Efficient Verifiable Top-k Queries in Two-tiered Wireless Sensor Networks

Hua Dai[1, 2], Geng Yang[1, 2], Haiping Huang[1, 2] and Fu Xiao[1, 2]
[1] College of Computer Science & Technology
Nanjing University of Posts and Telecommunications, Nanjing 210013 - China
[2] Key Lab of Broadband Wireless Communication and Sensor Network Technology
Ministry of Education, Nanjing 210013 - China
[e-mail: {daihua, yangg, hhp, xiaof}@njupt.edu.cn]
*Corresponding author: Hua Dai

Received July 4, 2014; revised February 25, 2015; accepted May 12, 2015; published June 30, 2015

Abstract

Tiered wireless sensor network is a network model of flexibility and robustness, which consists of the traditional resource-limited sensor nodes and the resource-abundant storage nodes. In such architecture, collected data from the sensor nodes are periodically submitted to the nearby storage nodes for archive purpose. When a query is requested, storage nodes also process the query and return qualified data as the result to the base station. The role of the storage nodes leads to an attack prone situation and leaves them more vulnerable in a hostile environment. If any of them is compromised, fake data may be injected into and/or qualified data may be discarded. And the base station would receive incorrect answers incurring malfunction to applications. In this paper, an efficient verifiable top-*k* query processing scheme called EVTQ is proposed, which is capable of verifying the authentication and completeness of the results. Collected data items with the embedded information of ordering and adjacent relationship through a hashed message authentication coding function, which serves as a validation code, are submitted from the sensor nodes to the storage nodes. Any injected or incomplete data in the returned result from a corresponded storage node is detected by the validation code at the base station. For saving communication cost, two optimized solutions that fuse and compress validation codes are presented. Experiments on communication cost show the proposed method is more efficiency than previous works.

Keywords: Two-tiered wireless sensor networks, top-*k* query, authenticity and completeness verification, hashed message authentication coding

A preliminary version of this paper appeared in IEEE MSN 2013, December 11-13, Dalian, China. This version includes a new optimization and concrete analysis on communication costs and security. This research was supported supported by the National Natural Science Foundation of China under the grant No. 61300240, 61402014, 61472193, 61373137, 61373138, 61201163, 61272084 and 61202004, the Natural Science Foundation of Jiangsu Province under the grant No. BK20141429, the Scientific and Technological Support Project (Society) of Jiangsu Province under the grant No. BE2013666, the Project of Natural Science Research of Jiangsu University under grant No.11KJA520002 and 14KJB520027, the Postdoctoral Science Foundation of China under the grand No. 2013M541703, and the Postdoctoral Science Foundation of Jiangsu Province under the grand No. 1301042B. We express our thanks to Dr. Rui Huang who checked our manuscript.

http://dx.doi.org/10.3837/tiis.2015.06.008 **ISSN : 1976-7277**

1. Introduction

We consider a two-tiered wireless sensor network [1, 2] as shown in **Fig. 1**, which has a few storage nodes at the upper tier surrounded by a variety of sensor nodes at the lower tier. Sensor nodes are the general data sensing devices limited in computation capacity, storage, energy and other resources, which are in charge of collecting and submitting data to affiliated storage nodes periodically. Storage nodes having abundant resources are responsible for answering the remote data queries from the base station through an on-demand wireless link (e.g., satellite). The two-tiered architecture is also known to be indispensable for increasing the capacity and flexibility of network, reducing system complexity, and prolonging the lifetime of network [1, 2]. However, since it is infeasible or difficult to keep a high-speed and always-on connection between the sensor networks and the base station, maintaining such in-network data storing and query processing in remote and tough environments is a necessary.

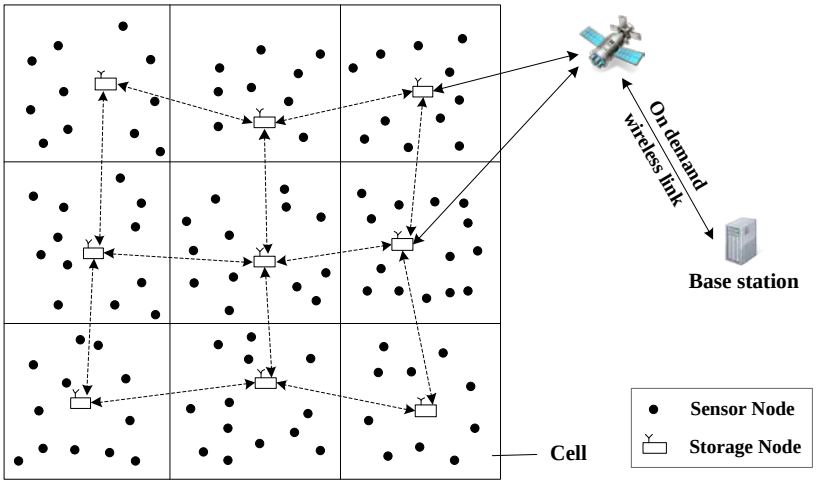


Fig. 1. A two-tiered sensor network architecture

Taking charge of in-network data storing and query processing, storage nodes are much more vulnerable and prone to be attacked in a hostile environment. Once a storage node is compromised, the intruder could return incomplete and/or fabricate fake information as the replies to the base station for data enquiring. Such application-level threats are very insidious and fatal, resulting to undesirable consequence in making critical decisions. Therefore, it is of importance to design an effective verifiable query processing mechanism, by which the authenticity and completeness of query results can be verified in the base station. Authenticity means detecting the fake data in returned information, while completeness is aiming at checking the result be intact in the base station.

Data query is a fundamental operation for events monitoring or data analysis in sensors networks, involving range query, top-k query, et al. Recently, verifiable range queries [3-7] have been well addressed, which require data within one or multiple attributes belongs to specified ranges. Whereas, research efforts on top-*k* queries [8-11] are limited, which is also an important type of data query commonly used in many sensor network applications. A top-*k* query requests the greatest *k* data items, where *k* is decided by the requesters. For example, “Reply the 5 highest temperature data items in warehouse B at every 5 minutes” is a typical

top-*k* query, which is used in fire monitoring applications. It is a challenge to deploy authenticity and completeness verifications for top-*k* queries in tiered sensor networks. First, the global information of the collected data in the interested query region is the base to compute the results of any top-*k* query, while only the storage nodes which could have been compromised are able to access such information. Second, the region in a top-*k* query is uncertain and unpredictable, which may cover from a node, multiple nodes, up to the entire network. As a result, it is impossible to predetermine the sensor nodes that may have data qualified with a top-*k* query and be prepared for verifications.

This paper focuses on verifiable top-*k* queries in two-tiered sensor networks. The key work is a scheme whereby the base station can verify the authenticity and completeness of any top-*k* query replies with more efficient communication cost. The basic idea of our scheme is that sensor nodes encode the ordering and adjacent relationships among their collected data items through a hashed message authentication coding function (HMAC-MD5 [23], HMAC-SHA1 [24], et al.), thus any abnormal query reply resulting from a compromised storage node injecting fake data into or discarding qualified data will be easily verified. Within this regulation, compromised storage nodes could only return both authentic and complete results for acceptance.

We summarize our contributions of this paper as follows. First, we propose a verifiable top-*k* query processing scheme including data submission protocol, query processing protocol and query result verifying algorithm. Second, we present two optimized solutions to which fusing and compressing validation codes are applied to save communication costs. Third, to evaluate the efficacy and efficiency of our scheme, detailed theoretical and quantitative analyses are given. Finally, we give the performance evaluation by comprehensive simulation, and the results indicate that our proposed solution shows better efficiency in communication costs and communication consumes much more energy than computation.

The paper is organized as follows. Section 2 gives a brief review of the related work. Section 3 describes the models and gives the problem statement. In Section 4, we present the details of verifiable top-*k* query protocols and its performance analysis. Two optimized solutions for reducing communication costs are provided in Section 5. We evaluate the performance of our approach in Section 6 and conclude this paper in Section 7.

2. Related Work

In sensor networks, top-*k* queries have been well investigated recently, and many protocols or schemes have been proposed. The pioneer work is done by Wu *et al.* [12]. To achieve continuous top-*k* query processing, a top-*k* monitoring approach called FILA was proposed by applying filter mechanism to reduce in-network communication. Malhotra *et al.* extend the same idea to an exact top-*k* query scheme, EXTOK [13] for accurate top *k* data among all the candidates. Those schemes are built on error-free data situation which is never the case in real world. The probability distribution model is considered for the collected data by many works such as [14, 15]. The proposed approximated top-*k* query schemes accept the query only if the error of query result is under certain threshold for upper level applications. Some other work such as [16] focus on history top-*k* queries, and we will not restate them in details since that topic is beyond our concern.

The security of data query in two-tiered sensor networks has been drawn attention only in this decade. Verifiable privacy-preserving range query [3-7] has been widely studied, aiming at protecting the privacy and integrity of range queries. Among the existing works, Sheng *et al.*

proposed a bucket partitioning [17] based scheme [3, 4], while collected data are stored into multiple continuous but not overlapping buckets. An optimized version focuses on the integrity verification is proposed by Shi *et al*. [5, 6]. The authenticity and completeness are preserved by crosschecking the spatial and temporal relationships among the returned data. For less communication cost of sensor nodes, Chen *et al*. proposed a secure and efficient range query processing protocol called SafeQ [7], which is based on prefix membership verification [18, 19] and neighborhood chains.

Secure top-*k* query in two-tiered sensor networks is the main topic of this paper. Zhang *et al*. proposed the verifiable fine-grained top-*k* query scheme [8] for the first time. They utilize the message authentication code generated by hashing three connected neighboring data items also known as chaining relationship for verification purpose. Each data item is followed by an additional code from which the query results will be verified. Because of the long bit length of the hashing value for each code, the communication cost of transmitting verification objects will be large. Liao and Li presented PriSecTopk [9], a secure top-*k* query processing scheme based on the message authentication code and order-preserving encryption [20]. Since a message authentication code is also needed to append to each data item, the communication overhead of PriSecTopk cannot be lowered comparing with [8]. By replacing message authentication code with the symmetric encryption which constructs smaller validation information, Ma *et al.* proposed a novel verification scheme VSFTQ [10] for fine-grained top-*k* queries. The communication cost of VSFTQ is less than [8] but still could be further optimized. Using the bucket partitioning scheme [17], Yu *et al*. proposed another secure top-*k* query processing protocol STQ [11], which does not support exact top-*k* queries since the bucket partitioning scheme is originally designed for range queries. Nevertheless, extra information caused by the partition work increases the communication cost. Therefore, low communication cost verifiable top-*k* query processing scheme in two-tiered sensor networks is still a challenge.

In this paper, we present a novel verifiable top-k query scheme EVTQ with lower extra communication cost. By appending the encoded ordering and adjacent relationship, any fabricated and incomplete result could be detected.

3. Models and Problem Statement

3.1 Network Model

We consider the same two-tiered sensor networks model as in [3-11]. As shown in **Fig. 1**, the network is partitioned into cells, and each cell contains several sensor nodes and a storage node. They are different in their capabilities. The storage nodes are powerful devices with abundant resources in power supply, storage and computation capacities, while the sensor nodes are cheap sensing devices with limited resource. Each sensor node transmits its collected data to the nearby storage node periodically. The base station converts users' questions into queries and then disseminates them to the corresponding storage nodes. Once the queries arrive, they are processed by the storage nodes. The results are replied to the base station through an upper-tier multi-hop network constructed by an on-demand wireless link connecting the storage nodes and the base station.

As in [3-11], we assume that sensor nodes know the relative location of their neighbors within one hop and their affiliating storage nodes, and storage nodes aware of all the sensor nodes in the same cells. Time is divided into identical slots. At the end of each slot, each sensor node submits all its collected data items to the affiliating storage node.

3.2 Top-*k* Query Model

We focus on the atomic top-*k* queries, each of which can be denoted as a four-element tuple:

$$Q_t = (C, \Gamma_t, t, k),$$

where C and t are the queried cell ID and time-slot number. Γ_t is the set of queried sensor nodes IDs indicating a query region in C, and k is the query index representing the quantity of required data items. The complex queries with multiple cells, query regions and/or time-slots can be easily decomposed into multiple atomic queries. For instance, **Fig. 2** shows that a network consists of 2 cells and 16 sensors. The complex query "obtaining the top 3 values within the round region at time-slot t" can be interpreted into 2 unique atomic queries, $Q_{t,1}=(C_1, \{s_2, s_7, s_8, s_{11}\}, t, 3)$ and $Q_{t,2}= (C_2, \{s_9, s_{12}, s_{16}\}, t, 3)$. And the result of complex query is the top 3 data items among the results of $Q_{t,1}$ and $Q_{t,2}$. In this paper, "top-*k* query" is short for " atomic top-*k* query" for simplicity.

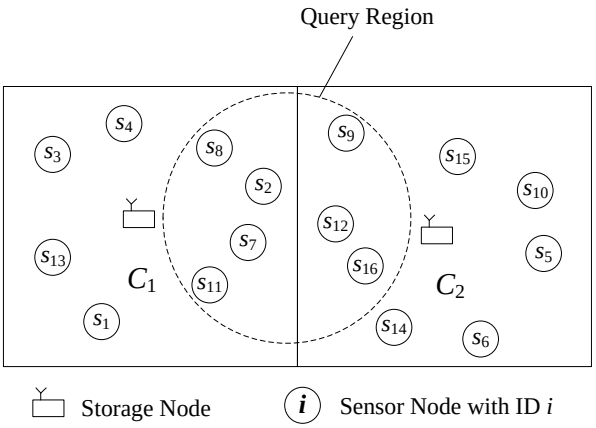


Fig. 2. A complex top-k query example

Since only one cell is considered in an atomic top-*k* query, let C denotes the cell consisting of a storage node M and n sensor nodes $\{s_1, s_2, \ldots, s_n\}$ whose IDs constitute the set $\Gamma=\{1, 2, \ldots, n\}$. Assuming each node $s_i \in S$ collects τ_i data items during a time-slot t, denoted as $D_{i,t}=\{d_{i,1}, d_{i,2}, \ldots, d_{i,\tau_i}\}$. Thus, M receives $\tau = \sum_{i=1}^{n} \tau_i$ data items in total at the end of t. To sort the collected data items for the top k data items, a scoring function such as [21] could be used to score each data item. We denote $f(x)$ as the score of x, where $f(*)$ is the public scoring function which satisfies that if $x \leq y$ then $f(x) \leq f(y)$. For simplicity, we subsequently assume that all the collected data items have unique scores, which is feasible if the node ID and the data collecting time are taken into consideration. Such assumption ensures that there must be a unique correct result for any top-*k* query.

3.3 Threat Model

Threat model such as [8, 10] assumes that the storage nodes could be compromised and instructed to return incomplete and/or fake data as the reply to the base station for ad-hoc top-*k* queries. In practice, sensor nodes could also be compromised, which may be used by the intruder to help the compromised storage nodes escape detection. It is an arduous task to prevent such attacks unless tamper proof hardware is integrated into detection. Under the

conventional assumption, however, uncompromised sensor nodes are always the majority and the data generated by the compromised sensor nodes just covers a little portion of the data stored by storage nodes, otherwise, there is no workable solution. Rather than the sensor nodes, if the storage nodes are compromised, greater damage could be incurred, since the storage nodes store the entire collected data items and are in charge of answering queries. Therefore, storage nodes are much attack prone and vulnerable to adversaries, which imply them to be the key points of secure data query processing.

Different from [3-7], data confidentiality preservation is not our focus in this paper. Many sensor network applications do not need data confidentiality but only query result authenticity and completeness [8]. For example, supposing the adversary intrudes in a battlefield, it is certainly not a secret that both know the intrusion events in a sensor network for battlefield reconnaissance. In other words, the intrusion being detected is known to the adversary, but in order to escape or delay the detection of his intrusions, he can compromise some storage nodes and force them to return fake and/or incomplete query responses. In such scenario, verifying the authenticity and completeness of query result turns to be a necessary.

Therefore, we concentrate on the countermeasures of query result authenticity and completeness verification against compromised storage nodes in this paper.

3.4 Problem Statement

Given a query Q_t=(C, Γ_t, t, k), we use $\Omega = \bigcup_{i \in \Gamma_t} D_{i,t}$ to represent candidate data set for the query result of Q_t, which has $\tau_t = \sum_{i \in \Gamma_t} \tau_i$ data items collected by the queried sensor nodes. The query result of Q_t is denoted as R_t which satisfies:

$$|R_t| = k \wedge \forall d_i \in R_t \left(d_i \in \Omega \wedge \forall d_j \in \left(\Omega - R_t \right) \rightarrow d_i > d_j \right)$$

The problem of interest is how the base station verifies whether or not R_t satisfies the requirements of authenticity and completeness conditions. Here, *authenticity* means all data items in R_t are indeed collected by the sensor nodes in Γ_t, while *completeness* means R_t has the exact top k data items among all the candidates. And only if R_t satisfies the above authenticity and completeness conditions simultaneously, then R_t is the verified query result.

To evaluate the proposed methods, we use the similar performance metrics as in [8, 10] which are listed as follows.

- In-cell communication cost, denoted as Δ_{ic}, is the total energy consumption in bits incurred by transmitting and receiving information between sensor nodes in cell C per time-slot. We also assume that the energy consumptions of transmitting and receiving every bit across each hop are equal.
- Query communication cost, denoted as Δ_{qc}, is the total information in bits transmitted between the storage nodes and the base station for top-k query processing. For simplicity, we assume the route between a storage node and the base station is a virtual hop, whose energy consumption differs from that between neighboring sensor nodes.

4. Verifiable Top-k Query Processing

In this section, we give the details of the verifiable top-*k* query processing scheme, EVTQ. Our basic idea is to enable authenticity and completeness verifications using the validation codes, which is returned along with the query result from the storage node *M*. Such validation code is formed from encoding the ordering and adjacent data items with a hashed message authentication coding function in the queried sensor node.

4.1 Assumptions and Lemmas

Assume that a sensor node s_i sorts its collected τ_i data items $D_{i,t}=\{d_{i,1}, d_{i,2}, \ldots, d_{i,\tau_i}\}$ into a list, such as $d_{i,1} > d_{i,2} > \ldots > d_{i,\tau_i}$, then we have the following Lemma 1 for a given top-*k* query $Q_t=(C, \Gamma_t, t, k)$ with its result R_t.

- ***Lemma* 1**: *For $d_{i,j} \in D_{i,t}$, if $d_{i,j} \in R_t$ then we have $\{d_{i,p} \mid 1 \le p < j\} \subset R_t$; likewise, if $d_{i,p} \notin R_t$ then we have $\{d_{i,p} \mid j<p \le \tau_i\} \cap R_t = \emptyset$.*

Lemma 1 implies that, for a list of adjacent data items, if one data item satisfies a top-*k* query, all its precedents also do.

We define the *outer-bound* of $D_{i,t}$ with regard to the query Q_t as the largest data item not satisfying Q_t. For instance, suppose that there are δ_i data items in $D_{i,t}$ satisfying Q_t, which must be $\{d_{i,1}, d_{i,2}, \ldots, d_{i,\delta_i}\}$ according to Lemma 1, then we have that d_{i,δ_i+1} is the outer-bound. Of course the outer-bound could not exist if all data items in $D_{i,t}$ satisfy Q_t or $D_{i,t}=\emptyset$. Additionally, we define $min(R_t)$ the lower bound of query result which is the smallest data item of R_t, then we have Lemma 2 as follows.

- ***Lemma* 2**. *Given a top-k query Q_t, any outer-bound of $D_{i,t}$ must be smaller than the lower bound of query result, i.e., $d_{i,\delta_i+1} < min(R_t)$, $i \in \Gamma_t$.*

Lemma 2 implies the stable relationship between outer-bound and the lower bound of query result, which will be utilized to perform completeness verification subsequently.

To achieve query result verification, we also assume that $H_K(\cdot)$ is a good hashed message authentication coding (HMAC) function with a key denoted in the subscription, and each sensor node s_i has a distinct secret key K_i sharing with the base station. For forward-secure authenticity, the key can be dynamic generated during each time-slot as in [3-5].

4.2 Data Submission Protocol

Data Submission Protocol is concerning about how a sensor node transmits its collected data items to *M*. For each sensor node s_i, after collecting τ_i data items $\{d_{i,1}, d_{i,2}, \ldots, d_{i,\tau_i}\}$ in each time-slot, s_i submits a message to *M* as follows depending on the quantity of its collected data items.

- If $\tau_i > 0$, the message is

$$s_i \rightarrow M: < i, t, d_{i,1}, H_{K_i}(t \| d_{i,1}), d_{i,2}, H_{K_i}(t \| d_{i,1} \| d_{i,2}), \ldots, d_{i,\tau_i}, H_{K_i}(t \| d_{i,1} \| d_{i,2} \| \cdots \| d_{i,\tau_i} \| \Psi) >,$$

- and if $\tau_i = 0$, the message is

$$s_i \rightarrow M: < i, t, H_{K_i}(t) >,$$

where Ψ *is* an extremely large value public to all sensor nodes.

The former case indicates that each collected data item $d_{i,j}$ will be submitted to M along with a HMAC which can be used for authentication verification in the base station. The HMAC embeds the time-slot along with all ordered data items which are no smaller than $d_{i,j}$ as the validation code. While the latter case implies that even if a sensor node collects nothing, it still needs to return a HMAC embedding only the time-slot, indicating none data collected. Since the HMAC function is with one-wayness and collision resistance features [22] and each sensor node only shares its secret key with the base station, it is computationally infeasible for the storage nodes to create a fake HMAC without been detected.

4.3 Query Processing Protocol

Query Processing Protocol is about how M processes a query and responses to the base station. When receiving a top-k query Q_t=(C, Γ_t, t, k), M selects and returns the largest k data items from the candidate data set Ω together with some additional information for query result verification. In particular, for each queried sensor node s_i ($i \in \Gamma_t$), assuming that if there are $\delta_i \in [0, \tau_i]$ data items collected by s_i satisfying Q_t , they must be $\{d_{i,1}, d_{i,2}, \ldots, d_{i,\delta_i}\}$ according to Lemma 1. Then M will return a message for each s_i in consistence with any of the following four cases.

- If $\delta_i = \tau_i = 0$, then
  $M \rightarrow$ Base Station : $< i, H_{K_i}(t) >$.
- If $0 \le \delta_i < \tau_i$ -1, then
  $M \rightarrow$ Base Station : $< i, d_{i,1}, \ldots, d_{i,\delta_i}, d_{i,\delta_i+1}, H_{K_i}(t \| d_{i,1} \| d_{i,2} \| \cdots \| d_{i,\delta_i+1}) >$.
- If $0 \le \delta_i = \tau_i$ -1, then
  $M \rightarrow$ Base Station : $< i, d_{i,1}, \ldots, d_{i,\delta_i}, d_{i,\delta_i+1}, H_{K_i}(t \| d_{i,1} \| d_{i,2} \| \cdots \| d_{i,\delta_i+1} \| \Psi) >$.
- If $\delta_i = \tau_i > 0$, then
  $M \rightarrow$ Base Station : $< i, d_{i,1}, \ldots, d_{i,\delta_i}, H_{K_i}(t \| d_{i,1} \| d_{i,2} \| \cdots \| d_{i,\delta_i} \| \Psi) >$.

The case 1 implies that if s_i has none data satisfying Q_t, M still needs to return a validation code. In case 2 and 3, the largest δ_i data items, $\{d_{i,1}, d_{i,2}, \ldots, d_{i,\delta_i}\}$, all belong to the aforementioned query result R_t, while d_{i,δ_i+1} is the outer-bound but not in R_t. Case 4 indicates that all the returned data items are in R_t. Specifically, there is no outer-bound in case 1 and 4. And the HMAC information in each case is to be utilized to verify the authenticity and completeness of R_t.

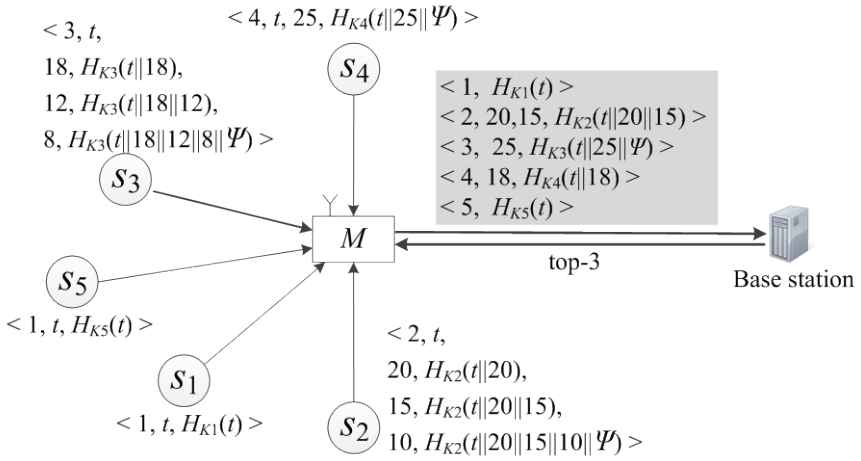


Fig. 3. An example of top-k query processing

An example is given as shown in **Fig. 3**. Assume that there is a cell C having four sensor nodes s_1, s_2, s_3, s_4 and s_5. In a time-slot t, s_1 and s_5 collects none data items while s_2, s_3 and s_4 collect {20, 15, 10}, {18, 12} and {25} respectively. According to Data Submission Protocol, they will send messages including node IDs, time-slot numbers, data items and validation codes (HMACs) to M, which are detailed in Fig.3. When the base station performs a top-3 query $Q_t = (C, \{s_1, s_2, s_3, s_4, s_5\}, t, 3)$, M responses as shown in the grey box of Fig. 3 to the base station. In practice, $<1, H_{k_1}(t)>$, $<2, 20, 15, H_{k_2}(t\|20\|15)>$, $<3, 18, 12, H_{k_3}(t\|18\|12\|\Psi)>$, $<4, 25, H_{k_4}(t\|25\|\Psi)>$ and $<5, H_{k_5}(t)>$ are returned for s_1, s_2, s_3, s_4 and s_5 respectively, following the above protocol. Obviously, there are two outer-bounds, 15 and 12, generated by s_2 and s_3.

4.4 Query Result Verification

Upon receiving query responses from M, the base station begins to compute the query result R_t and verify its authenticity and completeness described as the following *Query Result Verifying Algorithm* in three phases: pre-processing, query result computing, and verifying.

Query Result Verifying Algorithm

Input: *The messages received from M for a top-k query $Q_t=(C, \Gamma_t, t, k)$. Assume that $<i, d_{i,1}, d_{i,2}, \ldots, d_{i,l_i}, h_i>$ is a received message for a queried sensor node s_i ($i \in \Gamma_t$) where h_i is the HMAC data, $l_i \geq 0$ and $d_{i,1} > d_{i,2} > \ldots > d_{i,l_i}$.*

Output: *Query result R_t and its verification result*

Phase* I*: Pre-processing

Obviously, the situations that none message or more than one different messages returned from M regarding to a queried sensor node will be easily detected by the base station. And if so, the query result verification is failed and there is no need to continue the next phases.

Notice that, it is possible that M could return some messages for some nodes outside Γ_t. Since such messages can be easily recognized and have no impact to our protocols, such scenario is not discussed here.

Phase* II*: Query Result Computing

Assume that $ds_i=\{d_{i,1}, d_{i,2}, \ldots, d_{i,l_i}\}$ is a set of collected data items for s_i. If we denote DS as the set of collected data items that are received by the base station, then we have:

$$DS = \bigcup_{i \in \Gamma_t} ds_i$$

We can easily obtain the query result R_t by ranking scores of the data items in DS. Then we have:

$$R_t = \left\{ d_i \,\middle|\, d_i \in DS \wedge \forall d_j \in \left(DS - R_t\right)\left(d_i > d_j\right) \wedge \left|R_t\right| = k \right\}$$

Phase* III*: Query Result Verifying

To verify R_t, the base station first locates the key for each queried sensor node, with which it can re-compute the validation code to verify the collected data items (if any). In practice, if one of the following cases is satisfied for each received message $<i, d_{i,1}, d_{i,2}, \ldots, d_{i,l_i}, h_i>$ corresponding to s_i, then R_t is considered authentic and complete. Otherwise, R_t is unacceptable and the

query result verification fails.

- $l_i = 0 \wedge h_i = H_{k_i}(t)$
- $l_i > 0 \wedge h_i = H_{k_i}(t \| d_{i,1} \| \ldots \| d_{i,l_i}) \wedge d_{i,l_{i-1}} \geq min(R_t) \wedge d_{i,l_i} < min(R_t)$
- $l_i > 0 \wedge h_i = H_{k_i}(t \| d_{i,1} \| \ldots \| d_{i,l_i} \| \Psi) \wedge (d_{i,l_{i-1}} \geq min(R_t) \wedge d_{i,l_i} < min(R_t))$
- $l_i > 0 \wedge h_i = H_{k_i}(t \| d_{i,1} \| \ldots \| d_{i,l_i} \| \Psi) \wedge d_{i,l_i} \geq min(R_t)$

Phase I is to check whether the storage node has returned one message for each queried sensor node. Phase II is to obtain the top-*k* data items as the query result by their scores' ranking. While Phase III is to make sure that any falsifying and discarding of returned messages will be detected. The four cases in Phase III are respectively corresponded to Query Processing Protocol. In particularly, case 2 and 3 are the application of Lemma 2.

We take the same example as in **Fig. 3** to discuss the query result verification. For the returned messages, <1, G_{j_0}'s(>, <2, 20, 15, G_{j_1}'s{{1/ {{04(>, <3, 18, 12, G_{j_2}'s{{07{{01{{{(>, <4, 25, G_{j_3}'s{{14{{{(> and <1, G_{j_4}'s(>, we have that each of them satisfies one of the four cases of Phase III. Therefore, R_t ={25, 20, 18} are the authentic and complete top-3 query result. Because of the enforcement of query result verifying algorithm with HMAC keys which are unknown to *M*, it is impossible for *M* to discard or forge data without being detected.

4.5 Security Analysis

As long as none of the queried sensor nodes is compromised, our proposed EVTQ can detect any forged and/or incomplete top-*k* query result incurred by the compromised storage node *M*. We present the reasons through the following aspects.

For each queried sensor node s_i, assume s_i has δ_i sorted data items $\{d_{i,1}, d_{i,2}, \ldots, c_{h\mp_h}\}$ satisfying the query. As shown in Query Processing Protocol, such data items are encoded by the HMAC functions. And *M* does not have the corresponding key. Therefore, it is impossible for *M* to insert forged data items into or discard some from $\{d_{i,1}, d_{i,2}, \ldots, c_{h\mp_h}\}$ without being detected.

Now we are discussing replacing some qualified data items with other unqualified ones. Assume that sensor node s_i and s_j have δ_i and δ_j data items satisfying the query respectively, and *M* tries to replace some qualified data items of s_i with some unqualified ones from s_j. This equals to decreasing δ_i to δ_i' and increasing δ_j to δ_j'. We assume that the query results before and after data replacement are R_t and R_t'. Since data items of s_i and s_j are both ordered, the outer-bound of s_i and s_j will increase from $c_{h\mp_h*0}$ to $c_{h\mp_h'*0}$ and decrease from $c_{i\mp_i*0}$ to $c_{i\mp_i'*0}$. Then we have $c_{h\mp_h'} > c_{h\mp_h'*0} \geq c_{h\mp_h} > c_{h\mp_h*0}$ and $c_{i\mp_i} > c_{i\mp_i*0} \geq c_{i\mp_i'} > c_{i\mp_i'*0}$. According to Lemma 2, $\{c_{h\mp_h}, c_{i\mp_i}\} \subseteq R_t$ and $\{c_{h\mp_h'}, c_{i\mp_i'}\} \subseteq R_t$', we have $c_{h\mp_h'*0} \geq c_{h\mp_h} \geq min(R_t) > c_{h\mp_h*0}$ and $c_{i\mp_i} \geq min(R_t) > c_{i\mp_i*0} \geq c_{i\mp_i'} \geq min(R_t') > c_{i\mp_i'*0}$. Thus, $c_{h\mp_h'*0} > min(R_t')$ can be deduced, which contradicts Lemma 2 and fails the query result verification.

As a result, any fake and incomplete query result can thus be detected.

4.6 Communication Cost Analysis

We now derive the in-cell communication cost Δ_{ic} and query communication cost Δ_{qc}. The parameters and descriptions are listed in **Table 1**.

Table 1. Notifications of Paramaters

Para.	Descriptions
l_{id}	The bit length of each sensor node ID
l_t	The bit length of each time-slot number
l_d	The bit length of each collected data item
α	The bit length of a HMAC
L	The average number of hops between a sensor node and the storage node
μ	The quantity of queried sensor nodes of Q_t, $\mu = \|\Gamma_t\|$
ε	The quantity of queried sensor nodes with outer-bounds

Since each sensor node submits all its collected data items together with their HMACs to M during each time-slot, then we have:

$$\Delta_{ic} = \sum_{i=1}^{n} (l_{id} + l_t + \tau_i \cdot (l_d + \alpha)) \cdot L = (n \cdot (l_{id} + l_t) + \tau \cdot (l_d + \alpha)) \cdot L \quad (1)$$

We then conclude the query communication cost incurred by returning query responses to the base station. In consistent with any of the four cases in Query Processing protocol, M returns the HMAC and the sensor node ID for each queried sensor node, the top k data items and some outer-bounds. Thus, we have:

$$\Delta_{qc} = \mu \cdot (l_{id} + \alpha) + (k + \varepsilon) \cdot l_d \quad (2)$$

From *eq*.(1) and *eq*.(2), we have the in-cell communication cost depending linearly on the quantities of sensor nodes and their collected data items, theoretically. The query communication cost mainly depends on three aspects: the quantity of the sensor nodes covered by the query region, of queried sensor nodes with outer-bounds, and of the required data items.

5. Communication Cost Optimizations

To reduce the energy consumptions of the networks, we focus on the optimizing issues towards the communication cost in this section. Two novel works are introduced, which can be simultaneously applied to reduce the communication cost significantly.

5.1 Validation Codes Fusion

As shown in the Query Processing Protocol, the storage node should return a validation code for each queried sensor node no matter it collects any data or not. If some of the returned HMACs could be fused, the query communication cost would be saved.

Assume that the set of queried sensor nodes Γ_t has two sub-sets $\Gamma_{t,0}$ and $\Gamma_{t,1}$, $\Gamma_t = \Gamma_{t,0} \cup \Gamma_{t,1}$. $\Gamma_{t,0}$ and $\Gamma_{t,1}$ represent collecting none data or some data respectively in time-slot t. We modify the Query Processing Protocol in Section IV to achieve validation codes fusion. The optimized protocol has two parts.

Part I: For all queried sensor nodes in $\Gamma_{t,0}$, M fuses their HMACs by XOR operation ($\oplus$), and return the fused code to the base station as follows.

$$M \rightarrow \text{Base Station} : < \oplus_{i \in \Gamma_{t,0}} \left(H_{K_i}(t) \right) >$$

Part II: For each queried sensor nodes in $\Gamma_{t,1}$, M performs the same procedures as case 2 to 3 in the original Query Processing Protocol.

Obviously, the query communication cost will be reduced since fewer validation codes are returned, especially when many queried sensor nodes have none data collected. We take the scenario of Fig. 3 as an example. According to the optimized protocol, <2, 20, 15, $H_{k_2}(t||20||15)$>, <3, 18, 12, $H_{k_3}(t||18||12||\Psi)$> and <4, 25, $H_{k_4}(t||25||\Psi)$> are returned for s_2, s_3 and s_4 separately, while $< H_{k_1}(t) \oplus H_{k_5}(t) >$ is returned for s_1 and s_5. Comparing with the original protocol, cost for two node IDs and a validation code are waived.

Since the returned message for M is optimized, the verification procedure should be improved correspondingly. We optimize the Query Result Verifying Algorithm of Section IV of which the Phase I and II are kept unchanged and Phase III is modified.

Phase* III*: Query Result Verifying

According to the optimized Query Processing Protocol, there are two kinds of messages received by the base station. One is <i, $d_{i,1}$, $d_{i,2}$, …,d_{i,l_i}, h_i> where h_i is a HMAC data, l_i> 0 and $d_{i,1}$>$d_{i,2}$> …>d_{i,l_i}, corresponding to a queried sensor nodes s_i who has some collected data items in time-slot t. The other is the fused validation code *<h_f>. When the base station receives all the messages from M, it can easily obtain $\Gamma_{t,1}$, and so does $\Gamma_{t,0}$ where $\Gamma_{t,0}=\Gamma_t / \Gamma_{t,1}$.*

To verify the query result R_t, the base station first locates the key for each queried sensor node, with which it can re-compute HMAC to verify the returned validation codes. And only if the following two conditions hold simultaneously, R_t is considered authentic and complete. Otherwise, R_t is unacceptable and the query result verification fails.

Condition* 1: For each received message <i, $d_{i,1}$, $d_{i,2}$, …,d_{i,l_i}, h_i> corresponding to a queried sensor node in $\Gamma_{t,1}$, one of the following cases is met.*

- $h_i = G_{j_h}'s \| c_{h0} \| \prod \| c_{hk_h} (\wedge c_{hk_{h0}} \geq min(R_t) \wedge c_{hk_h} < min(R_t)$
- $h_i = G_{j_h}'s \| c_{h0} \| \prod \| c_{hk_h} \| (\wedge (c_{hk_{h0}} \geq min(R_t) \wedge c_{hk_h} < min(R_t))$
- $h_i = G_{j_h}'s \| c_{h0} \| \prod \| c_{hk_h} \| (\wedge c_{hk_h} \geq min(R_t)$

Condition* 2: For each queried sensor node s_j in $\Gamma_{t,0}$, the base station computes $H_{K_j}(t)$ by the shared key K_j. And the received fused validation code h_f satisfies*:

$$h_f = \oplus_{j \in \Gamma_{t,0}} \left(H_{K_j}(t) \right)$$

Under the optimized query result verifying algorithm mentioned above, it is impossible for the storage node to discard or forge data without being detected.

5.2 Validation Codes Compression

The query scheme in section IV will bring much in-cell and query communication cost on because of the transmission of the extra validation codes, which consists of several HMACs. E.g., a HMAC may have 128 bits with HMAC-MD5 [23] or 160 bits with HMAC-SHA1 [24].

We draw on the basic idea of compressing the HMAC data for the validation codes compression, which can reduce both the in-cell and query communication costs.

Assume that each HMAC have α bits uniformly distributed in I_h={0, 1, …, 2^{α}-1}. We use a simple hash function $\hbar$ to compress HMACs, where x is a HMAC data item and $\alpha<\beta$.

$$\hbar(x) = x \bmod (2^{\beta}-1) \tag{3}$$

After applying $\hbar$, each HMAC can be converted to a fewer-bits code uniformly distributed in {0, 1, …, 2^{β}-2} because of the uniform distribution of HMAC in I_h, by which the validation codes are compressed.

Although the hash function h could cause collisions, the impact is very limited to EVTQ. A collision means $\hbar(x)=\hbar(y)$, for any two HMAC data items x and y, if $x\neq y$. The probability of a collision Pr is:

$$Pr = \frac{1}{2^{\beta}-1} \tag{4}$$

Suppose that the 128 bits HMAC-MD5 is used (α=128), the results of impact of β on Pr are shown in **Fig. 4**. With an appropriate β, the collision probability could be omit. For instance, β=24, we have Pr = 5.96*10^{-8}. Under such a low probability, it is almost impossible for an adversary to construct a fake HMAC leading to a collision after applying $\hbar$, but the size of each validation code is significantly reduced, *i.e.*, each 128 bits HMAC is compressed into 24 bits and the compression rate is 18.75%.

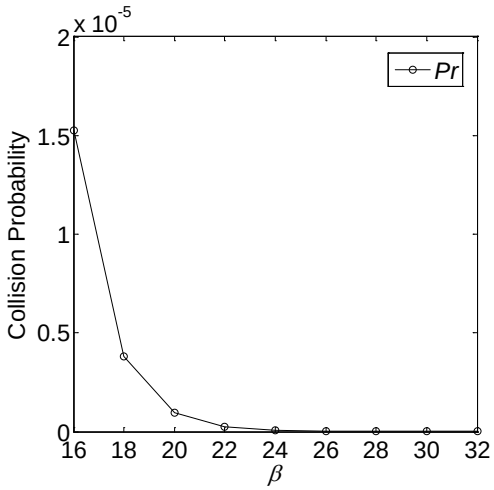


Fig. 4. Impact of β on Pr

5.3 Communication Cost Analysis

Comparing the validation codes fusion and compression, the former can only decrease the query communication cost, once some queried sensor nodes have no collected data. While the latter saves not only the query communication cost but also the in-cell communication cost. Fortunately, these two optimizations can be applied together, which effectively save the communication cost of the whole network.

Assume that there are λ queried sensor nodes with none collected data item. And each compressed HMAC is β bits. Then we have

$$\Delta_{ic} = (n \cdot (l_{id} + l_t) + \tau \cdot (l_d + \beta)) \cdot L \tag{5}$$

and

$$\Delta_{qc} = \begin{cases} \mu \cdot (l_{id} + \beta) + (k + \varepsilon) \cdot l_d & \lambda = 0 \\ (\mu - \lambda) \cdot (l_{id} + \beta) + (k + \varepsilon) \cdot l_d + \beta & \lambda > 0 \end{cases}. \tag{6}$$

From *eq* 5, β is smaller than α in *eq* 1, so the in-cell communication cost is lower. For *eq* 6, it has two conditions. If $\lambda = 0$, it is clear the query communication cost is reduced. For $\lambda > 0$, the cost is even smaller.

6. Performance Evaluation

In this section, we first use a comprehensive simulator on basis of [25] to evaluate the in-cell and query communication costs of the proposed scheme EVTQ, and compare it with the methods denoted as VFTQ [8] and VSFTQ [10]. In fact, there are three schemes proposed in VFTQ, and we only compare with the scheme 1 as the EVTQ is a direct improvement on scheme 1 and the basic idea can be easily adapted to the scheme 2 and 3. Then, we evaluate the communication and computation energy consumptions of the optimized EVTQ in sensor nodes with two common architectures MICAz and TelosB.

We implement EVTQ, VFTQ and VSFTQ on the simulator with random data set. EVTQ-bas and EVTQ-opt represent the proposed EVTQ with and without optimizations. We also assume error-free and collision-free package transmissions in our evaluations.

Assume that a top-*k* query is carried out in a cell with *n* sensor nodes and a storage node. The placement of sensor nodes in a cell follows a uniform distribution over a square region of 80×80 m^2 and the communication radius of a sensor is 10m. We use HMAC-MD5 as the HMAC algorithm, which generates 128 bits output. For simplicity, we assume each sensor node collects *N* data items during each time-slot. Other default parameters are summarized in **Table 2**.

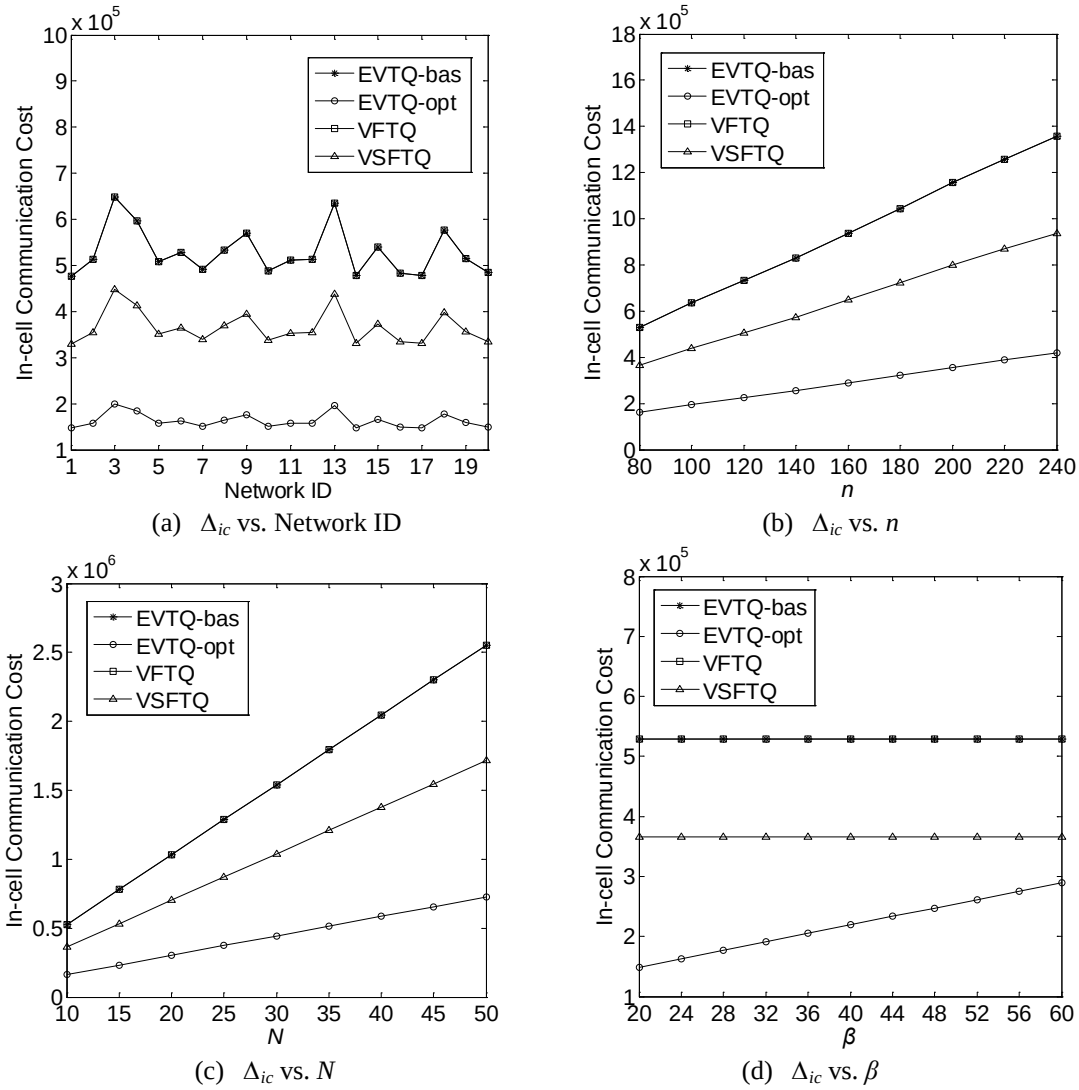
Table 2. Default Evaluation Parameters

Para.	n	l_t	l_{id}	l_d	α	β	N	λ	k
Val.	80	32 bit	32 bit	16 bit	128 bit	24 bit	10	0	20

In each measurement, we generate 20 networks with random topologies which are represnented by different network IDs. The total in-cell and query communication cost of each measurement is the average of 20 networks.

6.1 In-cell Communication Cost Evaluation

We compare the in-cell communication costs (Δ_{ic}) of EVTQ (with and without optimization), VFTQ and VSFTQ through four evaluations where network topology, *n*, *N* and β change respectively. Results of evaluations are shown in **Fig. 5**.



(a) Δ_{ic} vs. Network ID (b) Δ_{ic} vs. n

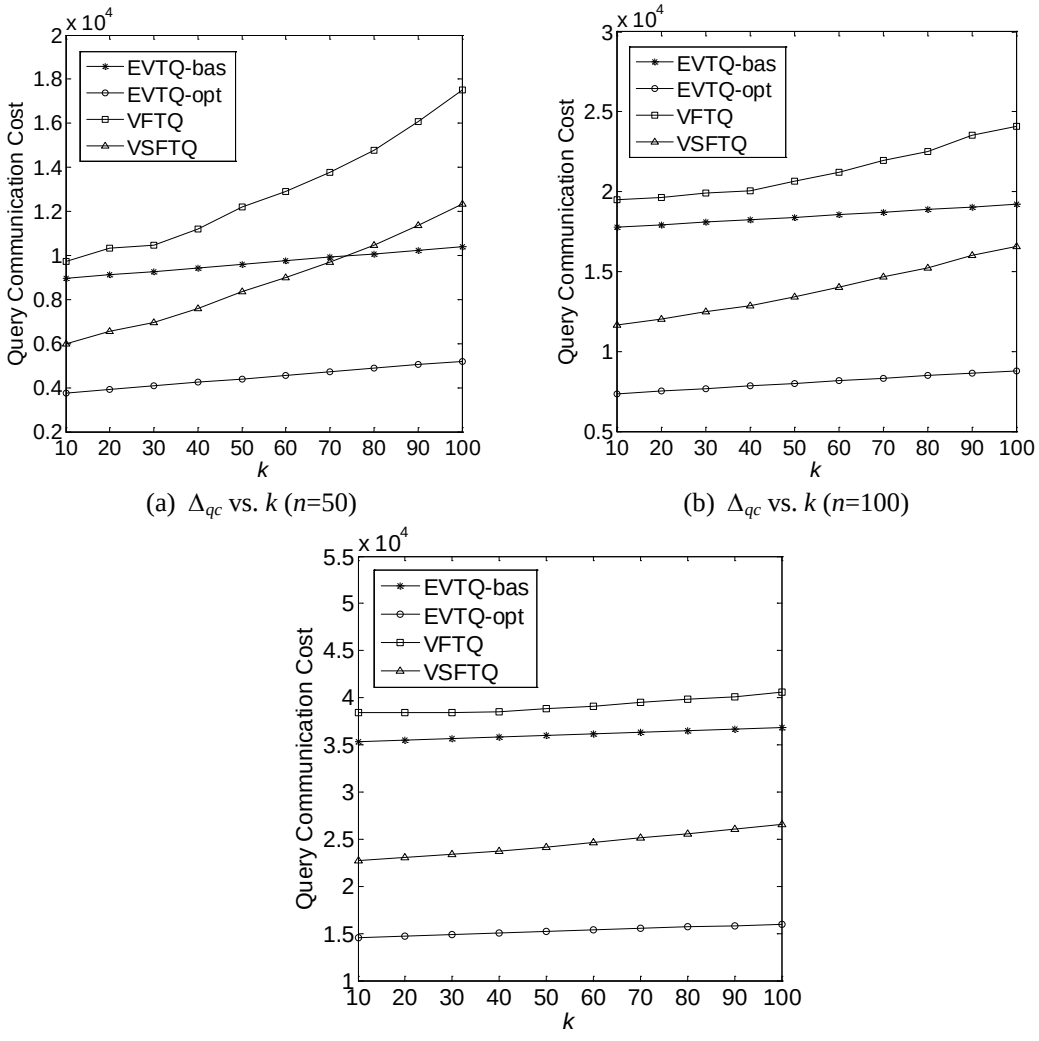
(c) Δ_{ic} vs. N (d) Δ_{ic} vs. β

Fig. 5. Evaluation results of in-cell communication costs

Fig. 5(a) shows that Δ_{ic} of EVTQ (with and without optimizations), VFTQ and VSFTQ are all uniformly distributed in different networks, while **Fig. 5(b)** and **Fig. 5(c)** show that they all increase as n and N grows. And **Fig. 5(d)** indicates that only the optimized EVTQ's Δ_{ic} is linearly proportion to β, where others are irrelevant to β. The plots of the proposed EVTQ without optimization overlap with those of VFTQ in all four figures, and the optimized EVTQ saves about 70% and 55% on average comparing with VFTQ and VSFTQ. Since both EVTQ without optimization and VFTQ only require each sensor node submit a collected data item together with a validation code, and after applying optimizations to our proposed EVTQ, validation codes will be compressed.

6.2 Query Communication Cost Evaluation

First, the query communication cost (Δ_{qc}) of EVTQ (with and without optimization), VFTQ and VSFTQ are evaluated against the growth of k. Three different network settings are configured with 50, 100 and 200 queried sensor nodes. Results of the evaluations are shown in **Fig. 6**.



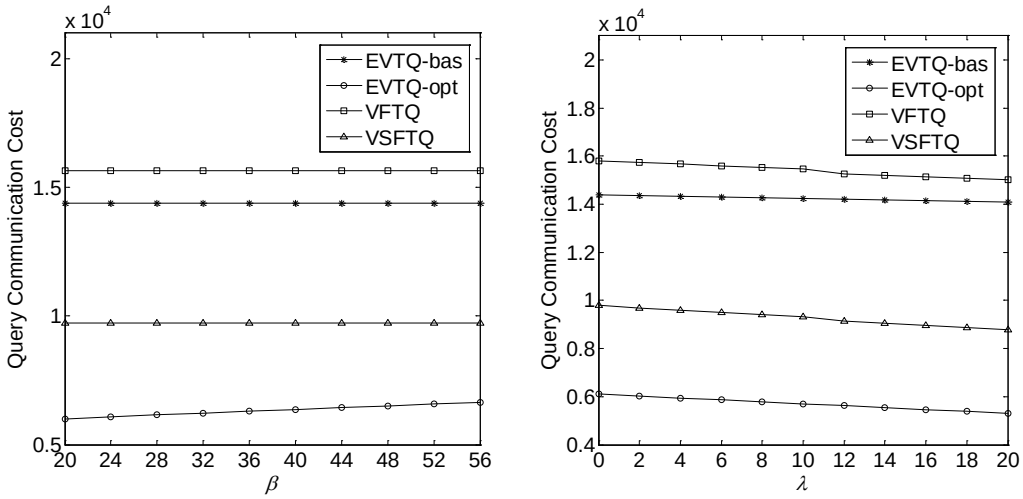
(a) Δ_{qc} vs. k (n=50) (b) Δ_{qc} vs. k (n=100)

(c) Δqc vs. k (n=200)

Fig. 6. Evaluation results of query communication costs

As shown in **Fig. 6(a)~(c)**, no matter with or without optimizations, the EVTQ's Δ_{qc} increase slowly and inconspicuously as k grows. The optimized EVTQ outperforms both the VFTQ and VSFTQ around 60% and 45% on average. In VFTQ and VSFTQ, the storage node needs return a validation code for both each qualified data items and the queried sensor nodes without qualified data items, but in optimized EVTQ only for every queried sensor node with data items, and one code for all the queried sensor nodes without data items. As a result, much more communication will be raised in VFTQ and VSFTQ, especially for more requested data items (k is enlarged) and/or more sensor nodes.

Then, we evaluate Δ_{qc} of EVTQ (with and without optimization), VFTQ and VSFTQ with the growth of β and λ. The result is shown in **Fig. 7**.



(a) Δ_{qc} vs. β (b) Δ_{qc} vs. λ

Fig. 7. Evaluation results of query communication costs

Fig. 7(a) shows that only the optimized EVTQ's Δ_{qc} increases as β grows, since others do not apply validation codes compression to reduce Δ_{qc}. And **Fig. 7(b)** describes that Δ_{qc} of the four schemes all decrease as λ grows, because the returned messages consisted of collected data items and validation codes are all reduced with more queried sensor nodes of none data items. Furthermore, **Fig. 7(a)-(b)** both indicate that the proposed EVTQ with optimizations takes an obviously improvements than VFTQ and VSFTQ in query communication cost for about 60% and 35% in **Fig. 7(a)** and about 63% and 39% in **Fig. 7(b)** on average, respectively.

6.3 Energy Consumption Evaluation

We apply two common mote architectures to evaluate the computation and communication energy consumptions of sensor nodes in the optimized EVTQ: MICAz and TelosB. The MICAz has an 8 bits bus width and a 7.37MHz processor frequency, while the TelosB has a 16 bits bus width and an 8MHz processor frequency. To obtain energy consumption of HMAC, we use the cost of common operations reported in [26] which are shown in **Table 3**, together with the general equation (7) from [27].

Table 3. Energy Consumption of Common Operations on the MICAz and TelosB [26]

Operation	MICAz	TelosB
Compute for 1 Clock Tick	3.5 nJ	1.2nJ
Transmit 1 bit	0.60μJ	0.72μJ
Receive 1 bit	0.67μJ	0.81μJ

$$time_{exec}(textlength) = \frac{a + b \cdot \lceil textlength / blocksize \rceil}{processorfreq \cdot buswidth} \tag{7}$$

where variables a and b are given as follows:

$$\begin{aligned} a &= a_{BASE} + a_{MUL} + a_{RISC} \\ b &= b_{BASE} + b_{MUL} + b_{RISC} \end{aligned} \tag{8}$$

Parameters a_{BASE}, a_{MUL}, a_{RISC}, b_{BASE}, b_{MUL} and b_{RISC} are given in [27]. With this information, we can estimate the time and hense the number of processor clock ticks, spent on various encryption/hash algorithms. MD5 is a block method that operate on block size of 512 bits. Because the input data size under the default parameter setting is less than the block size, padding is required. The time using MD5 to hash in microseconds is determined from (7). The number of processor clock ticks are determined by multiplying the time by processor frequency. And finally the energy consumption was determined from the number of clocks according to energy per clock tick given in [27] shown in **Table 3**. The final result, given in **Table 4**, shows time, clock ticks and energy consumption spent per node to hash data on the MICAz and TelosB with MD5.

Table 4. Time, clock ticks and energy consumption of MD5 on the MICAz and TelosB

Sensor	Time (ms)	Clock ticks	Energy (μJ)
MICAz	5000.37	36852.75	128.98μJ
TelosB	2303.30	18426.28	22.11μJ

According to [22], each HMAC computation has two hashing procedures, two XOR and two concatenating operations. Because the hashing algorithm MD5 is much more costly than XOR and concatenating operations, the cost of HMAC-MD5 can be estimated twice of MD5 approximately. With HMAC-MD5, the energy spent per node to process data on the MICAz and TelosB are 257.96μJ and 44.22μJ.

To determine the energy cosumption of sensor nodes in the optimized EVTQ, we adopt the above estimated energy cosumption of HMAC-MD5 with the costs of 1 processor clock tick, and 1 bit transmission and reception as shown in **Table 3**. Using the simulator, **Table 5** gives the evaluation results of communication and computation energy consumptions of sensors nodes with the default parameter setting.

Table 5. Communication and computation energy consumptions of MICAz and TelosB nodes

Energy consumption	MICAz	TelosB
computation	203.40mJ (18.88%)	34.87mJ (2.73%)
Communication	873.70mJ (81.12%)	1240.04mJ (97.27%)
Total	1077.10mJ	1274.91mJ

Table 5 indicates that MICAz and TelosB nodes spend much more energy in communication than computation. And the computation cost of TelosB nodes rarely affects the lifetime of networks since it only accounts for 2.73% in total. Due to having faster processor and wider bus, TelosB nodes consume less energy than MICAz nodes in computation. While the former consume more in communication because of the larger costs of 1 bit transmission and reception.

According to the above evaluations and analysis, we can conclude that the proposed EVTQ is more efficient than the existing works [8, 10] in communication costs. And communication consumes much more energy than computation.

7. Conclusion

As wireless sensor networks are popular in many important fields, verifying authentication and completeness of query result in query processing is a challenge in sensor network applications. In this paper, we propose EVTQ, a communication cost efficient scheme for

handling verifiable top-*k* queries in two-tiered sensor networks. To implement verifiable top-*k* query processing, sensor nodes are designed to encode their collected data items in an ordering and adjacent relationship by a hashed message authentication coding function. Thus no matter the storage node injects fake data into or discards qualified data from query result, they will be both detected. Furthermore, we also give validation codes fusion and compression based optimizations for less communication costs. The result of our evaluations shows that the proposed EVTQ outperforms the existing works in communication cost.

References

[1] O. Gnawali, K. Y. Jang, J. Paek, et al., "The tenet architecture for tiered sensor networks," in *Proc. of the 4th ACM Conference on Embedded Networked Sensor Systems*, pp. 153-166, 2006. Article (CrossRef Link)

[2] P. Desnoyers, D. Ganesan, P. Shenoy, "TSAR: A two tier sensor storage architecture using interval skip graphs," in *Proc. of the 5th ACM Conference on Embedded Networked Sensor Systems*, pp. 39-50, 2005. Article (CrossRef Link)

[3] B. Sheng, Q.Li, "Verifiable privacy-preserving range query in two-tiered sensor networks," in *Proc. of the 27th IEEE International Conference on Computer Communications*, pp. 46-50, Article (CrossRef Link)

[4] B. Sheng, Q. Li, "Verifiable privacy-preserving sensor network storage for range query," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1312-1326, 2011. Article (CrossRef Link)

[5] J. Shi, R. Zhang, Y. Zhang, "Secure range queries in tiered sensor network," in *Pro. of the 28th IEEE International Conference on Computer Communications*, pp. 945-953, 2009. Article (CrossRef Link)

[6] J. Shi, R. Zhang, Y. Zhang, "A spatiotemporal approach for secure range queries in tiered sensor networks," *IEEE Transaction on Wireless Communications*, vol. 10, no. 1, pp. 264-273, 2011. Article (CrossRef Link)

[7] F. Chen, A. X. Liu, "Privacy and integrity preserving range queries in sensor networks," IEEE/ACM Transaction on Networks, vol. 20, no. 6, pp. 1774-1787, 2012. Article (CrossRef Link)

[8] R. Zhang, J. Shi, Y. Liu, et al., "Verifiable fine-grained top-*k* queries in tiered sensor networks," in *Proc. of the 29th IEEE International Conference on Computer Communications*, pp.1199-1207, 2010. Article (CrossRef Link)

[9] X. Liao, J. Li. "Privacy-preserving and secure top-*k* query in two-tier wireless sensor network," in *Proc. of the 2012 Global Communications Conference*, pp. 335-341, 2012. Article (CrossRef Link)

[10] X. Ma, H. Song, J Wang, et al., "A novel verification scheme for fine-grained top-*k* queries in two-tiered sensor networks," *Wireless Personal Communications*, vol. 75, no. 3, pp. 1809-1826, 2014. Article (CrossRef Link)

[11] C. M. Yu, Y. T. Tsou, C. S. Lu, et al., "Practical and secure multidimensional query framework in tiered sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 241-255, 2011. Article (CrossRef Link)

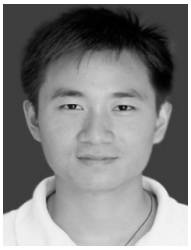
[12] M. Wu, J. Xu, X. Tang, et al., "Top-*k* monitoring in wireless sensor networks," *IEEE Transaction on Knowledge and Data Engineering*, vol. 19, no. 7, pp. 962-976, 2007. Article (CrossRef Link)

[13] B. Malhotra, M. A. Nascimento, I. Nikolaidis, "Exact top-*k* queries in wireless sensor networks," *IEEE Transaction on Knowledge and Data Engineering*, vol. 23, no. 10, pp. 1513-1525, 2011. Article (CrossRef Link)

[14] M. Ye, W. C. Lee, D. L. Lee, et al., "Distributed processing of probabilistic top-*k* queries in wireless sensor networks," *IEEE Transaction on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 76-91, 2013. Article (CrossRef Link)

[15] A. Silberstein, R. Braynard, C. Schlatter, et al., "A sampling-based approach to optimizing top-*k* queries in sensor networks," in *Proc. of the 22nd International Conference on Data Engineering*, pp.68, 2007. Article (CrossRef Link)

[16] J. Cheng, H. B. Jiang, J. C. LIU, et al., “On efficient processing of continuous historical top-*k* queries in sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 5, pp. 2363-2367, 2011. Article (CrossRef Link)
[17] H. Hacigümüş, B. Iyer, C. Li, et al., “Executing SQL over encrypted data in the database-service-provider model,” in *Proc. of the 2002 ACM SIGMOD International Conference on Management of Data*, pp. 216-227, 2002. Article (CrossRef Link)
[18] C. Jerry, Y. Hao, H. Y. Starsky, et al., “Design and implementation of cross-domain cooperative firewall,” in *Proc. of the IEEE International Conference on Network Protocols*, pp. 284-293, 2007. Article (CrossRef Link)
[19] A. X. Liu and F. Chen. “Collaborative enforcement of firewall policies in virtual private networks,” in *Proc. of the 27th Annual ACM Symposium on Principles of Distributed Computing*, pp. 95-104, 2008. Article (CrossRef Link)
[20] R. Agrawal, J. Kiernan, R. Srikant, et al., “Order-preserving encryption for numeric data,” in *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp. 563-574, 2004. Article (CrossRef Link)
[21] G. Das, D. Gunopulos, N. Koudas, et al., “Answering top-*k* queries using views,” in *Proc. of the 32nd International Conference on Very Large Data Bases*, pp. 451-462, 2006. Article (CrossRef Link)
[22] H. Krawczyk, R. Canetti, M. Bellare, “HMAC: keyed-hashing for message authentication,” RFC 2104. Reston: Internet Society, 1997. Article (CrossRef Link)
[23] R. Rivest. “The MD5 message-digest algorithm,” RFC 1321. Reston: Internet Society, 1992. Article (CrossRef Link)
[24] D. Eastlake, P. Jones, “US secure hash algorithm 1 (SHA1),” RFC 3174, Reston: Internet Society, 2001. Article (CrossRef Link)
[25] A. Coman, J. Sander and M. A. Nascimento, “Adaptive processing of historical spatial range queries in peer-to-peer sensor networks,” *Distributed and Parallel Databases*, vol. 22, no. 2, pp. 133-163, 2007. Article (CrossRef Link)
[26] G. de Meulenaer, F. Gosset, F. X. Standaert, et al., "On the energy cost of communication and cryptography in wireless sensor networks," in Proc*. of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*, pp.580–585, 2008. Article (CrossRef Link)
[27] P. Ganesan, R. Venugopalan, P. Peddabachagari, et al., "Analyzing and modeling encryption overhead for sensor network nodes," in *Proc. of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, pp.151–159, 2003. Article (CrossRef Link)



Hua Dai, born in 1982. PhD and Associate professor with the College of Computer Science and Technology, Nanjing University of Posts and Telecommunications. His current research interests include database security, distributed data management and security. E-mail: daihua@njupt.edu.cn.

Geng Yang, born in 1961. Professor and PhD supervisor with the College of Computer Science and Technology, Nanjing University of Posts and Telecommunications. His current research interests include computer communication and networks, parallel and distributed computing, and information security. E-mail:yangg@njupt.edu.cn

Haiping Hua, born in 1981. PhD and Associate professor with the College of Computer Science and Technology, Nanjing University of Posts and Telecommunications. His current research interests are wireless sensor networks and Information Security. E-mail: hhp@njupt.edu.cn.

Fu Xiao, born in 1980. Professor with the College of Computer Science and Technology, Nanjing University of Posts and Telecommunications. His current research interests are wireless sensor networks and multimedia information processing. E-mail: xiaof@njupt.edu.cn.