Hindawi
Journal of Electrical and Computer Engineering
Volume 2018, Article ID 9140937, 8 pages
https://doi.org/10.1155/2018/9140937

Research Article

Secure and Efficient Cluster-Based Range Query Processing in Wireless Sensor Networks

Liming Zhou,[1] **Yingzi Shan**,[2] **and Lu Chen**[1]

[1]*School of Computer and Information Engineering, Henan University, Kaifeng 475004, China*
[2]*Department of Finance, Yellow River Conservancy Technical Institute, Kaifeng 475004, China*

Correspondence should be addressed to Liming Zhou; lmzhou@henu.edu.cn

Received 2 February 2018; Revised 10 July 2018; Accepted 18 July 2018; Published 2 October 2018

Academic Editor: Stefano Vitturi

In wireless sensor networks, preserving privacy is more important and has attracted more attentions. Protecting data and sensor privacy while collecting and computing query results is a challenge. In cluster-based sensor networks, when a user queries a sensitive data, the adversaries can monitor original node or gain the data in cluster node. To deal with this problem, we propose a secure and efficient scheme for cluster-based query processing in wireless sensor networks. To preserve location privacy of sensors, we use anonymity method to confuse adversaries. To protect the sensitive data, we use prefix membership verification method to prevent adversaries from gaining sensitive messages collected by sensor nodes. And we analyze the security and communication cost. The results show that our scheme can efficiently protect privacy in query processing.

1. Introduction

Wireless sensor networks have been widely deployed in various applications, such as monitoring environment, collecting temperature data, and gaining information of battlefield. Each sensor node transmits sensed data to a base station for further processing. In some applications, clustering method has been extensively studied [1] and used to organize sensor nodes, which has been considered as a useful approach. And some nodes are grouped into clusters such that sensor node sends data to a cluster head in the same cluster. Many clustering applications aimed at enhancing the energy efficiency and extending the network lifetime in wireless sensor networks.

In wireless sensor networks, when sensor nodes collect information in our daily life, we should pay attention to protect data privacy and security. For instance, a user wants to query sensitive data from a certain sensor node according to his interests. The sensor network may leak private information about the user's interests to an adversary who can gain the content from the queried data. Meanwhile, the adversary can monitor the frequency of query to analyze the user's preferences and find the related sensor nodes. Then the adversary may attack and compromise the related nodes. And the compromised node may respond to a query and send fake data to the user, which is in conflict with the privacy requirement. So query processing brings serious security challenges.

When users monitor events or analyze sensed data, data query becomes an important operation in wireless sensor networks. Recently, many existing privacy techniques can be employed in sensor network scenarios. For example, a target region transformation technique [2], range query [3–6], and top-k query [7, 8] have been well addressed. However, these schemes are not suitable for cluster-based query processing in wireless sensor networks. And many techniques do not consider the computing power, power of sensors, and capacity, which are the limiting factors in wireless sensor networks. For limited availability resources, it is important to make the trade-off between the privacy preservation and the communication overhead.

Based on the above discussions, in this paper, we propose a novel cluster-based privacy preserving query processing in wireless sensor networks. We consider the privacy issue when processing data query in wireless sensor networks. If a user wants to gain and query information from the

cluster-based sensor network, we will use the anonymity method and the prefix membership verification scheme [9, 10] to protect the sensitive data against adversaries. When a cluster head receives a query message, the cluster head will randomly choose several cluster members which include the real queried node. So, it is unlikely that the adversary can monitor the real frequency of query in a cluster. Therefore, the adversary cannot gain the user's interests from analyzing the frequency of query or find the location of the real source node. Meanwhile, cluster members encode their sensed data and send them to their cluster head. The cluster heads can correctly process data queries over encoded data without knowing their real values. And an adversary cannot know the query results in the cluster heads. In our scheme, we make a balance between data confidentiality and query efficiency.

The rest of the paper is organized as follows. Section 2 gives the related work and the previous proposed techniques for data query. In Section 3, we describe the system model and the security model. Then, we present our secure and efficient cluster-based query processing scheme in Section 4. In Section 5, we present the security analysis and performance analysis. Finally, we have the conclusion in Section 6.

2. Related Work

Protecting querying region's privacy in wireless sensor networks has been drawn attention recently [2]. In [2], a querying region transformation technique is proposed to fuzzy the target region of the query according to predefined transformation functions. The transformation function maps one region into m regions so that the target region cannot be distinguished from the other uninteresting regions. Meanwhile, multiple transformation functions include uniform, randomized, and hybrid function.

A secure and efficient range query processing scheme is proposed in [6], called SafeQ. They use the prefix membership verification and neighborhood chains to encode both data and queries such that a storage node can correctly process encoded queries over encoded data without knowing their values. The prefix membership verification converts the verifications of whether a number is in a range to several verifications of whether two numbers are equal. The neighborhood chains allow a sink to verify whether the result of a query contains exactly the data items that satisfy the query. The SafeQ scheme can preserve privacy and integrity for processing range queries in two-tiered sensor networks.

Privacy preserving range query has been widely studied in two-tiered wireless sensor networks. Many range query schemes are proposed to protect privacy of range queries. CSRQ [3] employs an encoding mechanism and encrypted constraint chain to preserve data privacy and query result integrity. In [11], Zhang et al. provided an efficient secure range query protocol. In their scheme, 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 [12, 13], two optimized versions which verify query result completeness to reduce the communication overhead between sensors and storage nodes based on the bucketing technique are proposed. In their scheme, a bit map is broadcasted by each sensor node to the nearby sensors, which indicates which buckets have data. In each sensor node, the collected data items and the received bit maps are encrypted together. The sink can verify the completeness of the query result for a sensor by examining the bit maps. But the compromised storage nodes can estimate the values of data items by using the bucketing technique to achieve data privacy.

Privacy preserving max/min query schemes in two-tiered sensor networks are proposed in [14–17], which use the prefix membership verification scheme to privately compute the maximum or minimum data item. But their schemes cannot be suitable for cluster-based sensor networks. The power and storage are limited in cluster heads.

3. Network and Adversary Models

3.1. Network Model. Sensor networks consist of a number of different types of sensor nodes that have been deployed to monitor environment or collect data and send information to the sink in an area. Nodes are organized into clusters. A cluster head is selected in each cluster to receive and query data from cluster members. In each cluster, every sensor sends data to its cluster head. The sink collects data with a lot of resources in storage, energy, and computation.

In this paper, we assume that sensor nodes are evenly deployed in the sensor network and do not move after being deployed. All of the sensors have roughly the same capabilities, power sources, and expected lifetimes. The users can access the sensor network by the sink. The sink translates a query from a user into multiple queries which are sent to the cluster heads. The cluster heads process the queries and return the query results to the sink. All query results are sent to the sink which changes all results into a final query result and sends the final result back to the user. When a user makes a query request, the sink will send query request to each cluster head. The cluster heads collect all results and send them back to the sink. The results are forwarded through certain routing strategies that adopted the sensor networks.

3.2. Adversary Model. For various kinds of wireless sensor networks, we assume that an adversary is a motivated and funded attacker whose objective is to learn sensitive data information. The adversary has unbounded energy resource, adequate computation capability, and sufficient memory of data storage. The adversary can use the leaked sensitive data to threaten the sensor network, such as health monitoring networks. For a user's query, the adversary tries to generate fake message and send it back to the user.

Meanwhile, the adversary wants to gain the user's interests and the frequency of query in clusters. He wants to find the location information of queried nodes. The adversary may stay nearby the cluster to monitor and eavesdrop constantly. When the adversary monitors a message in a cluster, he will know the location of a sensor node. If the

frequency of transmitted messages is large, the adversary will find that certain sensor node is important for the user. When the adversary compromises the sensor node, the compromised node will send fake data to the user.

4. Secure and Efficient Cluster-Based Query Processing Scheme

In this section, we propose a scheme for preserving privacy query processing in cluster-based sensor networks. Each cluster head collects the data from sensor nodes in a cluster. To preserve privacy, sensor nodes encrypt or encode their collected data, for example, DES algorithm. So the adversary cannot gain the content of transmitted data.

4.1. The Basic Idea. In order to preserve privacy query processing, we propose a secure and efficient cluster-based query processing scheme to address this problem in wireless sensor networks. After the sensor network is deployed, the cluster heads are randomly chosen. Then the cluster heads broadcast their join messages. When a node firstly receives a join message from a cluster head, it will reply to the cluster head and join the cluster. The cluster head will record the sensor's ID. Meanwhile, the sink will record all cluster heads' ID.

However, if a cluster head has the less remaining energy, it will randomly select one of its members as the new cluster head in the cluster. And the new cluster head will record the ID of all members in the cluster. Then, the sink will replace the ID of the previous cluster head with the ID of the new cluster head.

When a user wants to gain the value of a sensor node s_i, he will make a query to the sink. The sink will send the query message to the cluster head which includes the sensor node s_i. The cluster head will randomly select several cluster members which include the real queried node s_i and gain the sensed data from them. It is aimed for preventing the adversary from monitoring the real frequency of the query in a cluster. We assume that each sensor s_i shares a secret key k_i with the sink in a network. A sensor s_i encrypts its sensed n data items $d_1, d_2, \ldots, d_n$ using key k_i in time slot t, the result of which is denoted as $(d_1)_{k_i}, \ldots, (d_n)_{k_i}$. Then, s_i encode $d_1, d_2, \ldots, d_n$ as $E(d_1, d_2, \ldots, d_n)$. Moreover, s_i sends the message that includes the encrypted data $(d_j)_{k_i}$ and the encoded data $E(d_1, d_2, \ldots, d_n)$ to its cluster head. The cluster head transmits the message to the sink. When the user wants to perform query $\{[a, b]\}$, the sink encodes the range $[a, b]$ as $G([a, b])$. Then, the sink applies a secret comparing method $C(E(d_1, d_2, \ldots, d_n), G([a, b]))$ to be used for query processing over encrypted and encoded data. A data d is in range $[a, b]$ if and only if $C(E(d_1, d_2, \ldots, d_n), G([a, b]))$ is true. Then, the sink decides whether $(d_j)_{k_i}$ should be included in the query result. Meanwhile, given $E(d_j)$ and $(d_j)_{k_i}$, it is infeasible for the sink to compute $d_j (1 \le j \le n)$. This condition can guarantee query privacy. Figure 1 illustrates the basic idea of cluster-based query processing scheme.

4.2. Prefix Membership Verification. We protect privacy query processing by using the prefix membership verification scheme which is first introduced in [8] and later formalized in [9]. In the prefix membership verification scheme, the key idea is to convert the verification of whether a number is in a range to several verifications of whether two numbers are equal. A $k-$prefix is in the form of $\{0, 1\}^k (*)^{w-k}$, which has k leading 0s and 1s, followed by $w-k*$s. For instance, $101*$ is a 3-prefix and it denotes the range [1010, 1011].

A prefix family consists of w bits binary number $b_1 b_2 \cdots b_w$, which is defined as the set of $w+1$ prefixes $\{b_1 b_2 \cdots b_w, b_1 b_2 \cdots b_{w-1}*, \cdots, b_1 * \cdots *, ** \cdots *\}$, where the ith prefix is $b_1 b_2 \cdots b_{w-i+1} * \cdots *$. The prefix family of x is denoted as $F(x)$. For example, the prefix family of number 11 is $F(11) = F(1011) = \{1011, 101*, 10**, 1***, ****\}$. In prefix membership verification scheme, for any number x and prefix P, $x \in P$ if and only if $P \in F(x)$.

In order to confirm whether a number x is in a range $[d_1, d_2]$, the range $[d_1, d_2]$ can be translated into a minimum set of prefixes, denoted as $S([d_1, d_2])$, the union of which is equal to $[d_1, d_2]$. Each prefix is a subrange of $[d_1, d_2]$, which follows the binary prefix format. For $[d_1, d_2]$, the number of prefixes in $S([d_1, d_2])$ is at most $2w-2$ [18], where d_1 and d_2 are two numbers of w bits. For example, $S([9, 15]) = \{1001, 101*, 11**\}$. We compute the prefix family $F(x)$ of number x and translate the range $[d_1, d_2]$ into a minimum set of prefixes $S([d_1, d_2])$. So, $x \in [d_1, d_2]$ if and only if $F(x) \cap S([d_1, d_2]) \neq \phi$.

In order to ensure whether$F(x) \cap S([d_1, d_2]) \neq \phi$, we use the operations of verifying whether two numbers are equal. Then, we convert each prefix to a corresponding unique number using the prefix numericalization scheme defined in [19]. A prefix numericalization function Nneeds to satisfy the following properties: (1) for any p, $N(p)$ is a binary string; (2) for any two prefixes p and q, $p = q$ if and only if $N(p) = N(q)$.Given a prefix $b_1 b_2 \cdots b_k * \cdots *$ of w bits, we insert 1 after b_k, then every $*$ is replaced by 0. Given a set of prefixes S, the resulting set of numericalized prefixes is denoted as $N(S)$. For example, $N(F(11)) = \{10111, 10110, 10100, 11000, 10000\}$ and $N(S([9, 15])) = \{10011, 10110, 11100\}$. Therefore, $x \in [d_1, d_2]$ if and only if $N(F(x)) \cap N(S([d_1, d_2])) \neq \phi$. For instance, $N(F(11)) \cap N(S([9, 15])) = 10110$, the number 11 is in the range [9, 15]. Figure 2 shows the process of $11 \in [9, 15]$.

4.3. Data Collection. In order to preserve sensitive data, sensor nodes send the sensed data to cluster heads and sink by a secure way. We assume that b_1 and b_2, respectively, denote the lower bound and the upper bound, the values of which are known to both sensors and the sink. And we assume that sensor s_i collects data item $d_j (1 \le j \le n)$ at a time slot t, and each data d_j is in the range $[b_1, b_2]$. When each sensor node s_i collects data, $s_i$sends the sensitive data by the following steps:

(1) Sort the n data, b_1, and b_2 in an ascending order. We assume $b_1 < d_1 < d_2 < \cdots < d_n < b_2$.

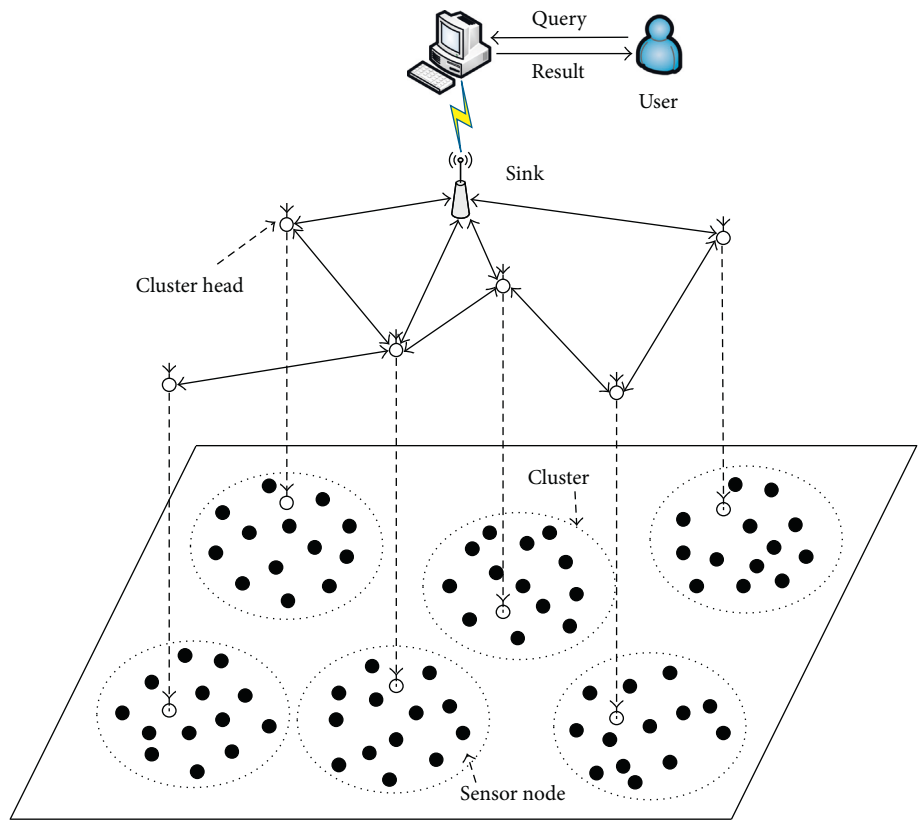

Figure 1: The cluster-based query processing scheme.

(2) Convert the $n+1$ ranges $[b_1, d_1], [d_1, d_2], \ldots, [d_n, b_2]$ to the corresponding prefix string, that is, compute $S([b_1, d_1]), S([d_1, d_2])), \ldots, S([d_n, b_2])$.

(3) Numericalize all prefixes. For example, compute $N(S([b_1, d_1])), N(S([d_1, d_2])), \ldots, N(S([d_n, b_2]))$.

(4) Compute the keyed-hash message authentication code (HMAC) [6, 20] of each data item in numericalize prefixes using key g, which is shared by all nodes and the sink. An HMAC function using key g is denoted as HMAC_g. Compute $\text{HMAC}_g(N(S([b_1, d_1]))), \text{HMAC}_g(N(S([d_1, d_2]))), \ldots, \text{HMAC}_g(N(S([d_n, b_2]))).$

(5) Encrypt every data item d_i to $(d_i)_{k_i}$ using key k_i.

(6) Sensor s_i sends the following packet to its cluster head (CH): $s_i \longrightarrow \text{CH} : \langle \text{id}, t, (d_i)_{k_i}, \text{HMAC}_g(N(S(b_1, d_1))), \text{HMAC}_g(N(S(d_1, d_2))), \ldots, \text{HMAC}_g(N(S(d_n, b_2)))\rangle$.

Because the HMAC function has the one-wayness and collision resistance properties, and data items are encrypted, the cluster head cannot obtain the real values of all data items.

4.4. Filter and Query Processing. When a cluster head receives collected packets from cluster members in a cluster, the cluster head will filter the packet by sensors' id. In the query phase, the cluster head randomly selects several cluster members which include the real queried node s_i and gain the sensed data from them. It is aimed for preventing the adversary from monitoring the real frequency of query in a cluster. Therefore, in the submission phase, the cluster head needs to filter out the useless packets and obtain the real packet. Then, the cluster head transmits the real packet to the sink.

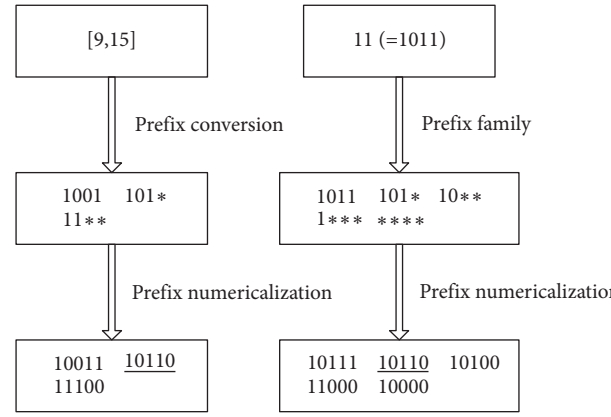

Figure 2: Prefix membership verification.

In the sink, it firstly converts the query range $[a, b]$ and computes prefix families $F(a)$ and $F(b)$. After the sink numericalize all prefixes as $N(F(a))$ and $N(F(b))$, it applies HMAC_g to each numericalized prefix as $\text{HMAC}_g(N(F(a)))$ and $\text{HMAC}_g(N(F(b)))$. When the sink receives a packet from cluster heads, it will process the packet based on the query range $[a, b]$ using the following theorem [21].

Theorem 1. *Given n numbers sorted in the ascending order* $d_1 < d_2 < \cdots < d_n$, *where* $d_j \in [b_1, b_2]\,(1 \le j \le n)$, *and a range* $[a, b]\,(b_1 < a \le b < b_2)$, $d_j \in [a, b]$ *if and only if there exist* $1 \le n_1 \le j < n_2 \le n + 1$ *such that the following two conditions hold:*

(1) $\mathrm{HMAC}_g(N(F(a))) \cap \mathrm{HMAC}_g(N(S([d_{n_1-1}, d_{n_1}]))) \neq \phi$

(2) $\mathrm{HMAC}_g(N(F(a))) \cap \mathrm{HMAC}_g(N(S([d_{n_2-1}, d_{n_2}]))) \neq \phi$

According to Theorem 1, the sink selects the smallest n_1 and the largest n_2 $(1 \le n_1, n_2 \le n + 1)$ such that $a \in [d_{n_1-1}, d_{n_1}]$ and $b \in [d_{n_2-1}, d_{n_2}]$. If $n_1 < n_2$, the data items d_{n_1}, $d_{n_1+1}, \ldots, d_{n_2-1}$ are in the range $[a, b]$; if $n_1 = n_2$, no data item is in the range $[a, b]$.

Based on the aforementioned description, Algorithm 1 shows a secure and efficient cluster-based query processing scheme. When a user wants to gain and check whether sensed data of certain node is in the range $[a, b]$ at a time slot t, the user will send a query message to the sink. And then the sink relays the message to a cluster head which include the queried node. In the cluster head, it randomly chooses several nodes which include the queried node. In the collection and submission phase, when the node collects the sensed data, it will process the data using the PMV and HMAC schemes. Then, the node sends the secure packet to the cluster head. After the cluster head filters out the useless packets, it sends the useful packet to the sink. Finally, the sink processes the packet and sends the final result to the user.

5. Performance Analysis

In order to protect privacy, we propose a secure and efficient query processing scheme to prevent an adversary from obtaining the sensitive data or finding the user's interests and location of sensor node in cluster-based sensor networks. In this section, we present the privacy analysis and communication overhead analysis. From the following analysis, we can see that our scheme brings a better network security and minimal communication overhead.

5.1. Privacy Analysis. For privacy of collected data, according to the data collection phase, sensor nodes convert the collected data by using encryption and HMAC scheme. So, the submitted information is not plaintext but encrypted and HMAC data. The HMAC function has one-wayness and collision resistance properties. And sensor nodes only share the secret key with the sink and encrypt sensitive data by the key. Therefore, it is computationally infeasible for cluster heads to obtain the value of d_i. It is difficult for the cluster head to break the privacy and gain the encryption and HMAC data. So, our scheme can efficiently protect collected data items.

For privacy of the query result, the sink obtains the query result by comparing the HMAC data items. For the HMAC data items and encrypted data, it is difficult for the adversary for computing and obtaining the values of the query result without keys. So, we can preserve the query result which is securely transmitted to the user.

For privacy of user's interests and location privacy of sensors in clusters, our scheme can efficiently preserve privacy information to prevent an adversary from monitoring user's interests and find the location information of sensors. And the adversary cannot use the content to trace the routing. We assume that an adversary monitors a local area with the intention of finding the interests of the user. We assume that each cluster has M members. The adversary wants to identify a set $D_T \subset M$ of nodes which represent the set of possible location in the local area. There is a close relationship between the analysis of query frequency of the adversary and the location privacy. When the adversary analyzes the query frequency uncertainly, it is secure to preserve the location information. In the eavesdropping area, the adversary will need to select the nodes of his analysis. We assume that the possible sensor nodes in D_T include queried nodes which send data to cluster head. If the size of D_T is very large, the adversary will find it difficult to analyze the user's interests. So, it is useful for preserving privacy information.

Let D_P be the set of the protected nodes. We use information-theoretic metric, called entropy [22], to measure the privacy protection provided by our scheme. The entropy of identifying the queried node in the wireless sensor network is defined as

$$c = -\sum_{i=1}^{D_\mathrm{T}} P_i \cdot \log_2(P_i), \tag{1}$$

where P_i is the probability that node i is the queried node, $|D_\mathrm{T}|$ is the number of uncertain nodes by the adversary, and $\sum_{i=1}^{|D_\mathrm{T}|} P_i = 1$. Therefore, the probability P_i of any sensor nodes in D_T being queried nodes can be estimated by $|D_\mathrm{P}|/|D_\mathrm{T}|$. Then, we denote the size of D_T as $R(|D_\mathrm{T}| = R)$. And let r be the size of the protected queried node's set $(|D_\mathrm{P}| = r)$. And we define the privacy as

$$\begin{aligned} c &= -\sum_{i=1}^{|D_\mathrm{T}|} \frac{|D_\mathrm{P}|}{|D_\mathrm{T}|} \cdot \log_2\left(\frac{|D_\mathrm{P}|}{|D_\mathrm{T}|}\right) \\ &= -\sum_{i=1}^{R} \frac{r}{R} \cdot \log_2\left(\frac{r}{R}\right) \\ &= r \cdot \log_2\left(\frac{R}{r}\right). \end{aligned} \tag{2}$$

The entropy represents the adversary's uncertainty about the user's interests and the location of sensors in a wireless sensor network. When the adversary believes that the nodes have the same probability to be the queried node in a cluster, the entropy is maximum value. Let D_T^* be the set of all nodes in a cluster and $|D_\mathrm{T}^*| = M$. We know that the size of D_T can influence the level of privacy. Therefore, the entropy is

(1) initial_cluster;
(2) sink_node;
(3) query_node_id;
(4) CH = cluster_head;
(5) query_range = [a,b];
(6) query_msg = QueryMsg(query_node_id, sink_node, query_range);
(7) **if** sink receives the query_msg **then**
(8) SendQueryMsgToCH(CH, query_node_id);
(9) **if** CH receives the query_msg **then**
(10) selected_nodes = SelectNodesRandom(query_node_id);
(11) **for** node in selected_nodes **then**
(12) d_i = CollectData();
(13) Sort d_i,b_1 and b_2 as $b_1 < d_1 < d_2 < \cdots < d_n < b_2$
(14) Compute prefixes and numericalize all prefixes as $N(S([b_1, d_1])), N(S([d_1, d_2])), \ldots, N(S([d_n, b_2]))$
(15) Compute the HMAC as $\mathrm{HMAC}_g(N(S([b_1, d_1]))), \mathrm{HMAC}_g(N(S([d_1, d_2]))), \ldots, \mathrm{HMAC}_g(N(S([d_n, b_2])))$
(16) Encrypt d_i by k_i as $(d_i)_{k_i}$;
(17) Send packet to CH $s_i \longrightarrow$ CH : $\langle \mathrm{id}, t, (d_i)_{k_i}, \mathrm{HMAC}_g(N(S(b_1, d_1))), \mathrm{HMAC}_g(N(S(d_1, d_2))), \ldots, \mathrm{HMAC}_g(N(S(d_n, b_2)))\rangle$
(18) **end for**
(19) packet = FilterPackets();
(20) SendToSink(packet);
(21) **end if**
(22) Compute and encode [a,b] as $\mathrm{HMAC}_g(N(F(a))), \mathrm{HMAC}_g(N(F(b)))$;
(23) GetQueryData(packet, $\mathrm{HMAC}_g(N(F(a))), \mathrm{HMAC}_g(N(F(b)))$);
(24) **end if**

Algorithm 1: Secure and efficient cluster-based query processing.

$$
\begin{aligned}
c &= -\sum_{i=1}^{|D_T|} \frac{|D_P|}{|D_T|} \cdot \log_2\left(\frac{|D_P|}{|D_T|}\right) \\
&= |D_P| \cdot \log_2\left(\frac{|D_T|}{|D_P|}\right) \\
&\le |D_P| \cdot \log_2\left(\frac{|D_T^*|}{|D_P|}\right) \\
&= r \cdot \log_2\left(\frac{M}{r}\right).
\end{aligned} \tag{3}
$$

Figure 3 shows the relationship between the level of privacy and the different number of nodes in a cluster. When M increases, the level of privacy (c) is higher. This is due to the increased number of nodes in a cluster. The probability that the adversary finds the protected queried nodes is decreased. So the level of privacy (c) increases. For the same M, when M is greater than 20, we can see that the number of protected queried nodes (r) increases, the level of privacy (c) increases.

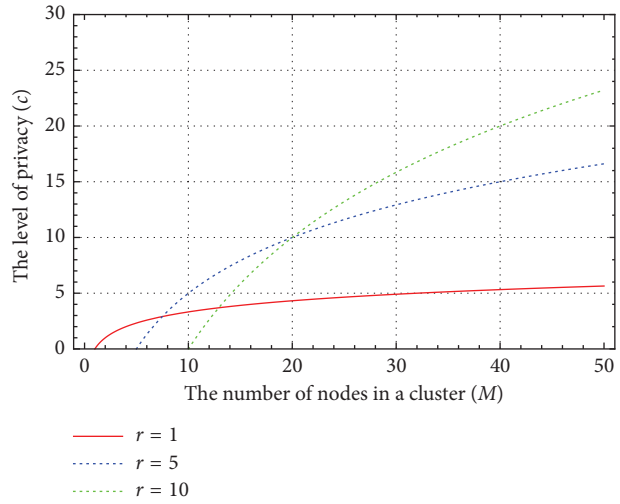

Figure 3: The different number of nodes in a cluster.

5.2. Energy Consumption Analysis. In cluster-based sensor networks, sensor nodes have limited energy resource. In this section, we discuss the energy consumption of sensor nodes in our scheme. In each phase, the total energy consumption includes the communication cost and computation energy consumption. We assume that the energy consumed by transmitting and receiving a data are e_t and e_r. And we assume that a cluster head randomly chooses N_r nodes to query.

In the data collection phase, sensor nodes will have extra computation overhead to preserve privacy of sensitive data. Given a range $[d_1, d_2]$, where d_1 and d_2 are two numbers of w bits, the number of prefixes in $S([d_1, d_2])$ is at most $2w - 2$ [18]. So a sensor computes at

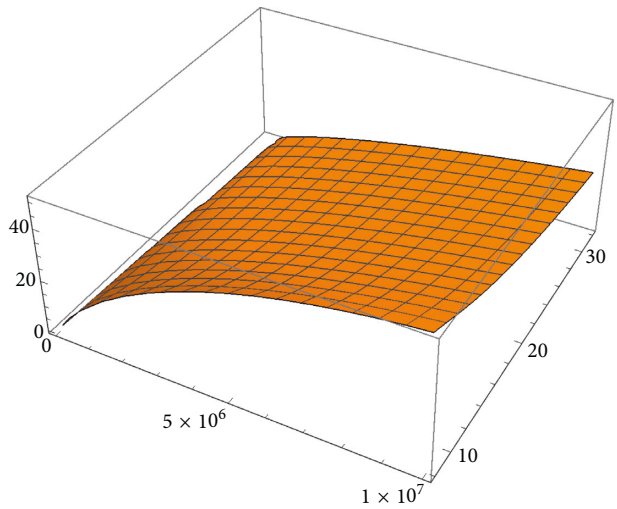

Figure 4: The relationship between energy consumption and the level of privacy.

most $(n+1)(2w-2)$ HMAC data. When a sensor node sends data to cluster head, it generates extra communication cost by sending encrypted data and HMAC data. We assume that each HMAC data is z_H bits and encrypted data is z_D bits. Let H_{hop} be the hop between a sensor node and a cluster head. In our scheme, $H_{hop} = 1$. In a cluster, the energy consumption E_{dc} is

$$\begin{aligned} E_{dc} &= \sum_{i=1}^{N_r} \left[(n+1)(2w-2)\cdot z_H + z_D + e_t + e_r\right]\cdot H_{hop} \\ &= N_r \cdot \left[(n+1)(2w-2)\cdot z_H + z_D + e_t + e_r\right]. \end{aligned} \tag{4}$$

In the query processing phase, the sink node computes the range $[a, b]$ and converts prefix families $F(a)$ and $F(b)$. For each value in the w bits, there are $w+1$ HMAC data items. So, the sink can perform at most $2(n+1)(2w-2)(w+1)$ comparisons. The energy consumption E_{qp} is

$$\begin{aligned} E_{qp} &= \left[2(n+1)(2w-2)(w+1) + e_t + e_r\right]\cdot H_{hop} \\ &= 2(n+1)(2w-2)(w+1) + e_t + e_r. \end{aligned} \tag{5}$$

Therefore, the total energy consumption E_{total} is

$$\begin{aligned} E_{total} &= E_{dc} + E_{qp} \\ &= N_r \cdot \left[(n+1)(2w-2)\cdot z_H + z_D + e_t + e_r\right] \\ &\quad + \left[2(n+1)(2w-2)(w+1) + e_t + e_r\right]. \end{aligned} \tag{6}$$

According to (3) and (6), Figure 4 shows the total energy expended in the systems as the prefix number bits increase from 8 bits to 32 bits and the energy expended in the level of privacy increases from 1 to 45, for the scenario where each HMAC data and encrypted data are 256 bits. We assume that each sensor collects 100 data items at each time slot. And we assume that the energy consumed by transmitting and receiving a data are 1. Each cluster head includes 100 sensor nodes. This shows that when the prefix number bits is the same, the higher level of privacy can increase the energy of the whole sensor network.

6. Conclusions

Wireless sensor networks have been widely deployed in many applications and drawn more attentions. It is an important problem to preserve the privacy of sensitive data in cluster-based query processing in wireless sensor networks. In this paper, we propose a secure and efficient scheme to protect query processing in cluster-based sensor networks. In order to preserve privacy, sensed data items are encrypted to prevent cluster heads from obtaining the content of data. We use the prefix membership verification method to query the result without plaintext data. Meanwhile, we use anonymity method to confuse adversaries and prevent adversaries from analyzing the user's interests and finding location of the queried node. Then, we perform the privacy analysis and energy consumption analysis.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper and confirm that the mentioned received funding in the acknowledgments did not lead to any conflicts of interest regarding the publication of this manuscript.

Acknowledgments

This work was supported by NSFC (Grant no. 61402015), the Science and Technology Development Plan Project of Henan Province (Grant no. 172102210189), and the Research Fund Project of Henan University (Grant no. 2016YBZR019).

References

[1] D. Gong, Y. Yang, and Z. Pan, "Energy-efficient clustering in lossy wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 73, no. 9, pp. 1323–1336, 2013.

[2] B. Carbunar, Y. Yu, L. Shi et al., "Query privacy in wireless sensor networks," in *Proceedings of 4th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks*, pp. 203–212, San Diego, CA, USA, 2007.

[3] H. Dai, Q. Ye, G. Yang et al., "CSRQ: communication-efficient secure range queries in two-tiered sensor networks," *Sensors*, vol. 16, no. 2, pp. 1–17, 2016.

[4] Y. T. Tsou, C. S. Lu, and S. Y. Kuo, "SER: Secure and efficient retrieval for anonymous range query in wireless sensor networks," *Computer Communications*, vol. 108, pp. 1–16, 2017.

[5] H. Dai, Q. Ye, X. Yi et al., "VP(2)RQ: efficient verifiable privacy-preserving range query processing in two-tiered wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 12, no. 11, pp. 1–15, 2016.

[6] F. Chen and A. X. Liu, "SafeQ: secure and efficient query processing in sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications*, pp. 1–9, Piscataway, NJ, USA, 2010.

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

[8] H. Wu and L. Wang, "Efficient and secure top-k query processing on hybrid sensed data," *Mobile Information Systems*, vol. 2016, Article ID 1685054, 10 pages, 2016.

[9] J. Cheng, H. Yang, S. H. Wong, 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, Piscataway, NJ, USA, October 2007.

[10] A. X. Liu and F. Chen, "Collaborative enforcement of firewall policies in virtual private networks," in *Proceedings of the 27th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pp. 95–104, New York, NY, USA, August 2008.

[11] X. Zhang, L. Dong, H. Peng et al., "Collusion-aware privacy-preserving range query in tiered wireless sensor networks," *Sensors*, vol. 14, no. 12, pp. 23905–23932, 2014.

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

[13] 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.

[14] 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 814892, 2013.

[15] Y. Yao, N. Xiong, J. H. Park, L. Ma, and J. Liu, "Privacy-preserving max/min query in two-tiered wireless sensor networks," *Computers and Mathematics with Applications*, vol. 65, no. 9, pp. 1318–1325, 2012.

[16] H. Dai, T. Wei, Y. Huang et al., "Random secure comparator selection based privacy-preserving max/min query processing in two-tiered sensor networks," *Journal of Sensors*, vol. 2016, Article ID 6301404, 13 pages, 2016.

[17] H. Dai, M. Wang, X. Yi et al., "Secure max/min queries in two-tiered wireless sensor networks," *IEEE Access*, vol. 5, pp. 14478–14489, 2017.

[18] P. Gupta and N. McKeown, "Algorithms for packet classification," *IEEE Network*, vol. 15, no. 2, pp. 24–32, 2001.

[19] Y. K. Chang, "Fast binary and multiway prefix searches for packet forwarding," *Computer Networks*, vol. 51, no. 3, pp. 588–605, 2007.

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

[21] F. Chen and A. X. Liu, "Privacy and integrity preserving range queries in sensor networks," Tech. Rep. MSU-CSE-09–26, Michigan State University, East Lansing, MI, USA, 2009.

[22] M. M. E. A. Mahmoud and X. Shen, "A cloud-based scheme for protecting source-location privacy against hotspot-locating attack in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1805–1818, 2012.