPP^t_{i,x_j} in R^t. If $\gamma^t_{i,x_j} = 1$, the data item included in DPP^t_{i,x_j} should be a qualified Top-k data item according to lines 24–26 in Algorithm 2. If $\gamma^t_{i,x_j} > 1$, according to lines 27–38 in Algorithm 2, the fog node must have made some illegal query-processing operations if any of the following cases happens (lines 33–35 in Algorithm 3): (a) n^t_{i,x_j} is not included in DPP^t_{i,x_j} in R^t; (b) no sensed data item in DPP^t_{i,x_j} is encrypted with a sequence number; (c) the sequence numbers encrypted in DPP^t_{i,x_j} are not sorted in ascending order from 1; (d) any sensed data item encrypted in DPP^t_{i,x_j} is not originally encrypted with Loc^t_{i,x_j}; and (e) $E_{\text{Key}_i}\left\{\mu^t_{i,x_j}, \text{Loc}^t_{i,x_j}\right\}$ is not originally included in DPP^t_{i,x_j}. Moreover, in the case that $\gamma^t_{i,x_j} > 1$, γ^t_{i,x_j} should be equal to either n^t_{i,x_j} or $n^t_{i,x_j} + 1$ according to lines 27–38 in Algorithm 2 where n^t_{i,x_j} is included in R^t. Thus, in lines 36–50 in Algorithm 3, the abovementioned two cases are considered, respectively, to detect the integrity of R^t.

5. Security Analysis

5.1. Analysis of STQ-SCS on Privacy Preservation

Theorem 1. *Our scheme STQ-SCS is able to preserve the privacy of both the sensed data items and its scores for fine-grained spatial-temporal Top-k query in FMSCSs under the security model presented in this paper.*

Proof. According to Algorithm 1, before being uploaded to fog nodes, all sensed data items are encrypted with the pairwise keys and all the data scores are encrypted with the master keys [35] by the sensor nodes in FMSCSs. Meanwhile, all the encryption keys should only be obtained from TA after authentication according to the key-distribution method used in STQ-SCS, and the fog nodes and the cloud servers are not able to obtain the keys and thus cannot disclose the values of the sensed data items and their scores. Since the cloud servers and the fog nodes are assumed to be curious and/or malicious while other parties in FMSCSs are assumed to be trustworthy in our security model, the privacy of the sensed data items and their scores can be preserved for fine-grained spatial-temporal Top-k query in FMSCSs using our scheme STQ-SCS. □

5.2. Analysis of STQ-SCS on Completeness Verification

Theorem 2. *Suppose a queried node S_i ($\forall i \in [1, N]$) generated $\mu^t_{i,j}$ ($\mu^t_{i,j} > 0$) data items at a queried location $\text{Loc}^t_{i,j}$ ($\forall j \in [1, \lambda^t_i]$) in epoch T^t, where there are $n^t_{i,j}$ ($0 < n^t_{i,j} \le k$) qualified Top-k data items. If at least one of those qualified Top-k data items is dropped from $\text{DPP}^t_{i,j}$ ($\forall i \in [1, N], \forall j \in [1, \lambda^t_i]$) in the query result R^t of $Q^t = \left\{I_{Q^t}, T^t, k, I_{MWSN}, QR_{I_{|MWSN|}}\right\}$ by the fog node or the cloud server which generates and/or transmits R^t, the incomplete R^t must be detected by end users with a 100% successful rate based on our scheme STQ-SCS.*

Proof. Since the fog node or the cloud server does not know Key_i, if it inserts the sensed data items that are encrypted with some other keys rather than Key_i into $\text{DPP}^t_{i,j}$ ($\forall i \in [1, N]$, $\forall j \in [1, \lambda^t_i]$), the incomplete R^t must be detected by the end user according to lines 6–9 in Algorithm 3. Moreover, according to lines 33–35 in Algorithm 3, R^t must be also considered as incomplete if the fog node or the cloud server puts any encrypted data item, which was generated by S_i in T^t at some other location rather than $\text{Loc}^t_{i,j}$, into $\text{DPP}^t_{i,j}$. Thus, in the following of this proof, we need only to consider the situation that all the encrypted sensed data items left in $\text{DPP}^t_{i,j}$ after being processed by the fog node are the real ones which were generated by S_i ($\forall i \in [1, N]$) at $\text{Loc}^t_{i,j}$ in T^t (but some or all of them may not be the qualified ones). Then, if at least one qualified sensed data items generated by S_i at $\text{Loc}^t_{i,j}$ in T^t is discarded by the fog node or the cloud server, one of

Ensure: $R_t = \left\{I_{Q^t}, \mathrm{RST}^t_{S_1}, \mathrm{RST}^t_{S_2}, \ldots, \mathrm{RST}^t_{S_{N-1}}, \mathrm{RST}^t_{S_N}\right\}$; $Q^t = \left\{I_{Q^t}, T^t, k, I_{\mathrm{MWSN}}, \mathrm{QR}_{I_{\mathrm{MWSN}}}\right\}$; $\{\mathrm{Key}^t_1, \mathrm{Key}^t_2, \ldots, \mathrm{Key}^t_{N-1}, \mathrm{Key}^t_N\}$.
Require: Completeness.

(1) $R_{tpk} = \varnothing$; $V_{\mathrm{nonTop}} = \varnothing$; Completeness = true;
(2) **for** $i = 1$ to N **do**
(3) **if** ($\mathrm{RST}^t_{S_i} \notin R_t$) $\|$ ($\mathrm{RST}^t_{S_i}$ contains no pairwise-key-encrypted target locations) **then**
(4) Set Completeness = false; return Completeness;
(5) **end if**
(6) Decrypt all the ciphertext in $\mathrm{RST}^t_{S_i}$ with Key_i;
(7) **if** The end user cannot decrypt the ciphertext normally **then**
(8) Completeness = false; return Completeness;
(9) **end if**
(10) Calculate the value of Ω_i which is the total number of the queried locations in $\mathrm{RST}^t_{S_i}$;
(11) **for** $j = 1$ to Ω_i **do**
(12) **if** DPP^t_{i,x_j} is not originally in $\mathrm{RST}^t_{S_i}$ (DPP^t_{i,x_j} is a Data-proof Packet corresponding to Loc^t_{i,x_j} which is in $\mathrm{QR}_{I_{\mathrm{MWSN}}}$ **then**
(13) Completeness = false; return Completeness;
(14) **end if**
(15) Calculate the value of γ^t_{i,x_j} which is the total number of the sensed data items in DPP^t_{i,x_j};
(16) **if** $\gamma^t_{i,x_j} = 0$ **then**
(17) **if** $E_{\mathrm{Key}_i}\left\{d^t_{i,x_j,1}, \mathrm{Loc}^t_{i,x_j}\right\}$ is originally in DPP^t_{i,x_j} in R^t **then**
(18) $V_{\mathrm{nonTop}} = V_{\mathrm{nonTop}} \cup \left\{d^t_{i,x_j,1}\right\}$;
(19) Continue;
(20) **else if** $E_{\mathrm{Key}_i}\left\{0, \mathrm{Loc}^t_{i,x_j}\right\}$ is originally in DPP^t_{i,x_j} in R^t **then**
(21) Continue;
(22) **else**
(23) Completeness = false; return Completeness;
(24) **end if**
(25) **end if**
(26) **if** $\gamma^t_{i,x_j} = 1$ **then**
(27) **if** $\mathrm{DPP}^t_{i,x_j} \neq \left\{E_{\mathrm{Key}_i}\left\{1, \mathrm{Loc}^t_{i,x_j}\right\}, E_{\mathrm{Key}_i}\left\{D^t_{i,x_j,1}, \mathrm{Loc}^t_{i,x_j}\right\}\right\}$ **then**
(28) Completeness = false; return Completeness;
(29) **end if**
(30) $R_{tpk} = R_{tpk} \cup \left\{D^t_{i,x_j,1}\right\}$;
(31) Continue;
(32) **end if**
(33) **if** (n^t_{i,x_j} is not included in DPP^t_{i,x_j} in R^t) $\|$ (no sensed data item in DPP^t_{i,x_j} is encrypted with a sequence number) $\|$ (the sequence numbers encrypted in DPP^t_{i,x_j} are not sorted in ascending order from 1) $\|$ (any sensed data item encrypted in DPP^t_{i,x_j} is not originally encrypted with Loc^t_{i,x_j}) $\|$ ($E_{\mathrm{Key}_i}\left\{\mu^t_{i,x_j}, \mathrm{Loc}^t_{i,x_j}\right\}$ is not originally included in DPP^t_{i,x_j}) **then**
(34) Completeness = false; return Completeness;
(35) **end if**
(36) **if** $n^t_{i,x_j} = \gamma^t_{i,x_j}$ **then**
(37) **if** $\gamma^t_{i,x_j} \neq \mu^t_{i,x_j}$ **then**
(38) Completeness = false; return Completeness;
(39) **else**
(40) $R_{tpk} = R_{tpk} \cup \left\{D^t_{i,x_j,1}, D^t_{i,x_j,2}, \ldots, D^t_{i,x_j,\gamma^t_{i,x_j}}\right\}$;
(41) **end if**
(42) **else if** $n^t_{i,x_j} = \gamma^t_{i,x_j} - 1$ **then**
(43) **if** ($E_{\mathrm{Key}_i}\left\{D^t_{i,x_j,\mu^t_{i,x_j}}, \mathrm{Loc}^t_{i,x_j}\right\}$ is included in DPP^t_{i,x_j}) && ($\gamma^t_{i,x_j} \neq \mu^t_{i,x_j}$) **then**
(44) Completeness = false; return Completeness;
(45) **end if**
(46) $R_{tpk} = R_{tpk} \cup \left\{D^t_{i,x_j,1}, D^t_{i,x_j,2}, \ldots, D^t_{i,x_j,n^t_{i,x_j}}\right\}$;
(47) $V_{\mathrm{nonTop}} = V_{\mathrm{nonTop}} \cup \left\{f\left(D^t_{i,x_j,\gamma^t_{i,x_j}}\right)\right\}$;
(48) **else**
(49) Completeness = false; return Completeness;

Algorithm 3: Continued.

```
(50)     end if
(51)   end for
(52)   end for
(53)   if ((V_nonTop = ∅) ‖ (SIZE(R_tpk) ≠ k)) then
(54)     Completeness = false; return Completeness;
(55)   end if
(56)     if f(MIN(R_tpk)) < MAX(V_nonTop) then
(57)       Completeness = false; return Completeness;
(58)   end if
(59)   Return Completeness.
```

Algorithm 3: Completeness verification of the query result R^t.

the following two cases must appear: (1) the fog node or the cloud server has dropped all the sensed data items from $\mathrm{DPP}^t_{i,j}$ when producing or transmitting R^t and (2) the fog node or the cloud server has just discarded only a part of the sensed data items from $\mathrm{DPP}^t_{i,j}$, and the discarded data items contain some qualified one/ones.

First of all, consider the case that the fog node or the cloud server has deleted all the sensed data items from $\mathrm{DPP}^t_{i,j}$. In this case, the fog node or the cloud server should leave $E_{\mathrm{Key}_i}\{d^t_{i,j,1}, \mathrm{Loc}^t_{i,j}\}$ in $\mathrm{DPP}^t_{i,j}$ in $\mathrm{RST}^t_{S_i}$ of R^t to avoid being detected according to lines 16–25 in Algorithm 3 because it cannot generate the legal encryption item $E_{\mathrm{Key}_i}\{0, \mathrm{Loc}^t_{i,j}\}$. Then, $d^t_{i,j,1}$ should be put into V_{nonTop} according to lines 17–18 in Algorithm 3, and some real but unqualified sensed data items generated in $\mathrm{QR}_{I_{\mathrm{MWSN}}}$ and T^t must be put into R_{tpk} to make the number of the elements in R_{tpk} equal to k according to lines 53–55 in Algorithm 3. If the discarded sensed data items contain some qualified one/ones, $d^t_{i,j,1}$ must be the score of a qualified Top-k data item. Then, $f(\mathrm{MIN}(R_{tpk}))$ must be smaller than $\mathrm{MAX}(V_{\mathrm{nonTop}})$ because the score of any qualified Top-k data item must be bigger than that of any real but unqualified one generated in $\mathrm{QR}_{I_{\mathrm{MWSN}}}$ and T^t assuming all data scores are distinct. Thus, according to lines 56–58 in Algorithm 3, the incomplete R^t must be detected by the end user.

Then, consider the case that the fog node or the cloud server has just deleted a part of the sensed data items from $\mathrm{DPP}^t_{i,j}$, and the deleted data items contain some qualified one/ones. In this case, two situations should be discussed. One is that all the sensed data items encrypted with sequence order numbers are deleted from, while the other is that at least one sensed data item encrypted with a sequence number is left in $\mathrm{DPP}^t_{i,j}$ after being processed. In the first situation, $E_{\mathrm{Key}_i}\{D^t_{i,j,\mu^t_{i,j}}, \mathrm{Loc}^t_{i,j}\}$ must be left in $\mathrm{DPP}^t_{i,j}$ after being processed, and there must be $\mathrm{DPP}^t_{i,x_j} \neq \{E_{\mathrm{Key}_i}\{1, \mathrm{Loc}^t_{i,j}\}, E_{\mathrm{Key}_i}\{D^t_{i,j,1}, \mathrm{Loc}^t_{i,j}\}\}$ since $\mu^t_{i,j} \neq 1$ in this situation and $E_{\mathrm{Key}_i}\{1, \mathrm{Loc}^t_{i,j}\}$ must not be included in $\mathrm{DPP}^t_{i,j}$. According to lines 26–29 in Algorithm 3, the incomplete R^t must be detected by the end user. Then, consider the second situation. To make the sequence numbers encrypted with the sensed data items in $\mathrm{DPP}^t_{i,j}$ in $\mathrm{RST}^t_{S_i}$ of R^t ascends from 1 orderly (Lines 33–35 in Algorithm 3), the fog node or the cloud server must delete all the sensed data items in one of the sets $\Phi_1, \Phi_2, \Phi_3, \Phi_4$, and Φ_5 from $\mathrm{DPP}^t_{i,j}$. The five sets are shown in equation (2), where $1 < w < \mu^t_{i,j} - 1$.

$$\begin{cases} \Phi_1 = \left\{D^t_{i,j,w}, D^t_{i,j,w+1}, \ldots, D^t_{i,j,\mu^t_{i,j}-1}\right\}, \\ \Phi_2 = \left\{D^t_{i,j,\mu^t_{i,j}-1}\right\}, \\ \Phi_3 = \left\{D^t_{i,j,w}, D^t_{i,j,w+1}, \ldots, D^t_{i,j,\mu^t_{i,j}}\right\}, \\ \Phi_4 = \left\{D^t_{i,j,\mu^t_{i,j}-1}, D^t_{i,j,\mu^t_{i,j}}\right\}, \\ \Phi_5 = \left\{D^t_{i,j,\mu^t_{i,j}}\right\}. \end{cases} \tag{2}$$

If the fog node or the cloud server discards the sensed data items/item in set Φ_1 or Φ_2 from $\mathrm{DPP}^t_{i,j}$ when processing $\mathrm{DPP}^t_{i,j}$, $E_{\mathrm{Key}_i}\{D^t_{i,j,\mu^t_{i,j}}, \mathrm{Loc}^t_{i,j}\}$ and $E_{\mathrm{Key}_i}\{1, D^t_{i,j,1}, \mathrm{Loc}^t_{i,j}\}$ must be left in $\mathrm{DPP}^t_{i,j}$ after being processed, which means that $\gamma^t_{i,j}$ is bigger than 1. According to lines 36–50 in Algorithm 3, the fog node has to either set $n^t_{i,j}$ to $\gamma^t_{i,j}$ or $\gamma^t_{i,j} - 1$ in $\mathrm{DPP}^t_{i,j}$ in $\mathrm{RST}^t_{S_i}$ of R^t to prevent the incomplete R^t from being detected. Even though, the incomplete R^t must also be detected by the end user according to lines 36–38 and 42–45 in Algorithm 3 because $\gamma^t_{i,j}$ must not be equal to $\mu^t_{i,j}$ in this case and $E_{\mathrm{Key}_i}\{D^t_{i,x_j,\mu^t_{i,j}}, \mathrm{Loc}^t_{i,j}\}$ is included in $\mathrm{DPP}^t_{i,j}$ at the same time.

If the fog node or the cloud server deletes the sensed data items/item in set Φ_3, Φ_4, or Φ_5 from $\mathrm{DPP}^t_{i,j}$, the encryption item $E_{\mathrm{Key}_i}\{\gamma^t_{i,j}, D^t_{i,j,\gamma^t_{i,j}}, \mathrm{Loc}^t_{i,j}\}$ should be left in $\mathrm{DPP}^t_{i,j}$. Then, if $\gamma^t_{i,j} = 1$, the incomplete R^t must be detected by the end user according to lines 26–29; if $\gamma^t_{i,j} > 1$, since $\gamma^t_{i,j} \neq \mu^t_{i,j}$ in this case, the fog node or the cloud server has to set $n^t_{i,j}$ to $\gamma^t_{i,j} - 1$ in $\mathrm{DPP}^t_{i,j}$ in $\mathrm{RST}^t_{S_i}$ of R^t to make the incomplete R^t free from being detected according to lines 36–50 in Algorithm 3. Then, $f(D^t_{i,j,\gamma^t_{i,j}})$ will be put into set V_{nonTop} according to lines 42–47 in Algorithm 3. Because some dropped sensed data item/items is/are qualified Top-k data item/items, $D^t_{i,j,\gamma^t_{i,j}}$ must also be a qualified Top-k data item. Since the number of the sensed data items in R_{tpk} should be k, some real but unqualified Top-k data items whose scores are smaller than $f(D^t_{i,j,\gamma^t_{i,j}})$ must be put into set R_{tpk}. Thus, there

must be $f(\mathrm{MIN}(R_{tpk})) < \mathrm{MAX}(V_{\mathrm{nonTop}})$, and the incomplete R^t must be detected by the end user according to lines 56–58 in Algorithm 3.

Thus, if the fog node drops at least one qualified sensed data items from $\mathrm{DPP}^t_{i,j}$, the end user in FMSCSs is able to detect the incomplete R^t with a successful rate of 100% based on STQ-SCS, and Theorem 2 holds. □

Theorem 3. *Under the security model presented in this paper, any end user in FMSCSs can detect the incomplete query results of fine-grained spatial-temporal Top-k queries with a 100% successful rate based on our scheme STQ-SCS.*

Proof. According to the security model, untrusted parties (the fog nodes and the cloud servers) cannot fabricate the pairwise-key-encrypted sensed data items, which cannot be detected by end users, because the untrusted parties cannot obtain the legal pairwise keys. Thus, for any fine-grained spatial-temporal Top-k query Q^t, if its query result R^t is incomplete, at least one qualified sensed data item must be discarded by the fog node or the cloud server when producing and/or transmitting R^t. In other words, there must be at least one queried sensor node $S_i\,(\forall i \in [1, N])$ whose corresponding $\mathrm{Data-proofPacket}\,\mathrm{DPP}^t_{i,j}$ at location $\mathrm{Loc}^t_{i,j}\,(\forall j \in [1, \lambda^t_i])$ satisfies the following condition: at least one qualified sensed data item was deleted from $\mathrm{DPP}^t_{i,j}$ by the fog node or the cloud server when producing and/or transmitting R^t. Then, according to Theorem 2, the incomplete R^t must be detected by the end user in FMSCSs based on our scheme STQ-SCS. Thus, Theorem 3 holds. □

6. Computation Complexity Analysis

This section analyzes the computation complexity of the three schemes presented above.

Firstly, the computation complexity of Algorithm 1 is analyzed as follows. Since most of the statements in Algorithm 1 are the loop body of the "for" loop statements in Algorithm 1, the computation complexity of Algorithm 1 should be that the loop numbers multiply the computation complexity of the loop body. In the loop body, there are only three conditional statements. Thus, the computation complexity of the loop body depends on the pairwise-key encryption methods used in STQ-SCS and the total length of the data that need to be encrypted as well as the computation complexity of OPE. Although different pairwise-key cryptography methods, such as [34, 37], may have different computation complexities, they are considered lightweight generally and fit for the resource-limited sensor nodes [38, 39], let alone the fog nodes which are much more powerful than the sensor nodes. Moreover, OPE also has low computation complexity according to [35]. For each $\mathrm{DPP}^t_{i,j}\,(0 < i \le N, 0 < j \le \mu^t_{i,j})$, the length of the data that need to be encrypted varies according to $\mu^t_{i,j}$, which symbolizes the total number of the sensed data items generated by S_i at $\mathrm{Loc}^t_{i,j}$ in T^t. Let l_D and l_d denote the bit length of a sensed data item and that of a data score, respectively, l_n symbolizes not only the bit length of a sequence number but also that of $\mu^t_{i,j}$, l_{Loc} refers to the bit length of a virtual location, and $l^{\mathrm{OPE}}_{i,j}$ and $l^{\mathrm{PW}}_{i,j}$ denote the bit length of the data that need to be encrypted using OPE and that of those encoded adopting the pairwise-key encryption method, respectively, in $\mathrm{DPP}^t_{i,j}$. Then, the values of $l^{\mathrm{OPE}}_{i,j}$ and $l^{\mathrm{PW}}_{i,j}$ can be worked out using equations (3) and (4), respectively, according to Algorithm 1.

Table 2: Default parameter settings.

Parameters	Default value
N	300
T (length of each epoch)	100 s
T_{mobile} (period for a sensor node to keep moving)	5 s
T_{static} (period for a sensor node to keep static)	5 s
m_{speed} (moving speed of each mobile sensor node)	5 m/s
r_{mobile} (ratio of the mobile sensor nodes to the total ones)	100%
$\mathrm{MWSN}_{\mathrm{size}}$ (size of the deployment field of each MWSN)	$400 \times 400\ \mathrm{m}^2$
R (communication radius of each sensor node)	50 m
r_D (data generation rate of each sensor node)	2 item(s)
q_{period} (period for the end user to launch a query)	5 s
q_{radius} (radius of the queried region which is a circle)	50 m
l_D (length of a sensed data item)	400 bits
l_d (length of a data score)	20 bits
l_n (length of a sequence number)	10 bits
l_{id} (length of an ID number)	10 bits
l_t (length of a time data)	32 bits
l_{Loc} (length of each two-dimensional location)	128 bits
l_{VLoc} (length of each virtual location)	16 bits
e_{send} (cost of sending one bit data)	1 mJ
e_{receive} (cost of receiving one bit data)	1 mJ

$$l^{\mathrm{OPE}}_{i,j} = \begin{cases} 0, & \text{if } \mu^t_{i,j} = 0, \\ l_d, & \text{if } \mu^t_{i,j} = 1, \\ \mu^t_{i,j} \times l_d, & \text{if } \mu^t_{i,j} \ge 2, \end{cases} \tag{3}$$

$$l^{\mathrm{PW}}_{i,j} = \begin{cases} l_n + l_{\mathrm{Loc}}, & \text{if } \mu^t_{i,j} = 0, \\ l_n + l_d + l_D + 3l_{\mathrm{Loc}}, & \text{if } \mu^t_{i,j} = 1, \\ (l_n + l_D + l_{\mathrm{Loc}})\mu^t_{i,j} + l_d + 2l_{\mathrm{Loc}}, & \text{if } \mu^t_{i,j} \ge 2. \end{cases} \tag{4}$$

Secondly, pay attention to Algorithm 2. The computation complexity of lines 1–9 is $O\,(\sum_{i=1}^{N} \lambda^t_i)$; the computation complexity of line 10 depends on the adopted sorting algorithm and the total number of sensed data items generated in T^t and $\mathrm{QR}_{I_{\mathrm{MWSN}}}$; that of line 11 is $O\,(\sum_{i=1}^{N} \lambda^t_i)$; that of lines 12–43 in Algorithm 2 is $O\,(N)$ in the best case (e.g., $n[i]$ is always 0 for each $i \in [1, N]$) and is $O\,(\sum_{i=1}^{N} n[i])$ in the worst case (e.g., $n[i]$ is not equal to 0 for each $i \in [1, N]$).

Finally, it is the turn of Algorithm 3, which mainly consists of one outer "for" loop whose loop body contains an inner "for" loop. In the loop body of the outer loop, the computation complexity of line 6 is the highest among all the statements that are in the loop body of the outer loop and out of the inner loop. If decrypting one encryption item $E_{\mathrm{Key}_i}\{*\}$ is taken as one operation, the operation number of line 6

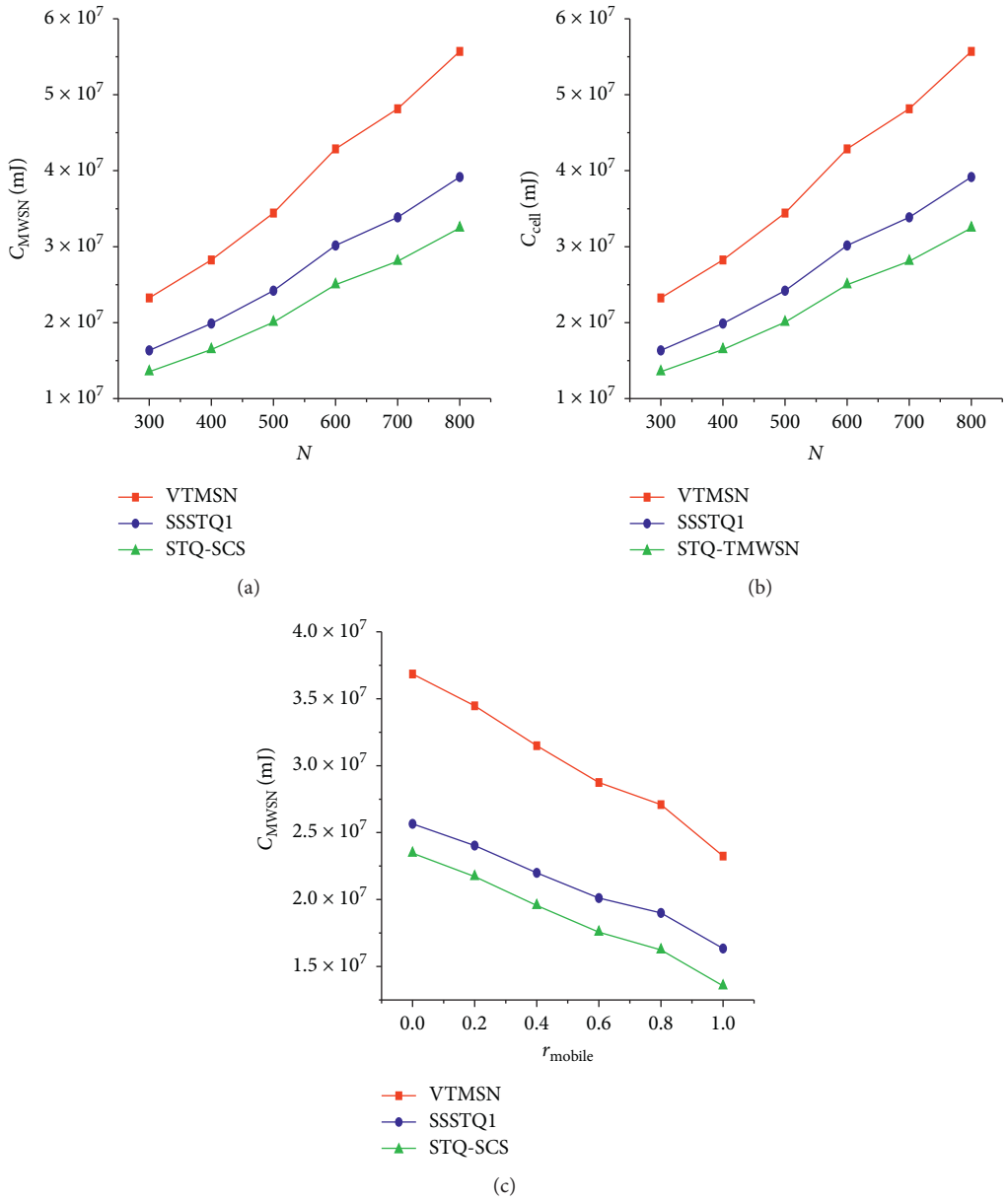

FIGURE 3: C_{MWSN} with different settings of r_D (a), N (b), and r_{mobile} (c).

should be $n[i] + 1$ according to line 40 in Algorithm 2. Then, the computation complexity of Algorithm 3 should be $O(\sum_{i=1}^{N}(n[i] + 1 + \Omega_i))$.

7. Performance Evaluation

In this section, we evaluate the performances of our proposed scheme STQ-SCS through extensive simulations taking OMNET++ as the simulation tool.

7.1. Metrics and Experimental Setup. The performance of STQ-SCS on energy efficiency is evaluated mainly by testing the additional communication cost, which is brought by transmitting the proof data, because other data such as the sensed data items always need to be transmitted no matter what kind of methods are used to ensure the security of the query. Specifically, the metrics used in our simulations are listed as follows.

(i) Additional communication cost in an MWSN (C_{MWSN}): total energy consumed by transmitting all the proof data produced in an MWSN and an epoch to the fog node in the MWSN. Since the sensor nodes are energy-limited, the additional energy cost brought by transmitting the proof data from each MWSN to its corresponding fog node should be given more attention to.

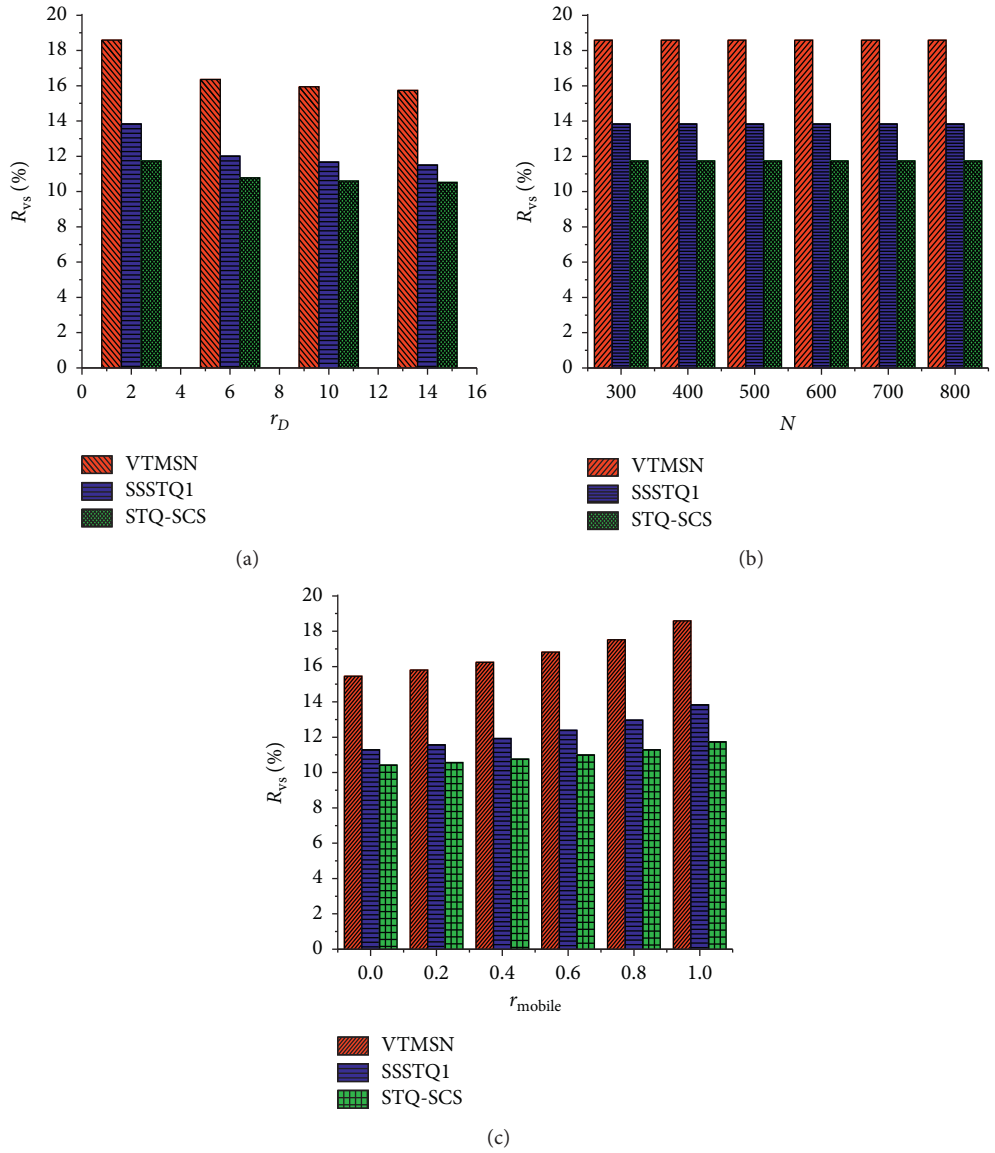

FIGURE 4: R_{vs} with different settings of r_D (a), N (b), and r_{mobile} (c).

(ii) Proof-data ratio (R_{vs}): the ratio of C_{MWSN} to C_{reports}. Here, C_{reports} refers to the total energy consumed by transmitting all the reports generated in an MWSN and an epoch to the fog node connecting to the MWSN, where the data reports include both the sensed data items and the proof data generated by all the sensor nodes in the MWSN and the epoch.

The parameters used in our simulation and their own default values are shown in Table 2, where the default values of some parameters are set by referencing [19]. In fact, static sensor nodes are also allowed to exist in FMSCSs. In the simulation, we adjust the ratio of the mobile sensor nodes to the total ones in the systems by changing the value of r_{mobile}.

7.2. Simulation Results. This section presents the simulation results of C_{MWSN} and R_{vs} with different settings of r_D, N, and r_{mobile}, respectively. We compare our scheme with VTMSN [29] and SSSTQ1 [32] in this section. VTMSN, which was proposed in 2015, is the earliest work on securing spatial-temporal Top-k query in FMSCSs, while SSSTQ1 can be considered as the state-of-the-art scheme proposed for securing spatial-temporal Top-k query in FMSCSs. Figure 3 shows the simulation results of C_{MWSN} under different settings of r_D, N, and r_{mobile}, and Figure 4 illustrates the simulation results of R_{vs} with different settings of r_D, N, and r_{mobile}, respectively. From Figure 3, we can see that the C_{MWSN} lines of STQ-SCS are all lower than those of VTMSN and SSSTQ1. This indicates that our proposed scheme STQ-SCS is more energy-efficient than the other two schemes. The C_{MWSN} lines in Figures 3(a) and 3(b) are on an upward trend because the quantity of sensed data items rises as r_D or N becomes larger and larger, which causes the increase of the proof data, while those in Figure 3(c) are on a downward trend as r_{mobile} rises from 0 to 1 because the sensor nodes are assumed to generate sensed data items only when they are

static or arrive at their target locations and the quantity of the sensed data items and the corresponding proof data must decrease when more sensor nodes are set to be mobile.

Thanks to the technology of virtual-location construction proposed in this paper, fewer bits of location information are included in the proof data in STQ-SCS than the other two schemes, which decrease the ratio of the proof data to the whole data including both sensed data items and their proof. From Figure 4, we can see that the values of R_{vs} of STQ-SCS are all under 12% which is within the acceptable range in real applications and also lower than those of the other two schemes.

8. Conclusions

This paper presents a privacy-preservation and integrity-verification scheme named STQ-SCS for fine-grained spatial-temporal Top-k query in FMSCSs. Thorough security analysis shows that STQ-SCS can make the end users in FMSCSs obtain the query results of fine-grained spatial-temporal Top-k queries without disclosing the privacy of both the sensed data items and their scores, considering that the fog nodes and the cloud servers are not trustworthy. Meanwhile, the security analysis also shows that, under the security model described in this paper, the end users in FMSCSs can detect the incomplete Top-k query results with a 100% successful rate based on our scheme STQ-SCS. Simulation results demonstrate that STQ-SCS is much more efficient than the related state-of-the-art schemes, and can be well used in FMSCSs in real applications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Jie Min conceptualized the study and wrote the original draft of the manuscript and was responsible for methodology; Junbin Liang investigated the study; Xingpo Ma performed simulation; Xingpo Ma and Hongling Chen reviewed and edited the manuscript; Xingpo Ma was involved in project administration and supervision; Hongling Chen performed formal analysis.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant no. 61972090), the Natural Science Foundation of Hunan Province (Grant no. 2019JJ40406), the Key Specialized Research and Development Project in Henan Province (Grant no. 202102210161), and Planning Subject for the 13th Five Year Plan of National Education Sciences (Grant no. 2019GXJK272).

References

[1] X. Li, Z. Ma, J. Zheng, Y. Liu, L. Zhu, and N. Zhou, "An effective edge-assisted data collection approach for critical events in the sdwsn-based agricultural Internet of Things," *Electronics*, vol. 9, no. 6, p. 907, 2020.

[2] A. Liu, X. Liu, and J. Long, "A trust-based adaptive probability marking and storage traceback scheme for wsns," *Sensors*, vol. 16, no. 4, p. 451, 2016.

[3] Y. Sun, D. Rehfeldt, M. Brazil, D. Thomas, and S. Halgamuge, "A physarum-inspired algorithm for minimum-cost relay node placement in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 681–694, 2020.

[4] S. Misra, S. Chatterjee, and M. S. Obaidat, "On theoretical modeling of sensor cloud: a paradigm shift from wireless sensor network," *IEEE Systems Journal*, vol. 11, no. 2, pp. 1084–1093, 2017.

[5] T. Aamir, H. Dong, and A. Bouguettaya, "Trust in social-sensor cloud service," in *Proceedings of the 2018 IEEE International Conference on Web Services (ICWS)*, pp. 359–362, Seattle, WA, USA, June 2018.

[6] Q. Zeng, Q. Duan, M. Shi, X. He, and M. M. Hassan, "Design framework and intelligent in-vehicle information system for sensor-cloud platform and applications," *IEEE Access*, vol. 8, pp. 201675–201685, 2020.

[7] A. Roy, S. Misra, and F. Nait-Abdesselam, "Range-price trade-off in sensor-cloud for provisioning sensors-as-a-service," *IEEE Transactions on Cloud Computing*, p. 1, 2020.

[8] X. Wei and L. Wu, "A new proposed sensor cloud architecture based on fog computing for internet of things," in *Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 615–620, Atlanta, GA, USA, July 2019.

[9] M. Bazm, M. Lacoste, M. Südholt, and J. Menaud, "Secure distributed computing on untrusted fog infrastructures using trusted linux containers," in *Proceedings of the 2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 239–242, Nicosia, Cyprus, December 2018.

[10] X. Yan, W. W. Y. Ng, B. Zeng et al., "Verifiable, reliable, and privacy-preserving data aggregation in fog-assisted mobile crowdsensing," *IEEE Internet of Things Journal*, p. 1, 2021.

[11] J. Ye and J. Wang, "Secure outsourcing of modular exponentiation with single untrusted server," in *Proceedings of the 2015 18th International Conference on Network-Based Information Systems*, pp. 643–645, Taipei, Taiwan, September 2015.

[12] K. N. Sevis and E. Seker, "Survey on data integrity in cloud," in *Proceedings of the 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 167–171, Beijing, China, June 2016.

[13] H. Li, Y. Yang, Y. Dai, S. Yu, and Y. Xiang, "Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data," *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 484–494, 2020.

[14] X. Meng, H. Zhu, and G. Kollios, "Top-k query processing on encrypted databases with strong security guarantees," in *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 353–364, Paris, France, April 2018.

[15] H. Quan, B. Wang, Y. Zhang, and G. Wu, "Efficient and secure top-k queries with top order-preserving encryption," *IEEE Access*, vol. 6, pp. 31525–31540, 2018.

[16] S. Su, Y. Teng, X. Cheng, K. Xiao, G. Li, and J. Chen, "Privacy-preserving top-*k* spatial keyword queries in untrusted cloud environments," *IEEE Transactions on Services Computing*, vol. 11, no. 5, pp. 796–809, 2018.

[17] D. Negi, S. Ray, and R. Lu, "Pystin: enabling secure lbs in smart cities with privacy-preserving top-*k* spatial–textual query," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7788–7799, 2019.

[18] X. Ding, P. Liu, and H. Jin, "Privacy-preserving multi-keyword top-*k* similarity search over encrypted data," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 344–357, 2019.

[19] R. Zhang, J. Shi, Y. Liu, and Y. Zhang, "Verifiable fine-grained top-*k* queries in tiered sensor networks," in *Proceedings of the 2010 IEEE INFOCOM*, pp. 1–9, 2010.

[20] X. Liao, J. Li, and Y. Lei, "Secure and efficient top-k query processing in two-tiered sensor network," *Journal of Computer Research and Development*, vol. 50, no. 3, pp. 490–497, 2013.

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

[22] D. Hua, Y. Geng, X. Fu, and Z. Qiang, "EVTQ: an efficient verifiable top-*k* query processing in two-tiered wireless sensor networks," in *Proceedings of the 2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 206–211, Dalian, China, December 2013.

[23] J. Liang, C. Jiang, X. Ma, G. Wang, and X. Kui, "Secure data aggregation for top-*k* queries in tiered wireless sensor networks," *Adhoc & Sensor Wireless Networks*, vol. 32, no. 1/2, pp. 51–78, 2016.

[24] R. Li, A. X. Liu, S. Xiao, H. Xu, B. Bruhadeshwar, and A. L. Wang, "Privacy and integrity preserving top-*k* query processing for two-tiered sensor networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2334–2346, 2017.

[25] R. Li, Y. Lin, Y. Yi, S. Xiong, and S. Ye, "Security top-*k* query protocol in two layer sensor networks," *Journal of Computer Research and Development*, vol. 49, no. 9, pp. 1947–1958, 2012.

[26] C. M. Yu, Y. T. Tsou, C. S. Lu, and S. Y. Kuo, "Practical and secure multidimensional query framework in tiered sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 241–255, 2011.

[27] W. Chen, L. Yu, and D. Gao, "A privacy preserving histogram aggregation algorithm with integrity verification support," *Chinese Journal of Electronics*, vol. 42, no. 11, pp. 2268–2272, 2014.

[28] X. Kui, J. Feng, X. Zhou, H. Du, and X. Ma, "Securing top-*k* query processing in two-tiered sensor networks," *Connection Science*, vol. 33, no. 1, pp. 1–19, 2020.

[29] F. Liu, X. Ma, J. Liang et al., "Verifiable top-*k* query processing in tiered mobile sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, Article ID 437678, 2015.

[30] H. Wu and L. Wang, "Efficient and secure top-*k* query processing on hybrid sensed data," *Mobile Information Systems*, vol. 201610 pages, 2016.

[31] X. Ma, X. Liu, J. Liang et al., "A comparative study on two typical schemes for securing spatial-temporal top-*k* queries in two-tiered mobile wireless sensor networks," *Sensors*, vol. 18, no. 3, p. 871, 2018.

[32] X. P. Ma, J. B. Liang, J. X. Wang et al., "Secure fine-grained spatio-temporal top-k queries in tmwsns," *Future Generation Computer Systems*, vol. 86, pp. 174–184, 2018.

[33] J. Li, Z. Guan, X. Du, Z. Zhang, and Z. Zhou, "A low-latency secure data outsourcing scheme for cloud-wsn," in *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, San Francisco, CA, USA, March 2017.

[34] B. Nagaraju and P. Ramkumar, "A new method for symmetric key cryptography," *International Journal of Computer Applications*, vol. 142, no. 8, pp. 36–39, 2016.

[35] E. Khoury, M. Medlej, C. A. Jaoude, and C. Guyeux, "Novel order preserving encryption scheme for wireless sensor networks," in *Proceedings of the 2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, pp. 1–6, Jounieh, Lebanon, April 2018.

[36] S. Jangirala, A. K. Das, M. Wazid, and A. V. Vasilakos, "Designing secure user authentication protocol for big data collection in iot-based intelligent transportation system," *IEEE Internet of Things Journal*, vol. 8, p. 1, 2020.

[37] S. Verma, R. Choubey, R. Soni, and P. Ogi, "An efficient developed new symmetric key cryptography algorithm for information security," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 7, pp. 18–21, 2012.

[38] S. Roy, J. Karjee, U. Rawat, N. Dayama Pratik, and N. Dey, "Symmetric key encryption technique: a cellular automata based approach in wireless sensor networks," *Procedia Computer Science*, vol. 78, pp. 408–414, 2016.

[39] M. Bala Krishna and M. N. Doja, "Deterministic *k*-means secure coverage clustering with periodic authentication for wireless sensor networks," *International Journal of Communication Systems*, vol. 30, no. 4, pp. 1–16, 2017.