Top-*k* Query Result Completeness Verification in Tiered Sensor Networks

Chia-Mu Yu, Guo-Kai Ni, Ing-Yi Chen, Erol Gelenbe, *Life Fellow, IEEE*, and Sy-Yen Kuo, *Fellow, IEEE*

***Abstract*— Storage nodes are expected to be placed as an intermediate tier of large scale sensor networks for caching the collected sensor readings and responding to queries with benefits of power and storage saving for ordinary sensors. Nevertheless, an important issue is that the compromised storage node may not only cause the privacy problem, but also return fake/incomplete query results. We propose a simple yet effective dummy reading-based anonymization framework, under which the query result integrity can be guaranteed by our proposed verifiable top-*k* query (VQ) schemes. Compared with existing works, the VQ schemes have a fundamentally different design philosophy and achieve the lower communication complexity at the cost of slight detection capability degradation. Analytical studies, numerical simulations, and prototype implementations are conducted to demonstrate the practicality of our proposed methods.**

***Index Terms*— Sensor networks, query result completeness, authentication.**

I. Introduction

A. Tiered Sensor Networks

In sensor networks for data collection, since there could be unstable connection between the authority (or network owner) and network, a middle tier with the purpose of caching the sensed data for data archival and query response becomes necessary.

The network model of this paper is illustrated in Fig. 1, where the authority can issue queries to retrieve the sensor readings. The middle tier is composed of a small number of storage-abundant nodes [24], called *storage nodes*. The bottom tier consists of a large number of resource-constrained ordinary sensors that sense the environment.

Manuscript received February 27, 2013; revised May 29, 2013, August 15, 2013, and October 29, 2013; accepted October 31, 2013. Date of publication November 14, 2013; date of current version December 19, 2013. This work was supported in part by the National Science Council of Taiwan under Grant NSC 99-2221-E-002-108-MY3 and in part by the Cloud Computing Systems and Software Development Project of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of China. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wanlei Zhou.

C.-M. Yu is with the Department of Computer Science and Engineering, Yuan Ze University, Taoyuan 320, Taiwan (e-mail: chiamuyu@saturn.yzu.edu.tw).

G.-K. Ni and S.-Y. Kuo are with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: t5599007@ntut.edu.tw; sykuo@cc.ee.ntu.edu.tw).

I.-Y. Chen is with the Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 106, Taiwan (e-mail: ichen@ntut.edu.tw).

E. Gelenbe is with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mail: e.gelenbe@imperial.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2013.2291326

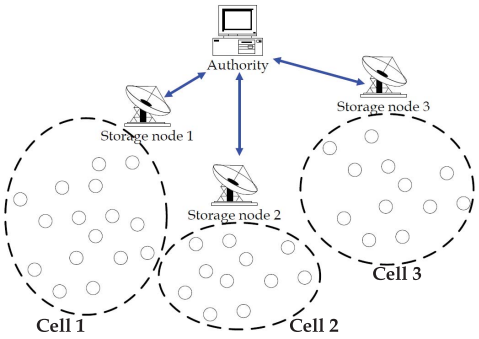


Fig. 1. A conceptual illustration of the tiered sensor network. Each cell (dashed circle) itself forms a multi-hop network.

In the above tiered architecture, sensor nodes are usually partitioned into disjoint groups, each of which is associated with a storage node. Each group of sensor nodes is called a *cell*. The sensor nodes in a cell form a multi-hop network and always forward the sensor readings to the associated storage node. The storage node keeps a copy of received sensor readings and is responsible for answering the queries from the authority. An example of the tiered architecture can be found in Fig. 1.

B. Security Issues in Tiered Sensor Networks

In tiered sensor networks, the authority issues proper queries to retrieve the desired portion of sensed data. We restrict ourselves in this paper to discussing top-*k* query, which is one of the most intuitive and commonly used queries. Top-*k* query [29] can be used to extract the extreme sensor readings.

Nonetheless, the storage nodes easily become the targets to be compromised because of their significant role in responding to queries. For example, by intercepting the sensor communications, the adversary can obtain the sensed data. By compromising storage nodes, the adversary can also return the falsely injected readings to the authority. The most challenging is that the compromised storage nodes can violate query result completeness, creating an incomplete query result for the authority by replacing some portions of the query result with the other genuine readings. For example, once the storage node 1 is compromised by the adversary, the storage node 1 can be configured by the adversary to always return unqualified sensor readings to the authority. Note that data integrity usually refers to both data authenticity and completeness.

C. Existing Works on Verifiable Queries

Two schemes, *additional evidence* and *crosscheck*, were proposed in [35] as solutions for securing top-*k* query in

tiered sensor networks. While the former generates hashes for each consecutive pair of sensed data for verification purpose, the latter performs network-wide broadcast such that the information about the readings is distributed over the entire network and therefore the query result cannot be manipulated. In particular, the idea behind additional evidence is that if each consecutive pair of sensed data is associated with a hash, once an unqualified sensor reading is used to replace the genuine query result, the authority may know because it can find that there are some missing sensor readings for hash verification. On the other hand, the idea behind crosscheck is that the genuine top-k results are distributed to several sensor nodes. With certain probability, the authority will find a query result incompleteness by checking the other sensor nodes' sensor readings. Hybrid method [35] is a combined use of additional evidence and crosscheck, attempting to balance the communication cost and the query result incompleteness detection capability.

Top-k query result integrity was also addressed in [37], where distributed data sources generate and forward the sensed data to a proxy node. Nevertheless, their network model differs from ours because in [37] it is a trusted single proxy node that generates the integrity verification materials whereas in our consideration there is no trusted central authority like proxy node in [37] for such responsibility.

Verifiable query processing is also considered in the context of range query. In [23], the query result completeness is achieved by requiring sensors to send cryptographic one-way hashes to the storage node even when they do not have satisfying readings. In [26], [36], crosscheck was also utilized to secure range query, as in [35]. By converting the verification of whether a number is in a range to several verification of whether two numbers are equal, SafeQ [8] offered an alternative for data retrieval in encryption domain. In SMQ [34], each sensor applies hash operation to the received data and its own data, generating a verifiable object of the sensor readings of the entire network. The basic idea behind SMQ is to construct an aggregation tree over the sensor nodes. Afterward, each sensor node simply aggregates and forwards the sensor readings of all its descendant nodes to its parent node. The notion of stream cipher is used in [27] to have a design of more efficient encrypted data retrieval.

The database community also conducted research on the completeness verification. Nevertheless, similar to [37], all the data to be queried are generated by the single entity. In addition, the prior works on top-k query in [5], [28] focus on the privacy issue, rather than integrity issue.

D. Efficiency and Security Gap

Despite the prior works on verifiable queries, we still have the following concerns:

- In a network of n sensors, Hybrid method [35] incurs tremendous $O(n^2)$ communications.
- Although SMQ [34] can be adapted to verify the top-k query result, an aggregation tree not only needs to be constructed but also needs to remain intact and unchanged. The exact information about the tree topology is also required by the authority. In real world deployment, these requirements are difficult to meet.
- The methods in [35] do not handle the data privacy issue. On the other hand, the *bucket index* used in SMQ [34] leaks the possible value range for each sensor reading, which could be a valuable information, to the adversary.

E. Naïve Approaches

Although the method in [35] can be extended in some straightforward way to the method with data confidentiality guarantee, such extension actually implies some of the other severe weaknesses, which are unacceptable in the design of a verifiable query scheme.

Consider the case that the sensor readings are encrypted by popular encryption functions, like DES and AES. In this case, the storage node is unable to answer the top-k query issued by the authority due to the lack of the numeric order of sensor readings.

On the other hand, consider the case that order-preserving encryption (OPE) [2] is used to encrypt sensor readings. In this case, the numeric order of sensor readings is preserved. Nevertheless, this is achieved by all of the sensors sharing a common OPE key. A consequence of doing so is that once a sensor is compromised, the OPE key is exposed to the adversary and the data confidentiality is completely breached.

As the above two plausible extensions offer the data confidentiality but fail to offer either the capability of answering the query or the resilience against the sensor compromises, we consider that the method in [35] does not have the confidentiality guarantee.

F. Contributions

The Verifiable top-k Query (VQ) schemes based on the novel dummy reading-based anonymization framework are proposed for privacy preserving top-k query result integrity verification in tiered sensor networks. In particular, we make the following contributions:

- A randomized and distributed version of Order Preserving Encryption, rdOPE, is proposed (in Sec. IV-A) to be the privacy foundation of our methods.
- AD-VQ-static (described in Sec. IV-D) achieves the lower communication complexity at the cost of slight detection capability degradation, which could be of both theoretical and practical interests.
- Analytical studies, numerical simulations, and prototype implementation are conducted to demonstrate the practicality of our proposed methods.

Table I shows the comparison among the prior solutions and our proposed methods.

II. System Model

A. Network Model

As shown in Fig. 1, the sensor network considered in this paper is composed of a large number of resource-constrained

TABLE I
COMPARISONS BETWEEN DIFFERENT SCHEMES FOR SUPPORTING VERIFIABLE TOP-k QUERY

	Communication cost	Detection probability	Data confidentiality guarantee	Resilience against topology change
Hybrid Crosscheck [35]	$O(n^2)$	vary significantly	no confidentiality guarantee	can tolerate topology change
SMQ [34]	$O(n)$	always ≈ 1	bucket scheme (partial information leakage)	cannot tolerate any topology change
GD-VQ (this paper)	$O(n)$	vary significantly	guaranteed by rdOPE	can tolerate topology change
LD-VQ (this paper)	$O(n)$	vary significantly	guaranteed by rdOPE	can tolerate topology change
AD-VQ (this paper)	$O(n)$	always ≈ 1	guaranteed by rdOPE	can tolerate topology change
AD-VQ-static (this paper)	$O(n)$	always ≈ 1	guaranteed by rdOPE	can tolerate topology change

sensors and a few *storage nodes*. A *cell* (the dashed circle in Fig. 1) is a connected multihop network composed of a storage node and a number of ordinary sensors. Storage nodes are storage-abundant, can communicate with the authority $\mathcal{A}$ via direct or multi-hop communications, and are assumed to know their affiliated cells. Time on the nodes has been synchronized and is divided into epochs. Note that time synchronization among nodes can be achieved by using algorithms like [16], [25].

With different types of data flow, two phases are considered; the first is *data submission phase*, during which the sensors submit the sensed data to the nearest associated storage node. At the end of each epoch, each sensor enters this phase. The second is *query response phase*, during which the storage node responds to the query issued by $\mathcal{A}$.

Without loss of generality, each node s_i generates μ_i distinct data readings, $d_{i,1} < \cdots < d_{i,\mu_i}$, at each epoch. The value range of data readings is assumed to be within $[1, r]$. This can be publicly known from hardware specification or domain knowledge. From the data collection point of view [10], [24], in a tiered sensor network, there are many cells, each of which operates individually and independently. There would be no difference between the case of single cell and the case of multiple cells. On the other hand, multiple storage nodes serve only the purpose of the data replication. Hence, throughout this paper, our discussions focus on a single cell $\mathcal{C}$, composed of n sensors, $\{s_i\}_{i=1}^n$, and a storage node $s_{\mathcal{M}}$, at a specific epoch t. Note that n could be huge as envisioned by HP CeNSE project [13].

B. Security Model

To achieve *forward security* [23], [26], [34], [36], a common setting is that each sensor s_i shares a secret key $\tilde{k}_i^t$ with $\mathcal{A}$ at each epoch t. This can be accomplished by storing a key $\tilde{k}_i^0$ before sensor deployment. Then, s_i computes $\tilde{k}_i^t = h(\tilde{k}_i^{t-1})$ and erases $\tilde{k}_i^{t-1}$ for each epoch $t > 0$, where $h(\cdot)$ denotes a cryptographic one-way hash function. Because we address the top-k query result at a specific epoch t, $\tilde{k}_i^t$ is replaced by $\tilde{k}_i$ for notational simplicity, unless otherwise stated (*e.g.*, Sec. V-A.2).

After node compromises, all the information stored in the compromised nodes will be exposed to the adversary. In particular, in Secs. IV-B, IV-C, and IV-D, the adversary is assumed to take full control of the storage node and be able to manipulate its computation result and communication contents. In Sec. IV-E, the scenario where the adversary compromises ordinary sensors is discussed. The goal of the adversary is to breach at least one of the data privacy, authenticity, and completeness. Since this paper focuses on the design for securing top-k query, we assume that the other security primitives such as broadcast authentication [20], key establishment [30], and anomaly detection [1], [11], [12] are applicable.

We particularly note that the keyed hash functions used in this paper are keyed-hash message authentication code (HMAC). Consider two parties sharing a secret key k. If the message m to be communicated is associated with $HMAC_k(m)$, the use of HMAC naturally guarantees the data authenticity and integrity.

C. Query Model

For the top-k query, although in general we need to consider a ranking function [29], which is used to output the ranking scores of data items, to ease the presentation, we assume that $\mathcal{A}$ instead asks $s_{\mathcal{M}}$ to return the data readings with the first k highest values of the corresponding cell.

D. Problem Statement

Suppose $\mathcal{A}$ issues a top-k query to $s_{\mathcal{M}}$. Let $\mathcal{B} = \{d_{i,j} | 1 \le i \le n, 1 \le j \le \mu_i\}$ be the set of sensor readings of entire network. The objective is to obtain the top-k result Ω_k that fulfills the following requirements:

- **privacy**: $\mathcal{B}$ cannot be known by $s_{\mathcal{M}}$, and moreover, $\{d_{i,j} | 1 \le j \le \mu_i\}$ can only be known by s_i.
- **authenticity**: $\Omega_k \subseteq \mathcal{B}$. The readings in Ω_k are from sensors $\{s_i\}_{i=1}^n$.
- **completeness**: $\min \Omega_k \ge \max(\mathcal{B} \setminus \Omega_k)$. No readings smaller than the minimum element in Ω_k will be accepted by $\mathcal{A}$.

E. Performance Metrics

The following performance metrics are used to evaluate the integrity verification methods:

- **detection probability**, $P_{\mathtt{det}}^{\mathtt{X}}$: the probability that an inauthentic or incomplete query result is detected by $\mathcal{A}$ in the X scheme.
- **communication cost**, $C^{\mathtt{X}}$: the communication cost $C^{\mathtt{X}}$ of the X scheme is defined as:

$$C^{\mathtt{X}} = C_T^{\mathtt{X}} + \beta C_V^{\mathtt{X}}, \tag{1}$$

where C_T^{X}, C_V^{X}, and β denote the number of bits transmitted between sensors and $s_{\mathcal{M}}$ in data submission phase (in-cell communication cost), the number of bits transmitted between $s_{\mathcal{M}}$ and $\mathcal{A}$ in query response phase (query communication cost), and the query frequency of $\mathcal{A}$ issuing the queries to retrieve data, respectively. For example, $\beta = 0.01$ means that on average the issues a query for every 100 epochs.

F. Evaluating Data Anonymity

There are many notions of anonymization. They are similar to each other but not the same. For example, k-anonymity [21] and differential privacy [9] are the notions of anonymity widely used in the literature. However, they are not suitable for defining data anonymization in our manuscript because the scenarios considered in k-anonymity and differential privacy research are different from the one considered in our manuscript. More specifically, k-anonymity and differential privacy, together with their algorithmic implementations, are designed for statistical databases as means to maximize the query accuracy and minimize the probability of identifying meaningful individual records. An example of statistical databases is the database of medical records, where each record consists of two attributes, name and a binary value indicating sick or not. k-anonymity and differential privacy are used to ensure that the adversary can hardly derive the information about who is sick by issuing aggregate queries.

From the above description, we can know that the statistical database scenario is different from ours; in our manuscript, though different dummy generation techniques are used, all of our proposed methods generate dummy readings, and then submit both genuine and dummy readings to $\mathcal{M}$. The objective of the adversary or the compromised $\mathcal{M}$ is to identify and remove the genuine readings from all of the readings transmitted.

To formally define data anonymization used in our manuscript, we find that the scenario considered in our manuscript is similar to the ones considered in the dummy-based location anonymity research [15], [33], where in LBSs, instead of submitting the exact positions or trajectory information to the service provider, the mobile device submits a number of dummy positions or trajectory information to the service provider, hiding the true information and increasing the user location privacy. Thus, our strategy is to follow the definitions of data anonymity defined in [15], [33] so as to evaluate the level of data anonymity of our proposed methods. In particular, we discuss the definition of data anonymity by considering the following quantities:

- Short term disclosure (SD): This quantity characterizes the ratio of the number of genuine readings and the total number of readings. In essence, SD also represents the probability of successfully identifying the genuine readings under the condition that the adversary does not have any prior knowledge about the distribution of genuine and dummy readings; i.e.,

$$SD = \frac{ND}{ND + NG}, \tag{2}$$

where NG and ND denote the numbers of genuine readings and dummy readings, respectively. SD can be used to define the level of data anonymity because of the observation that, in the case of the adversary without prior knowledge, more dummy readings imply that it is more difficult for the adversary to identify the genuine (or dummy) readings and therefore higher data anonymity.
- Ubiquity (UB): This means that the dummy readings exist in the entire value range. When dummy readings live in particular value intervals, the adversary is more likely to have a better guess of genuine (or dummy) readings. An extreme case is that the genuine and dummy readings occupy every single value in the value range. This forms the best ubiquity. The notion of UB is defined in [15] in a descriptive way but is not defined mathematically. We therefore define UB as

$$UB = |\{g_i, d_j | 1 \le i \le NG, 1 \le j \le ND\}|. \tag{3}$$

Note that some of g_i's and d_j's could be the same, and therefore UB is not necessarily equal to $NG + ND$. In essence, UB in Eq. (3) counts the number of values occupied by the readings. Thus, larger UB enhances the location anonymity of dummy readings in a value range. The rationale behind UB is that if the readings are scattered over the entire value range, it is unlikely for the adversary to identify the genuine (or dummy) readings.
- Uniformity (UN): Uniformity means that, in the ideal case, each value subinterval should contain the same proportional number of readings. When the distribution of readings satisfies the uniformity, the introduction of dummy readings offers higher reading anonymity because the adversary has no clue to differentiate between genuine and dummy readings. Similarly, the notion of UN is defined in [15] in a descriptive way but is not defined mathematically. We therefore define UN as a binary variable

$$UN = \begin{cases} 1, & \text{if accept } ks(\{g_i, d_j | 1 \le i \le NG, \\ & \qquad 1 \le j \le ND\}, U) \\ 0, & \text{if reject } ks(\{g_i, d_j | 1 \le i \le NG, \\ & \qquad 1 \le j \le ND\}, U) \end{cases} \tag{4}$$

where $ks(X_1, X_2)$ denotes two-sample Kolmogorov-Smirnov test used to compare the distributions of the values in the two data sets X_1 and X_2, and U is a set of values uniformly sampled from the value range. The rationale behind UN is similar to the one behind UB. Nonetheless, UB focuses more on the number of values the readings occupy whereas UN focuses more on the distribution shape the readings form.

Throughout this paper, we use SD_{X}, UB_{X}, and UN_{X} to denote the SD, UB, and UN of the X scheme, respectively.

III. A Brief Review of Order-Preserving Symmetric Encryption

We review a cryptographic essential, Order-Preserving symmetric Encryption (OPE) [2], used in our proposed algorithms.

OPE is defined as a deterministic encryption scheme over the numerical values with the characteristic that, if the plaintexts x_1 and x_2 satisfy $x_1 < x_2$, then $\mathcal{E}_K(x_1) < \mathcal{E}_K(x_2)$, where $\mathcal{E}_K(\cdot)$ denotes the OPE function with key K, is guaranteed.

A simple OPE was presented in [2]; given that y numbers, $x_1 < \cdots < x_y$, are the possible plaintexts, the simplest OPE is to generate an array of y uniformly random numbers $k_1 < \cdots < k_y$ as the key K. The encryption (decryption) is accomplished by searching in the key K for the ciphertext (plaintext) corresponding to the plaintext (ciphertext); e.g. $\mathcal{E}_K(x_i) = k_i$, $1 \le i \le y$. Note that our proposed randomized and distributed version of OPE (described in Sec. IV-A) can be regarded as a natural extension of the above simple OPE.

A distinguishing feature of OPE is that its key also acts as possible ciphertexts. The above encryption reveals nothing but the numerical order of plaintexts because all the possible ciphertexts $k_1, \ldots, k_y$ are distributed uniformly over a specific range. Despite the leakage of numerical order, OPE is in fact provably secure [3], [4]. Albeit the above OPE scheme has the drawback of large key size, we keep such a simple form of OPE in mind for the ease of presentation throughout this paper. More sophisticated OPE schemes can be found in [2]–[4].

IV. Proposed Methods

In this section, a novel use of Order Preserving Encryption (OPE), randomized and distributed OPE (rdOPE), is first developed to establish the privacy guarantee in the proposed Verifiable top-k Query (VQ) schemes. Our study evolves in a number of successive steps; we present Global Dummy reading-based VQ (GD-VQ) and Local Dummy reading-based VQ (LD-VQ), which constitute the foundation of our proposed dummy reading-based anonymization framework. Afterward, they are enhanced to be Advanced Dummy reading-based VQ (AD-VQ), which reduces the communication overhead significantly.

As stated in Sec. II, we only consider the storage node compromises in Sec. IV-B∼IV-D. The defense against sensor compromises will be discussed in Sec. IV-E.

Table II summarizes the notations frequently used in this paper. Throughout the paper, though $[1, c]$, $[1, r]$, and $[1, b]$ typically include all the real values in the corresponding intervals, we abuse the notation and they denote all of the integer values in the corresponding intervals only.

TABLE II
Notation Table

	Description
k	the top-k query
n	the number of sensor nodes
s_i	the i-th sensor node
$s_{\mathcal{M}}$	the storage node
$\mathcal{A}$	the authority (network owner) who issues the top-k query
$k^{(i)}$	rdOPE key possessd by s_i
μ_i	the number of sensor readings of s_i
$d_{i,1}, \ldots, d_{i,\mu_i}$	the sensor readings of s_i
$[1, r]$	the value range of sensor readings
$\mathcal{B}$	the set of sensor readings in the entire network
Ω_k	the set of top-k query result
$\tilde{k}_i$	the key in one-way hash chain possessed by s_i
β	the query frequency
c	the value range of the hash function $h_{rdOPE}(\cdot)$
b	the value range of the encryption function $\mathcal{E}_{k^{(i)}}(\cdot)$
$k^{(i)}_{j,v}$	the v-th possible ciphertext corresponding to the j-th plaintext in s_i
$e_{i,1}, \ldots, e_{i,\mu_i}$	the encryptions of $d_{i,1}, \ldots, d_{i,\mu_i}$
$\hat{e}_{i,1}, \ldots, \hat{e}_{i,\mu_i}$	the union $\{e_{i,1}, \ldots, e_{i,\mu_i}\} \cup \{\text{dummy readings}\}$ of encrypted and dummy readings
ℓ_{id}, ℓ_d, and ℓ_h	the number of bits required to represent sensor ID, sensor (dummy) reading, and hash output
L	the average hop distance between two sensor nodes
α_{gdvq}, α_{ldvq}, and δ	the security parameters in GD-VQ and LD-VQ
η	the value difference between maximum and minimum encrypted readings
$\mathcal{L}_i = \langle \mathcal{L}_{i,L}, \mathcal{L}_{i,U} \rangle$	the virtual line segment with the starting (left-end) point $\mathcal{L}_{i,U} = e_{i,\mu_i}$ and the ending (right-end) point $\mathcal{L}_{i,L} = e_{i,\mu_i} - \eta$
$\mathcal{L}'_i$	the virtual line segment fully covering $\mathcal{L}_i$
$\mathbb{N}^{(i)}$	the set $\{s_{\mathbb{N}^{(i)}_1}, \ldots, s_{\mathbb{N}^{(i)}_{\lvert\mathbb{N}^{(i)}\rvert}}\}$ of the neighboring sensors of s_i
$\mathbb{H}^{(i)}$	the set $\{h_{\tilde{k}_i}(e_{i,1}), \cdots, h_{\tilde{k}_i}(e_{i,\mu_i})\}$ of hashes corresponding to individual genuine encrypted readings

A. The rdOPE Scheme

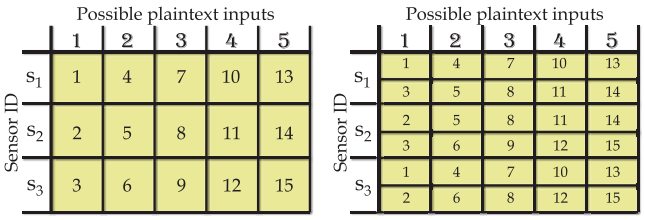
1) Motivation: OPE has been applied widely to encrypted database retrieval. Unfortunately, in the literature, the data are all assumed to be generated and encrypted by a single authority, which is not the case in our consideration. In addition, because the number of possible sensor readings could be limited and known from hardware specification, the relation between plaintexts and ciphertexts could be revealed. For example, if the sensors can only generate 20 kinds of possible outputs, then practically the adversary can derive the OPE key by investigating the numerical order of the eavesdropped ciphertexts despite the theoretical security guarantee.

2) Algorithmic Description of rdOPE: Our solution is a novel use of OPE, called rdOPE, which provides the randomness in the encryption outputs and is suitable for the case of distributed data generation with limited input value range. The technical challenge of rdOPE design is to maintain the numerical orders of encryptions from different sensors that use different OPEs. With the observation that the possible mapping between plaintexts and ciphertexts are fixed by $\mathcal{A}$ in advance, the ciphertexts can be determined prior to sensor deployment such that the numerical orders of ciphertexts in different sensors can be preserved.

More specifically, rdOPE for a network of n sensors with r possible sensor readings is defined as an encryption scheme $\langle \mathcal{E}, \mathcal{D}, k^{(i)}, h_{rdOPE}(\cdot), n, r, b, c \rangle$ such that

$$\mathcal{E}_{k^{(i)}}(x_1) < \mathcal{E}_{k^{(j)}}(x_2) \text{ if } x_1 < x_2, 1 \le i, j \le n, \tag{5}$$

where $k^{(i)}$ and $k^{(j)}$ denote the rdOPE keys possessed by s_i and s_j, respectively, and the value ranges of the hash output $h_{rdOPE}(\cdot)$ and encryption function output $\mathcal{E}_{k^{(i)}}(\cdot)$ are $[1, c]$ and $[1, b]$, respectively. Two rdOPE design examples are shown in Figs. 2a and 2b.



(a) The case of $c = 1$. (b) The case of $c = 2$.

Fig. 2. Examples of rdOPE.

An instance of rdOPE key construction (also called rdOPE table construction) works as follows. At first, rcn possibly distinct numbers, $k_1 \leq \cdots \leq k_{rcn}$, are chosen randomly from $[1, b]$ by $\mathcal{A}$. The numbers $k_1, \ldots, k_{rcn}$ are partitioned into r groups, $g_1, \ldots, g_r$, where $g_{\hat{i}}$ consists of $k_{1+(\hat{i}-1)cn}, \ldots, k_{\hat{i}cn}$, $1 \leq \hat{i} \leq r$. $\mathcal{A}$ randomly samples c numbers from $g_{\hat{i}}$ without replacement, and then stores them in s_i, $1 \leq i \leq n$. The c numbers from $g_{\hat{i}}$ are the possible ciphertexts of the plaintext input $\hat{i}$. As a result, the rdOPE key $k^{(i)}$ for s_i is a $c \times r$ array containing s_i's possible ciphertexts. In the above rdOPE key construction, if $k_1, \ldots, k_{rcn}$ are selected such that $k_{cn} \neq k_{cn+1}$, $k_{2cn} \neq k_{2cn+1}$, $\ldots$, and $k_{(r-1)cn} \neq k_{(r-1)cn+1}$, the constraint of Eq. (5) can always be fulfilled based on the partitioning rule of $g_1, \ldots, g_r$.

Let $k^{(i)}_{j,v}$ be the v-th possible ciphertext corresponding to the j-th plaintext in s_i. For example, $k^{(1)}_{3,2} = 8$ and $k^{(2)}_{5,1} = 14$ in Fig. 2b. rdOPE works as follows. When having a sensor reading x_j, the sensor s_i simply computes $\mathcal{E}_{k^{(i)}}(x_j) = k^{(i)}_{j,v}$, where v is calculated by $h_{rdOPE}(x_j||\tilde{k}_i)$. Once $k^{(i)}_{j,v}$ is received by $\mathcal{A}$, the decryption $\mathcal{D}_{k^{(i)}}(k^{(i)}_{j,v})$ can be accomplished by searching in $k^{(i)}$ for the plaintext corresponding to $k^{(i)}_{j,v}$.

Note that for a specific $1 \leq j \leq r$, the numbers for $k^{(i)}_{j,v}$, $1 \leq i \leq n$, $1 \leq v \leq c$, could be arbitrary because as shown in Eq. (5) there is no constraint on the keys for the same readings generated by different sensors.

3) Discussion of rdOPE: One can observe that rdOPE in fact offers only a way of using OPE in a distributed system. The distribution of $k^{(i)}$ on different sensors are still all uniform, fulfilling the requirement of conventional OPE. In the extreme case of $c = 1$, rdOPE degenerates to the simple OPE presented in [2] (also described in Sec. III). Thus, in the case of $c \geq 2$, the security of rdOPE can be guaranteed to be not less than that of OPE.

One can also observe that the c choices of each plaintext gives the flexibility in encryption outputs, offering the output randomness. As a result, even if the number of possible inputs is limited, as c is increased, the adversary is more difficult to infer the plaintext by correlating the eavesdropped ciphertexts to the possible plaintexts known from hardware specification.

A feature shared by both OPE and rdOPE is that there exist *illegitimate ciphertexts*, which is defined as the numbers not in rdOPE keys. Because all of the entries in $k^{(i)}$ are chosen manually by $\mathcal{A}$, when $\mathcal{A}$ receives a number that claims itself as an encryption of rdOPE, it is easy for $\mathcal{A}$ to verify such claim by checking whether the received number appears in $k^{(i)}$.

Two possible concerns of implementing rdOPE on sensor networks are:

- the additional computation burden for $\mathcal{A}$ to calculate the rdOPE table, and
- the additional space requirement for each sensor to store the corresponding rows of the rdOPE table.

The first concern involves the computation of rdOPE keys such that Eq. (5) can be fulfilled. Since the amount of computation linearly grows with the number of rdOPE keys, with the usual assumption of $\mathcal{A}$ with strong computation capability, for $\mathcal{A}$ the effort of calculating rdOPE table is affordable. On the other hand, the second concern is about the storage overhead. The rdOPE table is of size $r \times c$. When the sensor readings are two-byte integers, the table size is as much as $2^{16}c$ bits. In the case of $c = 4$, this results in additional 2^{18} bits space requirement. As the current generation of sensor nodes usually has near or even more than hundreds of kilo bytes, for ordinary sensors the effort of storing the rdOPE table can be deemed affordable as well.

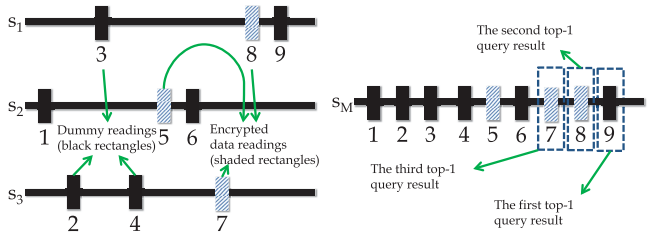
In the following, we will exploit the unique feature that there are illegitimate ciphertexts in rdOPE to construct our VQ schemes.

B. The GD-VQ Scheme

Basic Idea of GD-VQ The basic idea of GD-VQ is that the privacy, authenticity, and completeness are guaranteed by rdOPE, cryptographic hash, and the insertion of dummy readings, respectively. In particular, once the adversary cannot distinguish between genuine and dummy readings, the malicious removal of query results may cause the lose of dummy readings that are supposed to be included in the query result. Note that the "dummy readings" of the sensor s_i are defined as those readings sent from s_i to the storage node, generated by the program of s_i itself, but not collected from the sensor hardware to reflect the environment condition. This enables $\mathcal{A}$ to detect the query result incompleteness.

1) Algorithmic Description of GD-VQ: The μ_i sensor readings are encrypted by s_i with rdOPE key $k^{(i)}$ to form $e_{i,1} < \cdots < e_{i,\mu_i}$. Let α_{gdvq} be a security parameter of GD-VQ. Each sensor additionally generates α_{gdvq} distinct random dummy readings from $[1, b]$, resulting in $\hat{e}_{i,1} < \cdots < \hat{e}_{i,\mu_i+\alpha_{gdvq}}$, where μ_i of them are $e_{i,1}, \ldots, e_{i,\mu_i}$, and α_{gdvq} of them are the dummy readings. This can be implemented by calculating $h_{GDVQ}(\tilde{k}_i||1), \ldots, h_{GDVQ}(\tilde{k}_i||\alpha_{gdvq})$, where the output range of $h_{GDVQ}(\cdot)$ is $[1, b]$. To ease the analysis, the dummy readings are assumed to not collide with $\{e_{i,1}, \ldots, e_{i,\mu_i}\}$. An illustrative example is shown in Fig. 3a, where the rdOPE ciphertexts are generated from rdOPE key in Fig. 2b.

Since the dummy readings are generated randomly from $[1, b]$, they could collide with the legitimate ciphertext that s_i does not sense the corresponding reading. Without particular treatments, this kind of collision makes $\mathcal{A}$ accept false readings. For example, as shown in Fig. 3a, the dummy reading 9 generated by s_1 can be pruned easily by $\mathcal{A}$ because no entry 9 is in $k^{(1)}$ of Fig. 2b. Nonetheless, s_3 generates dummy readings 2 and 4 but actually does not have the corresponding sensor readings 1 and 2. This results in a circumstance where



(a) s_i point of view (b) $s_{\mathcal{M}}$ point of view

Fig. 3. Examples of GD-VQ.

the dummy readings collide with encrypted sensor readings. Under this circumstance, $\mathcal{A}$ may falsely accept 1 and 3 as s_1's readings. Our remedy is to generate different hashes depending on whether the reading is dummy. Specifically, in data submission phase, sensor s_i submits the following message to $s_{\mathcal{M}}$:

$$\begin{aligned} s_i \rightarrow s_{\mathcal{M}} : i, \quad & \hat{e}_{i,1}, h_{\tilde{k}_i}(\hat{e}_{i,1}||\Lambda_{i,1}), \\ & \vdots \\ & \hat{e}_{i,\mu_i+\alpha_{gdvq}}, h_{\tilde{k}_i}(\hat{e}_{i,\mu_i+\alpha_{gdvq}}||\Lambda_{i,\mu_i+\alpha_{gdvq}}), \end{aligned} \quad (6)$$

where $\Lambda_{i,j} = \emptyset$ if $\hat{e}_{i,j}$ is dummy and $\Lambda_{i,j} = \tilde{k}_i$ otherwise, for $1 \le j \le \mu_i + \alpha_{gdvq}$. Note that $h_{\tilde{k}_i}(e||\emptyset)$ is essentially equal to $h_{\tilde{k}_i}(e)$. With such different hash generations, $\mathcal{A}$ is able to know whether the received result is dummy. The details will be descried below.

2) Top-k Query Processing of GD-VQ: In GD-VQ, to retrieve Ω_k, $\mathcal{A}$ instead needs to issue top-1 queries repeatedly until $\mathcal{A}$ obtains Ω_k. In other words, from $s_{\mathcal{M}}$ point of view, for each received top-1 query, the current top-1 query is applied to the sensor readings excluding the previously returned top-1 results. An example of the query response phase of GD-VQ is shown in Fig. 3a, where $\mathcal{A}$ obtains two genuine readings, 7 and 8, by repeatedly issuing three top-1 queries.

On the other hand, from $\mathcal{A}$ point of view, for each received top-1 result, it follows the algorithm in Fig. 4 to verify the query result integrity and determine whether further top-1 query is needed. Let $(s_\pi, e_{\pi,j}, h_{\pi,j})$, where $e_{\pi,j}$, s_π, and $h_{\pi,j}$, $\pi \in [1,n]$, $j \in [1,\mu_\pi]$, denote the encrypted reading, the sensor that generates $e_{\pi,j}$, and the corresponding hash, respectively, be the currently received top-1 result. $\mathcal{A}$ first checks whether there are missing dummy readings larger than $e_{\pi,j}$ (the first **if** statement in Fig. 4); those dummy readings larger than $e_{\pi,j}$ are supposed to be received in previous queries. This can be accomplished because with the knowledge of $k^{(i)}$, $\mathcal{A}$ can generate all of the dummy readings by also calculating $h_{GDVQ}(\tilde{k}_i||1), \ldots, h_{GDVQ}(\tilde{k}_i||\alpha_{gdvq})$, $1 \le i \le n$. For example, when receiving the second top-1 result, 8, $\mathcal{A}$ checks whether it had received 9 with the knowledge of 1, 2, 3, 4, 6, and 9 being dummy. Subsequently, $\mathcal{A}$ checks whether $e_{\pi,j}$ is dummy by calculating $h_{\tilde{k}_\pi}(e_{\pi,j}||\tilde{k}_\pi)$ and $h_{\tilde{k}_\pi}(e_{\pi,j})$ (the second **if** statement in Fig. 4). $e_{\pi,j}$ is genuine sensor reading if $h_{\pi,j} = h_{\tilde{k}_\pi}(e_{\pi,j}||\tilde{k}_\pi)$, is dummy if $\pi, j = h_{\tilde{k}_\pi}(e_{\pi,j})$, and is inauthentic otherwise. Finally, depending on whether $\mathcal{A}$ has collected enough number of genuine sensor readings, $\mathcal{A}$ issues one more top-1 query or stops issuing query (the third **if** statement in Fig. 4).

Parameter: $\Omega_k = \emptyset$ is set for the first execution
1 **if** all dummy readings $\ge e_{\pi,j}$ have been received
2 **if** $e_{\pi,j}$ is a genuine sensor reading
3 $\Omega_k = \Omega_k \cup \{D_{k^{(\pi)}}(e_{\pi,j})\}$
4 **if** $|\Omega_k| < k$
5 issue one more top-1 query
6 **else** stop issuing top-1 query
7 **elsif** $e_{\pi,j}$ is a dummy reading
8 issue one more top-1 query
9 **else** alarm of inauthentic query result
10 **else** alarm of incomplete query result

Fig. 4. GD-VQ integrity verification.

The privacy and authenticity requirements mentioned in Sec. I can be fulfilled owing to the use of rdOPE and cryptographic hashes in Eq. (6). The completeness requirement of $\min \Omega_k \ge \max(\mathcal{B} \setminus \Omega_k)$ can also be fulfilled because replacing the elements in Ω_k by ones in $\mathcal{B} \setminus \Omega_k$ will be detected by $\mathcal{A}$ with certain probability, as shown below.

3) Detection Probability of GD-VQ: Assume that $k' \ge k$ top-1 queries are issued by $\mathcal{A}$ to retrieve Ω_k. From the above description, one can know that among these k' query results, k of them are genuine sensor readings and the remaining are dummy. Since the adversary is unable to distinguish between genuine and dummy readings, the only option for the adversary is to randomly choose and replace x of k' query result by the other smaller readings. The detection probability $P_{\mathtt{det}}^{GDVQ}$ of GD-VQ can be formulated as:

$$P_{\mathtt{det}}^{GDVQ} = Pr[\text{at least one of } x \text{ choices are dummy}] \quad (7)$$

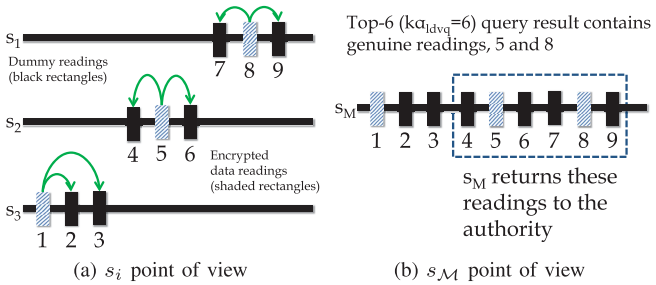
$$= 1 - \frac{k}{k'}\frac{k-1}{k'-1}\cdots\frac{k-(x-1)}{k'-(x-1)} = 1 - \prod_{\ell=0}^{x-1}\frac{k-\ell}{k'-\ell}. \quad (8)$$

4) In-Cell Communication Cost of GD-VQ: There are totally $\mu_i + \alpha_{gdvq}$ readings and $\mu_i + \alpha_{gdvq}$ hashes to be transmitted by s_i. Let ℓ_{id}, ℓ_d, and ℓ_h be the numbers of bits required for representing a sensor ID, a sensor (dummy) reading, and a hash, respectively. The average hop distance between sensor and $s_{\mathcal{M}}$ is L. As a consequence, the in-cell communication cost C_T^{GDVQ} of GD-VQ can be formulated as:

$$C_T^{GDVQ} = \sum_{i=1}^{n}(\ell_{id} + (\mu_i + \alpha_{gdvq})(\ell_d + \ell_h))L. \quad (9)$$

5) Query Communication Cost of GD-VQ: In the worst case, the minimum dummy reading is still larger than the maximum genuine sensor reading, resulting in the case that $\mathcal{A}$ needs to retrieve all of the dummy readings first before obtaining the genuine top-k result. With such consideration, the query communication cost of GD-VQ can be formulated as:

$$C_V^{GDVQ} = (k + n\alpha_{gdvq})(\ell_{id} + \ell_d + \ell_h). \quad (10)$$



(a) s_i point of view (b) $s_{\mathcal{M}}$ point of view

Fig. 5. Examples of LD-VQ.

6) Weakness of GD-VQ: Though the dummy reading insertion enables $\mathcal{A}$ to verify the result completeness, as the dummy readings are distributed over $[1, b]$, GD-VQ is in fact very communication inefficient, because 1) $\mathcal{A}$ is required to issue an uncertain number of top-1 queries to obtain the genuine top-k result, and 2) all of the dummy readings need to be returned in the worst case, leading to the overwhelming communication burden. Subsequently, a *local dummy reading* based scheme is proposed to conquer these two performance problems.

C. The LD-VQ Scheme

1) Basic Idea of LD-VQ: The LD-VQ design is the same as the GD-VQ design except that the dummy reading generation is dependent on the sensor readings and distributed over a limited range. By further taking advantage of the observation that the compromised storage node in most cases is unable to eavesdrop on sensor communications, such design has two benefits: 1) $\mathcal{A}$ can issue a single query to retrieve the genuine top-k result, reducing the need of two way communication between $s_{\mathcal{M}}$ and $\mathcal{A}$. 2) Even in the worst case, C_V can still be limited.

Though the level of anonymization of LD-VQ is weaker than GD-VQ, $\mathcal{A}$ is provided with an efficient way for the retrieval of Ω_k.

2) Algorithmic Description of LD-VQ: The μ_i sensor readings of s_i are encrypted by s_i with rdOPE key $k^{(i)}$ to form $e_{i,1} < \cdots < e_{i,\mu_i}$. Let α_{ldvq} be a security parameter of LD-VQ. In LD-VQ, each sensor additionally generates $\alpha_{ldvq} - 1$ distinct local dummy readings for each encryption, resulting in $\hat{e}_{i,1} < \cdots < \hat{e}_{i,\alpha_{ldvq}\mu_i}$, where μ_i of them are $e_{i,1}, \ldots, e_{i,\mu_i}$ while $(\alpha_{ldvq} - 1)\mu_i$ of them are dummy.

The dummy reading generation on each sensor s_i in LD-VQ is that, for each reading $e_{i,j}$, α_{ldvq} distinct dummy readings are selected randomly from $[e_{i,j} - \frac{\delta}{2}, e_{i,j} + \frac{\delta}{2}]$. This can be implemented by calculating $h_{LDVQ}(\tilde{k}_i||1), \ldots, h_{LDVQ}(\tilde{k}_i||\alpha_{ldvq})$, where the output range of $h_{LDVQ}(\cdot)$ is $[1, \delta]$ with δ being a system parameter affecting security and communication cost. The dummy readings are local in the sense that they are distributed over a restricted range. An illustrative example of LD-VQ is shown in Figs. 5a and 5b.

The local dummy readings might also lead to the collision mentioned in Sec. IV-B. The same technique as in Sec. IV-B can be utilized here to resolve the collision problem. We omit the repeated description for saving space. In data submission phase, s_i submits the following messages to $s_{\mathcal{M}}$:

$$\begin{aligned} s_i \rightarrow s_{\mathcal{M}}: \quad & \hat{e}_{i,1}, h_{\tilde{k}_i}(\hat{e}_{i,1}||\Lambda_{i,1}), \\ & \vdots \\ & \hat{e}_{i,\alpha_{ldvq}\mu_i}, h_{\tilde{k}_i}(\hat{e}_{i,\alpha_{ldvq}\mu_i}||\Lambda_{i,\alpha_{ldvq}\mu_i}), \end{aligned} \tag{11}$$

where $\Lambda_{i,j} = \emptyset$ if $\hat{e}_{i,j}$ is dummy and $\Lambda_{i,j} = \tilde{k}_i$ otherwise, for $1 \le j \le \alpha_{ldvq}\mu_i$. We particularly note the difference between Eqs. (6) and (11); when forwarding readings, each sensor does not include the ID information in the message. With the assumption that the compromised storage node cannot eavesdrop on sensor communications, this makes not only $s_{\mathcal{M}}$ and the adversary but also $\mathcal{A}$ unable to identify the sources of the readings. Nevertheless, $\mathcal{A}$ actually still can recover Ω_k and its source by taking advantage of determiniticity of rdOPE design. The details will be described later.

The rationale behind the ID information removal is that once the adversary can identify the sources of readings, it can remove all of the readings from the sensors generating the top-k result without being detected. For example, in Fig. 5a, if the adversary knows 7, 8, and 9 are from s_1, then it can return the incomplete result 1, 2, 3, 4, 5, and 6 that will succeed in the integrity verification below.

3) Top-k Query Processing of LD-VQ: In LD-VQ, to retrieve Ω_k, $\mathcal{A}$ instead needs to issue a top-$k\alpha_{ldvq}$ query to $s_{\mathcal{M}}$ because, in the worst case, the dummy readings induced by $e_{i,j}$ are all larger than $e_{i,j}$.

Let $\mathfrak{R} = \{(e_i, h_i)|1 \le i \le k\alpha_{ldvq}\}$, $e_1 \le \cdots \le e_{k\alpha_{ldvq}}$, be the received top-$k\alpha_{ldvq}$ result. $\mathcal{A}$ performs the following procedures to verify its integrity.

1) $\mathcal{A}$ prunes the illegitimate rdOPE ciphertexts; this can be accomplished by calculating
$$\mathfrak{I} = \Big(\bigcup_{1\le i\le k\alpha_{ldvq}} \{e_i\}\Big) \bigcap \Big(\bigcup_{1\le n,1\le j\le r,1\le v\le c} \{k_{j,v}^{(i)}\}\Big).$$
Then, $\mathcal{A}$ keeps only $\{(e_i, h_i)|e_i \in \mathfrak{I}\}$. For notational simplicity, $\{(e_i, h_i)|e_i \in \mathfrak{I}\}$ is written as $\{(e'_i, h'_i)\}$, where $1 \le i \le k''$, for some $k'' \le k\alpha_{ldvq}$.
2) Let $\mathfrak{N}(e'_i)$ be the candidate set of sensors that include e'_i in their rdOPE keys. $\mathcal{A}$ can compute $\mathfrak{N}(e'_i)$ by checking which of $k^{(1)}, \ldots, k^{(n)}$ contain e'_i. $\mathcal{A}$ can know that e'_i is generated by s_j if both $s_j \in \mathfrak{N}(e'_i)$ and either $h'_i = h_{\tilde{k}_j^t}(e'_i)$ or $h'_i = k_{\tilde{k}_j^t}(e'_i||\tilde{k}_j^t)$. If there exists e'_i such that the corresponding generating sensor cannot be found, $\mathfrak{R}$ is inauthentic. Finally, let $\mathfrak{G} = \{(e'_i, h'_i)|h'_i = k_{\tilde{k}_j^t}(e'_i||\tilde{k}_j^t)\}$ be the set of received genuine readings. $\mathfrak{R}$ is inauthentic if $|\mathfrak{G}| < k$.
3) $\mathcal{A}$ calculates the dummy readings induced by each encryption $e'_i \in \mathfrak{G}$. If all of the calculated dummy readings greater than $e_1 \in \mathfrak{R}$ can be found in $\mathfrak{R} \setminus \mathfrak{G}$, then $\mathcal{A}$ accepts $\mathfrak{R}$ as a complete result and decrypts the k first largest readings in $\mathfrak{G}$ with proper rdOPE keys to obtain Ω_k. If not, $\mathcal{A}$ considers $\mathfrak{R}$ incomplete.

LD-VQ can also fulfill the privacy, authenticity, and completeness requirements defined in Sec. I due to the similarity between LD-VQ and GD-VQ.

4) Detection Probability of LD-VQ: If δ is chosen properly, the adversary cannot distinguish between genuine and dummy readings because the value range of dummy readings is still $[1, b]$. Moreover, the adversary cannot identify the association between the reading and the sensor generating the reading because no ID information is attached to the forwarding messages and the adversary is assumed to be unable to monitor sensor communications of the entire network. Thus, the only option for the adversary is to randomly replace x readings in the query result with the other smaller readings. Hence, if the dummy readings induced by distinct encryptions are also distinct, the detection probability can be approximated by:

$$P_{\mathtt{det}}^{LDVQ} \approx Pr[\text{at least one of } x \text{ choices are dummy}] \quad (12)$$

$$= 1 - \prod_{\ell=0}^{x-1} \frac{k-\ell}{k\alpha_{ldvq} - \ell}. \quad (13)$$

5) In-Cell Communication Cost of LD-VQ: In LD-VQ, each sensor s_i has totally $\alpha_{ldvq}\mu_i$ genuine readings and $\alpha_{ldvq}\mu_i$ dummy readings to be reported. In addition, $\alpha_{ldvq}\mu_i$ hashes also need to be forwarded to $s_{\mathcal{M}}$. Hence, the in-cell communication cost C_T^{LDVQ} of LD-VQ can be calculated as:

$$C_T^{LDVQ} = \sum_{i=1}^{n} (\alpha_{ldvq}\mu_i)(\ell_d + \ell_h)L. \quad (14)$$

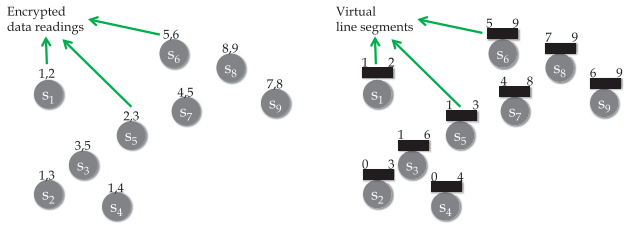
6) Query Communication Cost of LD-VQ: To retrieve Ω_k, $\mathcal{A}$ instead needs to issue a top-$k\alpha_{ldvq}$ query. Hence, the query communication cost C_V^{LDVQ} of LD-VQ can be calculated as:

$$C_V^{LDVQ} = k\alpha_{ldvq}(\ell_d + \ell_h). \quad (15)$$

7) Weakness of LD-VQ: The security of LD-VQ completely relies on the assumption of *local adversary* that cannot eavesdrop on sensor communications, which is not true in certain cases. In addition, as stated in Sec. IV-C.1, the level of anonymization of LD-VQ is weaker[1] than GD-VQ. Moreover, the parameter δ is difficult to set; if an improperly small δ is used, the genuine readings and the corresponding dummy readings can be separated and removed without being detected. For example, assume that a sensor generates readings 1, 2, 10, and 11, among which 1 and 10 are dummy. If $\delta = 4$ is known, the adversary can return the incomplete result 1 and 2 without being detected.

D. The AD-VQ Scheme

While GD-VQ incurs overwhelming communication burden, the security of LD-VQ completely relies on the assumption of local adversary. Moreover, the above two proposals share a common weakness that all of the readings, including genuine and dummy, need to be sent explicitly. In AD-VQ, we offer an alternative that can conquer the above problems simultaneously.



(a) Individual readings (b) Virtual line segments

Fig. 6. The conceptual illustrations of AD-VQ.

1) Basic Idea of AD-VQ: We observe a property of top-k result that the readings of neighboring sensors of the sensors generating Ω_k are either smaller than $\min \Omega_k$ or are included in Ω_k, as shown in Fig. 6a where 5 and 6, 4 and 5, and 7 and 8 in s_6, s_7, and s_9, respectively, are smaller than the top-1 result, 9, in s_8. A straightforward method for the query result completeness verification is to enable $\mathcal{A}$ to also have the readings of the neighboring sensors of the sensors claiming to generate Ω_k. Nevertheless, this method is flawed in that the *global adversary* that monitors every single communication of the entire network may exhaustively search for the "*hill sensor*", which is defined as the sensor whose maximum reading is larger than all of the readings of its neighboring sensors, but is not in Ω_k. For example, s_3 in Fig. 6a is the hill sensor for top-1 query because its reading 5 is the maximum of the readings in the proximity.

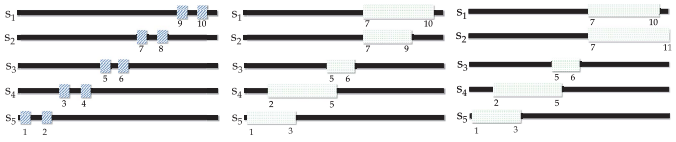
We propose a novel dummy reading-based technique to anonymize the readings such that the genuine readings cannot be identified and thus hill sensors cannot be found. A conceptual illustration is shown in Fig. 6b, where the readings are hidden in the virtual line segments (black rectangles) described below.

2) Algorithmic Description of AD-VQ: Each sensor s_i has the sensed data $d_{i,1} < \cdots < d_{i,\mu_i}$ and their encryptions $e_{i,1} < \cdots < e_{i,\mu_i}$. Let η be a system parameter denoting the difference between the maximum and minimum encrypted readings within an epoch.[2] Then, s_i constructs a virtual line segment $\mathcal{L}_i = \langle \mathcal{L}_{i,L}, \mathcal{L}_{i,U} \rangle$ with $\mathcal{L}_{i,L} = e_{i,\mu_i} - \eta$ and $\mathcal{L}_{i,U} = e_{i,\mu_i}$, where $\mathcal{L}_{i,L}$ and $\mathcal{L}_{i,U}$ are used to represent the starting and ending points of $\mathcal{L}$, respectively.

After that, two more points, $\mathcal{L}'_{i,L}$ and $\mathcal{L}'_{i,U}$ are selected randomly from $[\mathcal{L}_{i,L} - \eta, \mathcal{L}_{i,L}]$ and $[\mathcal{L}_{i,U}, \mathcal{L}_{i,U} + \eta]$, respectively, to form another line segment $\mathcal{L}'_i = \langle \mathcal{L}'_{i,L}, \mathcal{L}'_{i,U} \rangle$, where $\mathcal{L}'_{i,L}$ and $\mathcal{L}'_{i,U}$ are computed as: $\mathcal{L}'_{i,L} = \mathcal{L}_{i,L} - h_{ADVQ}(\tilde{k}_i || \mathcal{L}_{i,L})$ and $\mathcal{L}'_{i,U} = \mathcal{L}_{i,U} + h_{ADVQ}(\tilde{k}_i || \mathcal{L}_{i,U})$ with $h_{ADVQ}(\cdot)$ assumed to be a hash function with output range $[0, \eta - 1]$. The

[1]The level of anonymization depends on how many dummy readings are inserted and how the dummy readings are scattered. In this sense, α_{gdvq} and $(\alpha_{ldvq} - 1)\mu_i$, the total number of dummy readings inserted in GD-VQ and LD-VQ, respectively, can be used to quantify the anonymization level. Moreover, δ can be used to quantify the anonymization level because it precisely describe how each dummy reading is scattered. Given a fixed number of dummy readings inserted, the level of anonymization of LD-VQ is weaker than GD-VQ because the dummy readings in GD-VQ could be located anywhere, whereas the locations of the dummy readings in LD-VQ are restricted and determined by those genuine sensor readings. Indeed, in the case of an extremely large $\delta = r - 1$, LD-VQ degenerates to GD-VQ. Nevertheless, in the general setting of a small δ, LD-VQ achieves a lower level of anonymization than GD-VQ.

[2]The setting of η may need domain knowledge as to the phenomenon the sensors are sensing. For example, though a temperature sensor's possible readings range from -50 to 150, η could be 10 for a period of 100 seconds because the temperature does not fluctuate within a short period of time. In the worst case, if no domain knowledge can be utilized, $\eta = \max_{i,i',v,v'}\{k^i_{r,v} - k^{i'}_{1,v'}\}$ can be chosen with the inferiority of increased communication cost.



(a) Individual readings (b) $\mathcal{L}'_1 \geq \mathcal{L}'_2$ (c) $\mathcal{L}'_1 \leq \mathcal{L}'_2$

Fig. 7. Examples of AD-VQ.

purpose of $\mathcal{L}'_i$ is to anonymize $e_{i,1}, \ldots, e_{i,\mu_i}$; the elements in $\mathcal{L}'_i$ other than the genuine encrypted readings can be thought of as dummy readings. The advantage of using a virtual line segment is that there is no need to store each element in $\mathcal{L}'_i$ explicitly and two numbers, $\mathcal{L}'_{i,L}$ and $\mathcal{L}'_{i,U}$, are sufficient to represent $\mathcal{L}'_i$. Furthermore, let $\mathbb{N}^{(i)} = \{s_{\mathbb{N}^{(i)}_1}, \ldots, s_{\mathbb{N}^{(i)}_{|\mathbb{N}^{(i)}|}}\}$ and $\mathbb{H}^{(i)} = \{h_{\tilde{k}_i}(e_{i,1}), \cdots, h_{\tilde{k}_i}(e_{i,\mu_i})\}$ be the set of the neighboring sensors of s_i and the set of hashes corresponding to individual genuine encrypted readings, respectively. Subsequently, in data submission phase, s_i submits to $s_{\mathcal{M}}$ the following message:

$$s_i \rightarrow s_{\mathcal{M}} : i, \mathcal{L}'_i, \mu_i, \mathbb{N}^{(i)}, \mathbb{H}^{(i)}, \\ h_{\tilde{k}_i}(i||\mathcal{L}'_i||\mu_i||\mathbb{N}^{(i)}_1||\ldots||\mathbb{N}^{(i)}_{|\mathbb{N}^{(i)}|}), h_{\tilde{k}_i}(i||\mathcal{L}'_i||\mu_i). \quad (16)$$

For example, s_1 in Fig. 7a has genuine readings 9 and 10, which are transformed into $\langle \mathcal{L}'_{1,L}, \mathcal{L}'_{1,L} \rangle = \langle 7, 10 \rangle$ in Fig. 7b. We particularly note that there are no individual readings in AD-VQ. Instead, the encryptions $e_{i,1}, \ldots, e_{i,\mu_i}$ are transformed into a line segment $\mathcal{L}'_i$ containing $e_{i,1}, \ldots, e_{i,\mu_i}$.

3) Top-k Query Processing of AD-VQ: Let $\mathcal{L}_{i,U}$ be the representative of $\mathcal{L}_i$. In this sense, a virtual line segment $\mathcal{L}_a$ is smaller than another one $\mathcal{L}_{a'}$ ($\mathcal{L}_a < \mathcal{L}_{a'}$) if $\mathcal{L}_{a,U} < \mathcal{L}_{a',U}$.

To obtain Ω_k, $\mathcal{A}$ instead issues a top-$(\eta-1+k)$ query. Such a query transformation is needed because the line segments containing top-k result could be smaller than the other line segments due to the dummy reading insertion. For example, in Fig. 7c, though $\mathcal{L}'_1$ contains the top-1 result, 10, we can see that $\mathcal{L}'_1 < \mathcal{L}'_2$. With the assumption of $\mathcal{L}'_1 > \cdots > \mathcal{L}'_n$ for the ease of presentation, $s_{\mathcal{M}}$ in query response phase submits to $\mathcal{A}$ the following messages:

$$s_{\mathcal{M}} \rightarrow \mathcal{A} : i, \mathcal{L}'_i, \mu_i, \mathbb{N}^{(i)}, \mathbb{H}^{(i)}, \\ h_{\tilde{k}_i}(i||\mathcal{L}'_i||\mu_i||\mathbb{N}^{(i)}_1||\ldots||\mathbb{N}^{(i)}_{|\mathbb{N}^{(i)}|}), 1 \leq i \leq \eta - 1 + k. \quad (17)$$

Note that if there are multiple satisfying line segments with the same largest point, they are regarded as one. Furthermore, $s_{\mathcal{M}}$ also needs to present more verification materials to $\mathcal{A}$. In particular, for every $s_j \in \mathbb{N}^{(i)}$, $1 \leq i \leq \eta - 1 + k$, $s_{\mathcal{M}}$ additionally sends to $\mathcal{A}$ the following messages:

$$s_{\mathcal{M}} \rightarrow \mathcal{A} : j, \mathcal{L}'_j, \mu_j, \mathbb{H}^{(j)}, h_{\tilde{k}_j}(j||\mathcal{L}'_i||\mu_j). \quad (18)$$

Let $\mathfrak{M}$ and $\mathfrak{V}$ be the sets of results collected from Eq. (17) and (18), respectively. $\mathcal{A}$ performs the following three-step algorithm to verify the query result integrity.

1) With the knowledge of $\mathbb{N}^{(\pi)}$ provided in $\mathfrak{M}$, $\mathcal{A}$ calculates $h_{\tilde{k}_\pi}(\pi||\mathcal{L}'_\pi||\mu_\pi||\mathbb{N}^{(\pi)}_1||\ldots||\mathbb{N}^{(\pi)}_{|\mathbb{N}^{(\pi)}|})$ for $\mathcal{L}'_\pi \in \mathfrak{M}$. In addition, $\mathcal{A}$ calculates $h_{\tilde{k}_\pi}(\pi||\mathcal{L}'_\pi||\mu_\pi)$ for $\mathcal{L}'_\pi \in \mathfrak{V}$. The query result is inauthentic if not all of the calculated hashes can be found in $\mathfrak{M} \cup \mathfrak{V}$.
2) In this step, $\mathcal{A}$ extracts the genuine top-k readings from the received top-$(\eta-1+k)$ result. For each $\mathcal{L}'_\pi \in \mathfrak{M} \cup \mathfrak{V}$, $\mathcal{A}$ computes $\{h_{\tilde{k}_\pi}(e) | e \in [\mathcal{L}'_{\pi,L}, \mathcal{L}'_{\pi,U}]\}$. The query result is inauthentic if
$$|\{h_{\tilde{k}_\pi}(e) | e \in [\mathcal{L}'_{\pi,L}, \mathcal{L}'_{\pi,U}]\} \cap \mathbb{H}^{(\pi)}| \neq \mu_\pi. \quad (19)$$
Let $\mathfrak{E}_\pi = \{e_{\pi,1}, \ldots, e_{\pi,\mu_\pi}\}$ be the set of genuine encrypted readings extracted from $\mathcal{L}'_\pi$, and $\mathfrak{E}_{\mathfrak{M}} = \bigcup_{\mathcal{L}'_\pi \in \mathfrak{M}} \mathfrak{E}_\pi$.
3) In this step, $\mathcal{A}$ verifies the query result completeness. Let $\mathfrak{T}_k$ be the candidate set of top-k encrypted readings extracted from $\mathfrak{E}_{\mathfrak{M}}$, i.e., $\mathfrak{T}_k \subseteq \mathfrak{E}_{\mathfrak{M}}$ with $\min \mathfrak{T}_k \geq \max(\mathfrak{E}_{\mathfrak{M}} \setminus \mathfrak{T}_k)$. For each s_π contributing readings to $\mathfrak{T}_k$, $\mathcal{A}$ checks whether the elements in $\cup_{s_{\pi'} \in \mathbb{N}^{(\pi)}} \mathfrak{E}_{\pi'}$ are either smaller than $\min \mathfrak{T}_k$ or included in $\mathfrak{T}_k$. $\mathcal{A}$ considers the received result complete if and only if the above verification succeeds.

4) Detection Probability of AD-VQ: Because the genuine readings have been anonymized, the adversary does not know which elements in $\mathcal{L}'_\pi$ are genuine reading. Thus, the only option left for the adversary is to randomly replace x of top-$(\eta-1+k)$ line segments by the other line segments. Assume that the sensor readings are distributed uniformly over $[1, r]$ and s_i has $|\mathbb{N}^{(i)}|$ neighboring sensors. If the maximum genuine readings in the x line segments of sensors $s_{\bar{\pi}_1}, \ldots, s_{\bar{\pi}_x}$ used for malicious replacement are $m(s_{\bar{\pi}_1}), \ldots, m(s_{\bar{\pi}_x})$, respectively, then given that $\cap_{i=1}^{x}(s_{\bar{\pi}_i} \cup \mathbb{N}^{(\bar{\pi}_i)}) = \emptyset$, the detection probability $P^{ADVQ}_{\mathtt{det}}$ can be formulated as:

$$1 - Pr[m(s_{\bar{\pi}_i}) \geq e_{j,\mu_j}, 1 \leq i \leq x, s_j \in \mathbb{N}^{(\bar{\pi}_i)}] \quad (20)$$

$$= 1 - \prod_{i=1}^{x} \left(\frac{r - m(s_{\bar{\pi}_i})}{r} \right)^{|\mathbb{N}^{(\bar{\pi}_i)}|}. \quad (21)$$

5) In-Cell Communication Cost of AD-VQ: In AD-VQ, the encrypted readings are replaced by a line segment. s_i only needs to submit the parameters for representing the line segment and the necessary verification materials to $s_{\mathcal{M}}$. Hence, the in-cell communication cost C^{ADVQ}_T can be formulated as:

$$C^{ADVQ}_T = \sum_{i=1}^{n} (\ell_{id} + 3\ell_d + |\mathbb{N}^{(i)}|\ell_{id} + (\mu_i + 2)\ell_h)L. \quad (22)$$

6) Query Communication Cost of AD-VQ: The query communication cost C^{ADVQ}_V can be upper bounded by:

$$\sum_{i=1}^{\eta-1+k} \left((|\mathbb{N}^{(i)}| + 1)(2\ell_{id} + 3\ell_d + (\mu_i + 1)\ell_h) \right) - \ell_{id}. \quad (23)$$

Note that Eq. (23) is due to the communication cost of Eqs. (17) and (18).

7) Further Communication Cost Reduction: In the context of static sensor networks, the neighborhood relationship for each sensor remains unchanged in most of the time. Thus, the neighborhood relationship for each sensor only needs to be updated very infrequently, or even better, the communication cost for the submission of $\mathbb{N}^{(i)}$ can

be saved. Moreover, the communication cost for the submission of hashes can also be saved by allowing heavier computation burden on $\mathcal{A}$ (described below). As a result, in this variant of AD-VQ, AD-VQ-static, after calculating $\mathcal{L}'_{i,L} = \mathcal{L}_{i,L} - h_{ADVQ}(e_{i,1}||\dots||e_{i,\mu_i}||\tilde{k}_i)$ and $\mathcal{L}'_{i,U} = \mathcal{L}_{i,U} + h_{ADVQ}(\mathcal{L}'_{i,L}||e_{i,1}||\dots||e_{i,\mu_i}||\tilde{k}_i)$, s_i submits to $s_{\mathcal{M}}$ the following message:

$$s_i \rightarrow s_{\mathcal{M}} : i, \mathcal{L}'_i, \mu_i, \\ h_{\tilde{k}_i}(i||\mathcal{L}'_i||\mu_i||\mathbb{N}^{(i)}_1||\dots||\mathbb{N}^{(i)}_{|\mathbb{N}^{(i)}|}), h_{\tilde{k}_i}(i||\mathcal{L}'_i||\mu_i). \quad (24)$$

To obtain Ω_k, $\mathcal{A}$ still issues a top-$(\eta - 1 + k)$ query, and then, with the assumption of $\mathcal{L}'_1 > \dots > \mathcal{L}'_n$, $s_{\mathcal{M}}$ submits to $\mathcal{A}$ the following messages:

$$s_{\mathcal{M}} \rightarrow \mathcal{A} : i, \mathcal{L}'_i, \mu_i, h_{\tilde{k}_i}(i||\mathcal{L}'_i||\mu_i|| \quad \mathbb{N}^{(i)}_1||\dots||\mathbb{N}^{(i)}_{|\mathbb{N}^{(i)}|}), \\ 1 \le i \le \eta - 1 + k, \quad (25)$$

and for every $s_j \in \mathbb{N}^{(i)}$, $s_{\mathcal{M}}$ additionally sends to $\mathcal{A}$ the following messages:

$$s_{\mathcal{M}} \rightarrow \mathcal{A} : j, \mathcal{L}'_i, \mu_j, h_{\tilde{k}_i}(j||\mathcal{L}'_i||\mu_i). \quad (26)$$

Let $\mathfrak{M}$ and $\mathfrak{V}$ be the sets of results collected from Eq. (25) and (26), respectively. $\mathcal{A}$ performs the following three-step algorithm to verify the query result integrity.

1) Assuming the knowledge of $\mathbb{N}^{(\pi)}$, $\mathcal{A}$ calculates $h_{\tilde{k}_\pi}(\pi||\mathcal{L}'_{\pi,L}||\mathcal{L}'_{\pi,U}||\mu_\pi||\mathbb{N}^{(\pi)}_1||\dots||\mathbb{N}^{(\pi)}_{|\mathbb{N}^{(\pi)}|})$ for $\mathcal{L}'_\pi \in \mathfrak{M}$ and $h_{\tilde{k}_\pi}(\pi||\mathcal{L}'_{\pi,L}||\mathcal{L}'_{\pi,U}||\mu_\pi)$ for $\mathcal{L}'_\pi \in \mathfrak{V}$. The query result is inauthentic if not all of the calculated hashes can be found in $\mathfrak{M} \cup \mathfrak{V}$.
2) For each line segment $\mathcal{L}'_\pi \in \mathfrak{M} \cup \mathfrak{V}$, $\mathcal{A}$ performs $2^{\left(\frac{\mathcal{L}'_{\pi,U} - \mathcal{L}'_{\pi,L}}{\mu_\pi}\right)}$ hash computations to see if there is one combination of $e_{\pi,1}, \dots, e_{\pi,\mu_\pi}$ that can satisfy with $\mathcal{L}'_{\pi,L} = \mathcal{L}_{\pi,L} - h_{ADVQ}(e_{\pi,1}||\dots||e_{\pi,\mu_\pi}||\tilde{k}_\pi)$ and $\mathcal{L}'_{\pi,U} = \mathcal{L}_{\pi,U} + h_{ADVQ}(\mathcal{L}'_{\pi,L}||e_{\pi,1}||\dots||e_{\pi,\mu_\pi}||\tilde{k}_\pi)$. The query result is inauthentic if such satisfying combination cannot be found. Let $\mathfrak{E}_\pi$ be the set of encrypted readings extracted from $\mathcal{L}'_\pi$, and $\mathfrak{E}_{\mathfrak{M}} = \bigcup_{\mathcal{L}'_\pi \in \mathfrak{M}} \mathfrak{E}_\pi$.
3) Let $\mathfrak{T}_k$ be the candidate set of top-k encrypted readings extracted from $\mathfrak{E}_{\mathfrak{M}}$, i.e., $\mathfrak{T}_k \subseteq \mathfrak{E}_{\mathfrak{M}}$ with $\min \mathfrak{T}_k \ge \max(\mathfrak{E}_{\mathfrak{M}} \setminus \mathfrak{T}_k)$. For each s_π contributing readings to $\mathfrak{T}_k$, $\mathcal{A}$ checks whether the elements in $\cup_{s_{\pi'} \in \mathbb{N}^{(\pi)}} \mathfrak{E}_{\pi'}$ are either smaller than $\min \mathfrak{T}_k$ or included in $\mathfrak{T}_k$. $\mathcal{A}$ considers the received result complete if and only if the above verification succeeds.

In this variant (AD-VQ-static) of AD-VQ, the detection probability P^{ADVQs}_{det} remains the same as P^{ADVQ}_{det}, but C^{ADVQs}_T and C^{ADVQs}_V can be formulated as:

$$C^{ADVQs}_T = \sum_{i=1}^{n} (\ell_{id} + 3\ell_d + 2\ell_h)L, \quad (27)$$

and

$$C^{ADVQs}_V \le \sum_{i=1}^{\eta-1+k} \Big((|\mathbb{N}^{(i)}| + 1)(\ell_{id} + 3\ell_d + \ell_h) \Big). \quad (28)$$

E. Counteracting Sensor Compromises

Two attacks due to the sensor compromises, false data injection and false incrimination, are considered. The former is that the compromised sensor forges extremely large readings to deviate Ω_k, and the latter is that the compromised sensor submits false hashes to frame the innocent $s_{\mathcal{M}}$.

While GD-VQ and LD-VQ are vulnerable to the false data injection, AD-VQ and AD-VQ-static are more resilient against the false data injection than the existing works, because their design inherently forces the adversary to compromise all of the sensors in the proximity. Otherwise, the compromised sensors will be an outlier and attract $\mathcal{A}$'s attention because no knowledge of genuine readings is available for the adversary.

On the other hand, we propose to use the aggregate signature [6] to defend against the false incrimination. In particular, n pairs $\{\langle k_{pub,i}, k_{pri,i}\rangle | 1 \le i \le n\}$ of public and private keys are generated by $\mathcal{A}$ before the sensor deployment. $\{k_{pub,i} | 1 \le i \le n\}$ are stored in $s_{\mathcal{M}}$, and $k_{pri,i}$ is stored in s_i, $1 \le i \le n$. With $k_{pri,i}$, s_i calculates the signature of its own message, receives signatures, aggregates the received signatures, and then forwards the aggregated signature, resulting in the additional ℓ_h communication cost. $s_{\mathcal{M}}$ with $\{k_{pub,i} | 1 \le i \le n\}$ checks the legitimacy of the aggregate signature for each epoch and presents the signature to $\mathcal{A}$ once there are some disputes. We can prove that at least one compromised node can be identified by such a design once the signature is requested by $\mathcal{A}$, but omit the details here.

V. Security and Performance Evaluation

More security discussions of our proposed methods are presented in Sec. V-A. Numerical simulations and prototype implementation were conducted to demonstrate the practicality of our methods in Secs. V-B and V-C, respectively. Then a comparison among the prior solutions and our methods will be presented in Sec. V-D.

A. Security Discussion

1) Resilience Against Other Attacks: There are a large number of attacks aiming to subvert the sensor networks functionality. We cannot enumerate them all. However, node replication attack [19], Sybil attack [18], wormhole attack [14], and false data injection attack [31] could be the representatives of the attacks for sensor networks. In the following, though different security impacts can be caused by the above attacks, we restrict ourselves to discussing the possibility of the adversary generating an incomplete query result that can be accepted by $\mathcal{A}$.

In node replication attack, the legitimate sensor nodes are compromised and replicated. Many clones of the legitimate nodes with all of the corresponding security credentials are placed back in the strategic positions of the networks. By launching node replication attack, the adversary is assumed to be able to control a limited number of legitimate sensor nodes. There could be two cases.

- The node replicated by the adversary is a storage node.
- The nodes replicated by the adversary are ordinary sensor nodes.

In the first case, our security discussion remains unchanged because the replication of storage node does not give the adversary more security credentials or more ability to subvert our proposed methods. On the other hand, the second case is equivalent to the sensor compromises already discussed in Sec. III-E of the revised manuscript because, similarly, the replication of sensor nodes does not give the adversary more security credentials or more ability to subvert our proposed methods. As a whole, our proposed methods are resilient against node replication attack.

In Sybil attack, the nodes with different legitimate IDs are shown in the network. In wormhole attack, a pair of so-called wormhole nodes is connected by, for example, a high-bandwidth out-of-bound channel. A wormhole node transmits whatever it receives to another wormhole node. Sybil and wormhole attacks mainly cause the routing chaos. In false data injectoin attack, the adversary simply injects useless messages whose destinations are random distant positions. False data injection attack mainly aims to consume the sensor nodes' battery power, reducing the network lifetime. These issues, though important, are orthogonal to our top-k query result completeness problem because false routing and junk messages cannot give the adversary more security credentials or more ability to subvert our proposed methods. Hence, they cannot be used to defeat our proposed methods.

2) Forward Security: The notion of forward security in the key-evolution context originated from. The notion of forward security does not have a purely mathematic definition; it only states a property that the compromise of the current secret key does not enable the adversary to *manipulate the contents* generated before the compromise. Here, depending on different cryptographic primitives, the meaning of the *content manipulation* varies; for example, for a forward secure signature scheme [7], it means that the compromise of the current secret key does not enable the adversary to forge signatures related to the past.

The use of hash chain in our proposed methods perfectly fits the key evolution framework in [7], [17], [22]. In particular, the forward security of our proposed methods refers to the property that the adversary cannot find a HMAC corresponding to its forged sensor reading. The following argument can be used to prove the forward security of our proposed methods. For each epoch t, each node calculates $\tilde{k}_i^t = h(\tilde{k}_i^{t-1})$ and erases $\tilde{k}_i^{t-1}$. In GD-VQ, each encrypted reading $\hat{e}_{i,j}$ is associated with a HMAC $h_{\tilde{k}_i^t}(\hat{e}_{i,j}||\Lambda_{i,j})$, $1 \leq j \leq \mu_i + \alpha_{gdvq}$. In LD-VQ, each encrypted reading $\hat{e}_{i,j}$ is associated with a HMAC $h_{\tilde{k}_i^t}(\hat{e}_{i,j}||\Lambda_{i,j})$, $1 \leq j \leq \alpha_{ldvq}\mu_i$. In AD-VQ and AD-VQ-static, each virtual segment $\mathcal{L}_i'$ is associated with two HMACs, $h_{\tilde{k}_i^t}(i||\mathcal{L}_i'||\mu_i||\mathbb{N}_1^{(i)}||\ldots||\mathbb{N}_{|\mathbb{N}^{(i)}|}^{(i)})$ and $h_{\tilde{k}_i^t}(i||\mathcal{L}_i'||\mu_i)$. Since $\tilde{k}_i^1, \tilde{k}_i^2, \ldots, \tilde{k}_i^{t-1}$ have already been erased, the adversary compromising a sensor node s_i at the epoch t cannot obtain $\tilde{k}_i^1, \tilde{k}_i^2, \ldots, \tilde{k}_i^{t-1}$ due to the one-way property of $h(\cdot)$. Thus, the adversary cannot construct a legitimate HMAC to make the authority $\mathcal{A}$ accept a forged reading claiming to be generated before epoch t. The above fulfills the definition of forward security.

3) Data Confidentiality of rdOPE: rdOPE is extended from the *baseline OPE* described in Sec. III. The data confidentiality of the baseline OPE can be confirmed in [2]. In spite of the data confidentiality guarantee of the baseline OPE, the data confidentiality of rdOPE remains unproven. In essence, given y possible plaintexts, in the baseline OPE, how we determine the ciphertexts is to randomly select y numbers from a fixed interval without replacement, $x_1 < \cdots < x_y$. On the other hand, the selection of ciphertexts in rdOPE on each sensor node can be understood as follows. In rdOPE, how we determine the ciphertexts is to randomly select $y' \geq y$ number from a fixed interval without replacement $x_1 < \cdots < x_{y'}$. Then, these y' numbers are partitioned into y consecutive disjoint equal-sized groups. For each group, one number is sampled uniformly at random. Eventually, y numbers are chosen to be the ciphertexts for the sensor node under consideration.

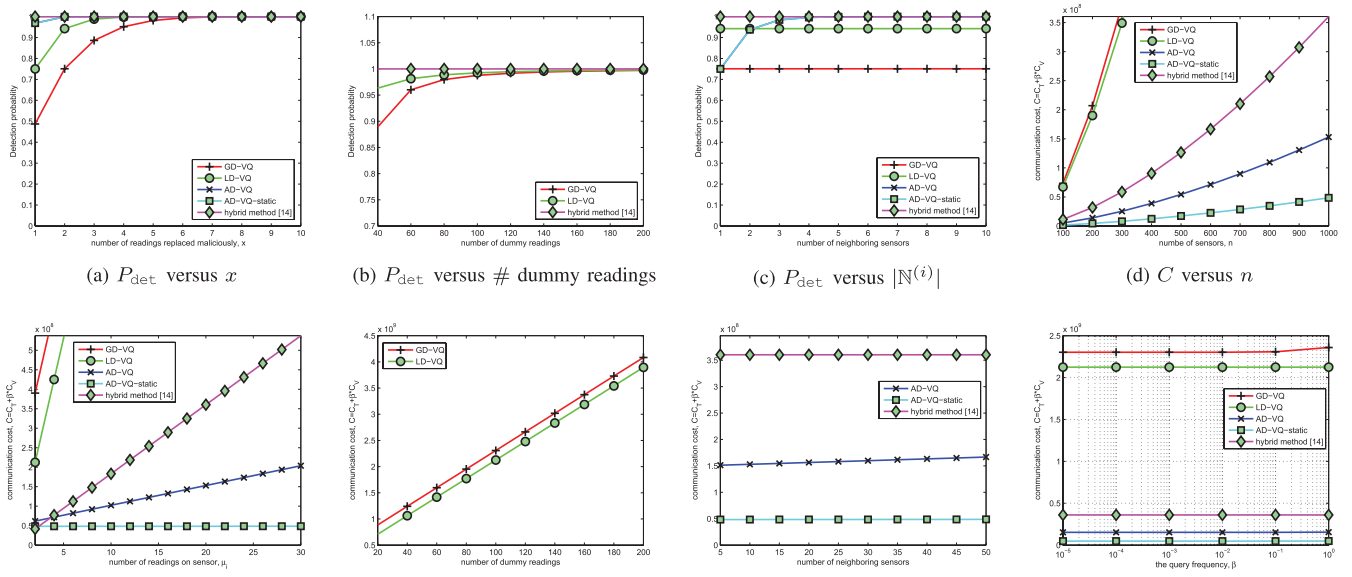
rdOPE can be thought of as each sensor node individually and independently performing the baseline OPE. In this sense, if we can prove that the ciphertexts in rdOPE of each sensor node are uniformly distributed, then we can claim that both the baseline OPE with the ciphertexts selected uniformly at random and rdOPE have the same data confidentiality level.

However, the ciphertexts determined by rdOPE are actually not uniformly distributed. Consider an example where $y' = 4$ numbers are randomly sampled from $[1, 10]$ and the objective is to obtain $y = 2$ numbers. The probability that the two numbers are eventually 9 and 10 is 0.

In spite of the above counterexample, we resort to hypothesis testing, trying to give an indirect proof that the ciphertexts randomly sampled by rdOPE are approximately uniformly distributed. In particular, we vary y from 10 to 1000 and y' from 20 to 5000. The underlying interval I is also varied from $[1, 100]$ to $[1, 50000]$. For every $\langle y, y', I\rangle$, we sampled y numbers from I uniformly at random, constituting a set A_1 of y numbers, and sampled y numbers from I in a way described in rdOPE, constituting a set A_2 of y numbers. Afterwards, we perform Kolmogorov-Smirnov test on A_1 and A_2 to test whether A_1 and A_2 have the same distribution. We ran the test 20 times for every $\langle y, y', I\rangle$. The results are all positive; A_1 and A_2 have the same distribution. Thus, with the prior knowledge that the numbers sampled in a way described in rdOPE are not uniformly distributed, such a result gives us the confidence and indirect proof that although the ciphertexts determined by rdOPE are not uniformly distributed, they are close to be uniformly distributed in the sense that they cannot be distinguished by Kolmogorov-Smirnov test. As a consequence, we claim that rdOPE has a level of confidentiality guarantee similar to the one offered by the baseline OPE with the ciphertexts selected uniformly at random, because the ciphertexts on each individual node are actually close to be uniformly distributed.

B. Numerical Results

The hybrid method [35] achieves the security and performance balance between additive evidence [35] and crosscheck [35]. It can be shown that the hybrid method can detect incomplete result with probability 1. The common parameter



(a) $P_{\mathtt{det}}$ versus x (b) $P_{\mathtt{det}}$ versus # dummy readings (c) $P_{\mathtt{det}}$ versus $|\mathbb{N}^{(i)}|$ (d) C versus n

(e) C versus μ_i (f) C versus # dummy readings (g) C versus $|\mathbb{N}^{(i)}|$ (h) C versus β

Fig. 8. Numerical results.

setting in our numerical results is $n = 100$, $\ell_{id} = 10$, $\ell_d = 400$, $\ell_h = 160$, $k = 10$, $\mu_i = 15$, and $|\mathbb{N}^{(i)}| = 10$. In addition, $L = \sqrt{n}$ (the average distance between two random nodes in a flat network), $\alpha_{gdvq} = 5\mu_i$, $\eta = 2\mu_i$, and α_{ldvq} are set such that the number of dummy readings added in GD-VQ is the same as the one in LD-VQ. The other parameter setting in the hybrid method is the same as the one shown in [35] for fair comparison.

1) Impact of x on P^X_{det}: Fig. 8a shows that, if more readings are replaced, then it is more likely that the dummy readings are chosen, and therefore, the incomplete result will be detected with higher probability. The reason of $P^{GDVQ}_{\mathtt{det}} < P^{LDVQ}_{\mathtt{det}}$ is that $\mathcal{A}$ in GD-VQ stops issuing the top-1 query as soon as it obtains $\mathbf{\Omega}_k$, while $\mathcal{A}$ in LD-VQ obtains a bunch of readings at a time. Thus, generally GD-VQ contributes relatively less dummy readings in response to queries, implying the worse detection probability.

2) Impact of Number of Dummy Readings on P^X_{det}: The parameter α_{gdvq} and α_{ldvq} are set such that the number of dummy readings in GD-VQ and LD-VQ will be the same in Fig. 8b. The same argument for the impact of x on $P_{\mathtt{det}}$ can explain the phenomenon of $P^{GDVQ}_{\mathtt{det}} \leq P^{LDVQ}_{\mathtt{det}}$ here with the same number of dummy readings added.

3) Impact of $|\mathbb{N}^{(i)}|$ on P^X_{det}: As $P^{GDVQ}_{\mathtt{det}}$ and $P^{LDVQ}_{\mathtt{det}}$ are independent of $|\mathbb{N}^{(i)}|$, they remain stable in Fig. 8c with varying $|\mathbb{N}^{(i)}|$. Since the neighboring sensors are utilized in AD-VQ and AD-VQ-static to detect the incomplete result, as $|\mathbb{N}^{(i)}|$ increases, it is more probable to find the sensor whose maximum reading is larger than the claimed top-k query result, resulting in the better detection probability.

4) Impact of n on C^X: The network size has direct impact on the communication cost. In particular, C^{ADVQ} and C^{ADVQs} grows slowly in Fig. 8d owing to their use of virtual line segments for representing readings.

5) Impact of μ_i on C^X: In AD-VQ, μ_i hashes need to be sent explicitly but the readings are all represented by, or equivalently, compressed to $\mathcal{L}'_i$ for s_i. Thus, C^{ADVQ} grows slowly with μ_i in Fig. 8c. In AD-VQ-static, the readings and hashes of s_i are all represented by $\mathcal{L}'_i$, which is independent of μ_i. Hence, C^{ADVQs} remains stable in Fig. 8e.

6) Impact of Number of Dummy Readings on C^X: C^{GDVQ} and C^{LDVQ} in Fig. 8f are linearly proportional to the number of dummy readings added because they are dominated primarily by the communication cost of μ_i readings and μ_i hashes.

7) Impact of $|\mathbb{N}^{(i)}|$ on C^X: The number of neighboring sensors has impact on C^{ADVQ} and C^{ADVQs} since $s_{\mathcal{M}}$ needs to report to $\mathcal{A}$ the line segments from the sensors around the sensors generating $\mathbf{\Omega}_k$. Moreover, s_i in AD-VQ additionally needs to send out $|\mathbb{N}^{(i)}|$ IDs in data submission phase. Thus, as $|\mathbb{N}^{(i)}|$ is increased, C^{ADVQ} grows faster than C^{ADVQs} in Fig. 8g.

8) Impact of β on C^X: The query frequency has influence on the long term communication cost. In fact, the AD-VQ achieves a significant reduction of C^{ADVQ}_T at the expense of a minor increase of C^{ADVQ}_V. Specifically, C^{ADVQ}_V is usually several orders of magnitude lower than C^{ADVQ}_T and will be amortized in different epochs. Hence, the change of β does not significantly affect the overall communication cost, as shown in Fig. 8h.

C. Prototype Implementation

GD-VQ and AD-VQ were implemented on TelosB motes on top of TinyOS (CPU: TI MSP430F1611; ROM: 48KB+256B; RAM: 10KB; Radio Chipset: ChipCon CC2420). Our program code was also run on TOSSIM in TinyOS 1.1.15 to evaluate the energy consumption. In our setting, together with the AES encryption function in a CC2420 chipset, CBC-MAC mode is

TABLE III
SUMMARY OF PROTOTYPE IMPLEMENTATION

	ROM	RAM	CPU
GD-VQ	13372 Bytes	603 Bytes	1223.755 mJ
AD-VQ	13036 Bytes	493 Bytes	1158.609 mJ

used to implement the hash function with $\mu_i = 25$ readings. Table III reports the results.

D. Comparison

1) Communication Cost: As Table I shows, the communication cost of AD-VQ and AD-VQ-static is significantly lower than the others. Hybrid Crosscheck [35] incurs tremendous communication cost because it involves the data broadcast over the cell. Though some parameters such as *broadcast probability* can be introduced to reduce the communication cost, it also dramatically lowers the detection probability. GD-VQ and LD-VQ need to send out individual dummy readings, resulting in a lot of additional packet transmissions. Our proposed AD-VQ-static and SMQ [34] achieve the lower communication cost because the former "encodes" the dummy readings in a virtual line segment only while the latter aggregates the hashes for the verification purpose along the packet forwarding of the tree.

2) Detection Probability: In Hybrid Crosscheck, the detection probability is strongly related to the communication cost. To achieve higher detection probability, the network needs to pay a lot of communication cost, causing the dilemma in choosing system parameters. GD-VQ and LD-VQ also have similar problems. Therefore, the detection probabilities of Hybrid Crosscheck, GD-VQ, and LD-VQ all vary significantly. In contrast to them, AD-VQ, AD-VQ-static, and SMQ all have stable detection probability irrespective of parameter settings.

3) Data Confidentiality Guarantee: Hybrid Crosscheck does not have the design about data confidentiality. On the other hand, SMQ achieves the data confidentiality through the use of bucket index. Though bucket index offers a simple way to access the encrypted data, it also leak the partial information about the value range of the sensor readings to the adversary and does not have provable security guarantee.

4) Resilience Against Topology Change: The effectiveness of SMQ completely relies on the underlying tree structure. Moreover, the $\mathcal{A}$ in SMQ needs the full knowledge of the tree topology. Any topology change would render the SMQ useless. On the other hand, Hybrid Crosscheck and our proposed VQ methods do not make such unrealistic assumptions.

5) Level of Data Anonymity: Here, we evaluate the level of data anonymity offered by our proposed methods based on the evaluation metrics SD, UB, and UN described in Sec. II-F. We have the following theorems.

Theorem 1: Assume that the condition that the genuine and dummy readings are all distinct holds. For arbitrary i, if $\mathcal{L}'_{i,U} - \mathcal{L}'_{i,L} - \mu_i \geq \alpha_{gdvq}$ and $\alpha_{gdvq} = (\alpha_{ldvq} - 1)\mu_i$, then $SD_{AD-VQ} \geq SD_{GD-VQ} = SD_{LD-VQ}$ and $UB_{AD-VQ} \geq UB_{GD-VQ} = UB_{LD-VQ}$. If $\mathcal{L}'_{i,U} - \mathcal{L}'_{i,L} - \mu_i < \alpha_{GD-VQ}$ and $\alpha_{gdvq} = (\alpha - 1)\mu_i$, then $SD_{AD-VQ} < SD_{GD-VQ} = SD_{LD-VQ}$ and $UB_{AD-VQ} < UB_{GD-VQ} = UB_{LD-VQ}$.

Proof: Under the constraints of $\mathcal{L}'_{i,U} - \mathcal{L}'_{i,L} - \mu_i \geq \alpha_{gdvq}$ and $\alpha_{gdvq} = (\alpha_{ldvq} - 1)\mu_i$, the relation $SD_{AD-VQ} \geq SD_{GD-VQ}$ holds because the number of dummy readings generated by the virtual segment in AD-VQ is larger than the number of dummy readings generated by GD-VQ. This can be attributed to the fact that every single value not occupied by the genuine readings are considered as dummy readings. The relation $SD_{GD-VQ} = SD_{LD-VQ}$ holds simply because of the constraint of $\alpha_{gdvq} = (\alpha_{ldvq} - 1)\mu_i$. The relation $UB_{AD-VQ} \geq UB_{GD-VQ}$ holds because the readings are all distinct and therefore there is no reading occupying the same value. Therefore, the number of readings is the same as the number of values occupied by the readings, which can be used to calculate UB. Finally, the relation $UB_{GD-VQ} = UB_{LD-VQ}$ holds because, similarly, the readings are all distinct. Therefore, the numbers of values occupied by the readings from GD-VQ and LD-VQ are the same.

The similar argument can be applied to the case of $\mathcal{L}'_{i,U} - \mathcal{L}'_{i,L} - \mu_i \leq \alpha_{gdvq}$ and $\alpha_{gdvq} = (\alpha - 1)\mu_i$. ■

Theorem 2: Assume that the condition that the genuine and dummy readings are uniformly distributed over the value range holds. The relations $Pr[UN_{GD-VQ} = 1] \geq Pr[UN_{LD-VQ} = 1]$ and $Pr[UN_{GD-VQ} = 1] \geq Pr[UN_{AD-VQ} = 1]$ hold.

Proof: According to the basic probability theory, the union of two sets, each of which is composed of the elements uniformly sampled from a given set, still follows uniform distribution. UN_{GD-VQ} is always 1 because both the genuine and dummy readings are assumed to be uniformly distributed. However, due to the procedures of LD-VQ and AD-VQ, obviously the dummy readings will not follow the uniform distribution. As a consequence, there is a nonzero probability that the Kolmogorov-Smirnov test rejects the null hypothesis that the two input samples follow the same distribution. Therefore, we can conclude that the relations $Pr[UN_{GD-VQ} = 1] \geq Pr[UN_{LD-VQ} = 1]$ and $Pr[UN_{GD-VQ} = 1] \geq Pr[UN_{AD-VQ} = 1]$ hold. ■

One can see from Theorems 1 and 2 that although AD-VQ is better than GD-VQ and LD-VQ in terms of SD and UB, GD-VQ works as the best method in terms of UN. The explanation for this is that GD-VQ, indeed, is the best way to anonymize the data only if there is no constraints on communication and computation overhead. Nonetheless, if we add more and more dummy readings in GD-VQ to gain the larger SD and UB, the communication and computation burden will also be increased. Therefore, AD-VQ is proposed to not only enhance the data anonymization by adding much more dummy readings without incurring significant overhead (via virtual line segment) but also reduce the communication and computation overhead by sacrificing UN requirement.

VI. CONCLUSION

A novel dummy reading-based anonymization framework is proposed to design Verifiable top-k Query (VQ) schemes. In particular, AD-VQ-static achieves the lower communication complexity with only minor detection capability penalty,

which could be of both theoretical and practical interests. With only symmetric cryptography involved and their low implementation difficulty, the VQ schemes are suitable and practical for current sensor networks.

REFERENCES

[1] O. H. Abdelrahman, E. Gelenbe, G. Görbil, and B. Oklander, "Mobile network anomaly detection and mitigation: The NEMESYS approach," in *Proc. 28th ISCIS*, Oct. 2013, pp. 429–438.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proc. ACM SIGMOD*, 2004, pp. 63–574.

[3] A. Boldyreva, N. Chenette, Y. Lee, and A. O'neill, "Order-preserving symmetric encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Tech.*, 2009, pp. 224–241.

[4] A. Boldyreva, N. Chenette, and A. O'neill, "Order-preserving encryption revisited: Improved security analysis and solutions," in *Proc. Int. Cryptol. Conf. CRYPTO*, 2011, pp. 1–18.

[5] M. Burkhart and X. Dimitropoulos, "Fast privacy preserving top-k queries using secret sharing," in *Proc. 19th ICCCN*, 2010, pp. 1–7.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and veriably encrypted signatures from bilinear maps," in *Proc. Int. Conf. Theory Appl. Cryptograph. Tech. Adv. Cryptol.*, 2003, pp. 416–432.

[7] M. Bellare and S. K. Miner, "A forward-secure digital signature scheme," in *Proc. 19th Annu. Int. Cryptol. Conf. Adv. Cryptol.*, 1999, 431–438.

[8] F. Chen and A. X. Liu, "SafeQ: Secure and efficient query processing in sensor networks," in *Proc. 24th IEEE Conf. Comput. Commun.*, Mar. 2010, pp. 1–9.

[9] C. Dwork, "Differential privacy," in *Proc. ICALP*, 2006, pp. 1–12.

[10] P. Desnoyers, D. Ganesan, and P. Shenoy, "TSAR: A two tier sensor storage architecture using interval skip graphs," in *Proc. ACM 3rd Int. Conf. Embedded Netw. Sensor Syst.*, 2005, pp. 39–50.

[11] E. Gelenbe, G. Görbil, D. Tzovaras, S. Liebergeld, D. Garcia, M. Baltatu, *et al.*, "NEMESYS: Enhanced network security for seamless service provisioning in the smart mobile ecosystem," in *Proc. ISCIS*, Oct. 2013, pp. 369–378.

[12] E. Gelenbe and G. Loukas, "A self-aware approach to denial of service defence," *Comput. Netw.*, vol. 51, no. 5, pp. 1299–1314, 2007.

[13] [Onllie]. Available: http://www.hpl.hp.com/news/2009/oct-dec/cense.html

[14] Y.-C. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks," in *Proc. 22nd Annu. Joint Conf. IEEE Comput. Commun. INFOCOM*, Apr. 2003, pp. 1976–1986.

[15] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proc. ICPS*, Jul. 2005, pp. 88–97.

[16] Q. Li and D. Rus, "Global clock synchronization in sensor networks," in *Proc. IEEE Conf. Comput. Commun. INFOCOM*, Jan. 2004, pp. 1–11.

[17] D. Ma and G. Tsudik, "A new approach to secure logging," *ACM Trans. Storage*, vol. 5, no. 1, pp. 1—3, 2009.

[18] J. Nesome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor network: Analysis & defense," in *Proc. 3rd Int. Symp. ISPN*, Apr. 2004, pp. 259–268.

[19] B. Parno, A. Perrig, and D. Johnson, "Distributed detection of node replication attacks in sensor networks," in *Proc. IEEE Symp. Security Privacy*, May 2005, pp. 49–63.

[20] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar, "SPINS: Security protocols for sensor networks," in *Proc. ACM Conf. Mobile Comput. Netw.*, 2001, pp. 521–534.

[21] L. Sweeney, "*k*-Anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl. Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[22] B. Schneier and J. Kelsey, "Cryptographic support for secure logs on untrusted machines," in *Proc. 7th USENIX Security Symp.*, 1998, pp. 53–62.

[23] B. Sheng and Q. Li, "Verifiable privacy-preserving range query in two-tiered sensor networks," in *Proc. 24th IEEE 27th Conf. Comput. Commun.*, Apr. 2008, pp. 743–766.

[24] B. Sheng, Q. Li, and W. Mao, "Data storage placement in sensor networks," in *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2006, pp. 344–355.

[25] K. Sun, P. Ning, C. Wang, A. Liu, and Y. Zhou, "TinySeRSync: Secure and resilient time synchronization in wireless sensor networks," in *Proc. 13th ACM Conf. CCS*, Feb. 2006, pp. 264–277.

[26] J. Shi, R. Zhang, and Y. Zhang, "Secure range queries in tiered sensor networks," in *Proc. 24th IEEE Conf. Comput. Commun.*, Jan. 2009, pp. 1–9.

[27] Y.-T. Tsou, C.-S. Lu, and S.-Y. Kuo, "Privacy- and integrity-preserving range query in wireless sensor networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2012, pp. 328–334.

[28] J. Vaidya and C. Clifton, "Privacy-preserving top-k queries," in *Proc. IEEE 21st ICDE*, Apr. 2005, pp. 545–546.

[29] M. Wu, J. Xu, X. Tang, and W.-C. Lee, "Top-k monitoring in wireless sensor networks," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 7, pp. 962–976, Jul. 2007.

[30] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "Noninteractive pairwise key establishment for sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 556–569, Sep. 2010.

[31] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proc. IEEE 23rd Annu. Joint Conf. Comput. Commun. INFOCOM*, Mar. 2004, pp. 2446–2457.

[32] C.-M. Yu, G.-K. Ni, I.-Y. Chen, E. Gelenbe, and S.-Y. Kuo, "Top-k query result completeness verification in sensor networks," in *Proc. IEEE Int. ICC Workshops*, Jun. 2013, pp. 1026–1030.

[33] T.-H. You, W.-C. Peng, and W.-C. Lee, "Protecting moving trajectories with dummies," in *Proc. Int. Conf. Mobile Data Manag.*, May 2007, pp. 278–282.

[34] C.-M. Yu, Y.-T. Tsou, C.-S. Lu, and S.-Y. Kuo, "Practical and secure multidimensional query framework in tiered sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 241–255, Jun. 2011.

[35] R. Zhang, J. Shi, Y. Liu, and Y. Zhang, "Verifiable fine-grained top-k queries in tiered sensor networks," in *Proc. 24th IEEE INFOCOM*, Mar. 2010, pp. 1–9.

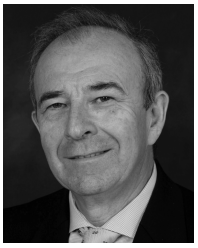
[36] R. Zhang, J. Shi, and Y. Zhang, "Secure multidimensional range queries in sensor networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2009, pp. 197–206.

[37] R. Zhang, Y. Zhang, and C. Zhang, "Secure top-k query processing via untrusted location-based service providers," in *Proc. 24th IEEE Conf. Comput. Commun.*, Mar. 2012, pp. 1170–1178.

Chia-Mu Yu received the Ph.D. degree from National Taiwan University in 2012. He was a Research Assistant with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, from 2005 to 2010. He was a Visiting Scholar with Harvard University, from 2010 to 2011, and a Visiting Scholar with Imperial College London from January 2012 to September 2012. He was a Post-Doctoral Researcher with the IBM Thomas J. Watson Research Center from 2012 to 2013. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Yuan Ze University. His research interests include cloud storage security, sensor network security, and smart grid security.

Guo-Kai Ni is currently a Post-Doctoral Researcher with the Department of Electrical Engineering, National Taiwan University.

Ing-Yi Chen received the B.Sc. degree in physics from National Central University, Taiwan in 1984, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Arizona, USA, in 1989 and 1992, respectively. He is currently a Professor with the Computer Science and Information Engineering Department, National Taipei University of Technology, Taiwan. Prior to joining Taipei Tech, he served as a Chief Technology Officer for China Times Inc., with responsibility for the corporate strategic technology planning and for handling university relations. His research interests include various topics in cloud computing and solution frameworks for building sensor network applications.



Erol Gelenbe (LF'11) is a fellow of ACM and IET and is the Professor in the Dennis Gabor Chair, and the Head of Intelligent Systems and Networks with Imperial College. He was a Chaired Professor and the ECE Department Head with Duke University, and the Director of the School of Electrical Engineering and Computer Science, UCF, Orlando. An expert on Computer and Network Systems Performance Evaluation, and the author of four books, he created a branch of queuing theory called G-Networks and invented the random neural network model. He developed the FLEXSIM manufacturing simulation concept and tool, and created the highly successful Performance Modeling and Engineering teams in France that developed the QNAP Modeling/Simulation software tool suite. He invented the cognitive packet network routing protocol for autonomic communications. He received the Best Paper Awards for Sustainable IT, and coordinates the EU FP7 Project NEMESYS on Mobile Network Security. He is also PI of recently funded projects on cloud computing and sustainability in ICT. He is an elected member of the French National Academy of Engineering, Hungarian Academy of Science, Academy of Sciences of Poland, and a fellow of the Turkish Science Academy. He won the IET's Oliver Lodge Medal in 2010 and the ACM's SIGMETRICS Life-Time Achievement Award in 2008. Prof. Gelenbe is an Associate Editor of other journals including the IEEE TRANSACTIONS ON CLOUD COMPUTING, *Acta Informatica*, and *Performance Evaluation*.

Sy-Yen Kuo (F'01) is currently the Dean with the College of Electrical Engineering and Computer Science, National Taiwan University, is a Distinguished Professor with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, and was the Chairman with the same department from 2001 to 2004. He was a Chair Professor and the Dean of the College of Electrical and Computer Engineering, National Taiwan University of Science and Technology from 2006 to 2009. He received the B.S. degree in electrical engineering from National Taiwan University in 1979, the M.S. degree in electrical and computer engineering from the University of California at Santa Barbara in 1982, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign in 1987. He was a Visiting Professor with the Computer Science and Engineering Department, Chinese University of Hong Kong, from 2004 to 2005, and a Visiting Researcher with AT&T Labs-Research, NJ, USA, from 1999 to 2000. He was the Chairman of the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan, from 1995 to 1998, a Faculty Member of the Department of Electrical and Computer Engineering, University of Arizona, from 1988 to 1991, and an Engineer with Fairchild Semiconductor and Silvar-Lisco, CA, USA, from 1982 to 1984. In 1989, he was a Summer Faculty Fellow with the Jet Propulsion Laboratory, California Institute of Technology. His current research interests include dependable systems and networks, mobile computing, and quantum computing and communications. He has published more than 300 papers in journals and conferences, and holds more than ten U.S. and Taiwan patents.

He received the Distinguished Research Award from 1997 to 2005 consecutively from the National Science Council in Taiwan and is currently a Research Fellow. He was a recipient of the Best Paper Award at the 1996 International Symposium on Software Reliability Engineering, the Best Paper Award in the simulation and test category at the 1986 IEEE/ACM Design Automation Conference, the National Science Foundation's Research Initiation Award in 1989, and the IEEE/ACM Design Automation Scholarship in 1990 and 1991.