Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 6301404, 13 pages
http://dx.doi.org/10.1155/2016/6301404

Research Article

Random Secure Comparator Selection Based Privacy-Preserving MAX/MIN Query Processing in Two-Tiered Sensor Networks

Hua Dai,[1,2] Tianyi Wei,[3] Yue Huang,[1] Jia Xu,[1,2] and Geng Yang[1,2]

[1]*College of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[2]*Jiangsu High Technology Research Key Lab for Wireless Sensor Networks, Nanjing 210003, China*
[3]*Bell Honors School, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*

Correspondence should be addressed to Hua Dai; daihua@njupt.edu.cn

Received 7 May 2015; Revised 4 August 2015; Accepted 24 August 2015

Academic Editor: Gwanggil Jeon

Privacy-preserving data queries for wireless sensor networks (WSNs) have drawn much attention recently. This paper proposes a privacy-preserving MAX/MIN query processing approach based on random secure comparator selection in two-tiered sensor network, which is denoted by RSCS-PMQ. The secret comparison model is built on the basis of the secure comparator which is defined by 0-1 encoding and HMAC. And the minimal set of highest secure comparators generating algorithm MaxRSC is proposed, which is the key to realize RSCS-PMQ. In the data collection procedures, the sensor node randomly selects a generated secure comparator of the maximum data into ciphertext which is submitted to the nearby master node. In the query processing procedures, the master node utilizes the MaxRSC algorithm to determine the corresponding minimal set of candidate ciphertexts containing the query results and returns it to the base station. And the base station obtains the plaintext query result through decryption. The theoretical analysis and experimental result indicate that RSCS-PMQ can preserve the privacy of sensor data and query result from master nodes even if they are compromised, and it has a better performance on the network communication cost than the existing approaches.

1. Introduction

As wireless sensor networks (WSNs) have been widely used in a variety of important areas such as environment monitoring, medical care, national defense, and military, various security problems of data privacy are becoming more and more critical. For example, in the rare animal monitoring, the location of rare animals could be obtained for illegal hunting; in the application of smart home, the information for use of family hydroelectricity could be stolen for burglary. Therefore, privacy-preserving has become a very important issue in WSNs.

Most large-scale WSNs are expected to apply a two-tiered architecture with the resource-limit sensor nodes at the lower layer and resource-abundant master nodes at the upper layer, and this architecture is used to construct our concerned two-tiered wireless sensor networks (TWSNs) in this paper, as shown in Figure 1. The master nodes have abundant resources of energy, computation, communication, and so forth, while the sensor nodes only have limited resources. The sensor nodes are only responsible of collecting data and periodically submitting it to a nearby master node for storage, which responds to the query requests from the base station (BS) and then returns the query results. Due to the simplicity of topological structure which contains multiple independent cells and the resource abundance of master nodes, TWSNs have a lot of advantages, such as stable link quality, simple route structure, and higher network scalability [1, 2].

However, because the master nodes are not only responsible of storing all the data from the sensor nodes, but also processing the query requests from BS, they are much more attractive and vulnerable to attackers in a hostile environment. Once a master node is compromised, serious threats could be brought to the data privacy of TWSNs. The attackers could utilize the compromised master nodes to obtain all the collected data of sensor nodes and the

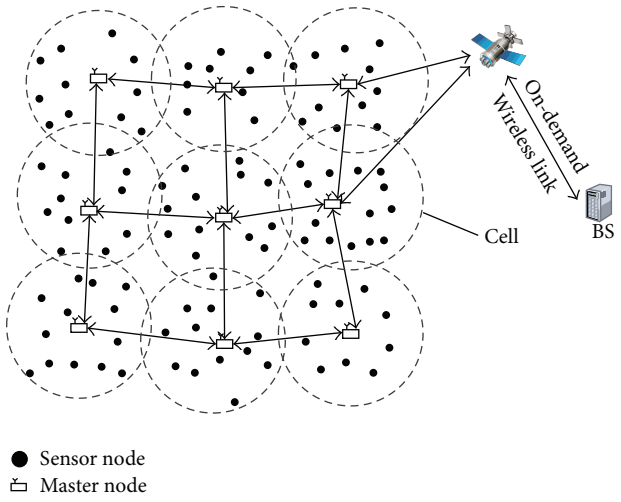

Figure 1: Architecture of TWSNs.

query results. Thus, it is necessary to investigate the privacy-preserving problems in TWSNs and develop efficient and effective solutions.

MAX/MIN query is a useful data query method to obtain the maximum or minimum data in the areas and epochs of interest. It can be utilized in event monitoring. For example, it can be applied to monitor the highest temperature in a warehouse so as to alarm the fire risk. The challenges to achieve privacy-preserving MAX/MIN query processing in TWSNs include the following:

(i) How to make the master nodes realize secure comparisons of data items without knowing their real values and then determine the maximum or minimum value, that is, the query result.

(ii) How to maximally reduce the communication cost of network, especially that of the sensor nodes due to their limited resources.

In this paper, we propose a privacy-preserving MAX/MIN query processing approach based on random secure comparator selection in TWSNs, which is denoted by RSCS-PMQ. The basic idea is as follows: once finishing data collection, the sensor nodes will encrypt the collected data into ciphertext and select the corresponding random secure comparators generated by using the 0-1 encoding [3] and hashed message authentication coding (HMAC) [4]. Then the ciphertext and the corresponding random secure comparators are submitted to the nearby master node. When the master node processes a query request from BS, it will utilize the algorithm MaxRSC to determine the minimal set of highest secure comparators, further determine the corresponding minimal set of candidate ciphertexts containing the query result, and return it to BS. After decrypting the received ciphertext, BS will obtain the query result in plaintext. Since the data storage and query response procedures in the master node do not involve the plaintext of collected data, the adversaries cannot read any hosted data or query result from the master node even if it is compromised. Thus, the proposed RSCS-PMQ can achieve the privacy protection in MAX/MIN query processing. In addition, the evaluation indicates that RSCS-PMQ has better performance than other existing works on the network communication cost.

The main contributions of this paper are as follows:

(i) We propose a comparison model based on random secure comparator selection through the 0-1 encoding and hashed message authentication coding, which supports data comparison without real values in master nodes.

(ii) We design an algorithm to generate the minimal set of highest secure comparators based on the former comparison model, MaxRSC, which is the key algorithm to accomplish RSCS-PMQ.

(iii) We provide the concrete protocols of achieving RSCS-PMQ, which consist of the data collection protocol and the query response protocol, and the latter protocol can protect data privacy from master nodes even if they are compromised.

(iv) We analyze the privacy protection and communication cost of RSCS-PMQ and conduct performance evaluation through comprehensive simulation.

The rest of this paper is organized as follows. Section 2 gives an overview of related works. Section 3 describes related models and problem statement. In Section 4, we present the privacy-preserving MAX/MIN query protocols based on random secure comparator selection. Then, Section 5 analyzes the privacy and communication costs of our approach. We evaluate the performance through simulation in Section 6 and conclude this paper in Section 7.

2. Related Works

Data query is an important operation for events monitoring or data analysis in TWSN. The security issues are the hot spots in data query researches, such as privacy protection, integrity, or completeness verification. Recently, the secure range query [5–11] and top-k query [12–17] have been broadly studied. However, there are limited researches on MAX/MIN query, and only [18, 19] propose solutions of privacy-preserving MAX/MIN query in TWSN.

Regarding the range query, a secure range query processing in TWSN is firstly proposed in [5], which employs bucket partition and symmetric encryption to achieve the privacy protection of collected data, and uses MAC to accomplish the completeness verification of query results. Based on [5], the spatiotemporal cross-check is introduced in [6, 7] to improve the efficiency of energy consumption. Furthermore, the spatiotemporal cross-check procedure of [6, 7] is optimized in [8], which balances the security and energy consumption and applies this method in multidimensional range query. A secure and energy-efficient range query processing protocol SafeQ is proposed in [9, 10], which is based on Prefix Membership Verification (PMV) [20] and neighborhood chain mechanism. In addition, the Bloom-Filter [21] is introduced to optimize the energy consumption. And a secure range

query protocol based on order-preserving function and link watermarking QuerySec is proposed in [11], which is capable of saving energy during query processing. However, these secure range query methods are not suitable for solving the privacy-preserving MAX/MIN query.

For top-k query, the fine-grained verifiable top-k query methods are proposed in [12, 13], whereby the network owner can verify the completeness and authenticity of query results in TWSNs. The verification code which embeds ordered and adjacent relationships of the collected data by HMAC is utilized in [14] to achieve the verifiable top-k query processing. The symmetric encryption is applied in [15] to reduce the communication cost in verifiable top-k query processing. Works in [12–15] merely support the completeness and authenticity verification of query results, but they cannot achieve privacy protection. To preserve privacy, the privacy-preserving top-k query processing approaches based on Order Preserving Encryption [22] are proposed in [16, 17]. Though the top-k query can be transformed into MAX/MIN query when $k = 1$, it is wasteful in energy consumption. The reason is that each sensor node should submit all the data collected in every epoch since top-k is designed for obtaining the highest k data, where k is variable. In contrast, MAX/MIN query only has to submit the sole maximum or minimum value. Therefore, taking top-k query as MAX/MIN query will result in high unnecessary data communication. In conclusion, the secure top-k query methods are not suitable for solving the privacy-preserving MAX/MIN query.

For MAX/MIN query, the same PMV as in [9, 10], the symmetric encryption and HMAC are used in [18] to achieve the privacy-preserving MAX/MIN query processing. Since more codes are transmitted in data submission, which are generated by using the PMV and HMAC functions, the energy consumption of [18] is high, which will reduce the lifetime of whole network. In contrast, our former work [19] adapts 0-1 encoding verification instead of PMV to achieve an energy-efficient privacy-preserving MAX/MIN query which is denoted by EMQP. In EMQP, the codes generated by using 0-1 encoding and HMAC of same data are significantly less than those generated in [18], which can reduce the energy consumption of sensor nodes. Furthermore, in this paper, we will adopt random selection of codes on the basis of EMQP to save much more energy and accomplish better privacy-preserving MAX/MIN query processing.

There are also many secure aggregation methods, such as [23–26]. However, most of these works adapt the traditional multihop wireless sensor networks, and they are not suitable for TWSNs.

3. Models and Problem Statement

3.1. Network Models. We consider a similar TWSN model as in [5–17], as shown in Figure 1. The network is divided into multiple cells, each containing master node $\mathscr{M}$ and several sensor nodes $\{s_1, s_2, \ldots, s_n\}$, which is named after cell $= (\mathscr{M}, \{s_1, s_2, \ldots, s_n\})$. In particular, the master nodes are powerful devices, which have abundant resources of energy, storage, and computation. Additionally, they are also responsible for receiving and storing data collected by the sensor nodes and processing the query requests from BS, while the sensor nodes are cheap devices with limited resources. Each sensor node merely submits its collected data to the master node in the same cell. The master node can apply its long-distance and high-frequency communication capacity to communicate with the nearby nodes, which should then construct the upper-tier multihop networks. The query results will be returned from the queried master nodes to BS through the above networks. There is an on-demand wireless link (e.g., satellite) between the master nodes and BS to interact with each other. However, such wireless link is usually unstable and is of high consumption and low speed.

We assume that the time is divided into nonoverlapping epochs, and in every epoch t, the sensor node s_i collects V sensor data $\{d_{i,1}, d_{i,2}, \ldots, d_{i,V}\}$. In TWSNs, BS owns the global network topological information, while a master node owns the network topological information of its located cell, and a sensor node only knows the locations of the master node in the same cell and 1-hop neighboring sensor nodes.

3.2. MAX/MIN Query Models. A MAX/MIN query in TWSN is a kind of query operation aimed at obtaining the maximum or minimum data among the data items collected in the specified epochs and area. Therefore, the following MAX/MIN query will be considered, which is denoted by a triple tuple:

$$Q = (\Theta, T, \Gamma), \tag{1}$$

where $\Theta \in \{\text{MAX}, \text{MIN}\}$ refers to the query type, T is the set of queried epoch numbers, and Γ denotes the set of queried sensor nodes IDs which indicate a query region. For example, query $Q = (\text{MAX}, t, \{s_1, s_4, s_6, s_{11}\})$ is aimed at getting the maximum data collected by sensor nodes s_1, s_4, s_6, s_{11} in the epoch t.

For simplicity, we focus on the simple MAX query aimed at one cell $(\mathscr{M}, \{s_1, s_2, \ldots, s_n\})$ and one epoch t; that is $Q = (\text{MAX}, t, \Gamma)$, where, $\Gamma \subseteq \{s_1, s_2, \ldots, s_n\}$. For other complicated queries covering multiple epochs and cells, it can be easily achieved by decomposing them into multiple simple ones. And we will conduct discussion in Section 4.5. Additionally, the MIN query is similar to the MAX query.

3.3. Problem Statement. In TWSNs, $\mathscr{M}$ is too vulnerable and tends to be easily under attacks from adversaries, since they are not only responsible for storing all the data collected by the sensor nodes, but also responsible for processing the query requests from BS. If the collected data is in plaintext and $\mathscr{M}$ is compromised, any data stored in $\mathscr{M}$ and the query results generated by $\mathscr{M}$ will be exposed to attackers, which tends to lead to privacy leakage. Therefore, it is necessary to take efficient and effective measures for privacy protection.

We adopt the same honest-but-curious threat model as in [27], where $\mathscr{M}$ may try to breach privacy to steal sensitive data but faithfully obey protocols while processing the query requests. Additionally, BS and sensor nodes are also assumed to be credible in contrast to $\mathscr{M}$. Based on

the above assumption, for the sake of achieving privacy-preserving MAX/MIN query processing, the following conditions should be satisfied:

(1) For the data collected by any sensor node in the network, only this sensor node and BS can obtain its real value in contrast to $\mathscr{M}$.

(2) For the real value of query results, only BS can obtain it in contrast to $\mathscr{M}$.

Moreover, $\mathscr{M}$ have abundant energy resources, while the sensor nodes are energy-limited, which results in the fact that the lifetime of the whole network totally relies on the energy consumption of sensor nodes. And most energy is consumed by communications according to [19]. Therefore, the communication cost of sensor nodes is a key metric for performance evaluation of query processing method in TWSN. We will conduct concrete evaluations of in-cell communication cost (C_S) and query response communication cost (C_M) in Section 6. The former represents the total energy consumption in bits incurred by data transmissions between the sensor nodes and $\mathscr{M}$ per epoch, while the later refers to the total information in bits transmitted between $\mathscr{M}$ and BS.

4. MAX/MIN Query Processing with Random Secure Comparator Selection

We use the 0-1 encoding verification [3], which can be utilized to compare data items without knowing their values. Let $x = b_1 b_2 \cdots b_{w-1} b_w \in \{0,1\}^w$ be a binary string with w bits. The 0-encoding and 1-encoding of x are denoted by $E_0(x) = \{b_1 b_2 \cdots b_{i-1} 1 \mid b_i = 0 \wedge 1 \le i \le w\}$ and $E_1(x) = \{b_1 b_2 \cdots b_i \mid b_i = 1 \wedge 1 \le i \le w\}$, respectively, where $|E_0(x)| + |E_1(x)| = w$. For two data items x and y, $x > y$ if and only if $E_1(x) \cap E_0(y) \neq \emptyset$; otherwise $x \le y$. Obviously, if codes of x and y are of different types, they can be compared; otherwise they are incomparable.

In order to improve the efficiency of intersection computing, the numeralization functions are usually applied to convert the 0-1 encoding binary strings into numbers. Thus we adopt the same numeralization function $\mathscr{N}(*)$ as in [19], which satisfies the idea that, for any two 0-encoding or 1-encoding binary strings, a and b, $a = b$ if and only if $\mathscr{N}(p) = \mathscr{N}(q)$. Additionally, we utilize HMAC to realize one-wayness and collision resistance of encoding data. We denote HMAC function by $H_g(*)$, where g is the secret key of HMAC, which is only shared in sensor nodes.

4.1. Comparison Model Based on Random Secure Comparator Selection

Definition 1. For data x, after applying 0-1 encoding, numeralization, and HMAC, the two generated code sets are denoted by the secure comparators of x, where $\mathrm{sc}_0(x)$ and $\mathrm{sc}_1(x)$ are type 0 and type 1 secure comparators, respectively; that is, $\mathrm{sc}_0(x) = H_g(\mathscr{N}(E_0(x)))$, $\mathrm{sc}_1(x) = H_g(\mathscr{N}(E_1(x)))$.

According to the data comparison property of 0-1 encoding verification, we do not have to compare any two data items based on their values, but only the corresponding secure comparators. Thus, Lemma 2 is established.

Lemma 2. *For data x and y, if $sc_1(x) \cap sc_0(y) \neq \emptyset$, then $x > y$; otherwise $x \le y$.*

Definition 3. The random secure comparator of data x is denoted by $\omega(x) = \mathrm{rnd}\{\mathrm{sc}_0(x), \mathrm{sc}_1(x)\}$, where $\mathrm{rnd}\{*\}$ is the random selection function of a set. Its value is denoted by $\omega(x).val$; that is, $\omega(x).val = x$, and its type is denoted by $\omega(x).type$, which is shown as follows:

$$\omega(x).type = \begin{cases} 0 & \omega(x) = \mathrm{sc}_0(x) \\ 1 & \omega(x) = \mathrm{sc}_1(x). \end{cases} \tag{2}$$

Given the random secure comparators of two different data items, we propose the *rsc_compare* algorithm, as shown in Algorithm 1, to conduct secure comparisons according to Lemma 2 and Definition 3.

Definition 4. For two secure comparators, a and b, if and only if $rsc\_compare(a,b) > 0$, then a is higher than b, and we denote it by $a \succ b$, which means $a.val \ge b.val$; if and only if $rsc\_compare(a,b) < 0$, then a is smaller than b, and we denote it by $a \prec b$, which means $a.val < b.val$; otherwise, a and b are incomparable.

According to Definition 4, only if two secure comparators are of different types, they are comparable. It is remarkable that the relations $\succ$ and $\prec$ do not have transitivity.

4.2. Algorithm for Generating Minimal Set of Highest Secure Comparators.

In this section, we firstly give the definition of the minimal set of highest secure comparators, which is the theoretical basis to achieve RSCS-PMQ. Then, we provide the generation algorithm of this set and analyze the probability of the amount of its elements.

Definition 5. Assume that $S = \{\omega(x_1), \omega(x_2), \ldots, \omega(x_n)\}$ is the corresponding set of random secure comparators of $\{x_1, x_2, \ldots, x_n\}$. The minimal set of highest secure comparators is denoted by Ψ, where

(1) $\Psi \subseteq S$;

(2) $\forall \omega(x_i), \omega(x_j) \in \Psi \rightarrow \omega(x_i).type = \omega(x_j).type$;

(3) $\forall \omega(x_i) \in \Psi \wedge \omega(x_j) \in S - \Psi \rightarrow \omega(x_i) \succ \omega(x_j) \vee (\exists \omega(x_e) \in S - \Psi \wedge x_j \neq x_e \rightarrow \omega(x_i) \succ \omega(x_e) \succ \omega(x_j))$.

According to Definition 5, the secure comparators in Ψ are of the same type; therefore, they are incomparable. Moreover, for secure comparators $a \in \Psi$ and $b \in S - \Psi$, if they are of different types, the former is obviously larger than the latter; if they are of the same type, there must exist another secure comparator $c \in S - \Psi$ with different type from a and b, which satisfies the idea that a is larger than b, while b is larger than c.

Input: secure comparators a and b
Output: if $a.val \geq b.val$, 1 is returned, if $a.val \leq b.val$,
-1 is returned, and *if* $a.type = b.type$, 0 is resturned,
Procedures:
If $a.type = b.type$, then return 0;
If $a.type = 1$, then
  If $a \cap b \neq \emptyset$, then return 1; //means $a.val > b.val$;
  Else, return -1; //means $a.val \leq b.val$;
Else
  If $a \cap b \neq \emptyset$, then return -1; //means $b.val > a.val$;
  Else, return 1; //means $b.val \leq a.val$;

Algorithm 1: *rsc_compare*.

Lemma 6. *Assume that Ψ is the minimal set of highest secure comparators of the data set $\{x_1, x_2, \ldots, x_n\}$; then we have*

$$\max\{x_1, x_2, \ldots, x_n\} \in \{a.val \mid a \in \Psi\}. \quad (3)$$

Lemma 6 can be easily deduced from Definition 5, which indicates that the maximum of $\{x_1, x_2, \ldots, x_n\}$ must exist in the corresponding data set of Ψ.

Lemma 7. *Given data set $\{x_1, x_2, \ldots, x_n\}$, its corresponding secure comparator set is $S = \{\omega(x_1), \omega(x_2), \ldots, \omega(x_n)\}$; then we have the probability of Ψ containing λ elements as follows:*

$$\Pr(|\Psi| = \lambda) = \left(\frac{1}{2}\right)^{\lambda}, \quad (4)$$

where $\lambda \in N^+$ and $\lambda \leq n$.

Proof. Assume that $x_1 \geq x_2 \geq \cdots \geq x_n$, and Ψ has λ secure comparators; then $\Psi = \{\omega(x_1), \omega(x_2), \ldots, \omega(x_\lambda)\}$. According to Definition 5, all the secure comparators in Ψ are of the same type. And $\omega(x_{\lambda+1})$ must be of different type from the ones in Ψ; otherwise $\omega(x_{\lambda+1})$ also belongs to Ψ, which contradicts to the given assumption. Therefore, the probability of Ψ having λ secure comparators is equivalent to the probability that $\omega(x_1), \omega(x_2), \ldots, \omega(x_\lambda)$ are of the same type while differing from $\omega(x_{\lambda+1})$. Apparently, the probability of all $\omega(x_1), \omega(x_2), \ldots, \omega(x_\lambda)$ being type 0 and $\omega(x_{\lambda+1})$ being type 1 is $(1/2)^{\lambda} \cdot 1/2 = (1/2)^{\lambda+1}$. Similarly, the probability is also $(1/2)^{\lambda+1}$ under the reverse circumstance. Thus, the probability of Ψ having λ secure comparators is $(1/2)^{\lambda+1} + (1/2)^{\lambda+1} = (1/2)^{\lambda}$. □

The generation algorithm of the minimal set of highest secure comparatorsis given as Algorithm 2, denoted by MaxRSC.

As shown in algorithm MaxRSC, T_0 and T_1 are used to store the current sets of the highest and second highest secure comparators. The variable *flag* is to indicate the higher set between T_0 and T_1, where $flag = 0$ indicates T_0 is higher; otherwise $T_1$is higher. When the algorithm ends, the final minimal set of highest secure comparators is T_0 if $flag = 0$; otherwise it is T_1. The algorithm is concise and direct, and its time complexity is only $O(n)$.

4.3. Data Collection Protocol. The data collection protocol is concerned with how a sensor node transmits its collected data items to $\mathcal{M}$. For each sensor node s_i, after collecting V data items $\{d_{i,1}, d_{i,2}, \ldots, d_{i,V}\}$ in epoch t, it performs the following procedure:

(i) Determine the maximum data among the collected data items; that is, $d_i = \max\{d_{i,1}, d_{i,2}, \ldots, d_{i,V}\}$.

(ii) Compute the random secure comparator $\omega(d_i)$, and set its type according to the random selection.

(iii) Encrypt d_i by using the key k_i shared with BS. The output ciphertext is denoted by $(d_i)_{k_i}$.

(iv) Submit the following message to $\mathcal{M}$, where $\mathrm{id}(s_i)$ is the ID of s_i:

$$s_i \longrightarrow \mathcal{M} : \left\langle \mathrm{id}(s_i), t, (d_i)_{k_i}, \omega(d_i) \right\rangle. \quad (5)$$

(v) Once $\mathcal{M}$ receives the above message from s_i, it will store the data of the message.

As shown in the above protocol, since the HMAC function is one-way and collision-resistant and sensor nodes only share the encryption and HMAC keys with BS, it is computationally infeasible to reveal the exact value to $\mathcal{M}$. Therefore, we can see that the data collection protocol can preserve data privacy from $\mathcal{M}$.

4.4. Query Response Protocol. The query response protocol is concerned with how $\mathcal{M}$ cooperates with BS to accomplish the query requests from users. The main idea is that $\mathcal{M}$ uses the MaxRSC algorithm to generate the minimal set of highest secure comparators on the basis of submitted random secure comparators of the queried sensor nodes and further determines the corresponding minimal set of candidate ciphertexts which is denoted by R. Then, $\mathcal{M}$ returns it to BS. And the final query result will be determined after BS performs decryption on R. The concrete steps of query response protocol are as follows:

(i) BS transmits the query request $Q = (\mathrm{MAX}, t, \Gamma)$ to $\mathcal{M}$, where $\Gamma = \{s_1, s_2, \ldots, s_m\}$.

(ii) Once $\mathcal{M}$ receives the query request, it firstly loads the ciphertext $(d_i)_{k_i}$ and the corresponding random

Input: The set of random secure comparators, S
Output: The minimal set of highest secure comparators, Ψ
Procedures:
(1) Initialize $T_0 = \emptyset$, $T_1 = \emptyset$ which are used to store type 0 and 1 secure comparators, respectively;
(2) Fetch the first secure comparator from S into the variable a, and set the variable $flag = a.type$;
(3) If $flag = 0$, add a into T_0, otherwise add it into T_1;
(4) For each $item \in S$,
    If $item.type = 0 \wedge flag = 0$, then
        If $T_1 = \emptyset$, then add $item$ into T_0;
        Else
            If $\forall b \in T_1 (b \prec item)$, then add $item$ into T_0;
            Else, delete every $b \in T_1$ if $b \prec item$;
    If $item.type = 0 \wedge flag = 1$, then
        If $\forall b \in T_1 (b \succ item)$, then add $item$ into T_0;
        Else if $\forall b \in T_1 (b \prec item)$,
            then clear T_0, add $item$ into T_0 and set $flag = 0$;
        Else, delete every $b \in T_1$ if $b \prec item$;
    If $item.type = 1 \wedge flag = 1$, then
        If $T_0 = \emptyset$, then add $item$ into T_1;
        Else
            If $\forall b \in T_0 (b \prec item)$, then add $item$ into T_1;
            Else, delete every $b \in T_0$ if $b \prec item$;
    If $item.type = 1 \wedge flag = 0$, then
        If $\forall b \in T_0 (b \succ item)$, then add $item$ into T_1;
        Else If $\forall b \in T_0 (b \prec item)$,
            then clear T_1, add $item$ into T_1 and set $flag = 1$;
        Else, delete every $b \in T_0$ if $b \prec item$;
    End For
(5) If $flag = 0$, then set $\Psi = T_0$; otherwise, set $\Psi = T_1$;

Algorithm 2: MaxRSC.

secure comparator $\omega(d_i)$ received from each sensor node $s_i \in \Gamma$ in epoch t. Assume that the loaded random secure comparators are $\{\omega(d_1), \omega(d_2), \ldots, \omega(d_m)\}$. With them as the input, then $\mathscr{M}$ generates the minimal set of highest secure comparators Ψ by using the MaxRSC algorithm, and the corresponding minimal set of candidate ciphertexts R is determined, where

$$R = \left\{ (d_i)_{k_i} \mid \omega(d_i) \in \Psi \right\}. \quad (6)$$

(iii) Finally, $\mathscr{M}$ constructs the following message and sends it to BS:

$$\mathscr{M} \longrightarrow \mathrm{BS} : \left\langle \left\{ \mathrm{id}(s_i), (d_i)_{k_i} \mid (d_i)_{k_i} \in R \right\} \right\rangle. \quad (7)$$

(iv) When BS receives the response message from $\mathscr{M}$, it uses the key shared with the sensor nodes to decrypt the ciphertext in $\Re$, and then the final query result will be determined.

Similar to data collection protocol, it is also computationally infeasible for $\mathscr{M}$ to get the real values and the query result in the query response protocol. Therefore, this protocol can also preserve data privacy from $\mathscr{M}$.

Lemma 8. *For the determined minimal set of highest secure comparators Ψ and the corresponding minimal set of candidate ciphertexts R in the query response protocol, we have the following:*

(1) $|\Psi| = |R|$, *where* $|*|$ *means the number of elements in the set* $*$.

(2) *The query result of Q must be embedded in the ciphertext of R.*

Proof. According to the construction of R in (6), we can easily have $|\Psi| = |R|$, and the values of secure comparators in Ψ are all embedded in the ciphertext of R. And Lemma 6 indicates that the maximum data among the data items collected by the queried sensor nodes (i.e., query result) must exist in the corresponding data set of Ψ. Therefore, the query result must be embedded in the ciphertext of R as well. □

Lemma 9. *Assume that there are λ elements in R; then we have its probability as follows:*

$$\Pr(|R| = \lambda) = \left(\frac{1}{2}\right)^{\lambda}. \quad (8)$$

Proof. From Lemma 7, we can obtain the idea that the probability of Ψ containing λ secure comparators is $(1/2)^{\lambda}$. Additionally, Lemma 8 indicates $|\Psi| = |R|$. Thus, the probability of R having λ elements is also $(1/2)^{\lambda}$. □

According to Lemma 9, we can easily deduce Lemma 10.

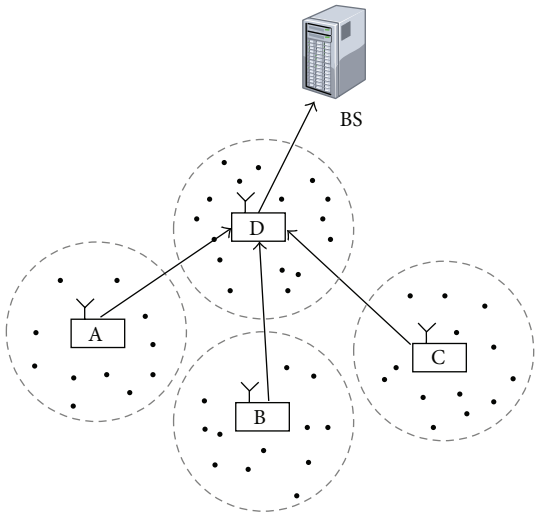

Figure 2: Complicated query example.

Lemma 10. *The mean quantity of elements in R is the mathematical expectation $E(\lambda)$ of the ciphertext quantity in R, where*

$$E(\lambda) = \sum_{\lambda=1}^{n} \lambda \cdot \Pr(\lambda) = \sum_{\lambda=1}^{n} \lambda \cdot \left(\frac{1}{2}\right)^{\lambda} = 2 - (n+2) \cdot \left(\frac{1}{2}\right)^{n} \tag{9}$$

and $E(\lambda) \approx 2$ when n is very large.

4.5. Complicated Query Processing Method. If complicated query Q involving multiple cells and epochs is applied, we can achieve it based on the basic ideas of the data collection and query response protocols. We give the overview of the complicated query processing method through an example.

As shown in Figure 2, there are four master nodes, A, B, C, and D, and BS composing the upper-tier tree-routing networks. Assume that the current query Q involves A, B, C, and D and several epochs. The main idea of processing Q is as follows. Firstly, A, B, C, and D use MaxRSC algorithm to determine their own minimal sets of highest secure comparators and the corresponding minimal set of candidate ciphertexts, which can be denoted by four pairs: (Ψ_A, R_A), (Ψ_B, R_B), (Ψ_C, R_C), and (Ψ_D, R_D), respectively. Then, A, B, and C submit (Ψ_A, R_A), (Ψ_B, R_B), and (Ψ_C, R_C) to D on their own. And D takes Ψ_A, Ψ_B, Ψ_C, and its own Ψ_D as inputs into MaxRSC algorithm to determine the global minimal set of highest secure comparators and the global corresponding minimal set of candidate ciphertexts R. Obviously, the global query result is embedded in R, and then D submits R to BS. Consequently, BS decrypts the ciphertext in R and gets the final query result of the complicated query Q.

5. Protocol Analysis

5.1. Privacy-Preserving Analysis

(1) Collected data privacy preservation: on the premise that BS and sensor nodes are credible in this paper, the privacy of data collected by the sensor nodes can be preserved from $\mathscr{M}$ only if it is ensured that it is impossible for $\mathscr{M}$ has to obtain the real value of any collected data. According to the data collection protocol, the data submitted by sensor nodes which are stored in $\mathscr{M}$ are the ciphertext and HMAC codes instead of the plaintext. Since the HMAC algorithm is one-way and collision-resistant and the encryption and HMAC keys are only shared by the sensor nodes and BS, given a random secure comparator $\omega(d_i)$ and ciphertext $(d_i)_{k_i}$, it is computationally infeasible for $\mathscr{M}$ to obtain the value of the collected data d_i. And, for $\mathscr{M}$, the complexity to peek the privacy is equal to cracking the HMAC and encryption. Thus, our proposed RSCS-PMQ can protect the privacy of the collect data from master nodes.

(2) Query result privacy preservation: as shown in the query response protocol, $\mathscr{M}$ cooperates with BS to achieve the MAX/MIN query processing. During the procedure, $\mathscr{M}$ takes the secure comparators as inputs to determine the minimal set of candidate ciphertexts embedding the plaintext query result through MaxRSC algorithm and then transmits it to BS. Consequently, BS decrypts the received ciphertext and gets the plaintext query result. Obviously, $\mathscr{M}$ has no chance to touch any plaintext query result except for cracking encryption or HMAC. Therefore, RSCS-PMQ can protect the privacy of query results from the master nodes.

5.2. Communication Cost Analysis. To analyze the communication costs of data collection and query response protocols, we present the parameters as follows:

n: the number of sensor nodes.

l_{id}: the bit-length of a sensor node ID.

l_t: the bit-length of an epoch.

l_c: the bit-length of an encrypted data item.

l_h: the bit-length of a HMAC data item.

l_q: the bit-length of a query request.

w: the bit-length of a collected data item.

L: the average hops from a sensor node to $\mathscr{M}$.

According to the 0-1 encoding properties, there are w type 0 and type 1 secure comparators for every w bits data. Thus, the random secure comparator of a w-bits data item contains $w/2$ HMAC data on average.

As shown in the data collection protocol, each sensor node submits a node ID, an epoch number, a ciphertext, and a secure comparator to $\mathscr{M}$. Therefore, the communication cost of data collection in the cell, denoted by in-cell communication cost (C_S), can be calculated with

$$C_S = n \cdot \left(l_{\text{id}} + l_t + l_h \cdot \frac{w}{2} + l_e\right) \cdot L. \tag{10}$$

As shown in the query response protocol, the communication cost for executing a query consists of two parts:

Table 1: Computation cost analysis of PMV-PMQ, EMQP, and RSCS-PMQ.

Privacy-preserving MAX/MIN query methods	The quantity of operations in sensor nodes		The quantity of comparison operations in a storage node
	Encryption	HMAC	
PMV-PMQ	1	$[w+2, 3w-1]$	$(n-1) * [w+1, (2w-2) * (w+1)]$
EMQP	1	$w+1$	$(n-1) * [1, w * w]$
RSCS-PMQ	1	$[1, w]$	$(n-1) * [1, w * w]$

Note: $[a, b]$ is an interval range between the low-bound a and the upper-bound b.

one part is the communication cost of BS for sending query requests to $\mathscr{M}$ and the other part is of $\mathscr{M}$ for returning the feedback messages to BS. Additionally, Lemma 10 indicates that the mean quantity of ciphertext returned to BS is the mathematical expectation $E(\lambda)$ of the ciphertext quantity in R. As a result, the calculation of query response communication cost is as shown which is denoted by C_M:

$$C_M = l_q + (l_{\text{id}} + l_e) \cdot E(\lambda). \tag{11}$$

According to Lemma 10, we have

$$\begin{aligned} C_M &= l_q + (l_{\text{id}} + l_e) \cdot \left(2 - (n+2) \cdot \left(\frac{1}{2}\right)^n\right) \\ &\approx l_q + 2(l_{\text{id}} + l_e), \end{aligned} \tag{12}$$

when n is very large.

Then, we have the total communication cost C_{total} as follows:

$$\begin{aligned} C_{\text{total}} &= C_S + C_M \\ &\approx n \cdot \left(l_{\text{id}} + l_t + l_h \cdot \frac{w}{2} + l_e\right) \cdot L + l_q + 2(l_{\text{id}} + l_e). \end{aligned} \tag{13}$$

5.3. Computation Cost Analysis. We analyze the computation cost of proposed RSCS-PMQ and compare it with other privacy-preserving MAX/MIN query methods: PMV-PMQ [18] and EMQP [19]. First of all, since all of the three methods use the complex algorithms of encryption and HMAC in sensor nodes, the computation cost of sensor nodes is mainly caused by the encryption and HMAC. Secondly, the storage node $\mathscr{M}$ determines the encrypted query results according to the intersections of paired code sets. To find out whether the intersection of two sets is null or not, many comparison operations are needed. As a result, the computation cost analysis of the three works is given in Table 1 on two aspects: the quantity of encryption and HMAC operations of a sensor node in an epoch and the quantity of comparison operations of $\mathscr{M}$ in a query.

As shown in Table 1, PMV-PMQ, EMQP, and RSCS-PMQ perform the same quantity of encryption operations in a sensor node, but RSCS-PMQ performs less HMAC operations than the other two methods. RSCS-PMQ and EMQP perform the similar quantity of comparison operations, but they have general better performance than PMV-PMQ. Therefore, the RSCS-PMQ approach proposed by us is more efficient in computation cost than PMV-PMQ and EMQP.

We will not discuss the robustness of our method since it is not the focus of this paper. And we assume that the robustness is supported by the low-layer protocols.

6. Performance Evaluations

To analyze and compare the performance of protocols, we implement the proposed RSCS-PMQ, PMV-PMQ, and EMQP on the improved simulator of [28]. According to the experimental results of [19], we know that the energy consumed by data communication is much larger than that by the computation of encryption and HMAC. Therefore, this paper will focus on the evaluation of communication cost. We perform the evaluations on the following two aspects:

(1) We firstly measure and analyze the in-cell communication cost (C_S) of these three methods. Since the amount of codes for each collected data item constructed in PMV-PMQ is within a certain range, we consider the upper and lower bounds of PMV-PMQ in our evaluations, respectively, that is, the highest and lowest in-cell communication costs, which are denoted by PMV-PMQ-T and PMV-PMQ-B, respectively. Additionally, since the hash-based optimization in EMQP is also suitable for RSCS-PMQ and PMV-PMQ, which is aimed at reducing the length of HMAC data, this paper only compares the C_S of three methods without the hash-based optimization.

(2) To evaluate the query response communication cost (C_M) generated by $\mathscr{M}$ and BS, we firstly measure the probability of R containing λ ciphertext and the average quantity of ciphertext in R in the RSCS-PMQ method. Then, we measure C_M of the three methods and calculate their proportions in the whole network communication costs while processing the MAX query.

The evaluations are performed on a PC with an Intel Core i5-3230M (quad-core 2.6 GHz) CPU and 8 G memory, running Windows 7 operating system, Eclipse, and Matlab. In addition, the experimental data set is randomly generated. In this simulation, the sensor nodes are assumed to be uniformly distributed in a cell covering a $100 \times 100\,\text{m}^2$ area, and the communication radius of a sensor is 20 m. The default setting of other parameters is as shown in Table 2.

6.1. In-Cell Communication Cost Evaluations. In each measurement, we randomly distribute the sensor nodes and generate 10 networks with different topologies represented by different network IDs. Then, we can determine the communication cost of a MAX/MIN query by computing the average communication cost of these 10 networks.

Table 2: Default values of parameters.

Parameter	l_{id}	l_t	n	w	l_c	l_h	l_q
Value	32 bits	32 bits	200	16 bits	128 bits	128 bits	256 bits

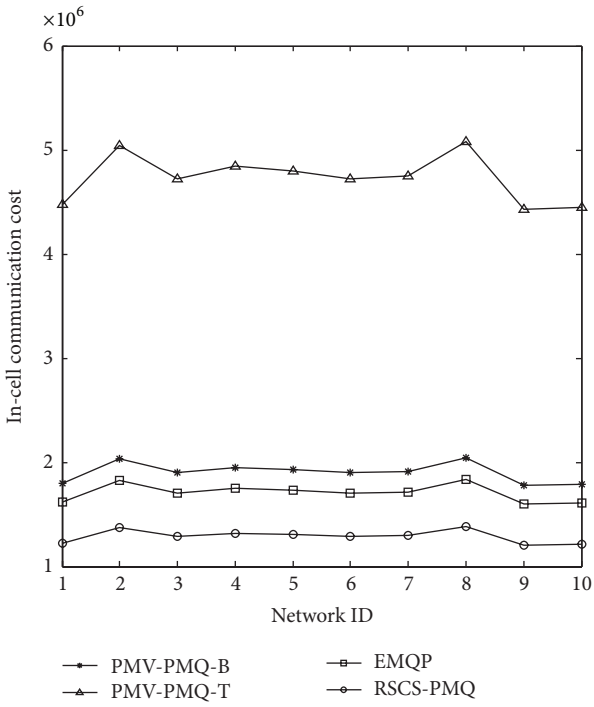

Figure 3: C_S versus network ID.

(1) C_S versus network ID: Figure 3 shows that the C_S of RSCS-PMQ, EPRQ, and PMV-PMQ are all uniformly distributed in different topology networks. And the C_S of PMV-PMQ is the highest, and EPRQ has the mediate C_S, while RSCS-PMQ has the lowest. Under the experiment setting, the C_S of RSCS-PMQ is 32.23% lower than the lower bound of PMV-PMQ and 24.54% lower than the lower bound of EPRQ, since the amount of HMAC data used for secure comparison submitted from the sensor nodes to $\mathscr{M}$ in the former method is smaller than that in the latter.

(2) C_S versus n and w: as shown in Figure 4, when the amount of sensor nodes n increases, the C_S of RSCS-PMQ, EPRQ, and PMV-PMQ also increase, since the amounts of ciphertext and HMAC data transmitted in the network both increase. In accordance with Figure 5, we can see that the C_S of three methods also increase as w increases, because the amount of HAMC data used for secure comparison is in proportion to w. In addition, Figures 4 and 5 indicate that the C_S of three methods are in linear proportion to n and w, which is consistent with the theoretical analysis result shown in (10). Moreover, we have the idea that the C_S of RSCS-PMQ is significantly lower than that of EPRQ and PMV-PMQ, and the former is about 30% lower than the lower bound of PMV-PMQ and about 25% lower than the lower bound of EPRQ.

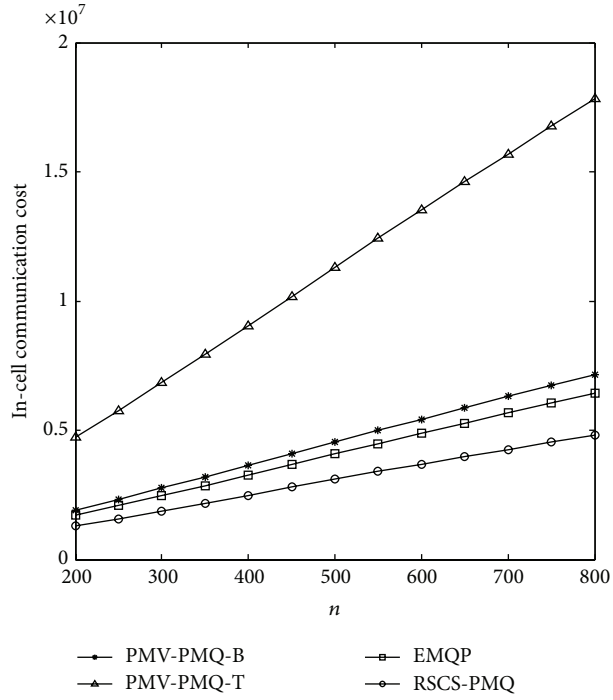

Figure 4: C_S versus n.

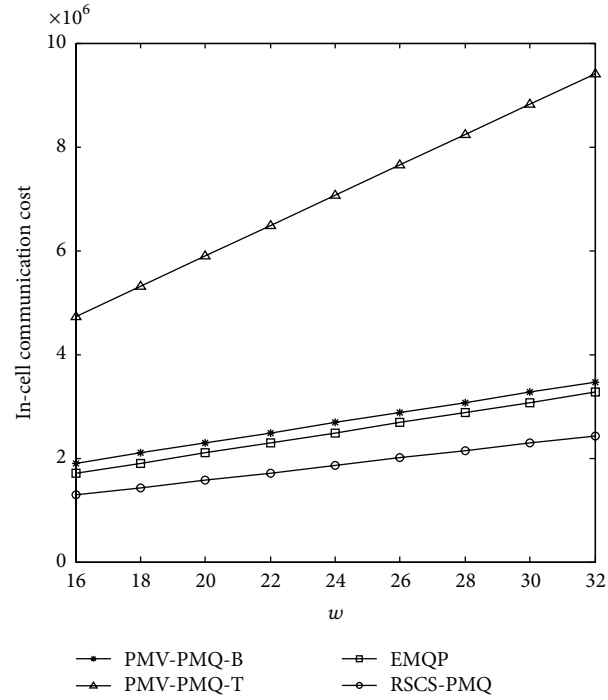

Figure 5: C_S versus w.

(3) C_S versus l_c and l_h: we adopt different encryption and HMAC algorithms to set different l_c and l_h, respectively. For example, l_c could be 64, 128, and

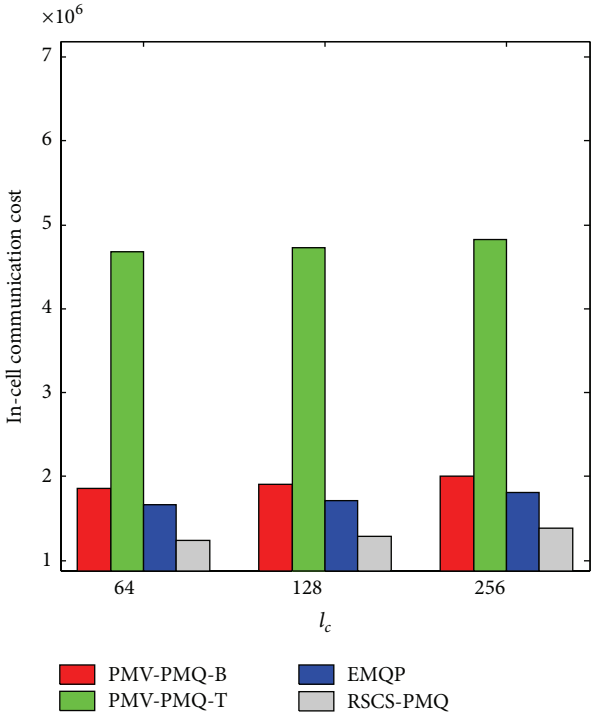

Figure 6: C_S versus l_c.

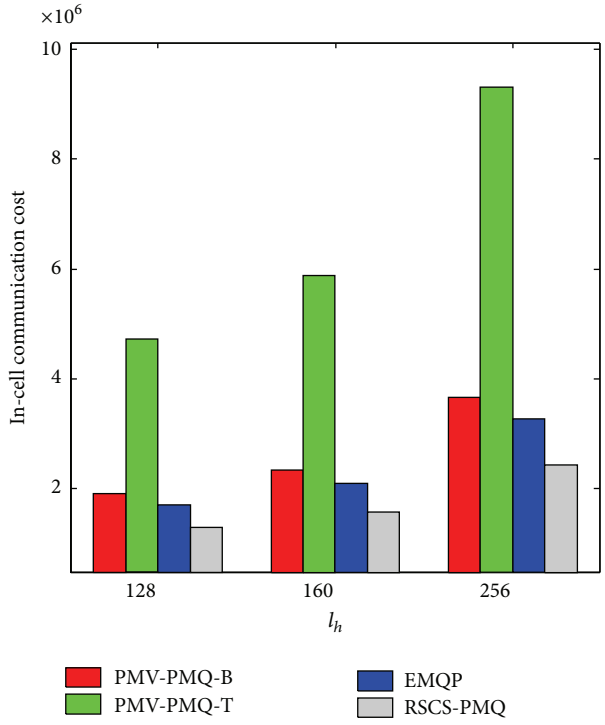

Figure 7: C_S versus l_h.

256 bits if DES, IDEA, and AES-256 are adopted, respectively, while l_h could be 128, 160, and 256 bits if HMAC-MD5, HMAC-SHA1, and HMAC-SHA256 are adopted, respectively.

Figure 6 shows that the C_S of RSCS-PMQ, EPRQ, and PMV-PMQ have slow and unapparent increase as l_c increases, while they increase obviously as l_h increases. The reason is that there is only one encrypted data item in the message submitted from each sensor node to $\mathscr{M}$, but the amount of HMAC data is in proportion to the length of collected data w, which is obviously bigger than the former one. And the increasing of l_h has a more obvious influence on C_S. Similar to the results of evaluations (1) and (2) in this section, Figures 6 and 7 indicate that RSCS-PMQ is significantly lower than EPRQ and PMV-PMQ in C_S, and the former one is about 30% lower than the lower bound of PMV-PMQ and about 25% lower than the lower bound of EPRQ.

6.2. Query Response Communication Cost Evaluations. (1) Assume that the sensor nodes collect data in 10000 epochs and transmit the corresponding ciphertext and HMAC data to $\mathscr{M}$. And $\mathscr{M}$ executes 10000 MAX queries aimed at each epoch mentioned above. We measure the probability and mean value of the amount of ciphertext in R returned from $\mathscr{M}$ to BS. We repeat the experimental process 10 times and get the results as shown in Figures 8 and 9.

From Figure 8, we can see that the probability of R containing λ ciphertexts in the practical experiment totally corresponds to the theoretical probability computed with (8) in Lemma 9, which also proves the correctness of Lemma 9 from the point of experimental statistics. Additionally, based on a large amount of experimental statistics, Figure 9 indicates that the mean quantity of ciphertext in R is in agreement with the mathematical expectation $E(\lambda)$ computed with (9) in Lemma 10, and it is close to 2 as the amount of test samples becomes very large. The result verifies the correctness of Lemma 10 from the point of experimental statistics.

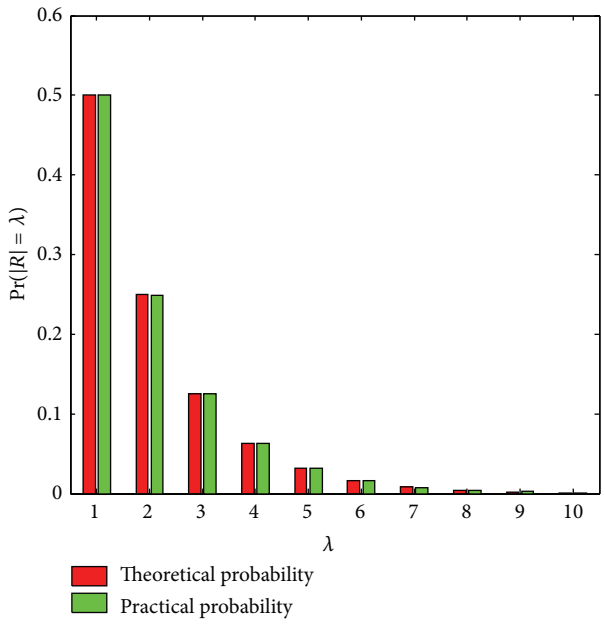

Figure 8: The probability of $|R| = \lambda$.

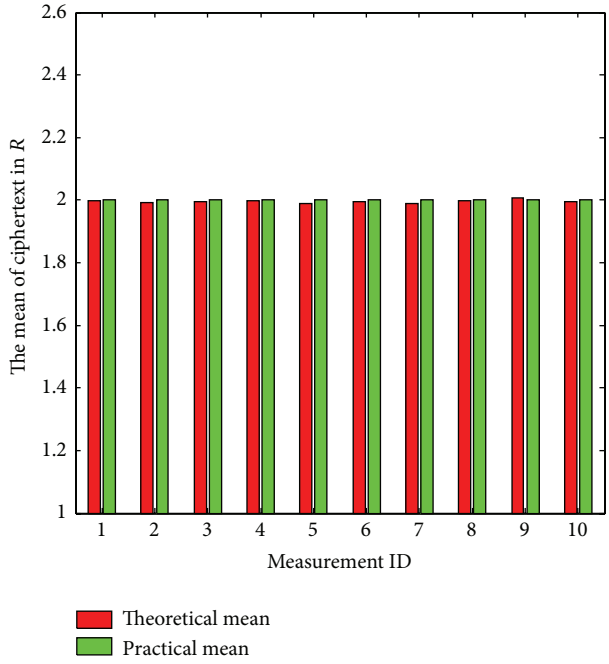

Figure 9: The mean of ciphertext in R.

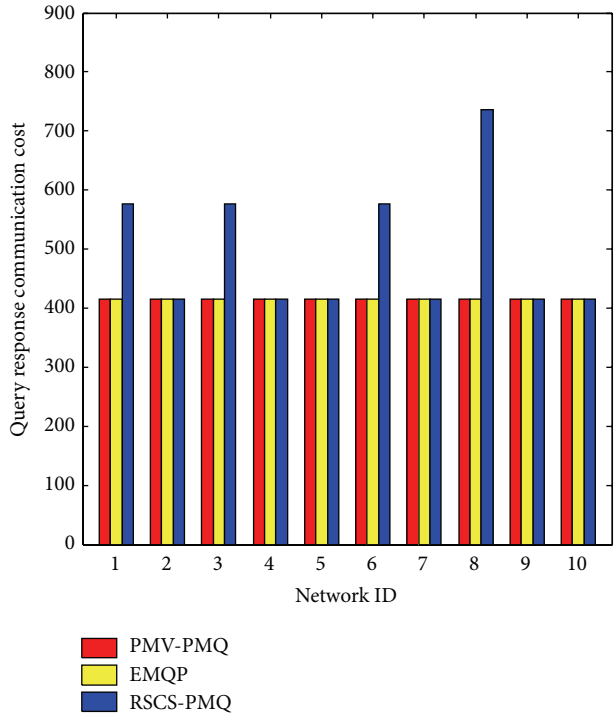

Figure 10: C_M versus network ID.

(2) Based on the 10 groups of data transmitted from the sensor nodes $\mathscr{M}$ under the 10 networks with random topologies in Section 6.1, we process 10 MAX queries, respectively. We test the query response communication costs (C_M) and the average proportion of them in the total network communication costs (C_{total}) for PMV-PMQ, EPRQ, and RSCS-PMQ, respectively. The experimental results are shown in Figures 10 and 11.

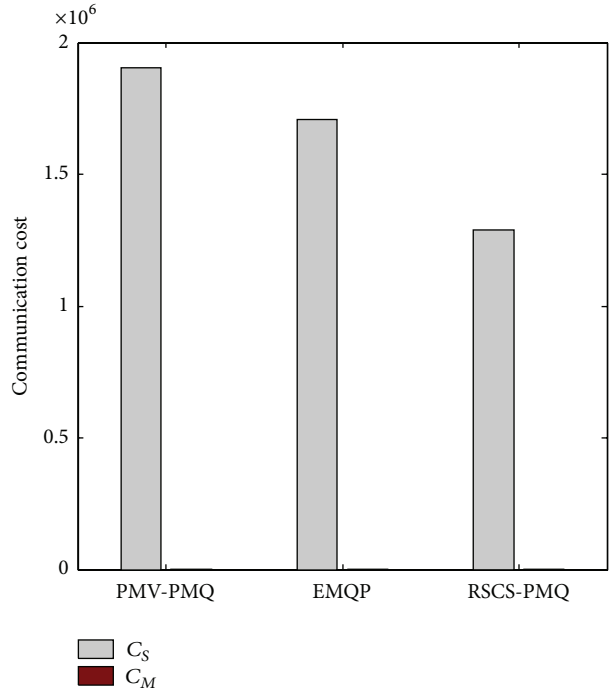

Figure 11: The average of C_S and C_M.

Figure 10 indicates that the C_M of EPRQ and PMV-PMQ are constant and equal, while the C_M of RSCS-PMQ is about 20% higher than the former two methods. The reason is as follows: $\mathscr{M}$ can only determine the ciphertext as the query result in EPRQ and PMV-PMQ, while the result returned by $\mathscr{M}$ in RSCS-PMQ is the set R containing multiple candidate ciphertext. The probability statistics of the amount of ciphertext in R is as shown in Figure 8, and the mean quantity of R is about 2 according to Figure 9.

However, as shown in Figure 11, in the average C_{total} of PMV-PMQ, EPRQ, and RSCS-PMQ where $C_{\text{total}} = C_S + C_M$, the mean value of C_M is significantly smaller than that of C_S, and they merely account for a very small proportion of C_{total}, only 0.22‰, 0.24‰, and 0.38‰ on average, respectively. Here, C_S of PMV-PMQ is the lower bound of its in-cell communication cost. In addition, C_M is generated by the resource-abundant master nodes and BS. As a result, C_M has little impact on C_{total} which is mainly determined by C_S in contract, and C_{total} of RSCS-PMQ is lower than that of PMV-PMQ and EPRQ.

From the above experimental results and analyses, we can obtain the following: compared with the existing EPRQ and PMV-PMQ, the in-cell communication cost of RSCS-PMQ is lower, which is about 30% lower than the lower bound of EPRQ and about 25% lower than the lower bound of PMV-PMQ. Additionally, although the query response communication cost of RSCS-PMQ is higher than that EPRQ and PMV-PMQ, it only accounts for a very small proportion of the total network communication cost, lower than 1%, and so do the later methods. And the total communication cost of

RSCS-PMQ is lower than EPRQ and PMV-PMQ. Thus, the RSCS-PMQ proposed in this paper has a better performance than the existing works.

7. Conclusion

In this paper, we propose a novel random secure compactor selection scheme and a minimal set of highest secure comparators generating algorithm to achieve privacy-preserving MAX/MIN queries in two-tiered wireless sensor networks. Our technique can prevent the compromised master node from peeking at the hosted data and also ensure high query efficiency in network communication cost. Moreover, the efficacy and efficiency of our method are confirmed through detailed evaluations and analysis. In the future works, we will focus on the verification of query result completeness and further develop the key technique of this paper to support other types of data queries.

Conflict of Interests

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

Acknowledgments

This research was supported by the National Natural Science Foundation of China under the Grants nos. 61300240, 61402014, 61572263, 61502251, 61472193, 61302157, 61373138, 61201163, and 61272084, the Natural Science Foundation of Jiangsu Province under the Grants nos. BK20151511 and BK20141429, the Project of Natural Science Research of Jiangsu University under Grants nos. 11KJA520002 and 14KJB520027, the Postdoctoral Science Foundation of China under the Grant no. 2013M541703, the Postdoctoral Science Foundation of Jiangsu Province under the Grant no. 1301042B, and Scientific & Technological Support Project (Society Development) of Lianyungang under the grant no. SH1306.

References

[1] O. Gnawali, K.-Y. Jang, J. Paek et al., "The tenet architecture for tiered sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 153–166, ACM, Boulder, Colo, USA, November 2006.

[2] Y. Diao, D. Ganesan, G. Mathur, and P. Shenoy, "Rethinking data management for storage-centric sensor networks," in *Proceedings of the 3rd Biennial Conference on Innovative Data Systems Research (CIDR '07)*, pp. 22–31, Asilomar, Calif, USA, January 2007.

[3] H.-Y. Lin and W.-G. Tzeng, "An efficient solution to the millionaires' problem based on homomorphic encryption," in *Applied Cryptography and Network Security: Third International Conference, ACNS 2005, New York, NY, USA, June 7–10, 2005. Proceedings*, vol. 3531 of *Lecture Notes in Computer Science*, pp. 456–466, Springer, Berlin, Germany, 2005.

[4] H. Krawczyk, R. Canetti, and M. Bellare, "HMAC: keyed-hashing for message authentication," Tech. Rep. RFC 2104, Internet Society, Reston, Va, USA, 1997.

[5] B. Sheng and Q. Li, "Verifiable privacy-preserving range query in two-tiered sensor networks," in *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM '08)*, pp. 46–50, Phoenix, Ariz, USA, 2008.

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

[7] J. Shi, R. Zhang, and Y. Zhang, "A spatiotemporal approach for secure range queries in tiered sensor networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 1, pp. 264–273, 2011.

[8] B. Sheng and Q. Li, "Verifiable privacy-preserving sensor network storage for range query," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1312–1326, 2011.

[9] F. Chen and A. X. Liu, "SafeQ: secure and efficient query processing in sensor networks," in *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, IEEE, San Diego, Calif, USA, March 2010.

[10] F. Chen and A. X. Liu, "Privacy and integrity-preserving range queries in sensor networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1774–1787, 2012.

[11] Y. Yi, R. Li, F. Chen, A. X. Liu, and Y. Lin, "A digital watermarking approach to secure and precise range query processing in sensor networks," in *Proceedings of the 32nd IEEE Conference on Computer Communications (INFOCOM '13)*, pp. 1950–1958, IEEE, Turin, Italy, April 2013.

[12] R. Zhang, J. Shi, Y. Liu, and Y. Zhang, "Verifiable fine-grained top-k queries in tiered sensor networks," in *Proceedings of 29th IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1199–1207, IEEE, San Diego, Calif, USA, March 2010.

[13] 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.

[14] H. Dai, G. Yang, H. P. Huang et al., "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.

[15] 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.

[16] 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.

[17] X. Liao and J. Li, "Privacy-preserving and secure top-k query in two-tier wireless sensor network," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, pp. 335–341, Anaheim, Calif, USA, December 2012.

[18] Y. Yao, N. Xiong, J. H. Park, L. Ma, and J. Liu, "Privacy-preserving max/min query in two-tiered wireless sensor networks," *Computers & Mathematics with Applications*, vol. 65, no. 9, pp. 1318–1325, 2013.

[19] H. Dai, G. Yang, and X. Qin, "EMQP: an energy-efficient privacy-preserving MAX/MIN query processing in tiered wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 814892, 11 pages, 2013.

[20] J. Cheng, H. Yang, S. H. Y. Wong, P. Zerfos, and S. Lu, "Design and implementation of cross-domain cooperative firewall," in *Proceedings of the 15th IEEE International Conference on Network Protocols (ICNP '07)*, pp. 284–293, Beijing, China, October 2007.

[21] O. Rottenstreich and I. Keslassy, "The Bloom Paradox: when not to use a bloom filter," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 703–716, 2015.

[22] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '04)*, pp. 563–574, Paris, France, June 2004.

[23] M. Rezvani, A. Ignjatovic, E. Bertino, and S. Jha, "Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 98–110, 2015.

[24] S. Roy, M. Conti, S. Setia, and S. Jajodia, "Secure data aggregation in wireless sensor networks: filtering out the attacker's impact," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 681–694, 2014.

[25] Q. Zhou, G. Yang, and L. He, "A secure-enhanced data aggregation based on ECC in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6701–6721, 2014.

[26] G. Yang, S. Li, X. Xu, H. Dai, and Z. Yang, "Precision-enhanced and encryption-mixed privacy-preserving data aggregation in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 427275, 12 pages, 2013.

[27] V. Bozovic, D. Socek, R. Steinwandt, and V. I. Villányi, "Multi-authority attribute-based encryption with honest-but-curious central authority," *International Journal of Computer Mathematics*, vol. 89, no. 3, pp. 268–283, 2012.

[28] A. Coman, M. A. Nascimento, and J. Sander, "A framework for spatio-temporal query processing over wireless sensor networks," in *1st International Workshop on Data Management for Sensor Networks, DMSN '04, in Conjunction with VLDB 2004*, pp. 104–110, can, August 2004.