SET: Secure and Efficient Top-k Query in Two-tiered Wireless Sensor Networks

Xiaoying Zhang[1,2], Hui Peng[3], Lei Dong[1,2], Hong Chen[1,2], and Hui Sun[1,2⋆]

[1] School of Information, Renmin University of China, Beijing, 100872, China,
[2] Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education, Beijing, China,
[3] The Fifth Electronic Research Institute of MIIT, Guangzhou, 510000, China

Abstract. Top-k query is one of important queries in wireless sensor networks (WSNs). It provides the k highest or lowest data collected in the entire network. Due to abundant resources and high efficiency, many future large-scale WSNs are expected to follow a two-tiered architecture with resource-rich master nodes. However, the sensor network is unattended and insecure. Since master nodes store data collected by sensor nodes and respond to queries issued by users, they are attractive to adversaries. It is challenging to process top-k query while protecting sensitive data from adversaries. To address this problem, we propose SET, a framework for secure and efficient top-k query in two-tiered WSNs. A renormalized arithmetic coding is designed to enable each master node to obtain exact top-k result without knowing actual data. Besides, a verification scheme is presented to allow the sink to detect compromised master nodes. Finally, theoretical analysis and experimental results confirm the efficiency, accuracy and security of our proposal.

Keywords: Top-k query, Wireless sensor networks, Privacy preservation, Integrity verification

1 Introduction

As the indispensable building block for Internet of Things, wireless sensor networks (WSNs) have been widely used in many applications, including smart home, e-health and environment monitoring. In these applications, top-k query is an important query which interests users. It is to seek for the k highest or lowest data in the sensor network, which is meaningful for monitoring extreme conditions. However, due to the openness and non-supervision of WSNs, severe privacy problems have been exposed in practical applications. For instance, in smart homes, sensors and actuators are distributed in houses to collect information and make our lives more comfortable. The power provider wants to know

⋆ Corresponding author. This work is supported by National Science Foundation of China (Grant No.61532021), National Basic Research Program of China (973 Program) (No.2014CB340403) and National High Technology Research and Development Program of China (863 Program) (No.2014AA015204).

which are the k highest electricity consumption of a community in the peak hours. During the process of data transmission and aggregation, household information may be overheard by an adversary such that privacy of every resident leaks. Therefore, privacy-preserving top-k query processing is greatly urgent.

On account of resource savings, rapid response and high scalability [16], the two-tiered architecture [10] will be adopted in future large-scale WSNs. As shown in Figure 1, a two-tiered WSN consists of a great many sensor nodes in the lower tier and a few master nodes, also called storage nodes, in the upper tier. Sensor nodes gather data while master nodes store data and return results to the sink. In spite of its advantages, this tiered network also brings difficulties in performing privacy-preserving top-k query. On the one hand, since master nodes store all data collected by sensor nodes, adversaries are apt to compromise them instead of sensor nodes. Once compromising a master node, the adversary can obtain all sensitive data, which violates **data privacy**. Moreover, the adversary can manipulate the compromised master node to submit fake or incomplete result, which breaches **result integrity**. On the other hand, master nodes are required to process queries efficiently and correctly, which is hindered if data are encrypted. As a result, it is challenging to preserve data privacy and result integrity as well as achieving efficient performance and correct result.

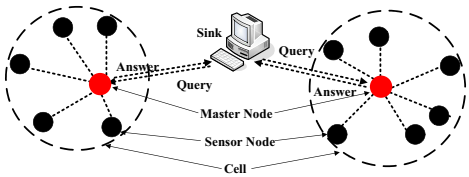


Fig. 1. Architecture of Two-tiered Wireless Sensor Networks

To address the above problem, this paper investigates secure top-k query in tiered WSNs with the following contributions:

- We first design a renormalized arithmetic coding scheme (RAC) suitable for WSNs. The RAC scheme encodes sensitive data of sensor nodes to tuples of codes which hold order-preserving property.
- We then propose SET, a framework for ***S***ecure and ***E***fficient ***T***op-k query in two-tiered WSNs. SET relies on the RAC scheme to preserve data privacy and enable master nodes to perform top-k queries efficiently and correctly without knowing actual data.
- We further present an integrity verification scheme to examine result integrity. Global correlation between data helps the sink to detect the incorrect top-k result.
- Theoretical analysis and experimental results indicate that our proposal can reduce more communication cost and save more energy while accomplishing accurate result, data privacy and result integrity.

The rest of paper is organized as follows. Section II summarizes related work. Section III describes models and requirements. Our SET framework is elaborated

in Section IV. We analyze and evaluate the proposed SET in Section V and Section VI, respectively. Finally, this paper is concluded in Section VII.

2 Related Work

Privacy-preserving query in WSNs has drawn more concerns. Existing work on privacy-preserving query mainly focuses on aggregation query, range query and top-k query. Our work focuses on top-k query. Privacy-preserving top-k query in two-tiered WSNs has been explored in [3,5,6,7,8,12,13,14,15,17].

Zhang *et al*. [14] for the first time present a verifiable fine-grained top-k query algorithm in tiered WSNs. Three schemes are proposed to verify the authenticity and completeness of result. These schemes rely on the basic idea that sensor nodes embed relationships among the data. The above work is further extended in [15] by proposing the random probing scheme, the query conversion scheme, and the random witness scheme. VSFTQ [8] sorts all data and constructs order relationships between data collected by each node. The final top-k result is verified using order relationships. Since work [8,14,15] only pays attention to authenticity and completeness of top-k result, serious privacy issues still exist.

SafeTQ [3] is a verifiable privacy-preserving top-k query algorithm in tiered WSNs. The whole network is separated into different cells. In each cell, the head node collaborates with the aided computing node to determine the k-th maximum (minimum) by using secure computation [11]. Probabilistic neighbor checking is to examine result integrity. Since its privacy preservation is ensured, the selection criterion of head nodes and aided computing nodes is not given.

Order-preserving encryption scheme is adopted in [7,12]. PriSecTopk [7] is a series of privacy-preserving top-k query mechanisms on the time slot data set in two-tier WSNs. The basic PriSecTopk use order-preserving symmetric encryption to hide original data and calculate results. To protect order-relation and distance-relation privacy, the basic PriSecTopk is improved by secret perturbation. One obvious drawback is that the result is imprecise. The sink verifies result integrity through sample-based hypothesis testing method combined with computing commitment information, and cannot detect the incomplete result completely. Work proposed in [12] transforms the input distribution to target distribution such that the final result may be also imprecise.

SVTQ [17] answers top-k queries in WSNs without leaking sensitive information. It computes two sets: the prefix family and the prefix range for each datum, and then performs prime aggregation to the numericalized prefixes. Comparison between two data is converted to checking whether their greatest common divisor equals to 1. Since there is no known efficient formula for primes, prime aggregation scheme requires sensor nodes to prestore enough prime numbers and their sequences. For instance, if the data domain is $[0, 127]$, each sensor node should prestore 512 prime numbers. Due to the limited storage space of sensor nodes, SVTQ is only suited to extremely limited domain.

ADVQ [13] can safely compute top-k result by data anonymization, but lots of extra information of neighboring nodes increases communication overhead of

both sensor nodes and master nodes, and introduces false positives. In addition, the randomized and distributed ordering preserving encryption also delays result response. ETQFD [5] supports secure top-k queries based on filters. Nevertheless, maintenance of filters in sensor nodes is costly and result integrity cannot be verified. Jonsson *et al.* [6] present a prototype system for secure distributed top-k aggregation, which is unsuitable for the tiered WSNs.

3 Models and Requirements

3.1 Network Model

The architecture of a two-tiered WSN is displayed in Figure 1. It is composed of three types of nodes: sensor nodes, master nodes and a sink. Sensor nodes are restricted in various resources, e.g., energy, storage and bandwidth. In contrast, master nodes have more but not inexhaustible resources. Only the sink has unlimited resources and powerful capabilities. Sensor nodes can communicate with their neighbors and their master nodes. Master nodes can also communicate with each other and the sink. In fact, the entire network is separated into several non-overlapping cells and each cell comprises one master node and some sensor nodes.

Sensor nodes collect data from surroundings with a specified frequency, and then periodically send data to their master nodes. The message that a sensor node s_i sends to its master node M is

$$s_i \rightarrow M : i, t, \{d_1, d_2, ..., d_\lambda\},$$

where i denotes the unique identifier of s_i, t denotes an epoch, λ is the number of sensed data and d_j $(1 \leq j \leq \lambda)$ is the data collected by s_i at epoch t. Suppose all sensors are synchronized so that they keep consistent in epoches.

The sink directly issues the top-k query $Q = \langle k, T, C \rangle$ to master nodes, where C is the set of interested cells, and k means that the user inquires the k highest data generated in area C at the time slot T. After receiving this query, master nodes in area C immediately search for proper data and submit the result to the sink. The result should contain the k highest data and the identifiers of corresponding sensor nodes. Unless otherwise stated, "top" in top-k query is defined as "highest" in this paper.

3.2 Adversary Model

As assumed in [3,5,6,7,8,12,13,14,15,17], the sink is reliable. During query processing, adversaries attempt to eavesdrop on sensitive data through the wireless link. It is unsafe to transmit data in the form of plaintext. Moreover, adversaries are more inclined to compromise master nodes than compromise sensor nodes. There are two reasons. First, master nodes store large quantity of sensitive data. Once compromising a master node, it enables the adversary to obtain all data

stored in this master node. Second, the adversary may manipulate the compromised master node to submit wrong result to the sink. It is obvious that the compromised master node leads to more harmful threats than the compromised sensor node. Therefore, as in state of the art work [3,6,7,8,12,13,14,15,17], our paper focuses on the problem of compromised master nodes.

3.3 Design Requirements

A good secure top-k query algorithm in tiered WSNs should fulfill the following requirements:

- Data privacy. Since all sensory data are private and sensitive, they should not be obtained by master nodes. Furthermore, data should only be known by its owner and the sink.
- Result integrity. Three conditions should be satisfied: (1) The number of data in the result should be equal to k; (2) All data in the result should be authentic, i.e., forged data are not permitted; (3) Any datum outside the result should not be greater than all data in the result. In other words, the result should include only data satisfying the query and exclude data dissatisfying the query. At least, the algorithm should guarantee that the sink is able to detect unauthentic or incomplete result.
- Efficiency. Communication cost is one important metric for energy consumption. It is generated during data submission from sensor nodes to master nodes and result submission from master nodes to the sink. The less communication cost, the higher efficiency.

4 Framework for Secure and Efficient Top-k Query

In this section, we propose a ***S***ecure and ***E***fficient ***T***op-k query framework (SET) for two-tiered WSNs. To prevent adversaries from knowing sensitive data, the naïve but effective method is to encrypt data using secret keys. However, it is impossible for master nodes to process the top-k query just based on encrypted data. To solve this problem, we design a renormalized arithmetic coding scheme (RAC) to encode data and preserve privacy while processing queries. In the following, we first present the RAC scheme, and then elaborate four phases of SET framework, including system initialization, data submission, result response and integrity verification. For clear description, Table 1 summarizes the notation used in this paper.

4.1 Arithmetic Coding

Arithmetic coding is one of entropy encoding for lossless data compression. It is first introduced in [9]. The key idea of Arithmetic coding is to represent the entire message by a number between 0 and 1. Figure 2 shows an example of arithmetic coding. To simplify description, assume any message is one combination of three

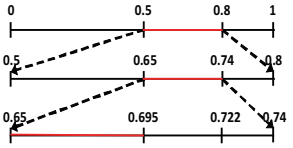
Table 1. Notation

Symbol	Meaning
k	the parameter for a top-k query
t	an epoch
T	a time slot
M	a master node
s_i	a sensor node with the unique identifier i
$k_{i,t}$	a secret key of s_i at epoch t
d	the sensory data
$E()$	a function for encryption
$\mathbb{B}()$	a function for binary conversion
$\mathbb{R}()$	a function for RAC scheme
$\mathbb{R}^0()$	the number of 0 bits before the first 1 bit in the RAC code
$\mathbb{R}^1()$	the 0-1 code beginning with the first 1 bit in the RAC code
$P(m)$	the probability of the source symbol m
$C(m)$	the cumulative distribution of the source symbol m

source symbols: A, B, C. Given a symbol m, $P(m)$ denotes its probability and $C(m)$ denotes its cumulative distribution with $C(m) = \sum_{i=1}^{m-1} P(i)$. The process of encoding a message "BBA" is detailed in 2(b). Initial interval is set to $[0, 1]$. For the first symbol "B", the interval is updated to $[0.5, 0.8]$ highlighted by red line. Then divide the new interval $[0.5, 0.8]$ into sub-intervals according to the probability and cumulative distribution. For the second symbol "B", the interval is further updated to $[0.65, 0.74]$ and divided. For the last symbol "A", the interval is finally updated to $[0.65, 0.695]$. Therefore, the message "BBA" can be represented by any number lying within the interval $[0.65, 0.695]$, e.g. 0.68.

Symbol(m)	P(m)	C(m)
A	0.5	0
B	0.3	0.5
C	0.2	0.8

(a) Symbol Distribution Table



(b) Encoding

Fig. 2. An Example of Arithmetic Coding

However, the final interval may collapse into a single point when the message is too long, which reduces the precision. Although renormalization is used to overcome this problem in [4], these algorithms are too complex to be applied to WSNs. Here we design a simple but efficient renormalized arithmetic coding (RAC) specialized for WSNs, as detailed in Scheme. 1. Assume the initial interval is set to $[0, I-1](I > 1)$, the main idea of RAC scheme is to keep the interval size

Scheme 1 Renormalized Arithmetic Coding

Input:
M : message
n : length of message
I: size of initial interval
$P(m)$: probability of symbol m
$C(m)$: cumulative distribution of symbol m
Output:
code : code of message

```
1: l ← 0
2: u ← I − 1
3: code ← null
4: count ← 0
5: for read a symbol m of M from 1 to n do
6:    r ← u − l + 1
7:    l ← l + r ∗ C(m)
8:    u ← l + r ∗ P(m) − 1
9:    r ← u − l + 1
10:   while r < I/4 do
11:      if l >= I/2 then
12:         code ← code + "1"
13:         u ← 2 ∗ u − I + 1
14:         l ← 2 ∗ l − I
15:         for from 0 to count do
16:            code ← code + "0"
17:         end for
18:         count ← 0
19:      else
20:         if u < I/2 then
21:            code ← code + "0"
22:            u ← 2 ∗ u + 1
23:            l ← 2 ∗ l
24:            for from 0 to count do
25:               code ← code + "1"
26:            end for
27:            count ← 0
28:         else
29:            u ← 2 ∗ u − I/2 + 1
30:            l ← 2 ∗ l − I/2
31:            count ← count + 1
32:         end if
33:      end if
34:      r ← u − l + 1
35:   end while
36: end for
37: if l >= I/2 then
38:   code ← code + "1"
39:   for from 0 to count do
40:      code ← code + "0"
41:   end for
42:   code ← code + "0"
43: else
44:   code ← code + "0"
45:   for from 0 to count do
46:      code ← code + "1"
47:   end for
48:   code ← code + "1"
49: end if
50: return code;
```

not smaller than $I/4$ by enlarging the interval once it becomes too small. Given a message, read each symbol m successively and update both the interval $[l, u]$ and interval size s (line 6-9). If the size of current interval is smaller than $I/4$, renormalization is required (line 10-35). There are three cases to be considered: (1) If the interval completely locates within the top half of $[0, I-1]$, which means the next interval gets closer to $I-1$ than 0, expand the interval and output a 1 followed by *count* 0s (line 11-18), where *count* is determined by the third case; (2) If the interval completely locates within the bottom half of $[0, I-1]$, which means the next interval gets closer to 0 than $I-1$, expand the interval and output a 0 followed by *count* 1s (line 19-27); (3) If the lower bound of the interval falls in the bottom half and the upper bound falls in the top half, the trend of next interval cannot be determined. Keep track the expansion by increasing *count* and expand the interval (line 28-31). After the final interval is calculated, choose a number of the interval (line 37-49). For example, the message "BBA" is encoded

to 10101. RAC scheme will be used to encode data in our SET. Now we begin to elaborate the SET framework.

4.2 System Initialization

The network is initialized before executing a task. Without loss of generality, let $[d_{min}, d_{max}]$ be the domain of data which are positive integers. Each sensor node s_i shares a seed key $k_{i,0}$ only with the sink. s_i encrypts its data by $k_{i,t}$ which is the secret key of s_i at epoch t. Let $k_{i,t} = hash(k_{i,t-1})$ and erase $k_{i,t-1}$ at epoch t. Let $\{00, 01, 10, 11\}$ be the set of source symbols. The sink constructs a symbol distribution table (SDT). SDT is a set of tuples $\langle m, P(m), C(m)\rangle$, where $P(m)$ and $C(m)$ respectively denote the probability and cumulative distribution of the symbol m. $k_{i,t}$ and SDT are preloaded in every sensor node. To avoid a brute force attack, all sensor nodes change SDT simultaneously and periodically based on the initial SDT and an agreed rule.

4.3 Data Submission

In data submission, we describe how a sensor node processes its data before sending them to the closest master node. Assume $d_1, d_2, ..., d_\lambda$ are the data collected by sensor node s_i at epoch t. s_i processes data in the following five steps:

- Step1. Sort $d_1, d_2, ..., d_\lambda$ in descending order with $d_1 > d_2 > ... > d_\lambda$.
- Step2. Convert each d_i to a binary form. For example, the integer 21 is converted to 10101. Assume the domain is $[0, 255]$, at least 8-bits (always even) is required to represent any value in this domain. In order to use the proposed RAC scheme, insert 000 to 10101 and shift 10101 to the right, i.e., 00010101. Let $\mathbb{B}()$ denote the binary conversion, and d_i is converted to $\mathbb{B}(d_i)$.
- Step3. Encode each $\mathbb{B}(d_i)$ using the RAC scheme (presented in Section IV.A) and the symbol distribution table SDT. For instance, assume $I = 100$ and 0.4, 0.3, 0.2, 0.1 respectively denote the probability of source symbols 00, 01, 10, 11, 00010101 is encoded to 0011101. Let $\mathbb{R}()$ denote the RAC scheme, and $\mathbb{B}(d_i)$ is encoded to $\mathbb{R}(\mathbb{B}(d_i))$.
- Step4. Represent each $\mathbb{R}(\mathbb{B}(d_i))$ by two parts: $\mathbb{R}^0(\mathbb{B}(d_i))$ and $\mathbb{R}^1(\mathbb{B}(d_i))$, where $\mathbb{R}^0(\mathbb{B}(d_i))$ denotes the number of 0 bits before the first 1 bit in $\mathbb{R}(\mathbb{B}(d_i))$ and $\mathbb{R}^1(\mathbb{B}(d_i))$ denotes the rest code of $\mathbb{R}(\mathbb{B}(d_i))$ excluding $\mathbb{R}^0(\mathbb{B}(d_i))$. For example, $\langle 2, 11101\rangle$ represents 0011101. Therefore $\mathbb{R}(\mathbb{B}(d_i))$ is represented by $\langle \mathbb{R}^0(\mathbb{B}(d_i)), \mathbb{R}^1(\mathbb{B}(d_i))\rangle$.
- Step5. Construct a structure $\langle f_i|d_i|b_i\rangle$ for each d_i, where $f_i = d_{i-1} - d_i$ ($f_1 = d_{max}$ for d_1) and $b_i = d_i - d_{i+1}$ ($b_\lambda = d_\lambda + 1$ for d_λ), which are useful for integrity verification. Then encrypt $\langle f_i|d_i|b_i\rangle$ using the secret key $k_{i,t}$. Let $\mathbb{E}()$ denote the encryption function, i.e., $\mathbb{E}(f_i|d_i|b_i)$.

After these five steps, the message that sensor node s_i submits to its master node M is

$$s_i \rightarrow M : i, t, \{\mathbb{E}(f_1|d_1|b_1), \mathbb{E}(f_2|d_2|b_2), ..., \mathbb{E}(f_\lambda|d_\lambda|b_\lambda)\},$$

$$\{\langle \mathbb{R}^0(\mathbb{B}(d_1)), \mathbb{R}^1(\mathbb{B}(d_1))\rangle, \langle \mathbb{R}^0(\mathbb{B}(d_2)), \mathbb{R}^1(\mathbb{B}(d_2))\rangle,$$
$$..., \langle \mathbb{R}^0(\mathbb{B}(d_\lambda)), \mathbb{R}^1(\mathbb{B}(d_\lambda))\rangle\}.$$

Without knowing correct SDT, RAC scheme and secret key $k_{i,t}$, even given $\mathbb{E}(f_i|d_i|b_i)$ and $\langle \mathbb{R}^0(\mathbb{B}(d_i)), \mathbb{R}^1(\mathbb{B}(d_i))\rangle$, the master node cannot obtain actual data.

4.4 Result Response

In result response, we concern how a master node searches for data required by a top-k query without knowing about actual data. Receiving a top-k query denoted as $Q = \langle k, T, C\rangle$, the master node M in C begins to process this query on all data transmitted from its affiliated sensor nodes at epoch $t \in T$. Given two positive integers d_i and d_j, let $\langle \mathbb{R}^0(\mathbb{B}(d_i)), \mathbb{R}^1(\mathbb{B}(d_i))\rangle$ and $\langle \mathbb{R}^0(\mathbb{B}(d_j)), \mathbb{R}^1(\mathbb{B}(d_j))\rangle$ be calculated according to the steps of data submission (in Section IV.C). If d_i and d_j are collected by the same sensor node at the same epoch, it is easily to determine which is the greater one according to the descending order of sorted data. If d_i and d_j are collected by different sensor nodes or at different epoches, there are three cases of comparison between d_i and d_j based on $\langle \mathbb{R}^0(\mathbb{B}(d_i)), \mathbb{R}^1(\mathbb{B}(d_i))\rangle$ and $\langle \mathbb{R}^0(\mathbb{B}(d_j)), \mathbb{R}^1(\mathbb{B}(d_j))\rangle$.

1) $\mathbb{R}^0(\mathbb{B}(d_i)) \neq \mathbb{R}^0(\mathbb{B}(d_j))$. If $\mathbb{R}^0(\mathbb{B}(d_i)) > \mathbb{R}^0(\mathbb{B}(d_j))$, there is $d_i < d_j$. For instance, integers 21 and 25 are encoded to $\langle 2, 11101\rangle$ and $\langle 1, 1000001\rangle$, respectively. Because $\mathbb{R}^0(\mathbb{B}(21))=2 > \mathbb{R}^0(\mathbb{B}(25))=1$, the master node knows encrypted 21 is less than encrypted 25.
2) $\mathbb{R}^0(\mathbb{B}(d_i)) = \mathbb{R}^0(\mathbb{B}(d_j))$ and $|\mathbb{R}^1(\mathbb{B}(d_i))| = |\mathbb{R}^1(\mathbb{B}(d_j))|$. If $\mathbb{R}^1(\mathbb{B}(d_i)) > \mathbb{R}^1(\mathbb{B}(d_j))$, there is $d_i > d_j$. For example, integers 21 and 20 are encoded to $\langle 2, 11101\rangle$ and $\langle 2, 11011\rangle$, respectively. There is $\mathbb{R}^0(\mathbb{B}(21)) = \mathbb{R}^0(\mathbb{B}(20)) = 2$ and $|\mathbb{R}^1(\mathbb{B}(21))| = |\mathbb{R}^1(\mathbb{B}(20))| = 5$. Because $\mathbb{R}^1(\mathbb{B}(21)) = 11101 > \mathbb{R}^1(\mathbb{B}(20)) = 11011$, the master node knows encrypted 21 is greater than encrypted 20.
3) $\mathbb{R}^0(\mathbb{B}(d_i)) = \mathbb{R}^0(\mathbb{B}(d_j))$ and $|\mathbb{R}^1(\mathbb{B}(d_i))| \neq |\mathbb{R}^1(\mathbb{B}(d_j))|$. Without loss of generality, assume $|\mathbb{R}^1(\mathbb{B}(d_i))| > |\mathbb{R}^1(\mathbb{B}(d_j))|$, insert $|\mathbb{R}^1(\mathbb{B}(d_i))| - |\mathbb{R}^1(\mathbb{B}(d_j))|$ 0 bits to $\mathbb{R}^1(\mathbb{B}(d_j))$ and shift $\mathbb{R}^1(\mathbb{B}(d_j))$ to the left. Now this case is equivalent to the second one. For instance, integers 21 and 22 are encoded to $\langle 2, 11101\rangle$ and $\langle 2, 111011\rangle$, respectively. There is $\mathbb{R}^0(\mathbb{B}(21)) = \mathbb{R}^0(\mathbb{B}(22)) = 2$ and $|\mathbb{R}^1(\mathbb{B}(21))| = 5 < |\mathbb{R}^1(\mathbb{B}(22))| = 6$. Convert $\mathbb{R}^1(\mathbb{B}(21))$ to 111010. As $111010 < 111011$, the master node knows encrypted 21 is less than encrypted 22.

Adopting the above scheme, master node M finds the highest k data $d'_1, d'_2, ..., d'_k$ respectively collected by sensor nodes $s'_1, s'_2, ..., s'_k$, and then responds to the sink with the message as follows:

$$M \rightarrow \ the\ sink : \{\langle s'_i, \mathbb{E}(f'_i|d'_i|b'_i)\rangle \,|d'_i \in \mathbb{D}, s'_i \in \mathbb{S}\},$$
$$\{\langle s_j, \mathbb{E}(f_1|d_1|b_1)\rangle \,|s_j \notin \mathbb{S}\},$$

where $\mathbb{D}=\{d'_1, d'_2, ..., d'_k\}$, $\mathbb{S}=\{s'_1, s'_2, ..., s'_k\}$, $\{\langle s'_i, \mathbb{E}(f'_i|d'_i|b'_i)\rangle \,|d'_i \in \mathbb{D}, , s'_i \in \mathbb{S}\}$ denotes the local top-k result set, and $\{\langle s_j, \mathbb{E}(f_1|d_1|b_1)\rangle \,|s_j \notin \mathbb{S}\}$ denotes the verification set. For the sensor node $s_j \notin \mathbb{S}$, M still needs to submit the first encrypted part $\mathbb{E}(f_1|d_1|b_1)$ of s_j for result verification. It can be seen that although the master nodes know nothing about actual data, they still get the correct result.

4.5 Integrity Verification

If each master node in area C submits its local top-k result honestly, the sink can calculate the exact final result on the basis of all local results. However, master nodes are easily to become the target of attacks. A compromised master node may return fake or incomplete result. Therefore, it is necessary for the sink to verify result integrity.

Suppose the sink receives a message:

$\{\langle s'_i, \mathbb{E}(f'_i|d'_i|b'_i)\rangle \,|d'_i \in \mathbb{D}, s'_i \in \mathbb{S}\}, \{\langle s_j, \mathbb{E}(f_1|d_1|b_1)\rangle \,|s_j \notin \mathbb{S}\}$. To simplify explanation, we let

$RS = \{\langle s'_i, \mathbb{E}(f'_i|d'_i|b'_i)\rangle \,|d'_i \in \mathbb{D}, s'_i \in \mathbb{S}\}$ and $VS = \{\langle s_j, \mathbb{E}(f_1|d_1|b_1)\rangle \,|s_j \notin \mathbb{S}\}$. The result is considered to be correct if and only if the following five conditions are all satisfied: (1) The sink can decrypt any encrypted parts $\mathbb{E}(f|d|b)$ correctly; (2) $|RS| = k$; (3) $\forall d(d \in RS)$ and $d'(d' \in VS)$, there must be $d \geq d'$; (4) $\forall d(d \in RS)$ with its corresponding f, there must be $d + f \in RS$; (5) $\forall d(d \in RS)$ with its corresponding b, $\exists d' \in RS$, if $d - b \geq d'$, there must be $d - b \in RS$. Condition (1) verifies the authenticity while other conditions verify the completeness. More analysis will be detailed in Section V. If all local results are valid, the sink further calculates the final top-k result.

5 Analysis

5.1 Privacy Analysis

In this paper, we only focus on the case where master nodes are compromised, because master nodes are more attractive to adversaries and it is difficult to protect sensor nodes from being compromised unless the hardware progresses.

If a master node M is compromised, M will attempt to obtain the real data stored in it through either the cyphertexts or the corresponding RAC codes. To decrypt the cyphertext, M has to get the correct key. Assume the length of the key is l_k, the probability that M guesses the correct key is 2^{-l_k}. The probability is negligible when l_k is large enough. To determine the mapping relationship between the real data and the RAC codes, M has to obtain the exact SDT. Assume SDT contains Y source symbols and P_i denotes the probability of the i-th symbol. There are infinite combinations of SDT satisfying $\sum_{i=1}^{Y} P_i = 1$, so the probability that M guesses the exact SDT is extremely tiny. Besides, keys and SDT vary constantly, which make the inference more difficult.

Therefore, even though master nodes are compromised, data privacy is well protected by our proposal.

5.2 Integrity Analysis

The local top-k result of a master node should satisfy the five conditions mentioned in Section IV.E. Now we begin to give the detailed integrity analysis.

We also assume the sink receives a message $\{\langle s'_i, \mathbb{E}(f'_i|d'_i|b'_i)\rangle \,|d'_i \in \mathbb{D}, s'_i \in \mathbb{S}\}, \{\langle s_j, \mathbb{E}(f_1|d_1|b_1)\rangle \,|s_j \notin \mathbb{S}\}$ from a master node M, let $RS = \{\langle s'_i, \mathbb{E}(f'_i|d'_i|b'_i)\rangle \,|d'_i \in \mathbb{D}, s'_i \in \mathbb{S}\}$ and $VS = \{\langle s_j, \mathbb{E}(f_1|d_1|b_1)\rangle \,|s_j \notin \mathbb{S}\}$. The sink is able to detect the unauthentic and incomplete result as follows:

1) If $\mathbb{E}(f'_i|d'_i|b'_i)$ of sensor node s'_i cannot be decrypted by the secret key $k_{i,t}$ shared with the sink, the sink can know the data is unauthentic. Because only data encrypted by the correct key is able to be decrypted by the same key.
2) If $|RS| \neq k$, the sink can detect this obvious error. Because k is much less than the number of sensor nodes such that exact k data must be included in RS.
3) If $\exists d_1$ collected by sensor node $s_j \notin \mathbb{S}$ and $d'_i \in RS$ satisfying: $d_1 > d'_i$, the sink is able to know the local result is incorrect for the reason that d_1 should be contained in the local top-k result as long as d'_i ($d'_i < d_1$) exists in this result.
4) If $\exists d'_i \in RS$ with its corresponding $f'_i (f'_i \neq d_{max})$ satisfying: $(d'_i + f'_i) \notin RS$, the sink can also find this error. Similar to the second case, $d'_i + f'_i$ ($d'_i + f'_i > d'_i$) should belong to the local top-k result.
5) If $\exists d'_i \in RS$ with its corresponding $b'_i (b'_i \neq d'_i + 1)$ and $d'_j \in RS$ satisfying: $d'_i - b'_i > d'_j$ and $(d'_i - b'_i) \notin RS$, the sink can still detect this error. Similar to the second case, $d'_i - b'_i$ should exist in the local top-k result.

Therefore, our integrity verification scheme is effective to detect the forged and incomplete result.

6 Performance Evaluation

In this section, we thoroughly evaluate the performance of the proposed SET by comparing with the state-of-the-art work—ADVQ [13], SVTQ [17] and PriSecTopk [7] in terms of communication cost and result accuracy.

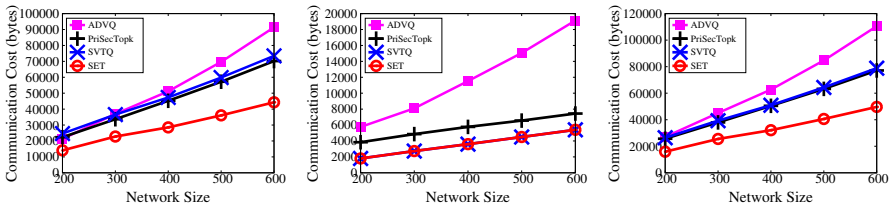
6.1 Experiment Setup

SET, ADVQ, SVTQ and PriSecTopk are implemented on the simulator OMNet++4.1 [2]. The area of sensor network is set to $400m \times 400m$. 200 more sensor nodes are uniformly deployed in the network. Assume the network is separated into four identical cells and a master node is placed at the center of each cell. The transmission radius of each sensor node is 50 meters. We build the dataset by randomly selecting different data from a real dataset LUCE[1].

For SET, ADVQ, SVTQ and PriSecTopk, we adopt the typical symmetrical encryption technique 64-bit DES to encrypt private data. For ADVQ and

PriSecTopk, we suppose the size of both $HMAC$s and MACs is 64 bits. In our SET, the parameters of SDT for RAC scheme are set as follows: the size of initial interval $I = 1000$ and the set of source symbols is $\{00, 01, 10, 11\}$ with probability $\{0.4, 0.3, 0.2, 0.1\}$ and cumulative distribution {0,0.4,0.7,0.9}.

In the experiments, there are two parameters, that is network size and submission period. Network size is defined as the number of sensor nodes in the whole network, ranging from 200 to 600. Assume each sensor node samples a datum every 10 seconds, then submission period is defined as the interval time between two successive submissions, ranging from 10 to 50 seconds.



(a) Sensor Nodes (b) Master Nodes (c) Whole Network

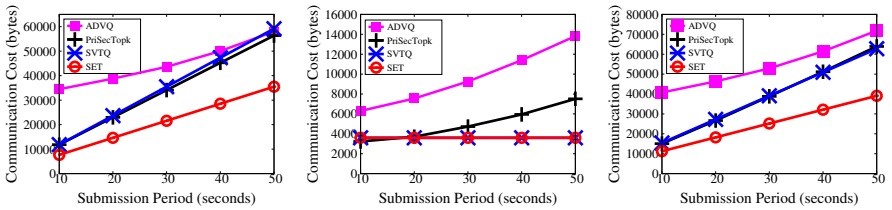
Fig. 3. Impact of Network Size on Communication Cost

6.2 Communication Cost

In WSNs, communication is the dominant factor to consume energy which affects network lifetime. The lower communication cost, the better efficiency.

Figure 3 displays the impact of the network size on the communication cost when the submission period is 40 seconds and k=10. As network size grows, more data are transmitted from sensor nodes to master nodes. The communication cost of sensor nodes in SET is much less than that in other three algorithms and increases more slowly. The reason is that: In ADVQ, sensor nodes send too much $HMAC$s, virtual line segments, and neighborhood information, and the number of neighbors rises rapidly as network density increases; In SVTQ, sensor nodes send the encrypted data and two big integers representing prime aggregation results; In PriSecTopk, sensor nodes send the encrypted data and the corresponding MACs; In our SET, sensor nodes only send the encrypted data and two simple codes generated upon the RAC scheme. With the growth of network size, more data are transmitted from master nodes to the sink, and the communication cost of master nodes in SET is still less than that in ADVQ and PriSecTopk. That is because master nodes in SET only submit k encrypted data as the top-k result and k' ($N - k \leq k' \leq N - 1$ and N denotes the network size) encrypted data for verification, whereas master nodes in ADVQ submit all data included in the $\eta - 1 + k$ highest virtual line segments and their neighbors' information, where η ($\eta > 1$) is a system parameter, and master nodes in PriSecTopk submit k MACs, the computing commitment information combined several sensing records for sample-based hypothesis testing method except k encrypted data. It is observed that the communication cost of master

nodes in SVTQ is equal to that in SET, because master nodes in SVTQ return the same message to the sink as SET does. Finally, the communication cost of whole network in SET is the least of four algorithms.



(a) Sensor Nodes (b) Master Nodes (c) Whole Network

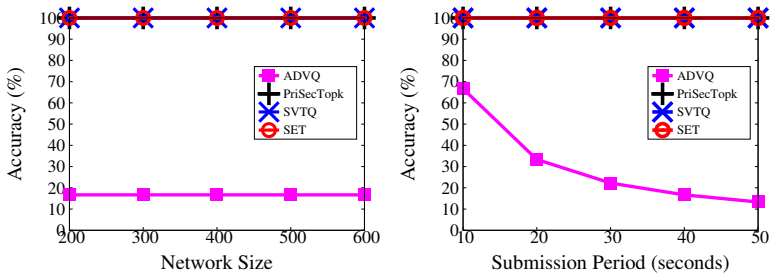
Fig. 4. Impact of Submission Period on Communication Cost

Figure 4 demonstrates the impact of the submission period on communication cost when the network size is 400 and k=10. As the submission period grows, the number of data sent by each sensor node increases. We can see that the communication cost of sensor nodes in SET is still much less than that in other algorithms. The reason is similar to that of 3(a). When a sensor node samples a datum, different algorithms will takes different action as follows. ADVQ will construct the $HMAC$ for the datum, update the data counter and two extra $HMAC$s, and may enlarge the virtual line segment. Except the encrypted part, SVTQ, PriSecTopk and our SET will respectively generate two prime aggregation results, a MAC and two smalll codes. The submission period has neglectable effect on the communication cost of master nodes in SET and SVTQ, because their communication cost of master nodes is nearly related to the network size for a given top-k query. In ADVQ, the number of data in a virtual line segment is proportional to the submission period. If the submission period becomes greater, the $\eta - 1 + k$ highest virtual line segments will cover more data so that master nodes will produce more communication. In PriSecTopk, we assume $s = k + 5$, which means master nodes select s sensing data in every selection of the query processing. As mentioned in work [7], master nodes will generate at most $\lceil \frac{\overline{N}-k}{s-k} \rceil$ computing commitment information for result verification, where $\overline{N}$ denotes the number of sensing data collected in one submission period. When the submission period increases by 10 seconds, $\overline{N}$ increases by the network size, and consequently, master nodes need to transmit more computing commitment information. The communication cost of whole network is mainly determined by that of sensor nodes. Therefore, SET saves the most communication cost for the whole network.

6.3 Result Accuracy

In the ideal condition, the sink can directly receive the top-k result including all data satisfying the query and excluding data dissatisfying the query. In fact, SET, ADVQ, PriSecTopk and SVTQ all adopt the two-tiered network model.

Due to this special network architecture, each master node first submits its local top-k result to the sink, and then the sink calculates the final result based on these local results, so it is crucial to find the accurate local top-k results. In the experiments, we divide the whole network into four identical cells. A single master node and its affiliated sensor nodes constitute a cell. Given a top-k query, the sink will receive four local top-k results, and we define result accuracy as the average ratio of the number of real local top-k data included in the local result to the sum of data in the local result in each cell. The higher accuracy, the better performance.



(a) Impact of Network Size (b) Impact of Submission Period

Fig. 5. Result Accuracy

Figure 5 displays the impact of different parameters on result accuracy. It can be seen that no matter how these parameters change, master nodes in SET, PriSecTopk and SVTQ always compute and return the correct local top-k data to the sink, i.e., the result accuracy of SET, PriSecTopk and SVTQ is 100%. In ADVQ, the network size has no impact on the accuracy as illustrated in 5(a) (submission period is 40 seconds and k=10), because its accuracy is only related to both k and the number of data included in the $\eta - 1 + k$ highest virtual line segments. In 5(b) (network size is 400 and k=10), the accuracy of ADVQ decreases sharply as the submission period grows. The reason is that lager submission period makes sensor nodes generate more data in each virtual line segment. As a result, master nodes need to send more unsatisfactory data to the sink.

In summary, experimental results demonstrate that our SET not only gains the accurate top-k result but also achieves low communication cost, which is more suitable for the practical applications of WSNs.

7 Conclusion

Privacy-preserving top-k query in WSNs is significant and challenging. In this paper, we present a secure and efficient top-k query framework—SET in two-tiered WSNs. In the proposed SET, data privacy is protected while the top-k result is correctly calculated by using the RAC scheme. Besides, the sink is able to verify result integrity through a series checking. Theoretical analysis and simulation results confirm the high efficiency, accuracy and security of SET.

References

1. Luce deployment. http://lcav.epfl.ch/cms/lang/en/pid/86035
2. Omnet++ 4.1. http://www.omnetpp.org
3. Fan, Y., Chen, H.: A secure topk query protocol in two-tiered sensor networks (in chinese). Chinese Journal of Computers 49(3), 1947–1958 (2012)
4. Hong, D., Eleftheriadis, A.: Memory-efficient semi-quasi renormalization for arithmetic coding. IEEE Transactions on Circuits and Systems for Video Technology 17(1), 106–110 (2007)
5. Huang, H., Juan, F., Wang, R., Qin, X.: An exact top-k query algorithm with privacy protection in wireless sensor networks. International Journal of Distributed Sensor Networks (IJDSN) 2014, 1–10 (2014)
6. Jnsson, K.V., Palmskog, K., Vigfusson, Y.: Secure distributed top-k aggregation. In: ICC. pp. 804–809. Ottawa, ON, Canada (Jun 2012)
7. Liao, X., Li, J.: Privacy-preserving and secure top-k query in two-tier wireless sensor network. In: IEEE Global Communications Conference (GLOBECOM). Anaheim, CA, USA (Dec 2012)
8. Ma, X., Song, H., Wang, J., Gao, J., Min, G.: A novel verification scheme for fine-grained top-k queries in two-tiered sensor networks. Wireless Personal Communications 75(3), 1809–1826 (2014)
9. Martin, G.N.N.: Range encoding: an algorithm for removing redundancy from a digitised message. In: Video and Data Recording Conference (1979)
10. Paek, J., Greenstein, B., Gnawali, O., Jang, K.Y., Joki, A., Vieira, M.A.M., Hicks, J., Estrin, D., Govindan, R., Kohler, E.: The tenet architecture for tiered sensor networks. ACM Transactions on Sensor Networks (TOSN) 6(4), 1–42 (2010)
11. Vaidya, J., Clifton, C.W.: Privacy-preserving kth element score over vertically partitioned data. IEEE Transactions on Knowledge and Data Engineering (TKDE) 21(2), 253–258 (2009)
12. Yao, Y., Ma, L., Liu, J.: Privacy-preserving top-k query in two-tiered wireless sensor networks. International Journal of Advancements in Computing Technology (IJACT) 4(6), 226– 235 (2012)
13. Yu, C.M., Ni, G.K., Chen, I.Y., Gelenbe, E., Kuo, S.Y.: Top-k query result completeness verification in tiered sensor networks. IEEE Transactions on Information Forensics and Security (TIFS) 9(1), 109–124 (2014)
14. Zhang, R., Shi, J., Liu, Y., Zhang, Y.: Verifiable fine-grained top-k queries in tiered sensor networks. In: IEEE Conference on Computer Communications (INFOCOM). pp. 1199–1207. San Diego, CA, USA (Mar 2010)
15. Zhang, R., Shi, J., Zhang, Y., Huang, X.: Secure top-k query processing in unattended tiered sensor networks. IEEE Transactions on Vehicular Technology (TVT) 63(9), 4681–4693 (2014)
16. 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: IEEE/ACM International Symposium on Quality of Service (2014)
17. Zhou, T., Lin, Y., Zhang, W., Xiao, S., Li, J.: Secure and verifiable top-k query in two-tiered sensor networks. In: Security and Privacy in Communication Networks (SecureComm). pp. 19–34. Sydney, NSW, Australia (Sep 2013)