Secure Top-\(k\) Query Processing in Unattended Tiered Sensor Networks

Rui Zhang, Member, IEEE, Jing Shi, Yanchao Zhang, Senior Member, IEEE and Xiaoxia Huang, Member, IEEE

Abstract—Many future large-scale unattended sensor networks (USNs) are expected to follow a two-tier architecture with resource-poor sensor nodes at the lower tier and fewer resource-rich master nodes at the upper tier. Master nodes collect data from sensor nodes and then answer the queries from the network owner on their behalf. In hostile environments, master and sensor nodes may be compromised by the adversary and return incorrect data in response to data queries. Such application-level attacks are more harmful and difficult to detect than blind DoS attacks on network communications, especially when the query results are the basis for critical decision making. This paper presents a suite of novel schemes to enable verifiable top-\(k\) query processing in USNs, which is the first work of its kind. The proposed schemes are built upon symmetric cryptographic primitives and enable the network owner to detect any incorrect top-\(k\) query results. Detailed theoretical and simulation results confirm the high efficiency and efficacy of the proposed schemes.

Index Terms—Unattended tiered sensor networks, top-\(k\) query, security.

I. INTRODUCTION

Unattended sensor networks (USNs) refer to the sensor networks operating without an online data collection entity [2], [3]. USNs are ideal for remote and extreme environments such as oceans, volcanos, animal habitats, and battlefields. Instead of maintaining a costly high-speed stable communication link between the network and its external network owner, the USN relies on in-network data storage [4]–[7] for continuously produced sensed data. The network owner can access the data via an on-demand wireless link to some master nodes. USNs may support various data queries and top-\(k\) queries [9], [10] is among the most important and also the focus of this paper. A top-\(k\) query asks for data items with numeric attributes or scores [9] among the \(k\) highest, where \(k\) is an application-dependent parameter. An exemplary top-10 query “Return the data whose temperature attribute is among the 10 highest between 2pm and 3pm.”

The unattended nature of USNs unfortunately renders top-\(k\) query processing very vulnerable to attacks in hostile environments. For example, master and/or sensor nodes in military or homeland security applications may be compromised by the adversary; those in commercial USNs may likewise be compromised by malicious business competitors to degrade their quality of data service.

The adversary may launch a number of attacks via compromised master and/or sensor nodes. For example, the adversary may instruct a compromised master node to return fake or juggled data in response to top-\(k\) queries from the network owner. Such application-level attacks are more subtle and harmful than blind Denial-of-Service (DoS) attacks, especially when query results are the basis for critical military or business decisions. As another example, compromised sensor nodes may forge sensed data

\[\text{Sensor Node}\]

\[\text{Master Node}\]

\[\text{Network Owner}\]

\[\text{Cell}\]

Fig. 1. A remote two-tier sensor network.
data with extremely large scores such that the data items generated by legitimate sensor nodes will have little chance to appear in the query result even if the master node well behave. Moreover, compromised sensor nodes may assist master nodes to escape detection.

The above situations necessitate proactive mechanisms to enable verifiable top-\(k\) query processing, by which the network owner can verify the authenticity and soundness of top-\(k\) query responses. Authentication check is needed to detect fake data in query responses, while soundness verification is necessary to make sure that the returned data items are indeed those satisfying the query conditions, i.e., indeed the ones with scores among the \(k\) highest. A query result is considered correct only if it is both authentic and sound.

This paper investigates verifiable top-\(k\) queries in UTSNs with the following contributions.

- We first propose VTQ, a novel scheme whereby the network owner can detect any incorrect top-\(k\) query results returned by a compromised master node. VTQ relies on sensor nodes embedding some relationships among the data items they generated so that the network owner can detect any incorrect top-\(k\) query result by examining the embedded information.
- We then propose a random probing (RP) scheme to detect possible colluding attack from compromised master and sensor nodes. RP works by letting the network owner probe some randomly chosen sensor nodes for additional proofs after a top-\(k\) query result passes the verification under VTQ.
- We further propose QC, a query conversion scheme to mitigate the impact of compromised sensor nodes forging data items with extremely high scores. The basic idea is that the network owner converts a top-\(k\) query into another such that the query result for the converted query contains the true top-\(k\) data items generated by legitimate sensor nodes with overwhelming probability.
- We also propose a lightweight scheme called RW to detect possible compromised sensor nodes framing a legitimate master node. RW relies on randomly chosen sensor nodes serving as witnesses for those submitting sensed data to the master node. In case of dispute, the network owner can detect framings by examining the testimonies from witness nodes.

All our proposed schemes are built upon symmetric cryptographic primitives and thus are very suitable for resource-constrained UTSNs. Their efficacy and efficiency are confirmed by detailed theoretical analysis and simulation results.

The rest of this paper is structured as follows. Section II introduces our network, query, and adversary models. Section II presents our problem formulation and the evaluation metrics. Section IV illustrates VTQ. Section V illustrates RP, QC, and RW for defending against compromised sensor nodes. All the proposed schemes are theoretically analyzed in Section VI and evaluated via detailed simulations in Section VII. Section VIII discusses the related work, and Section IX concludes this paper.

II. NETWORK, QUERY, AND ADVERSARY MODELS

A. Network Model

We assume a similar network model as in [7], [11]–[13]. The UTSN is partitioned into many cells, each consisting of many sensor nodes and one master node. We assume that master and sensor nodes know their respective locations and also affiliated cells. The localization requirement is fundamental in most sensor network applications and can be satisfied by many existing techniques such as [14], [15]. There might be sensor nodes in the overlapping area of multiple cells, in which case they are affiliated with all those cells.

Master and sensor nodes differ significantly in their resources. In particular, master nodes have abundant resources in storage, energy (e.g., a heavy-duty battery or solar panel), and computation, while sensor nodes are much more constrained in every regard. In addition, each master node can communicate with neighboring master nodes via relatively long-range and high-rate radios, thus forming an upper-tier multi-hop network.

As in [7], [11], [12], we assume that time is divided into epochs. At the end of each epoch, each sensor node submits to its affiliated master node all the data (if any) it generated during that epoch. We assume that there is no stable communication link connecting the sensor network to the external network owner, so data must be stored at master nodes. The network owner can issue top-\(k\) queries via an on-demand wireless (e.g., satellite) link to some master node(s), which is often both costly and relatively low-rate. As a result, the communication cost incurred by top-\(k\) queries over such on-demand wireless links should be kept as low as possible.

B. Top-k Query Basics

Data generated by sensor nodes may have multiple attributes, each corresponding to one type of sensors or one aspect of a detected event. Each data item can be scored by some scoring functions [9] and ranked based on its score. For sake of simplicity, the following primitive top-\(k\) queries will be considered.

\[(\text{cell} = C) \land (\text{epoch} = t) \land (\text{num} = k) \land (\text{query region} = I_t)\] .

Here \(C\) and \(t\) are the interested cell ID and epoch number, respectively, \(k\) refers to the number of desired data items, and \(I_t\) denotes the physical query region. We will subsequently abuse the notation \(I_t\) to also denote the set of sensor node IDs in the query region. Our assumption here is that both the network owner and master node know the mappings between sensor node IDs and their respective geographic locations. We aim to support fine-grained top-\(k\) queries, in which \(I_t\) may cover one or more random sensor nodes in cell \(C\).

C. Adversary Model

We aim to support authenticity and soundness verification of top-\(k\) query results and refer the readers to the existing rich literature (e.g., [2], [6], [15]–[24]) for other important security issues.
We assume that the adversary has compromised some master and sensor nodes in the UTSN. Since the operations in different cells are independent from each other, the adversary will not gain more from the collaboration of compromised master/sensor nodes in different cells. Without loss of generality, our subsequent discussion thus focuses on a cell $C$ consisting of a master node $M$ and $N$ sensor nodes $\{S_i\}_{i=1}^N$ whose IDs compose a set $\mathcal{I} = \{1, 2, \cdots, N\}$. Among them, we assume that $c < N$ sensor nodes are compromised.

The adversary may launch different attacks through compromised $M$, sensor nodes, or both. In particular, we consider the following attacks in this paper.

- **Attack 1**: Compromised $M$, with the possible assistance of compromised sensor nodes, may return incorrect query results in response to the network owner’s top-$k$ queries.
- **Attack 2**: Compromised sensor nodes may forge data items with extremely high scores such that the data items generated by legitimate sensor nodes will have little chance to appear in the query result.
- **Attack 3**: Compromised sensor nodes may frame a good master node by exploiting our verification mechanism, e.g., deviating from protocol execution, such that the network owner will falsely identify $M$ as malicious.

Different from [7], [11], [12], we do not intend to ensure data confidentiality against master nodes. Many sensor network applications do not require data confidentiality but only query-result authenticity and soundness. For example, intrusion events in a sensor network for battlefield reconnaissance are known to the adversary and thus need not be secret. In other words, the adversary knows that he has been detected, but he can instruct compromised master nodes to return fake and/or unsound query responses so that the network owner cannot precisely determine his itinerary. In such cases, enabling query-result authenticity and soundness verifications becomes a must. Achieving secure top-$k$ query-processing and data confidentiality is still an open challenge.

### III. Problem Statement

In this section, we formulate the problem and introduce our design goals.

#### A. Problem Formulation

To ease the presentation, we assume that during each epoch $t$, each node $S_i \in \{S_i\}_{i=1}^N$ generates $\mu$ data items, denoted by $D_t = \{D_{i,j}\}_{j=1}^\mu$. Our scheme can be easily adapted to support the case in which each node generates different number of data items. The master node $M$ thus receives $N\mu$ data items in the end of epoch $t$, which are denoted by $D = \bigcup_{t=1}^d D_t$. We assume that all the data items generated in cell $C$ during epoch $t$ have mutually different scores. For example, we can break a tie between two different data items by considering their corresponding node IDs or the times when they are generated.

This assumption implies that a unique correct response exists for any top-$k$ query. We will denote by $s_{i,j}$ the score of $D_{i,j}$, i.e., $s_{i,j} = f(D_{i,j})$, where $f(\cdot)$ is a public scoring function [9]. In addition, we will equate $D_{i,j} \leq D_{i',j'}$ with $s_{i,j} \leq s_{i',j'}$ for any $i, i', j, j'$.

Given a query $Q_t = \langle C, t, k, \mathcal{T} \rangle$ as introduced in Section II-B, we define the corresponding candidate data set as $D_t = \bigcup_{t=1}^d D_t$, which contains $\mu_t = n\mu$ candidate data items, where $n = |\mathcal{T}|$. It is possible that there are less than $k$ candidate data items, i.e., $\mu_t < k$. This situation, however, has very little impact on our schemes. For simplicity, we hereafter assume $\mu_t \geq k$ in most descriptions and will point out the additional actions that need be taken for $\mu_t < k$ when appropriate.

Assuming that $M$ returns a query response containing $k$ data items, denoted by $\mathcal{R}_t$, the problem of interest is how the network owner can efficiently verify the compliance of $\mathcal{R}_t$ with the following conditions.

- **Authenticity**: All data items in $\mathcal{R}_t$ were generated by nodes in the query region, or equivalently $\mathcal{R}_t \subseteq D_t$.
- **Soundness**: $\mathcal{R}_t$ contains the top $k$ data items among all the candidates, or equivalently $D_{i,j} > D_{i',j'}$ for all $D_{i,j} \in \mathcal{R}_t$ and $D_{i',j'} \in D_t \setminus \mathcal{R}_t$.

#### B. Performance Metrics

The following performance metrics will be used throughout.

- $P_{\text{det}}$: detection probability: the probability that an incorrect (i.e., forged and/or unsound) top-$k$ query result is detected.
- $C_{\text{cell}}$: in-cell communication cost: the total additional communication energy consumption in bits incurred by enabling verifiable top-$k$ queries in cell $C$ per epoch. Here we assume the same energy consumption in transmitting and receiving every bit across each hop.
- $C_{\text{query}}$: query communication cost: the total additional information in bits transmitted between $M$ and the network owner for enabling verifiable top-$k$ queries. The route connecting $M$ to the network owner may traverse multiple master nodes and the on-demand wireless link. For simplicity, we associate an energy cost of transmitting and receiving every bit with this route, which is usually much larger than that between neighboring sensor nodes.

### IV. Verifiable Top-$k$ Queries

In this section, we present VTQ, which enables the network owner to verify the authenticity and soundness of any top-$k$ query result in UTSNs against a compromised master node. For clarity, we defer the discussion of other attacks launched by compromised sensor nodes to Section V.

#### A. Overview

VTQ is essentially built upon the following two facts.

**Fact 1**: Suppose that each node $S_i$ sorts its data items in the descending order such that $D_{i,j} > D_{i,j+1}$ for all $j \in [1, \mu - 1]$. If $D_{i,j}$ is among the top $k$, so is $D_{i,s}$ for all $x \in [1, j]$; likewise, if $D_{i,j}$ is not among the top $k$, neither is $D_{i,y}$ for all $y \in (j, \mu]$.

**Fact 2**: Any top-$k$ data item is larger than any non-top-$k$ data item in the query region.

Fact 1 implies that adjacent data items generated by the same sensor node are very likely to satisfy or dissatisfy a top-$k$ query.
at the same time. If node $S_i$ has $k_i > 0$ data items among the top $k$, then they must be $D_{i,1}, \ldots, D_{i,k}$. On the other hand, Fact 2 implies that for any two nodes $S_i$ and $S_j$, $i \neq j$, if $D_{i,k+1} > D_{j,1}$, then node $S_j$ has no data item among the top $k$, i.e., $k_j = 0$.

To exploit these two facts, we let each sensor node sort their data items and exchange its highest score with its nearby nodes. Each node then chains adjacent data items with other nodes’ highest scores using a cryptographic hash function. On receiving a top-$k$ query $Q_i$, we require master node $M$ to return some additional information besides the top-$k$ data items in the query result whereby the network owner can verify both the authenticity and soundness of the query result.

For our purpose, we assume that each $S_i$ is preloaded with a distinct initial key $K_{i,0}$ uniquely shared with the network owner. At the end of each epoch $t \geq 1$, $S_i$ generates an epoch key by $K_{i,t} = h(K_{i,t-1})$ and erases $K_{i,t-1}$ from its memory, where $h(\cdot)$ denotes a good hash function. We also introduce an extremely small public value $\chi$ and an extremely large public value $\chi_2$, both out of the known domain of the data score. Assuming that $N = nm$, we partition each cell $C$ into $m$ virtual subcells of equal size and assume that each sensor node knows its affiliated subcell. We denote the $m$ subcells and their respective node ID sets by $\{C_y\}_{y=1}^m$ and $\{J_y\}_{y=1}^m$, respectively.

In what follows, we detail the VTQ design, which consists of four phases. In the data-submission phase, each sensor node preprocesses its sensed data using cryptographic methods for submission. In the subsequent query-processing phase, $M$ answers a top-$k$ query by returning the query result and certain proofs to the network owner. In the final verification phase, the network owner verifies the authenticity and soundness of the query result by examining the proofs.

### B. Data Submission

At the end of each epoch, sensor nodes in each subcell $C_y$ exchange some information about their sensed data. Consider node $S_i$ as an example. Node $S_i$ broadcasts its highest score and node ID within subcell $C_y$ as follows:

$$S_i \rightarrow \ast : i, s_{i,1}.$$  

Here we assume a suitable broadcast authentication protocol like multilevel $\mu$TESLA [16] for secure and reliable transmissions of such broadcast messages.

Node $S_i$ waits for sufficient time to receive all the highest scores $\{s_{j,1}\}_{j \in C_y \setminus \{i\}}$ from all the other nodes in $C_y$. It then sorts its own data scores and the received ones $\{s_{j,1}\}_{j=1}^m \cap \{s_{x,1}\}_{x \in J_y}$ in the descending order, resulting in a list of $\mu + n - 1$ scores, where $n$ is the size of each subcell. Recall our assumption that all the data items generated during each epoch in cell $C$ have different scores. Node $S_i$ then replaces the scores received from other nodes with their corresponding node IDs, resulting in $\mu + 1$ lists of node IDs $L_{i,1}, \ldots, L_{i,\mu+1}$, separated by $S_i$’s own scores $s_{i,1}, \ldots, s_{i,\mu}$. More specifically, for any node ID $x$ appears in $L_{i,1}, L_{i,j}$ $(2 \leq j \leq \mu)$, and $L_{i,\mu+1}$, we have $s_{x,1} > s_{i,1}, s_{i,j-1} > s_{x,1} > s_{i,j}$, and $s_{x,1} < s_{i,\mu}$, respectively. In addition, if $x$ and $y$ both appear in $L_{i,j}$, $x$ is on the left hand side of $y$ if and only if $s_{x,1} > s_{y,1}$. We call each $L_{i,j}$ a auxiliary ID list henceforth.

As a concrete example, suppose that subcell $C_1$ consists of sensor nodes $S_1, S_2$ and $S_3$ with data score sets $\{1, 5, 9\}$, $\{2, 3, 4\}$ and $\{6, 7, 8\}$, respectively. During data submission, node $S_1$ broadcasts its highest score with node ID $(1, 9)$ and receives $(2, 4)$ and $(3, 8)$ from nodes $S_2$ and $S_3$, respectively. Node $S_1$ then sorts its own data scores $\{1, 5, 9\}$ and the received $4$ and $8$ in the descending order, resulting in $(9, 8, 5, 4, 1)$. It then replaces data scores $4$ and $8$ with their corresponding node IDs to obtain $(9, 3, 5, 2, 1)$. The corresponding auxiliary ID lists are then $L_{1,1} = \emptyset, L_{1,2} = (3), L_{1,3} = (2)$, and $L_{1,4} = \emptyset$.

Let $h_*(\cdot)$ denote a message authentication code (MAC) computed using the key at the subscript. Node $S_i$ then binds adjacent data items as well as auxiliary ID sets by computing

$$V_{i,j} = \begin{cases} h_{K_i,1}((\chi || L_{i,1} || D_{i,1})) & j = 1, \\ h_{K_i,1}((D_{i,j-1} || L_{i,j-1} || D_{i,j})) & 2 \leq j \leq \mu \\ h_{K_i,1}((D_{i,\mu} || L_{i,\mu+1} || \chi_2)) & j = \mu + 1. \end{cases} \tag{1}$$

Finally, each $S_i$ submits all its data items to the master node $M$ in the following message.

$$S_i \rightarrow M : i, t, \{L_{i,1}, D_{i,1}, V_{i,1}\},$$

$$\cdots$$

$$\{L_{i,\mu}, D_{i,\mu}, V_{i,\mu}\},$$

$$\{L_{i,\mu+1}, V_{i,\mu+1}\}.$$  

### C. Query-Processing

After receiving a top-$k$ query $Q_i = \langle c, t, k, I_i \rangle$, the master node $M$ first locates the largest $k$ data items in the candidate data set $D_i$, whereby to determine the number of top-$k$ data items for each node $S_j$ (denote by $k_j$). It follows that $\sum_{i \in I_j} k_i = k$. For convenience, we will call a data item qualified (or unqualified) if it is (or not) among the top $k$. Similarly, we will call a sensor node qualified (or unqualified) if it has at least one (or no) qualified data item.

For each qualified node $S_j$ (i.e., $k_j > 0$), $M$ returns the following information as a part of the query response.

- Case 1: if $k_j < \mu$, the information is

$$M \rightarrow \text{network owner : } i, \langle L_{i,1}, D_{i,1}, V_{i,1}\rangle,$$

$$\cdots$$

$$\langle L_{i,k_j+1}, D_{i,k_j+1}, V_{i,k_j+1}\rangle,$$

where $D_{i,1}, \ldots, D_{i,k_j}$ are qualified data items and $D_{i,k_j+1}$ is unqualified but needed for later verification.

- Case 2: if $k_j = \mu$, the information is

$$M \rightarrow \text{network owner : } i, \langle L_{i,1}, D_{i,1}, V_{i,1}\rangle,$$

$$\cdots$$

$$\langle L_{i,\mu}, D_{i,\mu}, V_{i,\mu}\rangle, L_{i,\mu+1},$$

where $D_{i,1}, \ldots, D_{i,\mu}$ are all qualified data items.

In addition, if $M$ does not return any data item from one subcell, the network owner cannot differentiate whether
that subcell indeed has no qualified data or M purposefully skipped them. In view of this situation, VTQ requires M to return some additional information for each subcell without qualified data. Specifically, we call a subcell unqualified if it overlaps with the query region but has no qualified data. The master node M is required to return the largest data item in each unqualified subcell with nodes Jy as follows.

- Case 3: Assuming that node S1 generated the largest data item D1,1 in epoch t among all the nodes in Jy \bigcap I, M need return the following information in the query response.

\[ M \rightarrow \text{network owner} : i, \langle L_{i,1}, D_{i,1}, V_{i,1} \rangle. \]

D. Verification

Upon receiving the query result from M, the network owner first verifies its authenticity by checking the MACs. In particular, for each sensor node Si with at least one data item returned, the network owner derives the corresponding key Ki,t. Then for each data item Dij returned, the network owner recomputes the corresponding Vi,j according to Eq. (1) and compares it with the received one. If the two match, Dij is considered authentic. Since each data item is bound with adjacent data items using MACs, verifying each Vi,j also ascertains that master node M has not inserted any forged data items or skipped any legitimate ones. If all the verifications succeed, the network owner considers the query result authentic, as each key Ki,t is known only to himself and Si.

The network owner proceeds to check the soundness of the query result by examining the relationships among the returned data items and auxiliary ID lists as follows.

- First, the network owner checks if there is at least one data item returned for every subcell that overlaps with the query region.
- Second, the network owner checks if the query result is consistent with Fact 2. In particular, since the information returned for each node S1 follows one of the three cases, the network owner can easily determine ki for S1 as well as the qualified data items, i.e., D1,1, \ldots, D1,ki (Cases 1 or 2), and the unqualified data item D1,ki+1 (Cases 1 or 3), if any. He can then verify if there are indeed total k qualified data items returned. If so, he further checks if the smallest qualified data item is larger than the largest unqualified data item among all those returned.
- Finally, the network owner examines all the auxiliary ID lists Lij contained in the query response to see if M has skipped all the data items for some qualified node. In particular, for each qualified data item, say Di,j with a non-empty auxiliary ID list Lij, the network owner checks whether there is at least one data item returned from node Sz for all x ∈ Lij \bigcap I. If not, the query result is considered unsound. The underlying rationale is very simple. If x ∈ Lij \bigcap I, node Sz must have at least one data item scoring higher than si,j according to the definition of Lij.

If all the above verifications succeed, the network owner considers the query result both authentic and sound.

V. DEFENSES AGAINST COMPROMISED SENSOR NODES

So far we have not considered the impact of compromised sensor nodes for the sake of clarity. In this section, we discuss three attacks launched by compromised sensor nodes and propose corresponding defenses.

A. Forging Auxiliary ID List

Compromised sensor nodes may collude with M to overshadow some qualified data items by forging their auxiliary ID lists. In particular, a compromised sensor node can forge its auxiliary ID lists to cheat the network owner into believing that no other node in the same subcell has qualified data items. Consider the following example. Suppose that the network owner queries the top-2 data items generated by nodes S1 and S2, among which S1 is legitimate and has generated top-2 data items, and S2 is compromised. Node S2 can fake its auxiliary ID lists by setting L1,1 = L2,2 = ∅, meaning that S1 has no data item larger than D2,2. The master node M can then return the top-2 data items of S2 and provide necessary proofs to pass the authenticity and soundness verification as in VTQ.

We propose randomized probing (RP) for the network owner to ask for additional proofs from randomly chosen sensor nodes. In particular, after the query result passes all the verifications in Section IV-D, the network owner randomly chooses \( \theta \geq 1 \) candidate nodes in each subcell that overlaps with the query region, from which no data item has been returned. Let d be the number of subcells that overlaps with the query region. The network owner sends \( \theta d \) chosen node IDs to M, which in turn returns the largest data item and corresponding auxiliary ID list for each of them. More specifically, for each chosen node Si, the master node M need return \( \langle L_{i,1}, D_{i,1}, V_{i,1} \rangle \).

On receiving the \( \theta d \) largest data items and auxiliary ID lists, the network owner first verifies the authenticity for each of them by checking the corresponding MAC as in Section IV-D. If all the information returned is authentic, the network owner proceeds to check if each pair of returned data item and auxiliary ID list are consistent with the query result.

Consider as an example D1,1 and L1,1 returned from node S1 in subcell C1, with nodes Jy. According to VTQ query processing, M must have returned at least one data item from nodes among Jy \bigcap I, i.e., the intersection between subcell C1 and the query region I. Without loss of generality, assume that M has returned data items D1,y from nodes Jy \bigcap I. If there was an overshadowing attack in C1, then M must have omitted data items from at least one node in Jy \bigcap I with data items larger than the smallest one among returned D1,y.

The network owner first checks if Jy \bigcap I contains at least one data item returned has its ID in L1,1. If not, he considers that there was an overshadowing attack in C1. The reason is that any data item among D1,y must be larger than D1,1 and has its corresponding node ID embedded in L1,1 according to VTQ. Assume that L1,1 = (j1, \ldots, jz), where z = |L1,1|. The network owner finds the maximum \( x \in [1, z] \) such that at least one data item is returned from node \( S_{j_x} \).
Then for each \( w \in [1, x - 1] \), the network owner checks if node \( S_{j_w} \) satisfies one of the following two conditions.

- **Condition 1:** \( j_w \notin I_t \), i.e., \( S_{j_w} \) is not in the query region.
- **Condition 2:** At least one data item has been returned from node \( S_{j_w} \).

If not, the network owner considers that there was an overshadowing attack, i.e., node \( S_{j_w} \)’s data items have been overshadowed. If the above verifications succeed for each of the \( \delta \) returned largest data items and auxiliary ID lists, the network owner considers that there was no overshadowing attack. The efficacy of randomized probing is analyzed in Section VI-B.

**B. Forging Data with Extremely High Scores**

Compromised sensor nodes may also overshadow some qualified data items by forging data items with extremely high scores. In particular, compromised sensor nodes in cell \( C \) each submits \( \mu \) fake data items with extremely high scores to \( M \), which are properly authenticated and chained as in VTQ. If any compromised node appears in the query region and \( k \) is small, the data from legitimate sensor nodes will have little chance to appear in the query result and thus be overshadowed. It is fundamentally difficult to tell if a data item is fake or legitimate without special assumptions. The only feasible solution is to tolerate such fake data items while retrieving the true top-\( k \) data items generated by legitimate sensor nodes.

Our defense is to let the network owner query more data items than needed to tolerate possible forged data items from compromised sensor nodes. By doing so, the quality of data queries will not be significantly affected as long as the query result contains the true top-\( k \) data items generated by legitimate sensor nodes. Moreover, the network owner could analyze all the returned data items offline using advanced statistical technique to detect compromised sensor nodes.

The remaining challenge is how to minimize the query overhead while at the same time ensuring that the query result contains the true top-\( k \) data items without knowing which sensor nodes are compromised. In what follows, we introduce QC, a query conversion scheme that converts a original top-\( k \) query \( Q_t = (C, t, k, I_t) \), into a \( \delta \)-constrained top-\( k' \) query \( Q_t^\delta = (C, t, k', I_t, \delta) \), where \( C, t, k, I_t \) have the same meanings as in the original top-\( k \) query definition, and \( \delta \leq \min(\mu, k) \) denotes the maximum number of data items that can be returned from any single node. Alternatively, we can view \( Q_t^\delta \) as the top-\( k' \) query over the candidate data set \( \{D_{i,j} | i \in I_t, 1 \leq j \leq \delta \} \), which is a subset of the original candidate data set \( D_t \).

The network owner sends \( Q_t^\delta \) to \( M \), which in turn returns the corresponding query result under VTQ. On receiving the query result, the network owner can verify its authenticity and soundness verification as in VTQ. The probability that the query result contains the true top-\( k \) data items, denoted by \( P_{true} \), is jointly determined by the number of compromised sensor nodes in the query region and the choice of \( \delta \) and \( k' \), which will be analyzed in Section VI-C.

**C. Framing Legitimate Master Node**

Our previous discussion focuses on detecting a compromised master node \( M \) which might be assisted by some compromised sensor nodes. The adversary, however, may also exploit our techniques to frame some legitimate master nodes. For example, assume that the adversary only compromise some sensor nodes in cell \( C \) while the master node \( M \) is legitimate. The compromised sensor nodes can frame \( M \) by sending it data authenticated using incorrect keys. Since \( M \) does not know the correct keys, it cannot detect such misbehavior. Consequently, the network owner will falsely identify \( M \) as malicious.

Our previous works [12], [25] suggest that an effective countermeasure against the framing attack is to let each sensor node and master node digitally sign every message transmitted and received. In case of dispute, the network owner can detect the misbehaving entities by analyzing related messages and signatures. This solution, however, requires public-key operations not suitable for resource-constrained sensor nodes.

Now we introduce a symmetric-key based solution (called RW) to defend against the framing attack, which relies on
randomly chosen nodes serving as witnesses for sensor nodes submitting sensed data to the master node. We assume that every sensor node can work in the promiscuous mode. Consider Fig. 2 as an example. Suppose that node $S_i$ wants to submit a message $msg$ to the master node $M$ in epoch $t$. Each intermediate node $S_j$ along the route that overhears the message checks if

\[ h_{K_{j,t}}(i||j||t) \mod X \leq Y, \]  

where $X \geq Y$ are two integer-valued system parameters. If so, node $S_j$ computes a testimony for $msg$ as follows.

\[ T_{i,j,t} = h_{K_{j,t}}(msg||i||t). \]

Each node submits all the testimonies generated in epoch $t$ to $M$ at the beginning of epoch $t+1$. We can see that $\rho = Y/X$ determines the ratio of witness nodes of node $S_i$ in epoch $t$ among all the intermediate nodes that overheard the message $msg$. Since $K_{j,t}$ is only known to $S_j$ and the network owner, the adversary cannot predict which nodes will be chosen as witnesses for $msg$. The adversary thus cannot compromise all the witness nodes in advance before framing a legitimate master node.

Later if there is a dispute between node $S_i$ and $M$, the network owner can retrieve all the related testimonies to determine whether $M$ is malicious or framed. Continue the previous example. Suppose that $M$ later returns a top-$k$ query result based on $msg$ and is detected as inauthentic by the network owner. The network owner first derives the IDs of all the witnesses of node $S_i$ during epoch $t$ according to Eq. (3) and then requires $M$ to return the original message $msg$ as well as all the testimonies on message $msg$. The network owner then recomputes each testimony according to Eq. (4) using the corresponding key of each witness node $S_j$. If a majority of the testimonies indicate that $msg$ is indeed the original message submitted by node $S_i$, the network owner considers $M$ is framed and excludes node $S_i$ from future query region.

It is worth noticing that the nodes far away from the master node will have more witnesses than those close to the master node for the same ratio $\rho = Y/X$ because its messages will be overhead by more intermediate nodes. The network owner may assign different values of $\rho$ for different nodes according to their distances to the master node.

## VI. Performance Analysis

In this section, we analyze the efficacy and overhead of the proposed schemes.

### A. Analysis of VTQ

We first have the following theorem regarding the detection capability of VTQ against a compromised master node.

**Theorem 1:** Assuming that none of the sensor nodes are compromised, VTQ can detect any incorrect top-$k$ query result returned by a compromised master node.

**Proof:** Consider a queried node $S_i$, which has $k_i$ qualified data items $\{D_{i,j}\}_{j=1}^{k_i}$. Since the adjacent data items are bound with MACs for which $M$ does not have the corresponding key $K_{i,t}$, $M$ cannot insert forged data items into or omit legitimate ones from $\{D_{i,j}\}_{j=1}^{k_i}$ without being detected during the authenticity check.

Now assume that the master node has returned authentic but an unsound query result, from which the network owner derives an unsound top-$k$ query result containing $k$ data items with the lowest score $s'$ among them. Let $s$ denote the lowest score among the $k$ data items in the correct query result. If $s' > s$, there must be less than $k$ data items with score no lower than $s'$ in the query region, so it is impossible for $M$ to find $k$ authentic data items with the lowest score $s'$, leading to a contradiction. On the other hand, if $s' < s$, the master node $M$ should have deleted at least one data item in the query region with score higher than $s'$. Suppose that $M$ has deleted $D_{i,j}$ with $s_{i,j} > s'$ and that node $S_i$ is in subcell $C_Z$. There are two cases:

- If $M$ returned no data item from node $S_i$, then $M$ must have returned at least one data item generated by some other node in the same subcell with score lower than $s'$, say $D_{x,y}$ with $s_{x,y} < s'$. Since $s_{i,j} > s'$, we have $s_{i,j} > s_{x,y}$ and node ID $i$ must have been embedded into one auxiliary ID list among $I_{x,1}, \ldots, I_{x,y}$ and returned to the network owner, from which the network owner knows that $M$ omitted some valid data item from node $S_i$.
- If $M$ has returned some data items generated by node $S_i$, say $D_{i,j-y}$, it must have returned one $D_{i,y}$ with $s_{i,y} < s'$ to pass the soundness check, which means that it must also return $D_{i,1}, \ldots, D_{i,y}$ to pass the authenticity check. Since $s_{i,j} > s' > s_{i,j-y}$, we have $j < y$ and $D_{i,j}$ must have been returned, leading to a contradiction.

Therefore, the network owner can detect any unsound query result as well.

Assume that each node ID is of $l_{id}$ bits, each score is of $l_{score}$ bits, $h_s(\cdot)$ is of $l_{mac}$ bits, and the average number of hops between a sensor node and $M$ is $L$. We then have the following theorem regarding the in-cell communication costs of VTQ.

**Theorem 2:** The in-cell communication cost of VTQ is given by

\[ C_{\text{cell}} = Nn(l_{id} + l_{score} + l_{mac}) + N(\mu + 1)Ll_{mac} + N(n - 1)Ll_{id}, \]

where $n$ is the number of nodes in each subcell.
Proof: The in-cell communication cost of VTQ consists of two parts: $C_{\text{score}}$, the cost incurred by exchanging highest scores within each subcell, and $C_{\text{data}}$, the cost incurred by transmitting data items and embedded auxiliary node ID lists to $\mathcal{M}$. Note that we do not consider the cost for transmitting epoch number, original data items, and corresponding node IDs because they have to be submitted even without VTQ.

Under VTQ, each node need broadcast its node ID and highest score within its subcell. Assume that $\mu$ TESLA [16] is used for broadcast authentication. Each broadcasted message is of $l_{\text{id}} + l_{\text{score}} + l_{\text{mac}}$ bits. Assume that the simplest broadcasting mechanism is used, in which each node rebroadcasts the message it received once. The $C_{\text{score}}$ is then given by

$$C_{\text{score}} = Nn(l_{\text{id}} + l_{\text{score}} + l_{\text{mac}}),$$

Since each node ID appears in $n − 1$ auxiliary ID lists, the total number of node IDs in all auxiliary ID lists is thus $N(n − 1)$. In addition, each node need transmit $\mu + 1$ MACs to $\mathcal{M}$ (cf. Eq. (1)). We thus have

$$C_{\text{data}} = N(\mu + 1)Ll_{\text{mac}} + N(n − 1)Ll_{\text{id}}.$$  

It follows that

$$C_{\text{cell}} = C_{\text{score}} + C_{\text{data}} = Nn(l_{\text{id}} + l_{\text{score}} + l_{\text{mac}}) + N(\mu + 1)Ll_{\text{mac}} + N(n − 1)Ll_{\text{id}}.$$  

We have only been able to derive an upper bound for the query communication cost of VTQ for a special case.

**Theorem 3**: Assuming that the query region comprises $g$ subcells and that each candidate data item is equally likely to be among the top $k$. The expected query communication cost under VTQ is bounded by

$$C_{\text{query}} \leq kl_{\text{mac}} + gn(1 − p_o)(l_{\text{data}} + l_{\text{mac}}) + g(1 − \alpha)\beta(\beta − 1)l_{\text{id}} + g\alpha(l_{\text{id}} + l_{\text{data}} + l_{\text{mac}}),$$

where

$$p_o = \frac{(gn − 1)\mu}{(g\mu)^k}, \quad \alpha = \frac{(g−1)n\mu}{(g\mu)^k},$$

and $\beta = \frac{n(1−p_o)}{1−\alpha}$.

Proof: The query communication cost of VTQ consists of three parts: the communication cost incurred by transmitting data items and indexes, denoted by $C_{\text{1}}$, and the communication cost incurred by transmitting embedded auxiliary ID lists, denoted by $C_{\text{2}}$, and the communication cost incurred by transmitting data item and MAC for unqualified subcell, denoted by $C_{\text{3}}$.

We first analyze $C_{\text{1}}$. Since there are total $gn\mu$ data items generated in $I_\ell$ during epoch $t$, the probability of a node $S_i$ having no top-$k$ data item is given by

$$p_o = \frac{(gn−1)\mu}{(g\mu)^k}.$$  

There are thus $gnp_o$ qualified nodes and $gn(1−p_o)$ unqualified nodes on average. For each of the top-$k$ data items, one MAC need be transmitted. For each qualified node, at most one additional data item and one index need be transmitted. We thus have

$$C_1 \leq kl_{\text{mac}} + gn(1 − p_o)(l_{\text{data}} + l_{\text{mac}}),$$

where $p_o$ is given in Eq. (9).

We now analyze $C_{\text{2}}$. Similar to the analysis of $p_o$, the probability that a subcell has no top-$k$ data item is given by

$$\alpha = \frac{(g−1)n\mu}{(g\mu)^k}.$$  

The expected numbers of subcells with at least one top-$k$ data items is thus $g(1 − \alpha)$. On average, each such subcell has $\beta = \frac{n(1−p_o)}{1−\alpha}$ qualified nodes, each of which has its ID embedded in at most $\beta$ dual auxiliary ID lists. We thus have

$$C_2 \leq ng(1 − \alpha)\beta(\beta − 1)l_{\text{id}}.$$  

We now derive $C_{\text{3}}$. The expected number of subcells with no top-$k$ data item is $gα$. For each of them, the master node need return one node ID, one data item, and one MAC. We thus have

$$C_3 = g\alpha(l_{\text{id}} + l_{\text{data}} + l_{\text{mac}}),$$

where $\alpha$ is given in Eq. (11).

Combining Eqs. (10), (12), and (13), we have

$$C_{\text{query}} = C_{\text{1}} + C_{\text{2}} + C_{\text{3}} \leq kl_{\text{mac}} + gn(1 − p_o)(l_{\text{data}} + l_{\text{mac}}) + ng(1 − \alpha)\beta(\beta − 1)l_{\text{id}} + g\alpha(l_{\text{id}} + l_{\text{data}} + l_{\text{mac}}),$$

where $p_o$ is given in Eq. (9), $\alpha$ is given in Eq. (11), and $\beta = \frac{n(1−p_o)}{1−\alpha}$.

We have not been able to find a closed-form solution for more general cases, which we will evaluate using simulations in Section VII.

**B. Analysis of RP**

We have the following theorem regarding the detection probability of RP against the overshadowing attack.

**Theorem 4**: Assume that $c$ out of $N$ sensor nodes are compromised. The detection probability of RP against overshadowing attack is bounded by

$$P_{\text{det}} > 1 − \left(\frac{c}{N}\right)^\theta.$$  

Proof: Since $c \ll N$, we can view each probed sensor node as being compromised with probability $p_c = c/N$. Assume that the adversary launched overshadowing attacks in $e \geq 1$ subcells. Consider one such subcell $\mathcal{C}_y$ as an example. Since the network owner probes $\theta$ randomly chosen nodes in each subcell, he cannot detect the overshadowing attack in $\mathcal{C}_y$ if all the probed nodes are compromised, which happens with probability $\left(\frac{c}{N}\right)^\theta$. He cannot detect any overshadowing attack in the query region if all $\theta c$ probed nodes are compromised. We thus have

$$P_{\text{det}} = 1 − \left(\frac{c}{N}\right)^\theta.$$  


which achieves maximum value \( P_{\text{det}} = 1 - \left( \frac{c}{N} \right)^{\theta} \) when \( e = 1 \).

We now estimate the communication cost incurred by RP. Consider a probed node \( S_t \) as an example, from which one data item \( D_{i,t} \), one MAC \( V_{i,t} \), and one auxiliary ID list \( L_{i,t} \) need to be returned. Since \( S_t \) is randomly chosen, the expected number of IDs in \( L_{i,t} \) is \( (n - 1)/2 \), i.e., about half of the nodes have highest scores higher than \( s_{i,t} \). We thus have

\[
C_{\text{RP}} = \theta d(l_{\text{data}} + l_{\text{mac}} + \frac{(n - 1)l_{\text{id}}}{2}),
\]

where \( d \) is the number of subcells that overlap with the query region.

C. Analysis of QC

The following theorem is about the effectiveness of QC.

**Theorem 5:** Assume that \( I_t = \Omega \) and that \( c \) out of \( N \) sensor nodes are compromised, each of which generates up to \( \mu \) data items with extremely large values. If the network owner converts a top-\( k \) query \( Q_k = (C, t, k, I_t) \) into \( \delta \)-constrained top-\( k' \) query, the probability that the query result of \( Q_{k'} \) contains the true top-\( k \) data items generated by legitimate sensor nodes is given by

\[
P_{\text{true}} = \begin{cases} 
0 & \text{if } \delta c + k > k', \\
\frac{p_k}{P_1} & \text{otherwise},
\end{cases}
\]

where

\[
P_1 = \sum_{0 \leq x_j \leq \delta \forall j \in [1, N-c]} \prod_{j=1}^{N-c} \Pr[k_j = x_j],
\]

\[
P_2 = \sum_{\sum_{j=1}^{N-c} x_j = k} \prod_{j=1}^{N-c} \Pr[k_j = x_j],
\]

\[
\Pr[k_j = x] = \binom{\mu}{x} p^x (1 - p)^{\mu - x},
\]

\[
p = \frac{k}{(N - c) \mu}. \tag{17}
\]

**Proof:** First, we have \( P_c = 0 \) if \( \delta c + k > k' \), since \( k' \) is not large enough to tolerate all the forged data items from compromised nodes in \( \Omega \).

Now consider the case \( \delta c + k \leq k' \). Without loss of generality, denote by \( i_1, \cdots, i_{N-c} \) the IDs of legitimate sensor nodes. Also denote by \( k_j \) the number of true top-\( k \) data items generated by node \( S_{i_j} \). The query result of \( Q_{k'} \) contains the true top-\( k \) data items from the legitimate sensor nodes if \( k_j \leq \delta \), for all \( j \in [1, N-c] \). Assume that each data item is equally likely to be among the top-\( k \). Since there are total \( (N - c) \mu \) data items generated by legitimate sensor nodes, the probability of any data item being among the true top-\( k \) is given by

\[
p = \frac{k}{(N - c) \mu}. \tag{17}
\]

When \( p \) is small, whether each data item being among the true top-\( k \) can be viewed as independent event. We can thus approximate \( k_j \) as a Binomial random variable with p.d.f. given by

\[
\Pr[k_j = x] = \binom{\mu}{x} p^x (1 - p)^{\mu - x}, \quad 0 \leq x \leq \mu, \tag{18}
\]

Denote by \( E_1 \) the event that \( k_j \leq \delta \) for all \( j \in [1, N-c] \) and \( E_2 \) the event that \( \sum_{j=1}^{N-c} k_j = k \). We have

\[
\Pr[E_1 \mid E_2] = \frac{\Pr[E_1 \cup E_2]}{\Pr[E_2]} \tag{19}
\]

We then have

\[
\Pr[\text{true}] = \Pr[1 \leq \delta, \cdots, k_{N-c} \leq \delta, \sum_{j=1}^{N-c} k_j = k] = \sum_{0 \leq x_j \leq \delta, \forall j \in [1, N-c]} \Pr[k_j = x_j, \forall j \in [1, N-c]] \tag{20}
\]

where \( \Pr[k_j = x] \) is given in Eq. (18).

Similarly, we have

\[
\Pr[E_2] = \Pr[\sum_{j=1}^{N-c} k_j = k] = \sum_{\sum_{j=1}^{N-c} x_j = k} \prod_{j=1}^{N-c} \Pr[k_j = x_j] \tag{21}
\]

where \( \Pr[k_j = x_j] \) is given in Eq. (18).

Substituting Eqs. (20) and (21) into Eq. (19), we can then obtain Eq. (16) and prove the theory.

**VII. Simulation Results**

In this section, we evaluate the performance of the proposed schemes using simulations.

We assume a cell of \( 1000 \times 1000 \text{m}^2 \) with 400 sensor nodes randomly distributed and a master node at the center. Each sensor node has a transmission range of 100m, leading to an average distance to the master node of \( L = 3.7 \) hops. We partition the cell into 25 subcells, each containing 16 sensor nodes. We also assume error-free and collision-free packet transmissions. For our purpose, the simulation code is written in C++, and each data point represents an average of 100 simulation runs each with a different random seed. Table I

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>400</td>
<td>( k )</td>
<td>10</td>
<td>( n )</td>
<td>100</td>
<td>( \mu )</td>
<td>10</td>
</tr>
<tr>
<td>( l_{\text{data}} )</td>
<td>160</td>
<td>( l_{\text{score}} )</td>
<td>16</td>
<td>( l_{\text{mac}} )</td>
<td>160</td>
<td>( l_{\text{id}} )</td>
<td>16</td>
</tr>
<tr>
<td>( c )</td>
<td>10</td>
<td>( s_c )</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Moreover, the in-cell communication cost increases linearly with the theoretical results match the simulation results very well. Moreover, the in-cell communication cost decreases rapidly as the number of subcells. We can see that the theoretical results match the simulation results very well. Moreover, when $k$ is small, the larger $m$ is, the higher the query communication cost. This is because when $k$ is small, there will be many unqualified subcells, for each of which some information need be returned, leading to higher query communication cost. On the other hand, when $k$ is large, the smaller $m$ is, the higher the query communication cost. The reason is that as $m$ increases, the number of unqualified subcells decreases, and the average number of IDs in each auxiliary ID list increases. Therefore, more node IDs are embedded into the data items returned, leading to higher communication cost. In general, small $m$ may lead to higher query communication cost when $k$ is small, so does large $m$ when $k$ is large.

Fig. 5(b) shows the theoretical bound and simulation results of the query communication cost of VTQ varying with the number of nodes queried, denoted by $q$. We can see that the query communication cost increases as the number of nodes in the query region increases. The reason is that for fixed $k$, the larger the query region is, the more candidate subcells, the more unqualified subcells, and the higher the query communication cost, and vice versa. In addition, we can also see that when $k$ is relatively large, e.g., $k = 100$, the query communication cost increases rapidly as $q$ increases from 20 to 100 and then slowly as $q$ further increases. The reason is that the number of qualified nodes increases as $q$ increases before $q$ exceeds $k$. For each additional qualified node, one additional data item need be returned under VTQ, leading to a rapid increase in query communication cost. After $q$ exceeds $k$, the number of unqualified subcells increases slowly as $q$ further increases, leading to a slow increase in query communication cost.

**B. Performance of RP**

Fig. 5(a) shows the theoretical and simulation results of the detection probability of RP against the overshadowing attack varying with $\theta$, the number of nodes probed in each subcell. We can see that the theoretical results match the simulation result very well. Moreover, the more nodes probed in each candidate subcell, the higher the detection probability against
overshadowing attack. The reason is that the overshadowing attack cannot be detected only if all the probed nodes are compromised and the probability that at least one probed node is not compromised increases as $\theta$ increases. We can see that even 10% of the sensor nodes are compromised, the detection probability is higher than 0.98 when $\theta = 2$ and close to one as $\theta$ further increases. It is thus unnecessary to choose a large $\theta$ in practice.

Fig. 5(b) shows the theoretical and simulation results of the additional communication cost incurred by RP varying with the number of candidate subcells. It is easy to see that the communication cost increases linearly as the number of candidate subcells increases, which is anticipated. This also implies that for a fixed query region, the communication cost incurred by RP increases as the total number of subcells increases as there will be more candidate subcells.

C. Performance of QC

Fig. 6(a) shows the theoretical results and simulation results of $P_{true}$, the probability of the query result containing true top $k$ varying with $c$, the number of compromised sensor nodes, where $k = 100$ and $k' = 300$, respectively. We can see that $P_{true}$ first decreases slowly as $c$ increases and then drops to zero after $c$ exceeds 65. The reason can be explained as follows. When $c$ is smaller than the threshold $(k' - k)/\delta$ (cf. Eq. (16)), the query result contains the true top-$k$ data items if none of the legitimate sensor nodes have more than $\delta$ qualified data items. As the number of compromised nodes increases, the number of legitimate nodes decreases, and the probability of at least one legitimate sensor node has more than $\delta$ increases, as the same number of qualified data items are allocated among fewer legitimate nodes. Once $c$ exceeds $(k' - k)/\delta$, the query result can no longer tolerate all the $c\delta$ forged data items, and $P_{true}$ thus drops to zero. Moreover, we can see that the choice of $\delta$ affects the $P_{true}$. In particular, when $\delta = 2$, $P_{true}$ is about 0.5 even if none of the sensor nodes are compromised. The reason is that it is very likely that a legitimate sensor node can have more than two qualified data items. On the other hand, when $\delta = 3$, $P_{true}$ is higher than 0.95 when the number of compromised sensor nodes is smaller than $(k' - k)/\delta$, but drops to zero as $c$ exceeds 65.

Fig. 6(b) shows $P_{true}$ varying with the $k'$, the number of compromised sensor nodes, where $k = 10$. We can see that the probability remains zero before $k'$ exceeds the threshold $k + c\delta$, as $k'$ is not large enough to tolerate all the $c\delta$ forged data items. After $k'$ exceeds the threshold, the probability significantly increases and remains constant as $k'$ further increases. The probability is not one because it is still possible that one sensor node has more than $\delta$ qualified data items.

In general, smaller $\delta$ leads to lower $P_{true}$ but could tolerate more compromised sensor nodes.

D. Performance of WT

To simulate the performance of WT, we assume the worst case in which the sensor node that launches the framing attack is one hop away from the master node and thus has the least number of witnesses on average for fixed witness ratio $\rho = Y/X$.

Fig. 7(a) shows the detection probability against framing attack varying with the witness ratio $\rho$. We can see that the detection probability increases as the witness ratio increases. This is anticipated since the higher the ratio $\rho$ is, the more witnesses are selected for each message transmission. The network owner can detect the framing attack as long as the number of legitimate witnesses is larger than that of compromised witnesses. Moreover, the higher the node density, the more neighbors each node has, the more witnesses, and vice versa. In practice, the ratio $\rho$ should be chosen according to the node density, i.e., the higher the node density, the lower the ratio.

Fig. 7(b) shows the communication cost incurred by WT varying with $\rho$. We can see that the communication cost increases linearly as the witness ratio $\rho$ increases. The reason is that each witness node need transmit one testimony. So the higher the ratio $\rho$ is, the more witnesses for each message transmission, the higher the communication cost, and vice versa. Since each testimony $T_{i,j,t}$ is essentially a MAC, which is much shorter than a data item, the communication cost incurred by transmitting testimonies is relatively small in comparison with that incurred by data submissions.

E. Discussion

We summarize the evaluation results as follows.

- VTQ can detect any fake and/or unsound top-$k$ query result returned by a compromised master node provided that none of the sensor nodes are compromised. The in-cell and query communication costs of VTQ can be adjusted by choosing proper $m$, the number of subcells.
Small \( m \) leads to high in-cell communication cost and low query communication cost when \( k \) is small, while large \( m \) leads to low in-cell communication cost and high query communication cost when \( k \) is large.

- RP can detect unsound top-\( k \) query result returned by colluding compromised master and sensor nodes with very high probability and incurs a low communication cost.
- QC can tolerate forged data items from compromised sensor nodes by increasing the number of data items queried while limiting the number of qualified data items that can be returned from each candidate node.
- WT can detect possible framing attacks against a legitimate master node with high probability and incurs a low communication cost.

In practice, all four schemes should be deployed together to enable verifiable top-\( k \) query processing in UTSNs. Built upon symmetric cryptographic primitives, our schemes are very suitable and practical for resource-constrained sensor networks.

VIII. RELATED WORK

In this section, we discuss some work most germane to our work.

Top-\( k \) queries are a common and important type of queries in sensor networks. Tremendous efforts have been devoted to realize efficient top-\( k \) query processing in sensor network, see for example [9], [10], [26]–[29]. These works nevertheless do not take security issues into account.

Verifiable data queries in UTSNs received attention only recently. In [7], Sheng and Li proposed a novel scheme to enable verifiable privacy-preserving one-dimensional range queries in UTSNs, which is subsequently improved by Shi et al. in [11]. Secure multi-dimensional range queries are later addressed in [12], [13], [31], [32]. None of these schemes can be applied to top-\( k \) queries.

Secure top-\( k \) queries can be viewed as a special instance of secure aggregation. In [33], Nath et al. proposed a set of secure aggregation schemes for wide-area sensing, including top-\( k \) queries. Their schemes rely on public key cryptographic operation, i.e., RSA encryption, and are thus unsuitable for resource-constrained sensor networks.

Our work is also loosely related to secure data outsourcing [34], in which a data owner outsources its data to a third-party service provider answering the data queries on behalf of the data owner. Significant effort has been devoted to ensure query integrity, i.e., that a query result was indeed generated from the outsourced data and contains all the data satisfying the query (the soundness requirement). Many techniques were proposed to realize a wide range of data queries, such as relational query [35]–[37], location-based range queries [38], [39], shortest path queries [40], and moving kNN queries [40]. None of these schemes consider top-\( k \) queries and thus are not applicable to our scenario.

IX. CONCLUSION

In this paper, we present a suite of novel schemes to secure top-\( k \) queries in UTSNs against a wide range of attacks from compromised master and/or sensor nodes. The proposed schemes enable the network owner to verify the authenticity and soundness of any top-\( k \) query results. Detailed analysis and simulation results confirm the high efficacy and efficiency of the proposed schemes. In the future, we intend to investigate the verifiability of other types of data queries in UTSNs.

ACKNOWLEDGMENT

This work was supported in part by the US National Science Foundation under grants CNS-0844972 (CAREER) and CNS-1117462. We would also like to thank anonymous reviewers for their constructive comments and helpful advice.

REFERENCES


