Secure Cooperative Data Storage and Query Processing in Unattended Tiered Sensor Networks

Rui Zhang, *Student Member, IEEE,* Jing Shi, *Member, IEEE,* Yanchao Zhang, *Senior Member, IEEE,* and Jinyuan Sun, *Member, IEEE*

***Abstract*—We consider an unattended tiered sensor network (UTSN) consisting of resource-rich master nodes at the upper tier and resource-poor sensor nodes at the lower tier. Sensor nodes submit data to nearby master nodes which store the data and answer the queries from the network owner on behalf of sensor nodes. Such a cooperative data storage and query processing paradigm offers a number of advantages over traditional Homogeneous unattended sensor networks. Relying on master nodes for data storage and query processing, however, raises severe concerns about data confidentiality and query-result correctness when the sensor network is deployed in hostile environments. In particular, a compromised master node may leak hosted sensitive data to the adversary; it may also return juggled or incomplete query results to the network owner. In this paper, we take multidimensional range queries as an example to investigate secure cooperative data storage and query processing in UTSNs. We present a suite of novel schemes that can ensure data confidentiality against master nodes and also enable the network owner to verify with very high probability the authenticity and completeness of any query result by inspecting the spatial and temporal relationships among the returned data. Detailed performance evaluations confirm the high efficacy and efficiency of the proposed schemes.**

***Index Terms*—Unattended tiered sensor networks; cooperative data storage and query processing; range query; security.**

I. Introduction

UNATTENDED sensor networks (USNs) refer to the wireless sensor networks with intermittent sink presence [2], [3], which are expected to be deployed in remote and extreme environments such as oceans, volcanos, animal habitats, and battlefields. It is often impossible or prohibitive to maintain a stable always-on communication connection from a USN to its external network owner. This situation necessitates in-network data storage [4]–[7] such that data continuously produced by sensor nodes are stored inside the network. The network owner can access the data when needed via an ad-hoc communication connection (e.g., a satellite link) or by physical means like dispatching mobile sinks to the USN [5].

There are mainly two approaches to realize in-network storage in USNs. A simple approach is to furnish individual sensor nodes with abundant memory space where data produced over time can be stored locally and await the next mobile-sink visit. Despite rapid progress in storage technology, this approach remains economically infeasible for large-scale USNs, where the number of sensor nodes may be in tens of thousands. A more viable approach is to equip a few special nodes, which we call *master node*s, with several gigabytes of NAND flash storage for a few tens of dollars [5], which together with sensor nodes enable *cooperative data storage and query processing* and also form an unattended tiered sensor network (UTSN).

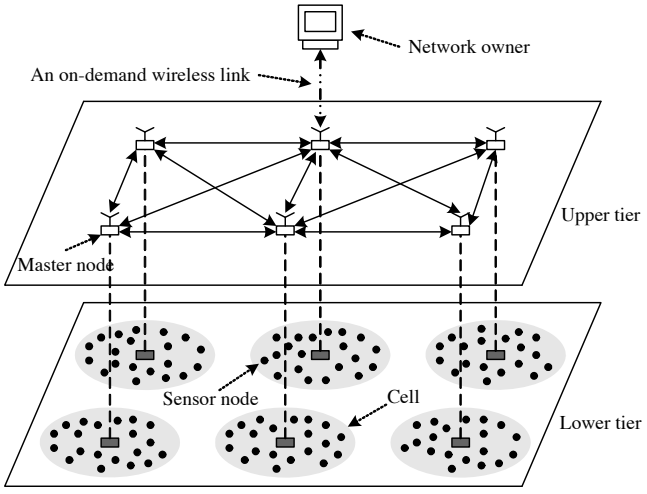


Fig. 1. A UTSN with cooperative data storage and query processing.

An exemplary UTSN with cooperative data storage and query processing is shown in Fig. 1. It consists of a large number of resource-poor sensor nodes at the lower tier and relatively fewer resource-rich master nodes at the upper tier. Sensor nodes perform sensing tasks and periodically submit sensed data to nearby master nodes for storage, while master nodes collect data from nearby sensor nodes, store them locally for extended periods of time, and answer various ad-hoc data queries from the network owner. Master nodes can also have abundant resources in energy and computation and form a multi-hop wireless mesh network among themselves using long-range high-bandwidth radios.

The above UTSN offers a number of advantages over a traditional homogeneous USN. First, the UTSN require less frequent mobile-sink visits (which are often costly) to offload sensor data due to the large storage space of the master nodes. Second, the network owner can query the data of interest via an on-demand wireless link at any time in the UTSN, while the only way to access the data in traditional homogeneous USNs is to dispatch mobile sinks, which could incur excessive

Manuscript received 1 February 2011; revised 20 July 2011. This work was supported in part by the US National Science Foundation under grants CNS-0716302 and CNS-0844972 (CAREER). The preliminary version of this paper appeared in ACM MobiHoc'09 [1].

R. Zhang and Y. Zhang are with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287 (e-mail: {ruizhang, yczhang}@asu.edu).

J. Shi is with the School of Public Administration, Huazhong University of Science and Technology, China (e-mail: js39@njit.edu).

J. Sun is with the Department of Electrical Engineering and Computer Science, University of Tennessee-Knoxville, Knoxville, TN 37996 (e-mail: jysun@eecs.utk.edu).

Digital Object Identifier 10.1109/JSAC.2012.120223.

delays. Last, such a two-tier network architecture is known to be indispensable for increasing network capacity and scalability, reducing system management complexity, and prolonging network lifetime [4], [8].

Relying on master nodes for data storage and query processing, however, raises significant security concerns. For instance, if a UTSN is deployed in hostile military or homeland security scenarios, master nodes are attractive targets of attack and may be compromised by the adversary. The adversary may launch two types of severe attacks through compromised master nodes. First, the adversary can read all the data stored on them which are possibly very sensitive (e.g., intrusion events). This attack calls for sound defenses to ensure data confidentiality while still enabling efficient data query processing. Second, the adversary may instruct compromised master nodes to return juggled and/or incomplete data in response to ad-hoc queries from the network owner. This attack is more subtle and harmful than blind DoS attacks on the sensor network, especially when the query results are used as the basis for making critical military or business decisions. To defend against this attack, we must enable the network owner to check the *authenticity* and *completeness* of any query result. The term authenticity means that all the data in the result originated from the purported sources and have not been tampered with, and completeness means that the result includes all the data satisfying the query. We refer to a query result as being *correct* if it is both authentic and complete.

In this paper, we take range queries as an example to investigate secure cooperative data storage and query processing in UTSNs. Range queries are an important and common type of queries in sensor networks which ask for data with one or multiple attributes falling in specified ranges (called *one-dimensional* or *multidimensional* range queries) [9], [10]. An exemplary multidimensional range query is "Return all observed objects with weights between 170 and 220 pounds and moving speeds between 3 and 5 miles per hour."

To the best of our knowledge, secure cooperative data storage and query processing in UTSNs have received attention only recently [7], [11], [12]. Aiming at one-dimensional range queries, these schemes [7], [11], [12] could ensure data confidentiality and also enable query-result authenticity and completeness verification with different communication and computation overhead. The only piece of work on secure multidimensional queries [13] relies on a common key shared among all sensor nodes, which is unfortunately vulnerable to compromised sensor nodes: the adversary can recover the common key after compromising any sensor node, whereby to further recover the original data. A more sound solution to secure multidimensional range queries thus remains an open challenge.

In this paper, we investigates techniques to secure multidimensional range queries in UTSNs against possibly compromised master nodes. We employ the bucketing technique [14], [15] to achieve data confidentiality and also query-result authenticity verification while ensuring efficient query processing. Our major contributions are a suite of novel techniques for the network owner to verify query-result completeness. In particular, our first construction is a deterministic approach that is an extension of the technique in [7] to multidimensional cases and also serves as our benchmark. It allows the network owner to immediately catch misbehaving master nodes at the cost of high communication overhead growing exponentially with the number of dimensions (or queriable data attributes). We then present two novel probabilistic techniques with significantly less communication overhead, including a spatial crosscheck technique and a temporal crosscheck technique. The former aims to create some relationships among data generated by sensor nodes affiliated with the same master node, while the latter aims to embed some relationships among data produced in different time periods. These two techniques can collectively allow the network owner to verify with overwhelming probability whether a query result is complete by examining the spatial and temporal relationships among the returned data. We further propose a random-probing technique as a complement to spatial and temporal crosscheck techniques to cope with compromised sensor nodes. With our countermeasures in place, compromised master nodes have to return authentic and complete query results to avoid being detected. Built upon symmetric cryptographic primitives, our techniques are shown to be very effective and efficient through comprehensive theoretical analysis and performance evaluations.

The rest of this paper is structured as follows. We introduces the network, query, and adversary models in Section II. Section III illustrates how to perform range queries over encrypted data based on the bucketing technique [14], [15]. A set of completeness verification techniques are then presented and thoroughly evaluated in Sections IV and V, respectively. This paper is finally concluded in Section VI.

II. Network, Query, and Adversary Models

A. Network Model

We assume a large-scale UTSN as shown in Fig. 1 and introduced in Section I. The network region is partitioned into physical *cells*, each containing a master node in charge of sensor nodes in that cell. Here we follow the conventional assumption that master nodes and sensor nodes know their respective geographic locations and also which cell they are in, which can be realized by many existing techniques such as [16], [17].

We do not assume an always-on communication connection to the external network owner. Instead, the network owner can query data by an on-demand wireless link (e.g., a satellite link) connected to some master node(s).

As in [7], [11], [13], we assume that time is divided into *epoches* and that sensor and master nodes are loosely synchronized. At the end of each epoch, each sensor node submits to its master node all the data (if any) it produced during that epoch. Without loss of generality, we subsequently focus on a cell C with N sensor nodes $\{S_i\}_{i=1}^{N}$ and a compromised (yet undetected) master node $\mathcal{M}$. It is worth noting that all the operations also apply to all the other cells with or without compromised master nodes.

B. Multidimensional Range Queries

Event data generated by sensor nodes can generally be described as a tuple of attribute values $\{A_j\}_{j=1}^{\mathbf{d}}$, where $\mathbf{d} \geq 1$ depends on concrete sensor network applications. Each

attribute A_j represents a sensor reading or an aspect of the event such as the weight of an observed object, its location, its speed, or its appearance or lingering time. For sake of simplicity, we will focus on the following type of *primitive* multidimensional range queries,

$$(\text{cell } = C) \wedge (\text{epoch } = t) \bigwedge_{j\in[1,\mathbf{d}]} (l_j \le A_j \le h_j), \quad (1)$$

where C and t denote the cell ID and the interested epoch, respectively, and $[l_j, h_j]$ is the interested range of attribute A_j.

C. Adversary Model

Tremendous efforts have been made to secure general sensor network as well as USNs, see for example [18]–[23]. This paper focuses on secure multidimensional range queries in UTSN. We resort to the existing rich literature for other important issues such as key management, secure routing, broadcast authentication, secure localization, DoS mitigation, and particularly secure and reliable message transmissions.

Although the adversary may directly compromise sensor nodes to read their data and manipulate their behavior, it is much more tempting to take over master nodes for their significant roles in the UTSN. The adversary is assumed to have compromised some master nodes whereby to launch attacks against data confidentiality and query-result authenticity and completeness. The adversary may also compromise sensor nodes to aid compromised master nodes. We, however, follow the conventional assumption that non-compromised sensor nodes are always the majority. Since every master node is only responsible for its own cell, the collusion of compromised master nodes will not do more harm. Our subsequent discussion thus concentrates on one compromised master node $\mathcal{M}$ in charge of a cell with N sensor nodes $\{S_i\}_{i=1}^{N}$.

III. Confidentiality-Preserving Range Queries

In this section, we illustrate how to realize data confidentiality, efficient range queries, and query-result authentication.

To ensure data confidentiality against $\mathcal{M}$, it is necessary to store encrypted data at $\mathcal{M}$ for which $\mathcal{M}$ has no decryption keys. For this purpose, node $S_i, \forall i \in [1, N]$, is preloaded with a distinct initial key $K_{i,0}$ uniquely shared with the network owner. At the end of epoch $t \ge 1$, S_i generates an epoch key by $K_{i,t} = h(K_{i,t-1})$ and erases $K_{i,t-1}$ from its memory, where $h(\cdot)$ denotes a good hash function.

We adapt the bucketing technique in [14], [15] to strike a balance between data confidentiality and query efficiency. Specifically, the domain of attribute $A_j, \forall j \in [1, \mathbf{d}]$, is divided into $\omega_j \ge 1$ consecutive non-overlapping intervals under a public partitioning rule known to the master node and all the sensor nodes, sequentially numbered from 1 to ω_j. A $\mathbf{d}$-dimensional bucket is defined by a tuple, $V = \langle v_1, v_2, \cdots, v_{\mathbf{d}}\rangle$ (called *bucket ID* hereafter), where $v_j \in [1, \omega_j], j \in [1, \mathbf{d}]$, is the interval index of A_j. Although it is possible that $\omega_i \ne \omega_j$ for $i \ne j$, we hereafter assume that $\omega_j = \omega, \forall j \in [1, \mathbf{d}]$, to simplify the presentation. When node S_i produces some data falling into some bucket, we say that S_i generated that bucket. We also denote by Y_i the number of buckets S_i generated during epoch t and by $V_{i,j}, j \in [1, Y_i]$, the jth bucket ID.

At the end of epoch t, every node in cell C encrypts all the data items falling into the same buckets as a whole and sends them with the corresponding bucket IDs to $\mathcal{M}$. Consider node S_i as an example. Assume that $Y_i = 2$ and that S_i has 3 and 2 data items in buckets $V_{i,1} = \langle 3, 5\rangle$ and $V_{i,2} = \langle 6, 3\rangle$, respectively. S_i sends the following message to $\mathcal{M}$ at the end of epoch t:

$$S_i \to \mathcal{M} : i, t, \langle\langle 3, 5\rangle, \{Data_1, Data_2, Data_3\}_{K_{i,t}}\rangle, \langle\langle 6, 3\rangle, \{Data_4, Data_5\}_{K_{i,t}}\rangle,$$

where $\{\cdot\}_\star$ denotes an OCB-like authenticated encryption primitive [24] using the key on the subscript. For conciseness, let us denote all the data items in bucket $V_{i,j}$ by $D_{i,j}$. Then the above message can be represented as

$$S_i \to \mathcal{M} : i, t, \langle V_{i,1}, \{D_{i,1}\}_{K_{i,t}}\rangle, \langle V_{i,2}, \{D_{i,2}\}_{K_{i,t}}\rangle.$$

The query process is straightforward. The network owner first converts the desired data ranges (see Eq. (1)) into a set of bucket IDs, denoted by $\mathcal{Q}_t$, and then sends $\langle C, t, \mathcal{Q}_t\rangle$ to $\mathcal{M}$. Upon receipt of the query, $\mathcal{M}$ returns all the encrypted data buckets received during epoch t whose IDs are within $\mathcal{Q}_t$ along with their corresponding sensor node IDs. The network owner can then derive all the corresponding epoch keys whereby to decrypt the received data buckets. Since the data ranges of interest may not exactly span consecutive full buckets, some buckets in the query reply may contain superfluous data items (false positives) the network owner does not want. We refer to [15] for optimal bucketing strategies which can achieve a good balance between false positives and data confidentiality.

In addition to ensuring data confidentiality, the authenticated encryption primitive allows the network owner to detect forged or juggled data in the query result. Unfortunately, $\mathcal{M}$ may still omit data from some nodes which satisfy the query, leading to query-result incompleteness. This issue is tackled in the following section.

IV. Query-Result Completeness Verification

In this section, we present a set of schemes for the network owner to verify the completeness of query results. Without loss of generality, we still consider cell C with sensor nodes $\{S_i\}_{i=1}^{N}$ and a compromised (yet undetected) master node $\mathcal{M}$. For clarity only, we first temporarily ignore compromised sensor nodes and then discuss their impact and corresponding defenses in Sections IV-D and IV-E, respectively.

To make subsequent theoretical analysis tractable, we make the following assumptions.

- There are totally μ (non-empty) data buckets generated in cell C during each epoch, and each node S_i on average produces $Y_i = \mu/N$ buckets.
- A query about cell C and epoch t is represented by $\langle C, t, \mathcal{Q}_t\rangle$, where $\mathcal{Q}_t \subseteq \Omega$ denotes the set of queried bucket IDs and $\Omega = \{\langle v_1, v_2, \cdots, v_{\mathbf{d}}\rangle | v_i \in [1, \omega], i \in [1, \mathbf{d}]\}$. We further define $\gamma = |\mathcal{Q}_t|/\omega^{\mathbf{d}}$ as the ratio of queried bucket IDs among all the $\omega^{\mathbf{d}}$ ones.
- $\mathcal{M}$ omits each data bucket satisfying $\mathcal{Q}_t$ from the query response with equal *dropping probability* $\delta < 1$. Note that $\mathcal{M}$ may use various δs for different queries.

In what follows, we first report a benchmark verification scheme as a direct extension of the encoding technique [7] to multidimensional cases. Then we introduce two novel probabilistic crosscheck schemes and further propose a random probing scheme to deal with compromised sensor nodes. The following two performance metrics will be used throughout.

- P_{det}**–detection probability**: the probability that $\mathcal{M}$ is detected as having returned incomplete data in response to a query $\mathcal{Q}_t$.
- $\overline{\mathsf{T}}$**–communication cost**: the total communication energy consumption in bits resulting from completeness verification in cell C. Here we assume the same energy to transmit and receive each bit across each hop.

A. Benchmark Scheme: Completeness Verification Based on Encoding Numbers

The basic idea of the encoding technique is to let each sensor node return some unforgeable proof for each empty bucket. Consider S_i as an example. Let $\mathcal{V}_i = \{V_{i,j}\}_{j=1}^{Y_i} \subseteq \Omega$ denote the buckets S_i generated in epoch t. At the end of epoch t, S_i generates a so-called *encoding number* for each empty bucket $V_{i,k} \in \Omega \setminus \mathcal{V}_i$ as

$$num(V_{i,k}, t) = h_{l_e}(i||t||V_{i,k}||K_{i,t}), \tag{2}$$

where $h_{l_e}(\cdot)$ denotes a good hash function of l_e bits and $K_{i,t}$ is S_i's epoch key. $num(V_{i,k}, t)$ is the proof that S_i did not generate data in bucket $V_{i,k}$ during epoch t and can be verified by the network owner who knows $K_{i,t}$. Finally, S_i submits to $\mathcal{M}$ all the data buckets and encoding numbers as follows.

$$S_i \rightarrow \mathcal{M} : i, t, \{V_{i,j}, \{D_{i,j}\}_{K_{i,t}} | V_{i,j} \in \mathcal{V}_i\}, \\ \{V_{i,k}, num(V_{i,k}, t) | V_{i,k} \in \Omega \setminus \mathcal{V}_i\}.$$

Upon receiving the query $\langle C, t, \mathcal{Q}_t \rangle$, $\mathcal{M}$ should generate a hash of all the concatenated encoding numbers from all the N nodes with corresponding empty bucket IDs in $\mathcal{Q}_t$.

$$\text{NUM}_{\mathcal{Q}_t} = h_{l_c}(\mathop{||}_{V_{i,k} \in \mathcal{Q}_t \cap \Omega \setminus \mathcal{V}_i, i \in [1,N]} num(V_{i,k}, t)),$$

where $h_{l_c}(\cdot)$ denotes a good hash function of l_c bits. Then $\mathcal{M}$ returns all the data buckets satisfying $\mathcal{Q}_t$ along with $\text{NUM}_{\mathcal{Q}_t}$.

On receiving the response, the network owner can infer the empty buckets of each node in cell C. Since it knows all the epoch keys $\{K_{i,t}\}_{i=1}^{N}$, it can proceed to generate all the corresponding encoding numbers whereby to recompute $\text{NUM}_{\mathcal{Q}_t}$. If the result matches what it received, the network owner considers $\mathcal{M}$ legitimate and malicious otherwise.

1) Detection probability: Assume that $\mathcal{M}$ chooses to omit some qualified data buckets from the query result. To escape the detection by the network owner, it has to return a correct $\text{NUM}'_{\mathcal{Q}_t}$ corresponding to the incomplete response. If the length l_c of $\text{NUM}'_{\mathcal{Q}_t}$ is sufficiently large, the probability 2^{-l_c} of directly guessing a $\text{NUM}'_{\mathcal{Q}_t}$ is negligible. Since $\mathcal{M}$ does not know the epoch keys of the corresponding sensor nodes to compute the encoding numbers, the only option left is to guess the l_e-bit encoding numbers of the omitted data buckets.

Now we derive $P_{\text{det},E}$, the probability that $\mathcal{M}$ can be detected. Since each of the μ buckets is queried with probability γ and omitted with probability δ, the total number of omitted buckets is approximately $\mu\gamma\delta$. In addition, $\mathcal{M}$ can guess the correct encoding number for each omitted bucket with probability 2^{-l_e}. The network owner cannot detect $\mathcal{M}$ if all the $\mu\gamma\delta$ buckets' encoding numbers are guessed correctly, which happens with probability $2^{-l_e\mu\gamma\delta}$. We thus have

$$P_{\text{det},E} = 1 - 2^{-l_e\mu\gamma\delta}. \tag{3}$$

If $l_e\mu\gamma\delta$ is sufficiently large, then the detection probability $P_{\text{det},E}$ will be very close to one. This encoding technique can thus be viewed as a deterministic approach.

2) Communication cost: The communication cost $\overline{\mathsf{T}}$ of this scheme is incurred by transmitting the encoding numbers to $\mathcal{M}$. Since there are totally μ non-empty buckets and $N\omega^{\mathbf{d}} - \mu$ empty buckets, $N\omega^{\mathbf{d}} - \mu$ encoding numbers along with the corresponding bucket IDs need to be transmitted. Assuming that the average number of hops between each sensor node and $\mathcal{M}$ is L_{avg}, then $\overline{\mathsf{T}}$ is given by

$$\overline{\mathsf{T}}_E = (N\omega^{\mathbf{d}} - \mu)(l_e + \lceil \log \omega \rceil \mathbf{d}) L_{avg}, \tag{4}$$

where $\lceil \log \omega \rceil \mathbf{d}$ is the length of a bucket ID in bits. As we can see, $\overline{\mathsf{T}}_E$ is acceptable when $\mu \simeq N\omega^{\mathbf{d}}$, i.e., there are almost no empty buckets. In practice, however, $\mu \ll N\omega^{\mathbf{d}}$ in event-driven sensor networks. This means that the communication cost of the encoding technique increases exponentially with $\mathbf{d}$ (the number of data attributes), i.e., $\overline{\mathsf{T}}_E = O(\omega^{\mathbf{d}}\mathbf{d})$. It is thus necessary to seek other techniques with better communication efficiency to cope with resource-constrained sensor nodes.

B. Probabilistic Spatial Crosscheck

The encoding technique enables deterministic detection of every query result's incompleteness with a communication cost $\overline{\mathsf{T}} = O(\omega^{\mathbf{d}}\mathbf{d})$. If the network owner can tolerate a small number of incomplete responses before detecting $\mathcal{M}$, it is feasible to design some probabilistic verification schemes with much less energy consumption.

The key idea of probabilistic spatial crosscheck is to embed some relationships among data generated by different sensor nodes. If $\mathcal{M}$ omits part of the data in the response, the network owner can decide with certain probability that the query result is incomplete by inspecting the relationships among other returned data. This technique thus forces $\mathcal{M}$ to either return all the data satisfying the query or risk being caught. To enable spatial crosscheck, we introduce a *gossip phase* at the end of each epoch t, in which sensor nodes exchange information about their sensed data before sending them to $\mathcal{M}$. Then each node that has data to submit appends some randomly-chosen received information to its own data buckets and then sends encrypted buckets to $\mathcal{M}$.

In particular, during the gossip phase of epoch t, each node with data for submission, say S_i, broadcasts a gossip message within cell C which contains each of the generated bucket IDs $\{V_{i,j}\}_{j=1}^{Y_i}$ with equal probability p_b. Here we assume a suitable broadcast authentication protocol like multilevel μTESLA [25] to ensure their authenticity.

At the end of the gossip phase, S_i receives a gossip message from every other node that has generated data in epoch t and thus knows the IDs of all the non-empty buckets which include its own ones and are denoted by $\mathcal{V}$. Recall that $\{D_{i,j}\}_{i=1}^{Y_i}$

denote the data items in bucket $V_{i,j}$. Then S_i appends each element in $\mathcal{V}$ along with the corresponding node ID to each element in $\{D_{i,j}\}_{i=1}^{Y_i}$ with equal probability p_e with the exception that $V_{i,j}$ will not be appended to $D_{i,j}$. With this measure, it is possible that a bucket ID may be appended to multiple data blocks at other sensor nodes and also to other buckets generated by the same sensor node. Subsequently, S_i sends encrypted data buckets to $\mathcal{M}$ as before.

For instance, suppose that S_i generated data blocks $\{D_{i,j}\}_{j=1}^{3}$ in buckets $\{V_{i,j}\}_{j=1}^{3}$ (i.e., $Y_i = 3$), respectively, and received $V_{k,1}$ from node S_k and $V_{u,1}$ and $V_{u,2}$ from node S_u. Also assume that S_i has decided to append $\langle i, V_{i,2}\rangle$ and $\langle k, V_{k,1}\rangle$ to $D_{i,1}$; append $\langle u, V_{u,2}\rangle$ and $\langle i, V_{i,3}\rangle$ to $D_{i,2}$; and append $\langle u, V_{u,1}\rangle$ and $\langle k, V_{k,1}\rangle$ to $D_{i,3}$. Finally, S_i sends the following message to $\mathcal{M}$.

$$\begin{aligned} S_i \rightarrow \mathcal{M} : i, t, &\langle V_{i,1}, \{D_{i,1}, \langle i, V_{i,2}\rangle, \langle k, V_{k,1}\rangle\}_{K_{i,t}}\rangle, \\ &\langle V_{i,2}, \{D_{i,2}, \langle u, V_{u,2}\rangle, \langle i, V_{i,3}\rangle\}_{K_{i,t}}\rangle, \\ &\langle V_{i,3}, \{D_{i,3}, \langle u, V_{u,1}\rangle, \langle k, V_{k,1}\rangle\}_{K_{i,t}}\rangle. \end{aligned}$$

Note that $\mathcal{M}$ cannot figure out the bucket IDs embedded in each bucket due to the encryption, that is, $\mathcal{M}$ does not know which buckets can crosscheck each other.

Upon receiving the query $\langle C, t, \mathcal{Q}_t\rangle$, $\mathcal{M}$ should return all the data buckets with IDs in $\mathcal{Q}_t$. The network owner then decrypts the data buckets in the response to get a set of embedded node/bucket ID pairs. If any such embedded bucket ID is in $\mathcal{Q}_t$ with the corresponding data bucket not being returned, the network owner decides that $\mathcal{M}$ omitted that data bucket and thus considers it malicious. For example, assume that $\mathcal{Q}_t$ contains $V_{i,1}$, $V_{i,2}$, and $V_{k,1}$ and that $\mathcal{M}$ did not return bucket $V_{k,1}$ of node S_k. If the network owner receives either of $\langle V_{i,1}, \{D_{i,1}, \langle i, V_{i,2}\rangle, \langle k, V_{k,1}\rangle\}_{K_{i,t}}\rangle$ and $\langle V_{i,3}, \{D_{i,3}, \langle u, V_{u,1}\rangle, \langle k, V_{k,1},\rangle\}_{K_{i,t}}$, it can find that it should also have received bucket $V_{k,1}$ of node S_k. The network owner can then consider $\mathcal{M}$ malicious.

1) Detection probability: Since each data buckets satisfies $\mathcal{Q}_t$ with probability γ and is omitted with probability δ, the network owner receives on average $(1-\delta)\mu\gamma$ buckets. Let $\mathcal{V}_r$ and $\mathcal{V}_d$ denote the set of buckets that the network owner receives and $\mathcal{M}$ omits, respectively. We can see that each bucket ID in $\mathcal{V}_d$ is embedded into each bucket in $\mathcal{V}_r$ with probability $p_b p_e$. The network owner cannot detect the incompleteness of the query result if no bucket in $\mathcal{V}_d$ whose ID has been embedded into any of bucket in $\mathcal{V}_r$, which happens with probability $(1-p_b p_e)^{|\mathcal{V}_r||\mathcal{V}_d|}$, where $|\mathcal{V}_r| = (1-\delta)\mu\gamma, |\mathcal{V}_d| = \delta\mu\gamma$. Therefore, we have

$$P_{\mathrm{det},S} = 1 - (1-p_b p_e)^{\mu^2\gamma^2\delta(1-\delta)}. \tag{5}$$

2) Communication cost: Now we estimate the communication cost $\overline{\mathsf{T}}$, incurred by the gossip messages and the transmission of embedded node/bucket IDs to $\mathcal{M}$. To simplify the analysis, we assume the simplest broadcast technique with which each node receives and broadcasts a message once.

Recall that each bucket ID is of $\lceil \log \omega \rceil \mathbf{d}$ bits. Since each node on average produces μ/N buckets, $\mu p_b/N$ bucket IDs will be inserted into a gossip message. Assuming that each node ID is of l_{id} bits, a gossip message is of $l_{id} + \mu p_b \lceil \log \omega \rceil \mathbf{d}/N$ bits. Since each of the N gossip messages will be transmitted and received N times, the associated communication cost is $N^2(l_{id} + \mu p_b \lceil \log \omega \rceil \mathbf{d}/N)$. According to the probabilistic gossiping and embedding process, each of the μ bucket IDs will have on average approximately $\mu p_b p_e$ copies at different nodes. The communication cost associated with transmitting the embedded node/bucket IDs to $\mathcal{M}$ is then $\mu^2 p_b p_e L_{avg}(l_{id} + \lceil \log \omega \rceil \mathbf{d})$. Therefore, we have

$$\overline{\mathsf{T}}_S = N(N l_{id} + \mu p_b \lceil \log \omega \rceil \mathbf{d}) + \mu^2 p_b p_e L_{avg}(l_{id} + \lceil \log \omega \rceil \mathbf{d}). \tag{6}$$

Apparently, the communication cost of spatial crosscheck is of order $O(\mathbf{d})$, which is significantly lower than that of the encoding technique, i.e., $O(\omega^{\mathbf{d}}\mathbf{d})$.

C. Probabilistic Temporal Crosscheck

In this section, we propose a probabilistic temporal crosscheck technique as a complement to spatial crosscheck. The key idea is to embed some relationship between data buckets generated in different epoches. If $\mathcal{M}$ did not correctly respond to query $\langle C, t, \mathcal{Q}_t\rangle$, but it answers query $\langle C, t', \mathcal{Q}_{t'}\rangle$, where $t' - t \leq \kappa$ and κ is a system parameter, then the network owner can catch $\mathcal{M}$'s misbehavior with very high probability.

To enable temporal crosscheck, we let each node $S_i, \forall i \in [1, N]$, maintain a fixed-length buffer of $\kappa L(l_{ep} + \lceil \log \omega \rceil \mathbf{d})$ bits, where L is a system parameter and l_{ep} denotes the length of an epoch ID. The buffer can thus hold κL bucket IDs and their corresponding epoch numbers. We subsequently call the pair $\langle t, V_{i,j}\rangle$ an *enhanced bucket ID*, which can indicate node S_i has generated bucket $V_{i,j}$ during epoch t. Denote the κL enhanced bucket IDs in node S_i's buffer by $\{E_{i,j}\}_{j=1}^{\kappa L}$. At the end of epoch t, S_i appends with probability p_t each element in $\{E_{i,j}\}_{j=1}^{\kappa L}$ to each of the buckets $\{D_{i,j}\}_{j=1}^{Y_i}$ it generated in epoch t, and also replaces the oldest $L_i = \min\{L, Y_i\}$ ones with L_i bucket IDs randomly chosen from $\{V_{i,j}\}_{j=1}^{Y_i}$.

Temporal crosscheck relies on queries issued in subsequent κ epoches to detect any incompleteness in the query result of $\mathcal{Q}_t$. Consider two queries $\langle C, t, \mathcal{Q}_t\rangle$ and $\langle C, t', \mathcal{Q}_{t'}\rangle$, where $t' - t \leq \kappa$. The network owner can verify the completeness of the query result for $\mathcal{Q}_t$ based on the query result for $\mathcal{Q}_{t'}$. In particular, if the decrypted result for $\mathcal{Q}_{t'}$ contains any $E_{i,j}$ satisfying $\mathcal{Q}_t$ while the corresponding data bucket was not found in the query result, the network owner considers $\mathcal{M}$ malicious.

1) Detection probability: Recall that there are on average totally μ data buckets generated during each epoch t. For simplicity, we assume that $Y_i = \mu/N \geq L, \forall i \in [1, N]$, so that the buffer at each node contains L enhanced bucket IDs generated in each of the past κ epoches.

Assume that $\mathcal{M}$ has returned an incomplete response to query $\mathcal{Q}_t$. Because $\mathcal{M}$ omits each qualified bucket with probability δ, there are on average $L\gamma\delta$ bucket IDs in the buffer of each node, say S_i, whose corresponding data buckets have been omitted by $\mathcal{M}$ in response to $\mathcal{Q}_t$.

Consider the query $\langle C, t+1, \mathcal{Q}_{t+1}\rangle$, $\mathcal{M}$ will return $\mu\gamma(1-\delta)/N$ buckets of S_i to the network owner. If none of them contains any omitted bucket ID in the buffer of epoch t, $\mathcal{M}$ cannot be detected as having omitted S_i's bucket in the response to query $\mathcal{Q}_t$, which happens with probability $(1-p_t)^{L\mu\gamma^2\delta(1-\delta)/N}$. Since there are N nodes, the network

owner cannot detect the incompleteness of the result for query $\mathcal{Q}_t$ by examining the query result of query $\mathcal{Q}_{t+1}$ with probability $(1-p_t)^{L\mu\gamma^2\delta(1-\delta)}$. Also note that $\mathcal{M}$ cannot be detected either if the network does not issue a query $\mathcal{Q}_t$, which is assumed to occur with probability $1-p_q$. It follows that

$$\begin{aligned} P_{\mathrm{det},T}^{\mathcal{Q}_{t+1}} &= 1-(1-p_q)-p_q(1-p_t)^{L\mu\gamma^2\delta(1-\delta)} \\ &= p_q(1-(1-p_t)^{L\mu\gamma^2\delta(1-\delta)}). \end{aligned} \tag{7}$$

Now consider all subsequent κ epoches. We have

$$\begin{aligned} P_{\mathrm{det},T} &= 1-\prod_{i=1}^{\kappa}(1-P_{\mathrm{det},T}^{\mathcal{Q}_{t+i}}) \\ &= 1-(1-p_q(1-(1-p_t)^{L\mu\gamma^2\delta(1-\delta)}))^{\kappa}. \end{aligned} \tag{8}$$

2) Communication cost: The communication cost $\overline{\mathsf{T}}$ of temporal crosscheck is incurred by transmitting the embedded bucket IDs from the buffer at each of the N nodes. Each node has on average μ/N buckets for submission to $\mathcal{M}$, into each of which $\kappa L p_t$ enhanced bucket IDs from the buffer will be inserted. We therefore have

$$\overline{\mathsf{T}}_T = \kappa L\mu p_t(l_{ep}+\lceil\log\omega\rceil\mathbf{d})L_{avg}, \tag{9}$$

which is of order $O(\mathbf{d})$.

D. Impact of Compromised Sensor Nodes

Now we discuss the impact of compromised sensor nodes on query-result completeness verification.

Case 1: Disobey the verification operations.

Both the encoding technique and spatial/temporal crosscheck rely on sensor nodes to provide some verification information for query-result completeness verification. Compromised sensor nodes can cooperate with $\mathcal{M}$ to misbehave as follows.

- For the encoding technique, $\mathcal{M}$ can use the known epoch keys to derive all the encoding numbers for the compromised nodes and omit arbitrary qualifying buckets from them in the query result without being detected.
- For spatial crosscheck, compromised sensor nodes can neither broadcast their own bucket IDs nor insert any received bucket ID into their own data buckets during the gossip phase.
- For temporal crosscheck, compromised sensor nodes can simply choose not to embed any bucket ID in the buffer into their newly generated data buckets.

We note that the above misbehavior will not affect the operations of non-compromised sensor nodes; therefore, the detection performance of the proposed techniques will not be affected much as long as non-compromised sensor nodes are always the majority.

Case 2: Return data only from compromised sensor nodes.

$\mathcal{M}$ may also defeat spatial and temporal crosscheck by returning data only from compromised sensor nodes. Since the buckets of compromised sensor nodes do not include the bucket IDs of non-compromised ones, the network owner cannot detect that $\mathcal{M}$ omitted the data of non-compromised sensor nodes. This attack is more intelligent and will be addressed in the next section.

E. Random Probing

In this section, we further propose a random-probing scheme as a complement to spatial and temporal crosscheck schemes, in which the network owner probes some random chosen nodes of which no data were returned in the query result.

To enable this, at the end of each epoch t, node $S_i, \forall i \in [1,N]$, randomly chooses $\min\{m, Y_i\}$ bucket IDs from $\{V_{i,j}\}_{j=1}^{Y_i}$ it generated in epoch t, and submits them to the master node after encrypting with its epoch key, where m is a system parameter. Specifically, let $\Pi_{i,t}$ denote the set of chosen bucket IDs. Node S_i additionally submits $\{\Pi_{i,t}\}_{K_{i,t}}$ to $\mathcal{M}$ at the end of epoch t. A special case is that when $Y_i = 0$, i.e., $\Pi_{i,t} = \emptyset$, S_i also submits a condensed proof $\{i,t\}_{K_{i,t}}$ showing that it has no data in epoch t.

Consider as an example a random probe for query $\mathcal{Q}_t$. After receiving the result for $\mathcal{Q}_t$, the network owner identifies the set $\mathcal{S}_t$ of nodes whose data appear in the query result. The network owner then randomly picks $\lambda' = \min\{\lambda, N-|\mathcal{S}_t|\}$ nodes from $\{S_i\}_{i=1}^{N}\backslash\mathcal{S}_t$ whose data do not appear in the query result, where λ is a system parameter. Denote by Λ the set of node IDs chosen by the network owner. The network owner sends a probing request $\langle t, \Lambda\rangle$ to $\mathcal{M}$. On receiving the probe, $\mathcal{M}$ returns $\langle i, t, \{\Pi_{i,t}\}_{K_{i,t}}\rangle$ or $\langle i, t, \{i,t\}_{K_{i,t}}\rangle, \forall i \in \Lambda$. After receiving all the λ' responses, the network owner decrypts them using the corresponding epoch keys. If any received bucket ID is in $\mathcal{Q}_t$, the network owner knows that $\mathcal{M}$ has omitted the corresponding data bucket and thus considers $\mathcal{M}$ malicious.

1) Detection probability: For simplicity, we assume that $\lambda' = \lambda$ and that $Y_i \geq m, \forall i \in [1,N]$. Then the network owner will receive m bucket IDs in each of the λ probe responses. $\mathcal{M}$ cannot be detected if none of the $m\lambda$ bucket IDs is in $\mathcal{Q}_t$, which happens with probability $(1-\gamma)^{m\lambda}$. Therefore, $\mathcal{M}$ can be detected with probability

$$P_{\mathrm{det},R} = 1-(1-\gamma)^{m\lambda}. \tag{10}$$

2) Communication cost: Since each node submits m bucket IDs to $\mathcal{M}$, we have

$$\overline{\mathsf{T}}_R = Nm\lceil\log\omega\rceil\mathbf{d}L_{avg}\,, \tag{11}$$

which is of order $O(\mathbf{d})$.

F. Hybrid Crosscheck

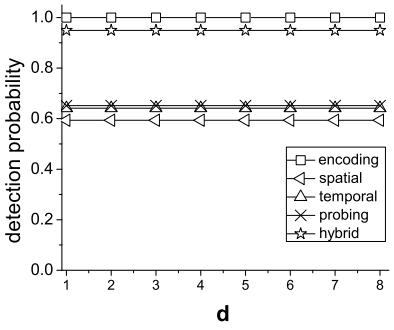
It is natural to build a hybrid scheme on spatial crosscheck, temporal crosscheck, and random probing. Given that the above three techniques fail with probabilities $\overline{P_{\mathrm{det},S}} = (1-p_b p_e)^{\mu^2\gamma^2\delta(1-\delta)}$, $\overline{P_{\mathrm{det},T}} = (1-p_q(1-(1-p_t)^{L\mu\gamma^2\delta(1-\delta)}))^{\kappa}$, $\overline{P_{\mathrm{det},R}} = (1-\gamma)^{m\lambda}$, respectively, we have

$$P_{\mathrm{det},H} = 1-\overline{P_{\mathrm{det},S}}\cdot\overline{P_{\mathrm{det},T}}\cdot\overline{P_{\mathrm{det},R}}\,. \tag{12}$$

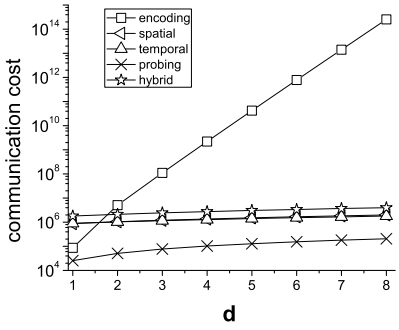
The communication cost of the hybrid scheme is simply given by $\overline{\mathsf{T}}_H = \overline{\mathsf{T}}_S + \overline{\mathsf{T}}_T + \overline{\mathsf{T}}_R$, where $\overline{\mathsf{T}}_S$, $\overline{\mathsf{T}}_T$ and $\overline{\mathsf{T}}_R$ are given in Eqs. (6), (9), and (11), respectively.

V. Performance Evaluation

In this section, we compare the detection probability P_{det} and the communication cost $\overline{\mathsf{T}}$ of the proposed schemes using numerical results.

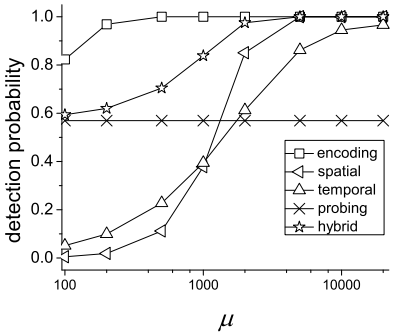


(a) detection probability P_{det}

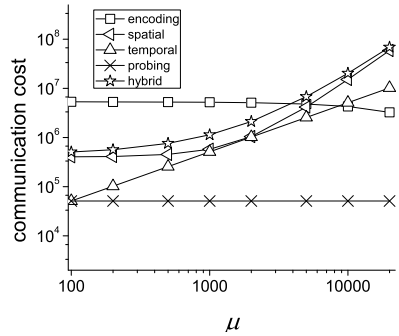


(b) communication cost $\overline{\mathsf{T}}$

Fig. 2. Impact of $\mathbf{d}$, the number of data dimensions.

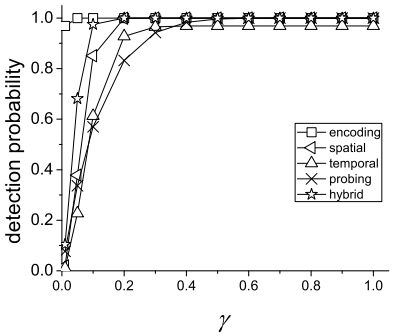


(a) detection probability P_{det}

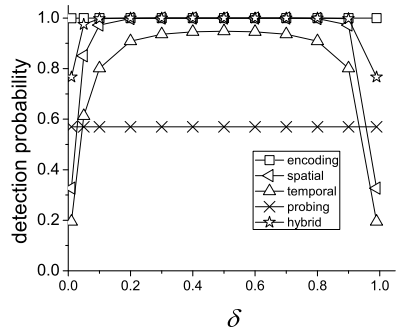


(b) communication cost $\overline{\mathsf{T}}$

Fig. 3. Impact of μ, the total number of buckets per epoch.



(a) detection probability P_{det}



(b) detection probability P_{det}

Fig. 4. Impact of γ and δ.

TABLE I
DEFAULT EVALUATION PARAMETERS

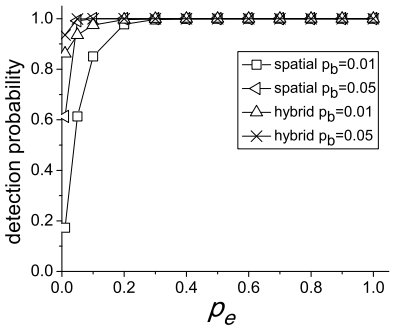
Para.	Val.	Para.	Val.	Para.	Val.	Para.	Val.
N	200	$\mathbf{d}$	2	ω	16	μ	2000
γ	0.1	δ	0.05	p_b	0.02	p_e	0.05
κ	5	L	4	p_t	0.2	p_q	0.5
m	4	λ	2	l_{id}	10	l_e	5
l_{ep}	24	L_{avg}	8				

A. Numeric Results

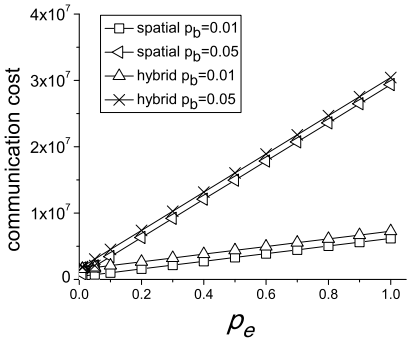
We assume a cell consisting of 400 sensor nodes and a master node. We also assume error-free and collision-free packet transmissions. Table I summarizes other default evaluation parameters unless specified otherwise.

1) Impact of $\mathbf{d}$*:* Fig. 2 shows the impact of $\mathbf{d}$, the number of dimensions or queriable attributes. We can see that the P_{det}s of the five schemes are all independent of $\mathbf{d}$. Under the default evaluation parameters, the encoding scheme has the highest detection probability $P_{\mathrm{det},E} = 1$. Although the other schemes have smaller P_{det}s, they can still enable quick detection of $\mathcal{M}$. The slight sacrifice in P_{det} leads to significant savings in the communication cost $\overline{\mathsf{T}}$, which is shown in Fig. 2(b) in $\log 10$ scale. Not surprisingly, the $\overline{\mathsf{T}}$ of the encoding scheme increases exponentially with $\mathbf{d}$, while the $\overline{\mathsf{T}}$ of each other scheme grows linearly with and is relatively insensitive to $\mathbf{d}$.

2) Impact of μ*:* Fig. 3 shows the the impact of μ, the total number of buckets generated in cell C per epoch. Since the P_{det} of the encoding scheme is still 1, we do not discuss



(a) detection probability $P_{\rm det}$



(b) communication cost $\overline{\mathsf{T}}$

Fig. 5. Impact of p_b and p_e.

it any further. The $P_{\rm det}$s of spatial and temporal crosscheck dramatically increase as μ increases from 100 to 1000 and finally converge to $P_{{\rm det},S} \approx 1$ and $P_{{\rm det},T} \approx 0.97$, respectively. This observation is anticipated for two reasons. First, the larger μ, the more buckets (i.e., $\mu\gamma\delta$) dropped by $\mathcal{M}$, and the more traces left for detection. Second, $P_{{\rm det},T}$ is upper-bounded by $1-(1-p_q)^\kappa$ (see Eq. (8)). Fig. 3(b) compares the $\overline{\mathsf{T}}$s of the five schemes. The encoding scheme's $\overline{\mathsf{T}}$ decreases with μ because the more data buckets generated, the fewer empty buckets and thus the fewer encoding numbers need be sent. In contrast, the $\overline{\mathsf{T}}$s of spatial crosscheck, temporal crosscheck, and the hybrid scheme all increase with μ. This is not surprising because the more buckets generated, the more relationships embedded among them. It is worth noticing that the random probing scheme has a much smaller $\overline{\mathsf{T}}$ with a moderate $P_{\rm dec}$, which seems to outperform all the other four schemes. However, because its $P_{\rm dec}$ is largely determined by γ and independent of μ, when γ is small and μ is large, $P_{\rm dec}$ of random probing could be low while $P_{\rm dec}$s of the spatial and temporal crosscheck schemes could still be sufficiently high. Therefore, we only consider it as a complementary scheme.

3) Impact of γ and δ: Fig. 4(a) shows the impact of γ, the interest ratio or the probability of a bucket being queried. As we can see, the $P_{\rm det}$s of the four probabilistic schemes increase with γ because the larger γ, the more buckets received by the network owner, the easier to detect data incompleteness.

Fig. 4(b) shows the impact of δ, the dropping probability. The $P_{\rm det}$s of spatial and temporal crosscheck are both maximized at $\delta = 0.5$ and approach zero at $\delta \approx 1$ due to the term $\delta(1-\delta)$ in $P_{{\rm det},S}$ and $P_{{\rm det},T}$ (cf. Eq.(5) and Eq.(8)). This means that the spatial and temporal crosscheck schemes cannot detect $\mathcal{M}$ alone in all cases. The hybrid scheme overcomes this by integrating the random probing scheme whose $P_{\rm det}$ is independent of δ because only the sensor nodes whose data do not appear in the response are probed.

4) Impact of p_b and p_e: Fig. 5 shows the impact of p_b (the probability of a bucket ID being broadcasted) and p_e (the probability of a received bucket ID being embedded) on the spatial and hybrid schemes. In general, the larger p_b and p_e, the higher the $P_{\rm det}$s, and the larger the $\overline{\mathsf{T}}$s. We can see in Fig. 5(a) that p_b need not be too large; what really matters is $p_b p_e$ (cf. Eq. (5)). Fig. 5(b) shows that p_b has larger influence on the $\overline{\mathsf{T}}$s of both schemes. Therefore, we can use smaller p_b with larger p_e to achieve a desirable $P_{\rm det}$.

5) Impact of p_q and p_t: Fig. 6 shows the impact of p_q (the probability that the network owner issues a query for a particular epoch) and p_t (the probability that a bucket ID is stored in the buffer for temporal crosscheck) on temporal crosscheck and the hybrid scheme. In general, the larger p_q and p_t, the higher $P_{\rm det}$s, the larger $\overline{\mathsf{T}}$s. However, as we can see in Fig. 6(a), since the $P_{\rm det}$ of temporal crosscheck is upper-bounded by $1-(1-p_q)^\kappa$, the temporal scheme is less effective when p_q is small. In contrast, p_q and p_t have less impact on the $P_{\rm det}$ of the hybrid scheme whose detection probability is guaranteed by the other two components. In addition, there is a linear relationship between p_t and the $\overline{\mathsf{T}}$ of temporal crosscheck, as shown in Fig. 6(b).

We also evaluate the impact of other parameters on our proposed schemes, such as $N, \omega, L, p_c, \kappa, m, \lambda$. Because their impacts are easy to understand from previous analysis, the results are thus omitted here due to the space constraints.

VI. Conclusion

In this paper, we took multidimensional range queries as an example to investigate secure cooperative data storage and query processing in UTSN by proposing a suite of novel schemes. The proposed schemes can not only ensure data confidentiality against master nodes and achieve query efficiency, but also enable the network owner to verify the authenticity and completeness of any query result with low cost. Detailed performance evaluations confirmed the high efficacy and efficiency of the proposed schemes.

References

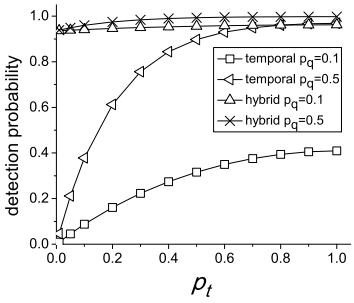
[1] R. Zhang, J. Shi, and Y. Zhang, "Secure multidimensional range queries in sensor networks," in *ACM MobiHoc'09*, New Orleans, LA, May 2009.

[2] R. D. Pietro, L. V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Catch me (if you can): Data survival in unattended sensor networks," in *PerCom'08*, Hong Kong, China, Mar. 2008, pp. 185–194.

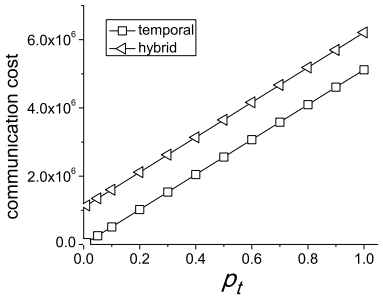
[3] D. Ma, C. Soriente, and G. Tsudik, "New adversary and new threats: security in unattended sensor networks," *IEEE Network*, vol. 23, no. 2, pp. 43-48, 2009.

[4] P. Desnoyers, D. Ganesan, and P. Shenoy, "TSAR: A two tier sensor storage architecture using interval skip graphs," in *ACM SenSys'05*, San Diego, California, USA, Nov. 2005, pp. 39–50.

[5] B. Sheng, Q. Li, and W. Mao, "Data storage placement in sensor networks," in *ACM MobiHoc'06*, Florence, Italy, May 2006.



(a) detection probability P_{det}



(b) communication cost $\overline{\mathsf{T}}$

Fig. 6. Impact of p_q and p_t.

[6] M. Shao, S. Zhu, W. Zhang, and G. Cao, "pDCS: Security and privacy support for data-centric sensor networks," in *IEEE INFOCOM'07*, Anchorage, Alaska, USA, May 2007, pp. 1298–1306.

[7] B. Sheng and Q. Li, "Verifiable privacy-preserving range query in sensor networks," in *IEEE INFOCOM'08*, Phoenix, AZ, Apr. 2008, pp. 46–50.

[8] O. Gnawali, K.-Y. Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler, "The tenet architecture for tiered sensor networks," in *ACM SenSys'06*, Boulder, Colorado, USA, Oct. 2006.

[9] X. Li, Y. J. Kim, R. Govindan, and W. Hong, "Multi-dimensional range queries in sensor networks," in *ACM SenSys'03*, Los Angeles, California, USA, Nov. 2003, pp. 63–75.

[10] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: an acquisitional query processing system for sensor networks," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, Mar. 2005.

[11] J. Shi, R. Zhang, and Y. Zhang, "Secure range queries in tiered sensor networks," in *IEEE INFOCOM'09*, Rio de Janeiro, Brazil, Apr. 2009.

[12] ——, "A spatiotemporal approach for secure range queries in tiered sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 1, pp. 264–273, Jan. 2011.

[13] F. Chen and A. Liu, "SafeQ: Secure and efficient query processing in sensor networks," in *INFOCOM'10*, San Diego, CA, Mar. 2010, pp. 1–9.

[14] H. Hacigümüş, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in *ACM SIGMOD'02*, Madison, Wisconsin, 6 2002, pp. 216–227.

[15] B. Hore, S. Mehrotra, and G. Tsudik, "A privacy-preserving index for range queries," in *VLDB'04*, Toronto, Canada, Aug. 2004, pp. 720–731.

[16] D. Liu, P. Ning, and W. Du, "Attack-resistant location estimation in sensor networks," in *IPSN'05*, Los Angeles, CA, Apr. 2005.

[17] Y. Zhang, W. Liu, Y. Fang, and D. Wu, "Secure localization and authentication in ultra-wideband sensor networks," *IEEE J. Sel. Areas Commun., Special Issue on UWB Wireless Communications - Theory and Applications*, vol. 24, no. 4, pp. 829–835, Apr. 2006.

[18] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE S&P'03*, Oakland, CA, May 2003.

[19] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *ACM CCS'03*, Washington, DC, Oct. 2003, pp. 62–72.

[20] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE J. Sel. Areas Commun., Special Issue on Security in Wireless Ad Hoc Networks*, vol. 24, no. 2, pp. 247–260, Feb. 2006.

[21] L. Ma, X. Cheng, F. Liu, F. An, and M. Rivera, "iPAK: An in-situ pairwise key bootstrapping scheme for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 8, pp. 1174–1184, Aug. 2007.

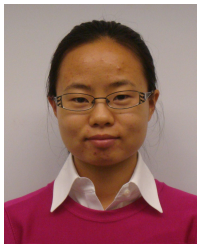
[22] R. Zhang, Y. Zhang, and K. Ren, "DP^{2}AC: Distributed privacy-preserving access control in sensor networks," in *IEEE INFOCOM'09*, Rio de Janeiro, Brazil, Apr. 2009.

[23] R. Zhang, J. Shi, Y. Liu, and Y. Zhang, "Verifiable fine-grained top-k queries in tiered sensor networks," in *INFOCOM'10*, San Diego, CA, Mar. 2010.

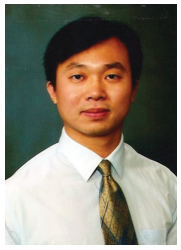
[24] P. Rogaway, M. Bellare, and J. Black, "OCB: A block-cipher mode of operation for efficient authenticated encryption," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 3, pp. 365–403, Aug. 2003.

[25] D. Liu and P. Ning, "Multilevel μTESLA: Broadcast authentication for distributed sensor networks," *Trans. Embedded Computing Sys.*, vol. 3, no. 4, pp. 800–836, 2004.

Rui Zhang received the B.E. in Communication Engineering and the M.E. in Communication and Information System from Huazhong University of Science and Technology, China, in 2001 and 2005, respectively. He is currently a Ph.D. student in School of Electrical, Computer, and Energy Engineering at Arizona State University. His research interests are network and distributed system security, wireless networking, and mobile computing. He was a software engineer in UTStarcom Shenzhen R&D center from 2005 to 2007.



Jing Shi received the B.E. in Communication Engineering and the M.E. in Communication and Information System from Huazhong University of Science and Technology, China, in 2003 and 2006, respectively, and the Ph.D. in Electrical and Computer Engineering from New Jersey Institute of Technology in 2010. She is currently a lecturer in School of Public Administration at Huazhong University of Science and Technology, China. Her research interests are network and distributed system security, wireless networking, and mobile computing.



Yanchao Zhang received the B.E. in Computer Science and Technology from Nanjing University of Posts and Telecommunications, China, in 1999, the M.E. in Computer Science and Technology from Beijing University of Posts and Telecommunications, China, in 2002, and the Ph.D. in Electrical and Computer Engineering from the University of Florida in 2006. He is currently as an Associate Professor in School of Electrical, Computer, and Energy Engineering at Arizona State University. Before ASU, he was an Assistant Professor of Electrical and Computer Engineering at New Jersey Institute of Technology from 2006 to 2010. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is (was) an Associate Editor of IEEE Transactions on Vehicular Technology, a Feature Editor of IEEE Wireless Communications, a Guest Editor of IEEE Wireless Communications Special Issue on Security and Privacy in Emerging Wireless Networks in 2010, and a TPC Co-Chair of Communication and Information System Security Symposium, IEEE GLOBECOM 2010. He received the NSF CAREER Award in 2009.

Jinyuan (Stella) Sun received the PhD degree in Electrical and Computer Engineering from University of Florida with Dr. Yuguang Michael Fang, in 2010. She has been an assistant professor in the Department of Electrical Engineering and Computer Science at University of Tennessee since August 2010. Her research interests include the security protocol and architecture design of wireless networks and critical systems.