sensors

MDPI

Article

CSRQ: Communication-Efficient Secure Range Queries in Two-Tiered Sensor Networks

Hua Dai [1,2,*,†], Qingqun Ye [1,†], Geng Yang [1,2,†], Jia Xu [1,†] and Ruiliang He [1,†]

[1] School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, No.66 Xinmofan Road, Nanjing 210013, China; 1214043108@njupt.edu.cn (Q.Y.); yangg@njupt.edu.cn (G.Y.); xujia@njupt.edu.cn (J.X.); 13041015@njupt.edu.cn (R.H.)

[2] Key Laboratory of Broadband Wireless Communication & Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education, Nanjing 210013, China

* Correspondence: daihua@njupt.edu.cn; Tel.: +86-158-9597-7337

† These authors contributed equally to this work.

Academic Editor: Rongxing Lu

Received: 13 November 2015; Accepted: 15 February 2016; Published: 20 February 2016

Abstract: In recent years, we have seen many applications of secure query in two-tiered wireless sensor networks. Storage nodes are responsible for storing data from nearby sensor nodes and answering queries from Sink. It is critical to protect data security from a compromised storage node. In this paper, the Communication-efficient Secure Range Query (CSRQ)—a privacy and integrity preserving range query protocol—is proposed to prevent attackers from gaining information of both data collected by sensor nodes and queries issued by Sink. To preserve privacy and integrity, in addition to employing the encoding mechanisms, a novel data structure called encrypted constraint chain is proposed, which embeds the information of integrity verification. Sink can use this encrypted constraint chain to verify the query result. The performance evaluation shows that CSRQ has lower communication cost than the current range query protocols.

Keywords: two-tiered sensor networks; range query; privacy and integrity preserving; encrypted constraint chain

1. Introduction

Wireless sensor networks (WSNs) provide effective and convenient solutions for various applications, such as environment sensing, military target tracking, intelligent transportation system, *etc*. Two-tiered wireless sensor network is a kind of practical WSN, and its architecture is illustrated in Figure 1 [1,2]. The lower tier of two-tiered WSNs is composed of massively sensor nodes with limited storage and energy that are responsible for collecting data items, while the upper tier is composed of fewer resource-rich storage nodes, whose main tasks are storing data submitted by the nearby sensor nodes and answering queries issued by Sink. Compared to traditional wireless sensor networks, two-tiered WSNs have some significant advantages by interposing storage nodes as an intermediate tier. Firstly, the network topology of two-tiered WSNs is simpler. Secondly, two-tiered WSNs have a higher efficiency on query processing because Sink only communicates with storage nodes for queries.

However, it brings some security challenges to sensor networks where the storage nodes serve as an intermediate tier between the sensor nodes and Sink. Storage nodes not only receive information from the nearby sensor nodes, but also answer queries issued by Sink. Thus, the storage nodes are more attractive to attackers in two-tiered WSNs. Once it is compromised, the sensitive information stored in storage node will be obtained or guessed by the attackers, and the compromised storage node will make the query result incorrect or incomplete by maliciously inserting, deleting or tampering with

the information. Therefore, it is important to study how to protect both the privacy of sensory data and integrity of the query result, even if the storage node is compromised.

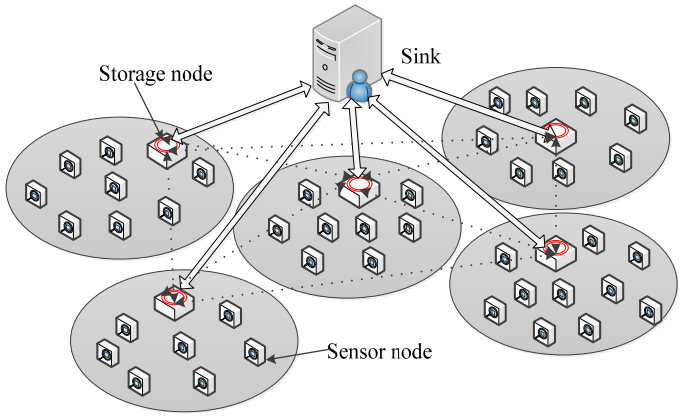

Figure 1. Architecture of two-tiered Wireless Sensor Networks.

Range query is an important type of query in sensor networks. In this paper, we focus on the secure range query processing. The security goals of secure range query include: (1) preserving the privacy of sensory data items and the interested range issued by Sink; and (2) preserving the integrity of the query result. There are two challenges that need to be addressed during the process to achieve these goals above: one is how to process queries without knowing the exact values of data items in sensor networks, and the other is how to verify whether or not the query result is correct and complete.

Therefore, we propose a communication-efficient secure range query processing method, denoted as CSRQ, which is a novel protocol for preserving the privacy and integrity of range query. When deployed in a hostile environment, we use a new data structure named Encrypted Constraint Chain to submit sensory data to the storage nodes. It ensures that the storage nodes cannot disclose the data stored on them. In addition, the message embedded in this chain makes the juggled and/or incomplete data items in queries detectable.

The main contributions in this paper are as follows: (1) A novel encrypted constraint chain model is proposed. Data items are submitted with a complete encrypted chain, which can preserve the privacy of sensitive data in sensor networks. Furthermore, adjacent relations of factors are embedded in chains, which can be used to verify the integrity of query result; (2) Based on the encrypted constraint chain model, a new scheme named CSRQ is proposed. CSRQ can protect sensitive data in two-tiered WSNs from the compromised storage nodes and allow Sink to verify whether the query result is complete and correct; (3) We evaluate our solutions by comprehensive simulation based on real datasets, and the results show that our scheme has a better performance on communication cost.

The rest of this paper is organized as follows. Section 2 gives a brief review of related works. Section 3 describes the system model and attack model. Section 4 proposes the scheme of encrypted constraint chain. Section 5 introduces the process of our protocol in detail. Section 6 analyzes the performance of our approach. We evaluate our approach using thorough experiments in Section 7 and conclude this paper in Section 8.

2. Related Works

Secure queries in two-tiered WSNs have drawn wide attention recently. Prior solutions for secure query in two-tiered WSNs include Top-k queries [3–9], MAX/MIN queries [10,11], range query processing, data aggregation in [12], k-NN processing in [13], *etc*.

The processes of security range queries have been investigated in [14–24]. In [14], Sheng and Li, which is described as S&L below, employed the bucket partition idea to preserve privacy and an authentication encoding mechanism to verify the integrity of query result in two-tiered WSNs. The

basic idea is to divide the domain of data values into multiple buckets and distribute data items into these buckets. In each time slot, the sensor nodes encrypt data items together in each bucket and send them along with bucket ID to the nearby storage node. Then, Sink finds the minimal set of bucket IDs that contains the range in query, and sends the set as the query to storage. The storage nodes find encrypted data items in these buckets and send them to Sink. Finally, Sink decrypts the encrypted buckets. However, the communication cost and memory cost will increase exponentially with increase of the number of buckets in this scheme. In [15,16], Shi *et al.* proposed an optimized version of S&L's scheme to reduce the communication cost between the sensor nodes and storage nodes. The main contribution of their optimization is that a new spatiotemporal crosscheck approach is proposed to verify the integrity of query result, which reduces the communication costs.

However, there are two main drawbacks existing in the schemes of both S&L and Shi *et al*, which are inherited from the bucket partitioning technique. (1) The compromised storage nodes could obtain reasonable estimation of actual values of both data items and queries [17]; (2) The communication cost increases exponentially with the number of dimensions of collected data. For the sake of these problems, Chen and Liu proposed SafeQ in [17,18], which has a better safety performance in preserving the privacy and integrity of range queries. The basic idea of SafeQ is that a prefix-encoding scheme is proposed to encode both data items and queries such that it could not be estimated by compromised storage nodes, and a new data structure called neighborhood chain is proposed to generate integrity verification information, and Sink can verify the integrity of query result using this information.

Although the prefix-encoding scheme and neighborhood chains structure proposed in SafeQ can solve the problems in S&L's scheme, they will increase the memory and communication costs for both sensor nodes and storage nodes. The main reason is that each data item needs to be stored twice in the structure of neighborhood chains. Therefore, Yi *et al.* proposed a new link watermarking scheme named QuerySec in [19], a protocol based on two new techniques: (1) a scheme based on order preserving function for preservation of data privacy; and (2) a new link watermarking scheme for verification of query results. In [21], Nguyen *et al.* proposed a novel model based on a d-disjunct matrix, an order-preserving function and a permutation function to preserve the privacy of sensitive information for the range queries, while it fails to consider the verification of query result.

In [22], an efficient secure range query protocol named ESRQ is proposed to realize more efficient and correct process of range query. In [23], Dong and Zhang provided an extend version of [22], and they were the first ones to focus on collusion attacks for range queries in two-tiered WSNs. The basic idea is that different sensor nodes have different hash functions to encode data items for the protection of data privacy and the correlation among data is used for verification of result. In [24], Dong and Chen *et al.* proposed SecRQ, which not only protects the privacy of data, but also consider the collusion attacks and probability attacks in two-tiered WSNs. It adopts generalized inverse matrices and distance-based range query mechanism for the security of data. Besides, a mutual verification scheme is proposed to verify the integrity of query results in this paper, and it verifies the integrity of query result with lower false positive rate and lower communication cost than the schemes mentioned above.

3. Models and Problems Statement

3.1. Network Model

The adopted architecture of two-tiered WSNs is shown in Figure 1, which is similar to [17]. Two-tiered WSN is a special kind of wireless sensor network, in which the storage node M is the intermediate tier of networks, and the lower tier is composed of sensor nodes. The whole network is divided into a number of *cells*. Each *cell* consists of M and a number of sensor nodes $S = \{s_1, s_2, \ldots, s_n\}$, which can be denoted as $cell = \{M,\{s_1, s_2, \ldots, s_n\}\}$. Sensor nodes are inexpensive sensing devices with limited storage and energy resource and are in charge of collecting data items from a *cell* and submitting information of data items to nearby M. M is resourceful device with relatively high storage and energy resources and is responsible for receiving and storing information submitted by nearby

sensor nodes and answering queries issued by Sink. Sink gathers query results from multiple M and computes the final query result.

3.2. Query Model

The range query refers to accessing all the sensory data items included in a specified range, which can be denoted as a three-tuple:

$$Q_t = (\psi, t, [low, high])$$

where ψ denotes the set ID of queried sensor nodes, t is the queried time slot, and *low* and *high* refer to the lower and upper bounds of query range, respectively. For example, if $Q_t = (\{s_1, s_2, \ldots, s_8\}, t, [10,20])$, it means that Sink finds all data items in [10,20] collected by sensor nodes s_1–s_8 during a time slot t.

For the sake of brevity, we only discuss the range query that Sink queries in a $cell = \{M, \{s_1, s_2, \ldots, s_n\}\}$ during time slot t in this paper, which can be denoted as $Q_t = (\psi, t, [low, high])$, where $\psi = \{s_1, s_2, \ldots, s_n\}$. To get the results of queries in multiple time slots and/or multiple cells, we can simply get the final result by resolving and merging the single results.

3.3. Threat Model

By using the same threat model in [17–22], we assume that Sink and the sensor nodes are trusted in two-tiered WSNs, but M is not. In fact, the sensor nodes can also be compromised in a hostile environment. Attackers may obtain sensitive data items from compromised sensor nodes. A sensor node only contains a small fraction of data items collected by all sensor nodes, while a great deal of sensitive data items are stored in M. Therefore, attackers can steal less information from a compromised sensor node than from M. Therefore, we are mainly concerned with the scenario of the compromised M in this paper.

If M is compromised in two-tiered WSNs, we consider that attackers can attack networks in the following two ways:

(1) The attackers obtain sensitive information stored in M directly or indirectly, which violates the privacy of data.
(2) The attackers forge or exclude the legitimate data items stored in compromised M, which makes the query result incorrect or incomplete.

3.4. Problems Statement

The goal of secure range query is not only to preserve the privacy of data items collected by sensor nodes and queries issued by Sink, but also to ensure that the integrity of query result can be verified by Sink. The details are as follows:

(1) The privacy issues: M could not obtain the actual values of any data items collected by sensor node and the values of lower and higher bounds of query range in Q_t.
(2) The integrity issues: If $Q_t = (\psi, t, [low, high])$, the query result, which can be denoted as QR, should contain all data items satisfying $[low, high]$ in ψ. All data items in QR are collected by the sensor nodes in ψ.

The keys to achieve the security goals above are as follows. First, M could decide whether a data item collected by sensory node should be included in query result by comparing it with *low* and *high* without knowing the actual values of them. Second, Sink could detect whether all data items satisfying $[low, high]$ are included in QR and whether all data items included in QR satisfy $[low, high]$, and that all of them are collected by the sensor nodes in ψ. Thus, we propose CSRQ, a query protocol with better performance in terms of both security and communication cost in preserving the privacy of data items and the integrity of query results.

Moreover, as an index used to evaluate the performance of security range query protocol, the communication costs include two aspects: one is the communication cost for sensor node, which plays a decisive role in the lifecycles of sensor networks; the other is the communication cost between M and Sink, which directly affects the operating costs of networks. We conduct a detailed analysis for the index of performance in Sections 6 and 7.

4. Encrypted Constraint Chain Model

In this section, we will introduce the encrypted constraint chain in detail, which is proposed to protect the privacy and integrity of two-tiered WSNs.

Definition 1. *Encrypted constraint chain: Given n numbers stored in the ascending order* $D = \{d_1, d_2, \ldots, d_n\}$, *where* $d_1 < \ldots < d_n$, *we partition these numbers into several parts with parameter* τ, *and encrypt every part. These encrypted parts can easily be brought together to form encrypted constraint chain* C_τ.

$$C_\tau = F_1 \bowtie F_2 \bowtie \ldots \bowtie F_\delta \tag{1}$$

Here "$\bowtie$" $\bowtie$denotes concatenation, and δ is the number of items in C_τ. We call F_i the constraint factor of C_τ, and $F_i.ds$ represents the dataset in F_i. C_τ satisfies following conditions:

(1) F_i has τ sensory data items, where $1 \leqslant i \leqslant \delta - 1$, while F_δ has no more than τ sensory data items.
(2) The upper and lower bounds of F_i are denoted as $UB(F_i)$ and $LB(F_i)$, respectively. Hence, for any two adjacent constraint factors F_i and F_{i+1}, we can determine $UB(F_i) = LB(F_{i+1})$.
(3) The computation formula of δ is

$$\delta = \begin{cases} 1 & \tau \geqslant n \\ 1 + \left\lceil \dfrac{n-\tau}{\tau-1} \right\rceil & \tau < n \end{cases} \tag{2}$$

The form of F_i in C_τ is as shown in Equation (3), in which k represents an encryption key, and "$||$" denotes the concatenation of data items.

$$F_i = \begin{cases} \left(d_{1+(i-1)\cdot(\tau-1)} \middle|\middle| \ldots \middle|\middle| d_{1+i\cdot(\tau-1)}\right)_k & 1 \leqslant i < \delta \\ \left(d_{1+(\delta-1)\cdot(\tau-1)} \middle|\middle| \ldots \middle|\middle| d_n\right)_k & i = \delta \end{cases} \tag{3}$$

Definition 2. *Let* $C_\tau = F_1 \bowtie F_2 \bowtie \ldots \bowtie F_\delta$ *be an encrypted constraint chain, where* δ *is called the length of* C_τ *and it is denoted as* $|C_\tau| = \delta$. F_i *satisfies that* $F_i \in C_\tau$, *and* F_{i-1} *and* F_{i+1} *are called the left and right neighbor constraint factors of* F_i, *respectively. The head factor of* C_τ *is denoted as* $head(C_\tau) = F_1$ *and the tail factor is denoted as* $tail(C_\tau) = F_\delta$.

Definition 3. *Given two encrypted chains* C_τ *and* C_τ'. *If each factor of* C_τ' *is included in* C_τ, *then* C_τ' *is called a sub-chain of* C_τ. *It can be denoted as* $C_\tau' \subseteq C_\tau$. *Thus, we have*

$$\forall F_i \in C_\tau' \, (F_i \in C_\tau) \Rightarrow C_\tau' \subseteq C_\tau \tag{4}$$

According to Definition 3, we can easily deduce the following property.

Property 1. *The sub-chain relation*$\subseteq$ *is transitive, which means*

$$C_\tau'' \subseteq C_\tau' \wedge C_\tau' \subseteq C_\tau \Rightarrow C_\tau'' \subseteq C_\tau \tag{5}$$

Definition 4. *Given an encrypted constraint chain* C_τ, *let* C_τ' *be a sub-chain of* C_τ, *which means* $C_\tau' \subseteq C_\tau$. *For a query range* [*low, high*], *if* C_τ' *simultaneously satisfies the following conditions, then* C_τ' *is called a Maximum Encrypted Constraint Sub-chain (MECS) of* C_τ.

(1) $LB(head(C_\tau{}')) < low \leqslant UB(head(C_\tau{}')) \wedge LB(tail(C_\tau{}')) \leqslant high < UB(tail(C_\tau{}'))$

(2) $\forall F_i \in (C_\tau{}' - head(C_\tau{}') - tail(C_\tau{}')) \left(\forall d_j \in F_i.ds(d_j \in [low, high])\right)$

(3) $\forall F_i \in C_\tau \wedge F_i \notin C_\tau{}' \left(\forall d_j \in F_i.ds \left(d_j \notin [low, high]\right)\right)$

Given a *MECS* of C_τ that satisfies [*low*, *high*], according to *Definition* 4, we can know that *low* is between the lower and upper bounds of head constraint factor in *MECS*, and *high* is between the lower and upper bounds of tail constraint factor. Except the head and tail, the data items in each factor of *MECS* are between *low* and *high*, and the data items included in C_τ but outside of *MECS* are not included in [*low*, *high*].

Now we give a further instruction of the above definitions and properties with some examples. For example, D = {3, 6, 11, 23, 38, 42}. Given a parameter τ = 3, the encrypted constraint chain is $C_\tau = (3\,|\,|\,6\,|\,|\,11)_k \bowtie (11\,|\,|\,23\,|\,|\,38)_k \bowtie (38\,|\,|\,42)_k$, where $|\,C_\tau\,| = 3$, $UB((3\,|\,|\,6\,|\,|\,11)_k) = LB((11\,|\,|\,23\,|\,|\,38)_k)$ and $UB((11\,|\,|\,23\,|\,|\,38)_k) = LB((38\,|\,|\,42)_k)$. Given a range [7,23], the *MECS* of C_τ which satisfies [7,23] is $C_\tau{}' = (3\,|\,|\,6\,|\,|\,11)_k \bowtie (11\,|\,|\,23\,|\,|\,38)_k$.

As demonstrated by the previous definitions and properties, if F_i and F_{i+1} are two adjacent factors in the encrypted constraint chain, the upper bound of F_i equals to the lower bound of F_{i+1}. Thus, it provides a theoretical basis for the integrity verification of query results.

5. Secure Range Query Protocols

In this paper, we employ the 0-1 encoding mechanism [25] to preserve privacy. The basic idea of 0-1 encoding mechanism is to convert the verification of whether a data item is within a range to the verification of whether there are intersections between two sets. Given a number x whose binary format is $b_1 b_2 \ldots b_{n-1}$, $b_n \in \{0,1\}^n$, where n is the bit length of number x. The 0-coding of x is defined as $E^0(x) = \{b_1 b_2 \ldots b_{i-1}\,|\,b_i = 0 \wedge 1 \leqslant i \leqslant n\}$ and the 1-coding of it is defined as $E^1(x) = \{b_1 b_2 \ldots b_i\,|\,b_i = 1 \wedge 1 \leqslant i \leqslant n\}$. If and only if $E^1(x) \cap E^0(y) \neq \emptyset$, $x > y$, else, $x \leqslant y$. According to the above definitions, given two numbers x and y, they can be compared with each other using their 0-1 codings. What is noteworthy is that x and y can be compared only if they are of different encoding types, which means that they have the encoding types of 0-coding and 1-coding, respectively.

In this paper, we convert each 0-1 coding to a corresponding unique number using the numerical function $\mathcal{N}$(*) similarly to that used in [18] and we encode the 0-1 coding data items using the keyed-Hash Message Authentication Code (HMAC) to ensure that it is infeasible for M to steal sensitive data items. We denote HMAC function as $HMAC_g$(*), where g is a key for HMAC which is only known to sensor node and Sink. A data processed by 0-1 encoding and HMAC is denoted as a comparator. $HNE^0(x) = HMAC_g\mathcal{N}(E^0(x)))$ is a comparator of 0-coding and $HNE^1(x) = HMAC_g(\mathcal{N}(E^1(x)))$ is a comparator of 1-coding. Thus, we can easily know $x > y$ if and only if $HNE^1(x) \cap HNE^0(y) \neq \emptyset$.

5.1. Submission Protocol

The submission protocol concerns how a sensor node submits its data to the nearby M. First, the sensor node encrypts all the data items collected during a time slot t, then builds the corresponding encrypted constraint chain and computes the comparators of encrypted data items. After that, the sensor node sends the encrypted constraint chain to the nearby storage node.

We denote d_{max} and d_{min} as the lower and upper bounds of a sensory data item, respectively. Let τ be the partition parameter of encrypted constraint chain. Each sensor node s_i in a network shares a secret key $k_{i,t}$ with Sink. The submission protocol is illustrated as the following Protocol 1.

Protocol 1: Submission Protocol

Let $d_{i,1}, d_{i,2}, \ldots, d_{i,N}$ be N data items collected by s_i during time slot t. For simplicity, we assume $d_{min} \leqslant d_{i,1} \leqslant \ldots \leqslant d_{i,N} \leqslant d_{max}$. Let $D_i = \{d_{min}, d_{i,1}, \ldots, d_{i,N}, d_{max}\}$. Then s_i performs the following steps.

(1) Compute the 0-1 code of each data item.

(2) Build the encrypted constraint chain $C_{\tau,i}$ of D_i with τ and $k_{i,t}$. Assuming $C_{\tau,i} = F_{i,1} \bowtie F_{i,2} \bowtie \ldots \bowtie F_{i,\delta}$, we can compute δ and $F_{i,j}$ as follows.

$$\delta = \begin{cases} 1 & \tau \geqslant N+2 \\ 1 + \left\lceil \dfrac{N+2-\tau}{\tau-1} \right\rceil & \tau < N+2 \end{cases} \tag{6}$$

$$F_{i,j} = \begin{cases} (d_{\min} || d_{i,1} || \ldots || d_{i,\tau-1})_{k_{i,t}} & j = 1 \\ (d_{i,(j-1)\cdot(\tau-1)} || \ldots || d_{i,j\cdot(\tau-1)})_{k_{i,t}} & 1 < j < \delta \\ (d_{i,(\delta-1)\cdot(\tau-1)} || \ldots || d_n || d_{\max})_{k_{i,t}} & j = \delta \end{cases} \tag{7}$$

(3) Compute the comparator of $UB(F_{i,j})$, where $1 \leqslant j \leqslant \delta - 1$. It means computing $HNE^0(UB(F_{i,j}))$ and $HNE^1(UB(F_{i,j}))$. Then, add the comparator set which contains the fewest elements into Ω_i. Thus we have,

$$\Omega_i = \{min\{HNE^0(UB(F_{i,j})), HNE^1(UB(F_{i,j})) \mid F_{i,j} \in C_{\tau,i} \wedge 1 \leqslant j \leqslant \delta - 1\}\} \tag{8}$$

where $min(X, Y)$ denotes the set containing the fewest elements.

(4) Send the following message to M, where $id(s_i)$ denotes the ID of s_i in networks.

$$s_i \rightarrow M : < id(s_i), t, C_{\tau,i}, \Omega_i >$$

In order to build the encrypted constraint chain easily, we denote an encrypted group as a constraint factor in this paper. Thus, the more sensory data are contained in an encrypted group, the less encrypted data and HMAC data are contributed by a sensor node. It contains at most $\left\lfloor le/w \right\rfloor$ sensory data items in an encrypted group, where the length of sensory data items is w and the length of an encrypted group is l_e. Thus, we can set $\tau = \left\lfloor le/w \right\rfloor$ to reduce the communication cost consumed by the data submitting of sensor nodes.

5.2. Query Protocol

The query protocol concerns how Sink and M process the queries correctly. To ensure that both the privacy of data and the integrity of query result can be preserved, we will give the basic idea of our query protocol. First, Sink computes the comparators of *low* and *high* in $Q_t = (\psi, t, [low, high])$, and then replaces the *low* and *high* in Q_t with their comparators respectively. After that, Sink sends the modified Q_t to M. Second, after receiving a query, M decides whether the value of comparator in encrypted constraint chain contributed by sensor node is included in $[low, high]$ using the 0-1 coding mechanism, and computes the minimal set which contains all data items satisfying the queries. Then, M sends this set to Sink. Third, after receiving this minimal set, Sink decrypts each encrypted data in dataset, and computes the QR. Finally, Sink verifies the authenticity and completeness of QR. The above steps show the basic idea of query protocol, and the detailed performance of query is as follows:

Protocol 2: Query Protocol

Phase 1: Sink sends query to *M*

Sink firstly computes $\{HNE^0(low), HNE^1(low), HNE^0(high), HNE^1(high)\}$, the comparator of *low* and *high* in $Q_t = (\psi, t, [low, high])$, and replaces the *low* and *high* in Q_t with their corresponding comparators, respectively. Then, Sink sends $Q_t = (\psi, t, \{HNE^0(low), HNE^1(low), HNE^0(high), HNE^1(high)\})$ to *M*.

Phase 2: *M* processes the query

Upon receiving Q_t from Sink, *M* performs the following two steps. We denote *CS* as the minimal encrypted dataset received by Sink, which contains the query results. The initial *CS* is empty, which is denoted as $CS = \emptyset$.

(1) Let $C_{\tau,i} = F_{i,1} \bowtie F_{i,2} \bowtie ... \bowtie F_{i,\delta}$ be an encrypted constraint chain contributed by s_i during a time slot, and $min\{HNE^0(UB(F_{i,j})), HNE^1(UB(F_{i,j}))\}$ is the comparator of $UB(F_{i,j})$, which is a factor in Ω_i. According to *Definition* 1, $LB(F_{i,j}) = UB(F_{i,j-1})$ $(1 < j \leqslant \delta)$, it is clear that the comparator of $LB(F_{i,j})$ is $min\{HNE^0(UB(F_{i,j-1})), HNE^1(UB(F_{i,j-1}))\}$ in Ω_i. Here, the 0-1 encoding technology is employed to compare $UB(F_{i,j})$ and $LB(F_{i,j})$ with *low* and *high*, where $F_{i,j}$ is a constraint factor in $C_{\tau,i}$. If one of the following three conditions can be satisfied, add $F_{i,j}$ into $\Im_i$, where $\Im_i$ is a set of constraint factor. We set $\Im_i = \emptyset$ initially.

Condition 1: $low \leqslant LB(F_{i,j}) \leqslant high$

Condition 2: $low \leqslant UB(F_{i,j}) \leqslant high$

Condition 3: $LB(F_{i,j}) \leqslant low \wedge high \leqslant UB(F_{i,j})$

After all constraint factors in $C_{\tau,i}$ are processed through the above steps, then all the constraint factor set $\Im_i$ will be added into another set *CS*.

(2) After all sensor nodes are processed through step (1), *M* will send the following message to Sink.

$$M \rightarrow \text{Sink} : < t, \{id(s_i), \Im_i \mid s_i \in \psi \wedge \Im_i \subseteq CS\} >$$

Phase 3: Sink receives the message

After receiving the message from *M*, Sink decrypts the encrypted message in *CS* and computes the query result *QR*, and then verifies the integrity of *QR*. The process of verification is detailed in Algorithm 1 of Section 5.3.

The Protocol 2 shows that the *CS* received by Sink is as follows:

$$CS = \cup_{s_i \in \Psi} \{\Im_i\} \tag{9}$$

Property 2. *In CS, it includes at least one item in* $\Im_i$ *contributed by* $s_i \in \psi$, *so we have*

$$\forall s_i \in \Psi \rightarrow |\Im_i| \geqslant 1 \tag{10}$$

Proof: Let $C_{\tau,i} = F_{i,1} \bowtie F_{i,2} \bowtie ... \bowtie F_{i,\delta}$ be a *MECS* received by *M* from s_i. According to *Definition* 1, we can easily know that any constraint factor's upper bound equals to the next factor's lower bound, which means $UB(F_{i,j}) = LB(F_{i,j+1})$. $[LB(F_{i,j}), UB(F_{i,j})]$ is a range interval composed of the lower and upper bounds of $F_{i,j}$. Thus, the composition process of $C_{\tau,i}$ in Protocol 1 shows that the following Equation (11) is true.

$$[d_{\min}, d_{\max}] = \cup_{1 \leqslant j \leqslant \delta} \left[LB(F_{i,j}), UB(F_{i,j})\right] \tag{11}$$

Because d_{min} and d_{max} are the lower and upper bounds of sensory data, the query range [*low*, *high*] must be included in $[d_{min}, d_{max}]$. According to this, we know that there is at least one constraint factor $F_{i,j}$ that satisfies $[LB(F_{i,j}), UB(F_{i,j})] \cap [low, high] \neq \emptyset$. There exist the following four possible cases.

Case 1. $LB(F_{i,j}) \leqslant low \leqslant UB(F_{i,j}) \leqslant high$
Case 2. $LB(F_{i,j}) \leqslant low \leqslant high \leqslant UB(F_{i,j})$
Case 3. $low \leqslant LB(F_{i,j}) \leqslant high \leqslant UB(F_{i,j})$
Case 4. $low \leqslant LB(F_{i,j}) \leqslant UB(F_{i,j}) \leqslant high$

According to Protocol 2, it is easy to know that $F_{i,j}$ should be added into $\Im_i$ if any one of the above four cases is satisfied. Thus, there is at least one item included in $\Im_i$, which means Property 2 is true.

Property 3. *We assume $C_{\tau,i}$ is an encrypted constraint chain that M receives from $s_i \in \psi$. Thus $\Im_i$ in Protocol 2 is a MECS of $C_{\tau,i}$ which satisfies* [*low, high*].

Proof: Assume that $\Im_i = \{F_{i,j}, F_{i,j+1}, \ldots, F_{i,j+p-1}\}$, where $p \geqslant 1$. Based on the three conditions that must be satisfied in *MECS* of Definition 4, we give a proof of Property 3. (1) According to the process of forming $\Im_i$ in Protocol 2, if $F_{i,j-1}$ and $F_{i,j}$ are included in $C_{\tau,i}$, $UB(F_{i,j-1}) < low$. If $UB(F_{i,j-1}) \geqslant low$, $F_{i,j-1}$ should also be added into $\Im_i$. Thus, it would be contradictory to the assumption that $\Im_i = \{F_{i,j}, F_{i,j+1}, \ldots, F_{i,j+p-1}\}$. What is more, Definition 1 shows that $UB(F_{i,j-1}) = LB(F_{i,j})$. If $F_{i,j} \in \Im_i$, then $LB(F_{i,j}) < low \leqslant UB(F_{i,j})$. Similarly, if $F_{i,j+p}$ and $F_{i,j+p-1}$ are included in $C_{\tau,i}$, $LB(F_{i,j+p}) \leqslant high < UB(F_{i,j+p})$. Furthermore, $head(\Im_i) = F_{i,j}$ and $tail(\Im_i) = F_{i,j+p}$ show that $\Im_i$ satisfies the first condition of Definition 4. (2) The conclusion that $LB(F_{i,j}) < low \leqslant UB(F_{i,j})$ and $LB(F_{i,j+p}) \leqslant high < UB(F_{i,j+p})$ in (1) indicate that any factors between $F_{i,j}$ and $F_{i,j+p}$ are included in [*low*, *high*], and the values of data in factors before $F_{i,j}$ are smaller than *low*, and those in factors after $F_{i,j+p}$ are bigger than *high*. It means that factors out of $\Im_i$ are not included in [*low*, *high*]. Thus, it illustrates that $\Im_i$ can also satisfy Conditions 2 and 3 in Definition 4.

Based on (1) and (2), it can be proven that $\Im_i$ is the *MECS* of $C_{\tau,i}$ which satisfies [*low*, *high*].

Theorem 1. *CS is the minimal encrypted dataset, which includes all data items satisfying* [*low, high*], *where CS is contributed by sensor nodes in* ψ.

Proof: Because all of $\Im_i$ contributed by $s_i \in \psi$ are *MECS* of $C_{\tau,i}$ that satisfy [*low*, *high*], according to the definition of *MECS*, we can know that any data items in constraint factors of $\Im_i$ are included in [*low*, *high*], and any data items out of $\Im_i$ are not included in [*low*, *high*]. Thus, $\Im_i$ is the minimal encrypted dataset which includes all data items satisfying [*low*, *high*] in $C_{\tau,i}$. As shown in Equation (9), $CS = \cup_{s_i \in \Psi} \{\Im_i\}$, therefore, *CS* is the minimal encrypted dataset which satisfies [*low*, *high*] in ψ.

5.3. The Computation of Query Result and the Algorithm of Integrity Verification

After receiving *CS* from *M*, Sink will decrypt the encrypted data in *CS* using the keys shared with sensor nodes and compute the query result *QR*, and then, it will verify the integrity of *QR*. Algorithm 1 shows the details of integrity verification.

Algorithm 1: The algorithm of integrity verification

Let $CS = \cup_{s_i \in \Psi} \{\Im_i\}$ be an encrypted dataset that Sink receives from *M*. Sink verifies the integrity of *QR* as follows.
Sink performs the following three steps to verify each $\Im_i$ contributed by $s_i \in \psi$.
(1) If $\Im_i = \emptyset$, it could not satisfy Definition 2. Thus the integrity of *QR* is violated. Quit the algorithm.
(2) If $\Im_i \neq \emptyset$, Sink decrypts all factors in $\Im_i$ using $k_{i,t}$ only shared with s_i, and checks whether both of following two conditions are satisfied. If so, add all the data within [*low*, *high*] into *QR*, and then turn to Step (3). Otherwise, the integrity of *QR* is violated so quit the algorithm.
1) Each factor $F_{i,v}$ in $\Im_i$ satisfies following condition:

$$low \leqslant LB(F_{i,v}) \leqslant high \vee low \leqslant UB(F_{i,v}) \leqslant high \vee LB(F_{i,v}) \leqslant low \wedge high \leqslant UB(F_{i,v})$$

2) $F_{i,k}$ and $F_{i,v+1}$ satisfy the following formula, where $F_{i,v}$ and $F_{i,v+1}$ are two adjacent factors in $\Im_i$.

$$UB(F_{i,v}) = LB(F_{i,v+1})$$

(3) If all factors contributed by $s_i \in \psi$ are processed through Steps (1) and (2), and all of them satisfy the query, then the *QR* satisfies the query, thus return the *QR*. Otherwise, continue to process the next $\Im_i$ contributed by unprocessed sensor node, and turn to Step (1).

Algorithm 1 shows that the key to verify the integrity of the *QR* is to check whether all of following three conditions can be satisfied. First, each sensor node s_i to be queried contributes a non-empty set $\Im_i$. Second, all factors in *QR* satisfy the query range [*low*, *high*]. Third, the upper bound of each factor equals to the lower bound of next one. If and only if all of the above three conditions are satisfied will *QR* satisfy the integrity of query results.

6. Protocol Analyses

In two-tiered WSNs, we mainly evaluate the performance of a security query protocol from following two aspects: one is security, and the other is the communication cost. In this section, we will analyze the performance of CSRQ from these two aspects.

6.1. Security Analysis

6.1.1. Privacy Analysis

(1) The privacy of sensory data. The key to preserve the privacy of data items in two-tiered WSNs is to ensure that *M* cannot steal the actual values of encrypted data items without knowing the secret keys. If a storage node is compromised, CSRQ can effectively preserve the privacy of sensitive data. Because s_i encrypts the collected data items using its private keys only shared with Sink before sending data items to *M* in CSRQ, it is very difficult for attackers to obtain the actual values of sensory data. Furthermore, the HMAC mechanism is employed in CSRQ to ensure that it is computationally infeasible to compute the actual values of sensory data without knowing both the Hash key and its secret key. Therefore, CSRQ has a better performance in protecting the privacy of sensory data.

(2) The privacy of query result. Similar to the privacy protection of sensory data, the key to protect privacy of query result is to ensure that *M* cannot get the actual values of results. In CSRQ, the sensor nodes send data items to *M* with the form of encrypted constraint chain and their corresponding comparators, and *M* compares the query range with data items without knowing the actual value of them, all data items included in *CS* are encrypted. Upon receiving *CS*, only Sink can decrypt the data items in *CS* and compute the query result. Therefore, without knowing the key used in the encryption, it is very difficult to steal the value of query result.

(3) The privacy of query range. In CSRQ, it does not allow attackers to obtain the actual values of query range either. Sink sends the query to *M* after replacing the query range with their corresponding comparators, which ensures that it is very difficult for *M* to leak the information of query range.

Thus, the CSRQ proposed in this paper can ensure that the privacy of sensory data, query result and query range can be protected.

6.1.2. Integrity Analysis

In CSRQ, we propose a novel encrypted constraint chain to ensure that the integrity of query result can be verified by Sink. The main idea is that the data items collected by all sensor nodes during a time slot *t* will be sent to the nearby *M* with the form of a complete encrypted constraint chain, which allows Sink to verify the integrity of query result by checking the relationship of adjacent factors in the chain. The integrity verification includes the following two-fold: one is verifying whether the data item satisfying the query is forged, and the other is verifying whether the data items satisfying query are deleted by attackers.

Let $\Im_i = F_{i,j} \bowtie F_{i,j+1} \bowtie \ldots \bowtie F_{i,v}$ be the *MECS* which satisfies $Qt = (\psi, t, [low, high])$ received by Sink from $s_i \in \Psi$ during time slot *t*. We assume that *M* is compromised and it attempts to attack $\Im_i$. Next, we will analyze the integrity of CSRQ from the following cases.

(1) If data item in *QR* satisfying the query is forged by the compromised *M*:

① If $F_{i,u}'$ is a tampered data item which replaces the original $F_{i,u}$ in $\Im_i$, Sink will find that $UB(F_{i,u\text{-}1}) \dotplus LB(F_{i,u}')$ or $UB(F_{i,u}') \dotplus LB(F_{i,u+1})$ after decrypting $\Im_i$. It could not satisfy the definition of encrypted constraint chain (*Definition* 1). Therefore, $\Im_i$ can be determined as incomplete. Or Sink will detect that

$d_\alpha \in F_{i,u}' \cdot ds$, where $d_\alpha \notin [low, high]$, which contraries to the definition of *MECS* (Definition 4). Thus *CR* can also be determined as incomplete.

② Similar to ① above, if $F_{i,u}'$ is inserted as a tampered data item between $F_{i,u}$ and $F_{i,u+1}$, Sink will detect that $UB(F_{i,u}) \neq LB(F_{i,u}')$ or $UB(F_{i,u}') \neq LB(F_{i,u+1})$ after decrypting $\Im_i$. It could not satisfy *Definition* 1 either, which means that *CR* is incomplete.

(2) If data item that satisfies the query range is deleted by *M*:

① If all data items in $\Im_i$ are deleted by attackers, Sink will detect that $\Im_i = \emptyset$, which is contrary to Property 2. Thus, Sink can judge that $\Im_i$ has been attacked.

② If the data items between $F_{i,a}$ and $F_{i,b}$ deleted by *M*, where $j \leqslant a < b \leqslant v$, Sink will detect that $UB(F_{i,a}) \neq LB(F_{i,b})$, which dissatisfies Definition 1. Thus, $\Im_i$ will be determined as incomplete.

③ If the head constraint factor of $\Im_i$ $F_{i,j}$ is deleted by attackers, $F_{i,j+1}$ will be the new head($\Im_i$) of $\Im_i$. Then, Sink will detect that all data items in $F_{i,j+1}$ are included in [*low*, *high*] after decrypting $\Im_i$, which could not satisfy the Condition (1) of Definition 4. Therefore, the incomplete $\Im_i$ can be detected by Sink. Similarly, if the deleted constraint factor $F_{i,j}$ is tail ($\Im_i$), it can also be detected by Sink.

In conclusion, Sink can verify the correctness and completeness of query result effectively in CSRQ.

6.2. Communication Cost

6.2.1. Communication Cost of Sensor Node

In two-tiered WSNs, the communication cost of sensor node is mainly incurred by transferring data items from the sensor nodes to *M*. Here, let E_c be the communication cost during a data submission for each sensor node.

Let n be the size of two-tiered WSNs inquired about, and l_t be the bit length of a time slot. We assume that each sensor node collects N data items during a time slot. According to the definition of encrypted constraint chain in our scheme, N data items will be divided into δ constraint factors by parameter τ, where δ can be calculated from Equation (2). Let l_e, l_h and l_{id} be the average length of an encrypted constraint factor, a HMAC encoding and an ID of encrypted node, respectively, and L be the average hops from each sensor node to *M*. By analyzing Protocol 1, the formula to calculate the communication cost is gained as follows.

$$E_C = \sum_{i=1}^{n} (l_{id} + l_t + \delta \cdot l_e + (\delta - 1) \cdot l_h)) \cdot L \tag{12}$$

6.2.2. Communication Cost of Query

The query Protocol 2 shows that the query is a collaborative process between *M* and Sink, thus the communication costs of query should contain two aspects: one is the cost for sending the query from Sink to *M*, and the other is the cost for sending the message from *M* to Sink. We assume that the minimal encrypted dataset contributed by s_i includes ρ_i constraint factors, and all other parameters have the same meaning given above. Similarly, we can know the communication cost E_Q for the query is as follows.

$$\begin{aligned} E_Q &= l_{id} + l_t + 4 \cdot l_h + n \cdot (l_{id} + l_t) + l_e \cdot \sum_{i=1}^{n} \rho_i \\ &= 4 \cdot l_h + (n+1) \cdot (l_{id} + l_t) + l_e \cdot \sum_{i=1}^{n} \rho_i \end{aligned} \tag{13}$$

In conclusion, we can finally obtain E_{total}, which is the total communication cost for CSRQ, simply by adding E_C to E_Q. Then, we have

$$\begin{aligned} E_{total} &= E_C + E_Q \\ &= \sum_{i=1}^{n} (l_{id} + l_t + (\delta_i - 1) \cdot l_h + \delta_i \cdot l_e) \cdot L + 4 \cdot l_h + (n+1) \cdot (l_{id} + l_t) + l_e \cdot \sum_{i=1}^{n} \rho_i \end{aligned} \tag{14}$$

In the next section, we will provide some further details about the performance analysis.

7. Experimental Results

For a further analysis of the performance of communication cost in CSRQ, we compare CSRQ with SafeQBloom, QuerySec, ESRQ and SecRQ by implementing these five schemes on a large real dataset from Intel Lab [26], and present the results of detailed performance evaluation obtained using MATLAB.

7.1. Contrastive Experimentfor Communication Cost

Since the sensor nodes in wireless sensor networks are mainly powered by battery, their energy is limited [27]. It is important to reduce the communication cost of sensor nodes in order to prolong the life of network. Thus, we analyze the communication costs of sensor nodes by comparing the performance of CSRQ with that of SafeQBloom, QuerySec, ESRQ and SecRQ on six aspects, including the network topology, the number of sensor nodes in a cell, the number of data items collected by a sensor node during a time slot, the length of an encoding data, an encrypted constraint factor and the number of constraint factors. We assume some default values of parameters, which are shown in Table 1 below.

Table 1. Experiment parameters.

Parameter	Value	Parameter	Value
The area covered networks/m^2	80 × 80	Length of a time-slot/b	4 × 8
Radius of sensor node communication/m	10	Length of a sensor node ID/b	4 × 8
Number of collected data item in a time-slot (N)	20	Network IDs	20
Number of factor in a Constraint Chain (τ)	4	Length of a data (w)/b	16
Length of a constraint factor (l_e)/b	128	Number of sensor nodes (n)	400

In our experiment, we assume that there are 20 groups of networks with various network topologies randomly distributed in the networks. The network ID of each group is unique. Then, the communication costs of query can be determined by computing the average costs of the 20 groups of networks. The details of experimental results and analysis are as follows:

(1) Impact of network ID. Figure 2 shows the communication cost of sensor nodes impacted by ID. Let other parameters be the default values.

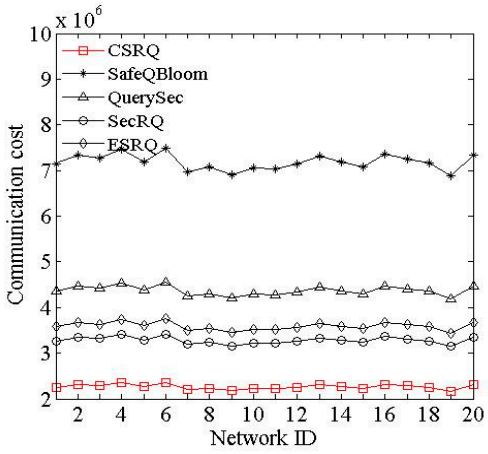

Figure 2. Impact of network ID on communication cost.

According to Figure 2, there is little change caused by different network topologies in these five mechanisms, and all of their communication costs of sensor nodes fluctuate within a small scope.

The average communication cost for sensor nodes of SafeQBloom is relatively high, while the costs of CSRQ and SecRQ are lower. The communication cost of CSRQ is lowest, which is 68.5% lower than that of SafeQBloom and 9.4% lower than that of SecRQ. The reasons are as follows. In CSRQ, it contains $\tau - 1$ data items in each constraint factor except the first and last ones. Therefore, fewer messages will be submitted to M than those in SafeQBloom and SecRQ. What is more, during each time slot, the sensor nodes only need to send the minimal set of each factor's upper boundary along with encrypted constraint chain to M, which can also reduce the communication cost of sensor node.

(2) Impact of n and N. We conducted experiment with different n and N, respectively, while other parameters are default values.

Figures 3 and 4 show the communication cost of sensor nodes under the impact of n and N, respectively, where n is the number of sensor nodes in a cell, and N is the number of data items collected by a sensor node during a time slot. The communication cost of sensor nodes increases with n and N in these five schemes. In CSRQ, the 0-1 encoding scheme is used for comparison, which requires fewer messages to be transferred. It can significantly reduce the communication cost of sensor node. Thus, compared to other four schemes, CSRQ has the lowest communication cost of sensor nodes. In conclusion, the experimental data demonstrates that CSRQ can achieve security range query with lower communication cost than the existing security range query schemes.

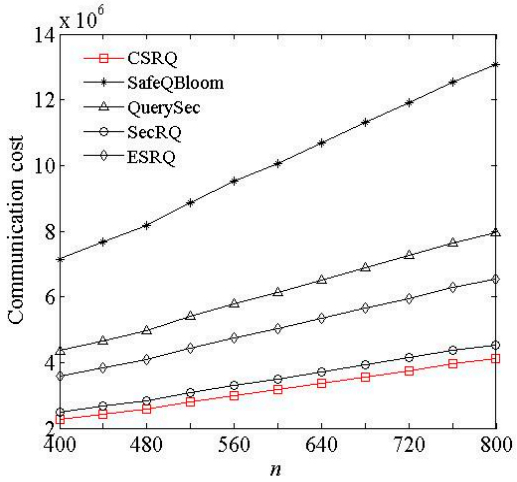

Figure 3. Impact of n on communication cost.

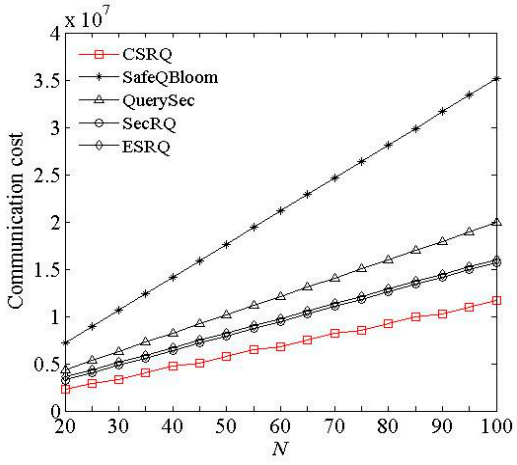

Figure 4. Impact of N on communication cost.

(3) Impact of w and l_e. As w and l_e increase, the changes of communication costs of sensor nodes are shown in Figures 5 and 6 where w denotes the bit length of data collected by sensor node, and l_e denotes the average length of an encrypted constraint factor.

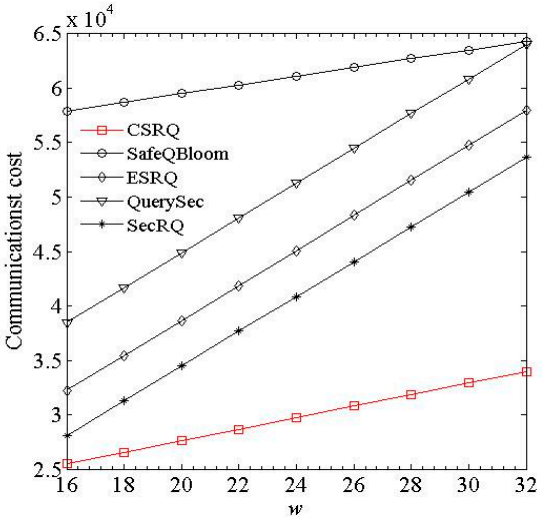

Figure 5. Impact of w on communication cost.

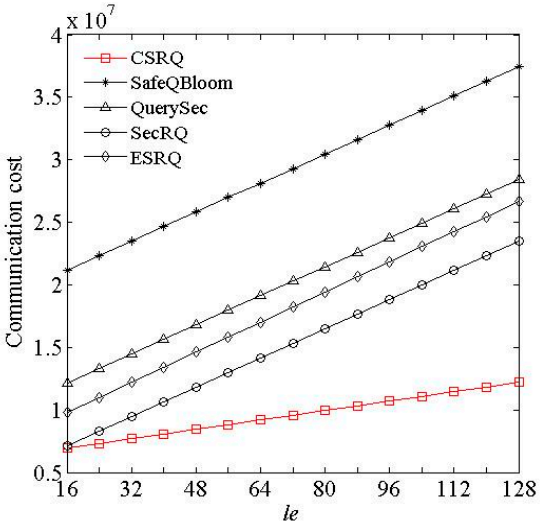

Figure 6. Impact of l_e on communication cost.

Figures 5 and 6 show that in these five schemes, longer lengths of sensory data and encrypted constraint factor will both cause greater communication cost of sensor nodes. CSRQ has lower communication costs than the others. The reason is similar to Figures 2 and 3.

(4) Impact of δ. With δ changed, where δ denotes the number of encrypted constraint factor, the change of communication costs of sensor nodes is shown in Figure 7.

Figure 7 reveals that the sensor node's communication costs in the five schemes increase with δ, and CSRQ has a lower communication cost than the others. The reason is as follow. The larger δ means that the sensor nodes submit more messages to M. In CSRQ, each sensor node only needs to submit $\delta - 1$ minimal sets of boundary messages along with encrypted constraint chain to M. It means that fewer messages need to be transferred in CSRQ than in other schemes.

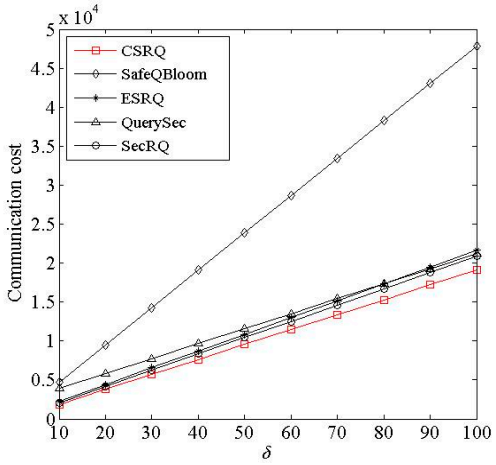

Figure 7. Impacted of δ on communication cost.

7.2. Contrastive Experiment for False Positive Rate

We define the false positive rate as the ratio of the number of unsatisfactory data items received by Sink to the number of data items satisfying query range. Therefore, the lower the false positive rate is, the higher the accuracy is.

Figure 8 reveals the false positive rate impacted by the network size, where the network size refers to the data items collected by the sensor nodes and transmitted to M. We can see that QuerySec, ESRQ and SecRQ have no false positive, and the average false positive rates of SafeQBloom and CSRQ are 0.47% and 0.51%, respectively. In both CSRQ and SafeQBloom, the query results received by Sink may contain constraint factors in which only most parts of data items satisfy the query range. What is more, in CSRQ, it contains at most $2(\tau - 1)$ unsatisfied data items in result received by Sink, and in general, $\tau > 2$, which is a little more than that in SafeQBloom. Therefore, the false positive rate of former scheme is slightly higher than that of CSRQ. Furthermore, the false positive rate of CSRQ is very low, and it decreases with the increase of network size, and then gradually approaches 0. Thus, it has little effect on the performance of range queries.

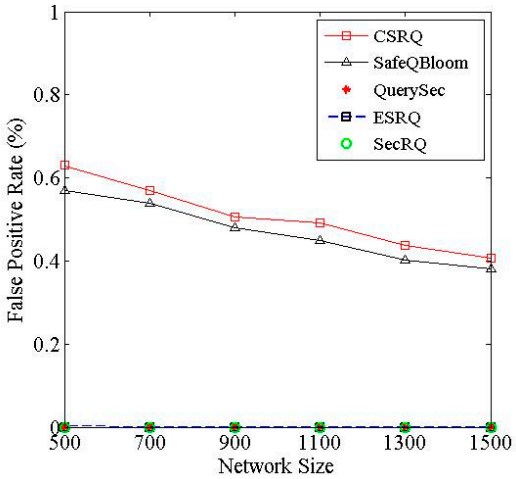

Figure 8. Impact of network size on false positive rate.

The experiment results show that CSRQ provides a better performance in communication cost than current query protocols, such as SafeQBloom, QuerySec, ESRQ and SecRQ, in terms of efficiency.

8. Conclusions

In this paper, we propose CSRQ, a novel efficient protocol for processing range queries in two-tiered WSNs, which has great performance in privacy and integrity preservation. To preserve the privacy of data items in networks, we encrypt the data items collected by the sensor nodes through the encoding mechanisms. To preserve the integrity of query range and result, we present a novel encrypted constraint chain scheme to link data items collected by a sensor node to each other, which allows Sink to verify the integrity by checking the adjacent relations embedded in the encrypted constraint chains. The results of our experiment show that CSRQ has a better performance in terms of efficiency than current query protocols.

Acknowledgments: This research was supported by the National Natural Science Foundation of China under the grant Nos. 61300240, 61402014, 61572263, 61502251, 61472193, 61302157, 61373138, 61201163 and 61272084; the Natural Science Foundation of Jiangsu Province under the grant Nos. BK20151511 and BK20141429; the Project of Natural Science Research of Jiangsu University under grant Nos. 14KJB520027; the Postdoctoral Science Foundation of China under the grand No. 2013M541703; the Postdoctoral Science Foundation of Jiangsu Province under the grand No. 1301042B; and CCF-Tencent Open Research Fund No. CCF-Tencent RAGR20150107.

Author Contributions: Hua Dai and Qingqun Ye conceived and designed models and protocols. Qingqun Ye analyzed the data and wrote the manuscript. Geng Yang contributed ideas for the experiment and analyzed the performance of our scheme. Jia Xu and Ruiliang He contributed to contrastive experiments and English language correction. All authors of the manuscript provided substantive comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gnawali, O.; Jang, K.Y.; Paek, J.; Vieira, M.; Govindan, R.; Greenstein, B.; Joki, A.; Estrin, D.; Kohler, E. The tenet architecture for tiered sensor networks. In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys'06), Boulder, CO, USA, 31 October–3 November 2006; pp. 153–166.
2. Desnoyers, P.; Ganesan, D.; Shenoy, P. TSAR: A two-tier sensor storage architecture using interval skip graphs. In Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys'05), San Diego, CA, USA, 2–4 November 2005; pp. 39–50.
3. Dai, H.; Yang, G.; Huang, H.P.; Xiao, F. Efficient Verifiable Top-k Queries in Two-Tiered Wireless Sensor Networks. *KSII Trans. Internet Inf. Syst.* **2015**, *9*, 2111–2131.
4. Zhang, R.; Shi, J.; Liu, Y.; Zhang, Y. Verifiable fine-grained top-k queries in tiered sensor networks. In Proceedings of the 29th IEEE International Conference Computer Communications, San Diego, CA, USA, 14–19 March 2010; pp. 1199–1207.
5. Fan, Y.; Chen, H. Verifiable privacy-preserving top-k query protocol in two-tiered sensor networks. *Chin. J. Comput.* **2012**, *35*, 423–433. [CrossRef]
6. Li, R.; Lin, Y.; Yi, Y.; Xiong, S.; Ye, S. A secure top-k query protocol in two-tiered sensor networks. *J. Comput. Res. Dev.* **2012**, *49*, 1947–1958. (In Chinese).
7. Yao, Y.; Ma, L.; Liu, J. Privacy-preserving Top-k Query in Two-tiered Wireless Sensor Networks. *Int. J. Adv. Comput. Technol.* **2012**, *4*, 226–235.
8. He, R.L.; Dai, H.; Yang, G.; Wang, T.C.; Bao, J. An Efficient Top-k Query Processing with Result Integrity Verification in Two-Tiered Wireless Sensor Networks. *Math. Probl. Eng.* **2015**, *2015*, 538482. [CrossRef]
9. Yu, C.M.; Ni, G.K.; Chen, Y.; Gelenbe, E.; Kuo, S.Y. Top-Query Result Completeness Verification in Tiered Sensor Networks. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 109–124. [CrossRef]
10. Yao, Y.; Xiong, N.; Park, J.H.; Ma, L.; Liu, J. Privacy-preserving max/min query in two-tiered wireless sensor networks. *Comput. Math. Appl.* **2012**, *65*, 1318–1325. [CrossRef]
11. Dai, H.; Wei, T.Y.; Huang, Y.; Xu, J.; Yang, G. Random Secure Comparator Selection based Privacy-Preserving MAX/MIN Query Processing in Two-tiered Sensor Networks. *J. Sens.* **2016**, *2016*, 6301404. [CrossRef]
12. Yao, Y.; Liu, J.; Xiong, N.N. Privacy-Preserving Data Aggregation in Two-Tiered Wireless Sensor Networks with Mobile Nodes. *Sensors* **2014**, *14*, 21174–21194. [CrossRef] [PubMed]

13. Peng, H.; Zhang, X.; Chen, H.; Wu, Y. Enable privacy preservation for k-NN query in two-tiered wireless sensor networks. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 6289–6294.
14. Sheng, B.; Li, Q. Verifiable privacy-preserving range query in two-tiered sensor networks. In Proceedings of the International Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008; pp. 46–50.
15. Shi, J.; Zhang, R.; Zhang, Y. Secure range queries in tiered sensor networks. In Proceedings of the IEEE International Conference on Computer Communications, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 945–953.
16. Zhang, R.; Shi, J.; Zhang, Y. Secure multidimensional range queries in sensor networks. In Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hong Kong, China, 18–21 May 2009; pp. 197–206.
17. Chen, F.; Liu, A.X. SafeQ: Secure and efficient query processing in sensor networks. In Proceedings of the IEEE International Conference on Computer Communications, San Diego, CA, USA, 15–19 March 2010; pp. 2642–2650.
18. Chen, F.; Liu, A.X. Privacy and integrity-preserving range queries in sensor networks. *IEEE/ACM Trans. Netw.* **2012**, *20*, 1774–1787. [CrossRef]
19. Yi, Y.; Li, R.; Chen, F.; Liu, A.X.; Lin, Y. A digital watermarking approach to secure and precise range query processing in sensor networks. In Proceedings of the 2013 IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 1950–1958.
20. Tsou, Y.T.; Lu, C.S.; Kuo, S.Y. Privacy-and integrity-preserving range query in wireless sensor networks. In Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, USA, 3–7 December 2012; pp. 328–334.
21. Nguyen, T.D.; Bui, T.V.; Dang, V.H.; Choi, D. Efficiently preserving data privacy range queries in two-tiered wireless sensor networks. In Proceedings of the International Conference on Ubiquitous Intelligence and Computing and International Conference on Autonomic and Trusted Computing, Fukuoka, Japan, 4–7 September 2012; pp. 973–978.
22. Zhang, X.; Dong, L.; Peng, H.; Chen, H.; Li, D.; Li, C. Achieving efficient and secure range query in two-tiered wireless sensor networks. In Proceedings of the IEEE/ACM International Symposium on Quality of Service, Hong Kong, China, 26–27 May 2014; pp. 380–388.
23. Zhang, X.; Dong, L.; Peng, H.; Chen, H.; Zhao, S.; Li, C. Collusion-Aware Privacy-Preserving Range Query in Tiered Wireless Sensor Networks. *Sensors* **2014**, *14*, 23905–23932. [CrossRef] [PubMed]
24. Dong, L.; Chen, X.; Zhu, J.; Chen, H.; Wang, K.; Li, C. A Secure Collusion-Aware and Probability-Aware Range Query Processing in Tiered Sensor Networks. In Proceedings of the IEEE 34th Symposium on Reliable Distributed Systems (SRDS), Montreal, Quebec, Canada, 28 September–1 October 2015; pp. 110–119.
25. Lin, H.Y.; Tzeng, W.G. An efficient solution to the millionaires' problem based on homomorphic encryption. In *Applied Cryptography and Network Security*; Springer Berlin Heidelberg: Berlin, Germany, 2005; pp. 456–466.
26. Bodik, P.; Hong, W.; Guestrin, C.; Madden, S.; Paskin, M.; Thibaux, R. Intel Lab Data. Available online: http://db.csail.mit.edu/labdata/labdata.html (accessed on 28 February 2004).
27. Xie, S.; Wang, Y. Construction of Tree Network with Limited Delivery Latency in Homogeneous Wireless Sensor Networks. *Wirel. Personal Commun.* **2014**, *78*, 231–246. [CrossRef]