Secure and Fault-Tolerant Event Boundary Detection in Wireless Sensor Networks

Kui Ren, *Member, IEEE*, Kai Zeng, *Student Member, IEEE*, and Wenjing Lou, *Member, IEEE*

***Abstract*— Event boundary detection is in and of itself a useful application in wireless sensor networks (WSNs). Typically, it includes the detection of a large-scale spatial phenomenon such as the transportation front line of a contamination or the diagnosis of network health. In this paper, we present SEBD, a fully distributed and light-weight Secure Event Boundary Detection scheme, which implements secure and fault-tolerant detection of event boundaries in an adversarial environment. An efficient key establishment protocol is first proposed which establishes location based keys at each sensor node to secure the communications. The idea of location-based keys also effectively minimizes the impact of node compromise such that a compromised node cannot impersonate other nodes at locations other than where it is. Then a collaborative endorsement scheme is designed to allow multiple nodes collectively endorsing a valid boundary claim for increased resilience against node compromise. SEBD further develops an enhanced (nonparametric) statistical model that supports localized detection and shows a much better accuracy and fault tolerance property as compared to previous models. The security strength and performance of SEBD are evaluated by both analysis and simulations.**

***Index Terms*— Security, wireless sensor network, event boundary detection.**

I. Introduction

An important application of WSNs is to monitor, detect, and report the occurrences of events of interest, such as forest fire, environment temperature, chemical spills, network health, etc. [1]–[4]. Due to the strict resource limitations (e.g., battery power, bandwidth, etc.) of sensor nodes and the nature of some events, it is not feasible to collect all sensor measurements and compute event boundaries in a centralized manner [4], [5]. A localized approach that allows in-network processing is therefore demanded. Sensor nodes are expected to collaborate with each other based on each own local view and provide a global picture for spatially distributed phenomena with greatly improved efficiency. Recently, several localized boundary detection schemes have been proposed [3], [4], [6]–[8]. All these schemes assume a trustworthy environment, and would fail in adversarial environments. Their resilience to node random measurement error is also very limited. However, for WSNs deployed in security-sensitive environments, it is critical for an event boundary detection scheme to be highly resilient against both node compromise and random fault.

In this paper, we study how to securely detect event boundary in WSNs under adversarial environments with enhanced fault-tolerance. In a trustworthy environment, each node reports its measurements honestly and a node with erroneous measurements will suppress/abort its own observation based on the information collected from other nodes in its neighborhood. However, this is not true in an adversarial environment where malicious compromised nodes exist. Compromised nodes can always lie about its measurements, claim to be a boundary node when it is not, or refuse to report itself as a boundary node when it is. Moreover, compromised nodes may also collude to fabricate non-existing event boundary to deceive the sink and cause erroneous actions taken. Sensor random measurement error further complicates the problem. Hence, to fully address these problems, the interferences from both node compromise and random measurement fault should be taken into consideration.

We propose a Secure Event Boundary Detection (SEBD) scheme, which allows secure detection of event boundaries in a localized manner, and is highly resilient against both node compromise and random measurement fault. In SEBD, with an efficient key establishment protocol, each sensor node establishes a unique secret key shared only with the sink, and several pairwise secret keys each shared with one of its neighbors. Those keys are bound to a node's physical location so even if the node is compromised, the impact is effectively confined to that particular node and at its particular location only. In SEBD, each node senses its local environment independently. Once an event of interest is detected, senor nodes first exchange their measurements among neighbors and benign nodes suppress possible faulty measurements following a majority rule. To enhance fault tolerance and prevent fabrication, once a node is detected as a boundary node, a number of its neighbors will collaboratively endorse the corresponding boundary claim message. A neighbor node endorses a boundary claim message only if the contained information is consistent with its own knowledge. The sink accepts a claim only when it contains a required number of valid endorsements. A nonparametric statistical boundary detection model is also developed, which is seamlessly integrated with the proposed security mechanism. It facilitates localized boundary node determination, and helps to suppress random measurement fault and malicious false readings. It shows a much higher accuracy and better fault-tolerance and compromise-resilience as compared to previous schemes [3], [4], [6]. The performance and security strength of the proposed

Manuscript received August 6, 2006; accepted December 16, 2006. The associate editor coordinating the review of this paper and approving it for publication was X. Shen.

Kui Ren is with the Electrical and Computer Engineering Department, Illinois Institute of Technology (e-mail: kren@ece.iit.edu).

K. Zeng and W. Lou are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute (e-mail: {wjlou, kzeng}@ece.wpi.edu).

Digital Object Identifier 10.1109/TWC.2008.060550.

SEBD are examined by analysis and extensive simulation study.

A. Related Work

Several localized event boundary detection schemes have been proposed recently [3], [4], [6]–[9]. Among them, Clouqueur, et. al. [9] sought algorithms to collaboratively detect the presence of a target in a region. Each sensor obtains the target energy (or local decision) from all other sensors in the region, drops extreme values if faulty sensors exist, computes the average, and then compares it with a predetermined threshold for final decision. Krishnamachari et. al. [3] proposed several localized threshold based decision schemes to detect both faulty sensors and event regions. The 0/1 decision predicates from the neighborhood are collected and the number of neighbors with the same predicates are calculated. This number is used for the final decision based on a majority vote. Another work that targets localized boundary detection in sensor networks was proposed by Chintalapudi et. al. in [6]. All of the three schemes in [6] take as inputs the 0/1 decision predicates from neighboring sensors. The statistical approach computes the number of 0's and 1's in the neighborhood and a boundary sensor is detected if its neighbors contain a "similar" number of 0's and 1's. Here the "similarity" is defined based on a threshold value that can be obtained based on a lookup table. Ding et. al. further proposed a similar approach [6] that takes as input not only binary 0/1 decision predicates but also real values that abstract sensor readings or sensor behaviors [4]. Note that all these schemes work in trustworthy environments.

Meanwhile, several schemes have been proposed to provide secure discrete event detection under adversarial environments [10]–[15]. In these schemes, every single event of interest is assumed to be detectable by at least T nodes, where T is a predefined threshold value and usually very small (< 10). The approach adopted in these schemes is to let every valid event detection report be collaboratively generated and independently endorsed by T nodes that have detected the event simultaneously. Cryptographic techniques are then used to generate such endorsements to allow both en-route and sink verification, while keeping the event report as short as possible. However, this approach can not be applied to a large-scale event directly, since it is neither feasible nor necessary for all the nodes in the event region to report its detection back to the sink due to the stringent resource constraints in WSNs. So far, there is no published work on secure protocols that aim to correctly identify and communicate event boundaries, rather than the event itself, in the presence of both node compromise and random node measurement faults.

B. Contributions

This paper makes the following contributions:

- We introduce the problem of securing event boundary detection in WSNs for applications related to large-scale spatial phenomena monitoring, and show how existing boundary detection schemes would fail in adversarial environments.
- We present a Secure Event Boundary Detection (SEBD) scheme, which is to the best of our knowledge the first protocol of its kind to secure event boundary detection in WSNs. SEBD withstands many types of attacks as will be discussed in Section III.
- We propose an enhanced statistic model for localized event boundary detection with proactive faulty measurements correction. Our model is more accurate and robust against node compromise and random fault as compared to existing schemes [3], [4], [6]. Moreover, it is nonparametric without relying on any prior knowledge of node compromise and fault probability, which, however, is required by existing schemes to achieve optimal results [4], [6].
- We use extensive simulations to evaluate SEBD, and show a very good performance and security strength, even when node compromise and fault probability reaches as high as 20%.

The remaining part of this paper is organized as follows. Section II details the proposed SEBD. The security analysis of SEBD is given in Section III. Section IV reports the simulation results of SEBD regarding both performance and security strength. Section V concludes the paper.

II. SEBD: The Scheme

A. Problem Identification

In this part, we describe how the event boundary detection schemes proposed under trustworthy environments would fail in adversarial environments. In adversarial environments, sensor nodes could be compromised and controlled by the attacker [16]. These compromised nodes will lie about their measurements and result in severe security threat, which greatly jeopardizes boundary detection functionality of a WSN. Both faulty nodes and compromised nodes may inappropriately cause non-boundary nodes (including themselves) to be recognized as boundary nodes due to the nature of statistical method used by most of exiting schemes. However, the damage caused by the compromised nodes is much worse than that of faulty nodes. This is because a *faulty* node is still a benign node, and would suppress its own measurements after referring to other measurements in its neighborhood. However, a *compromised* node will always lie about its measurements, report itself as a boundary node when it is not, and suppress such claims when it is[1]. A collection of compromised nodes could prevent the event boundary from being correctly detected by presenting false measurement information. Moreover, compromised nodes may collude to fabricate non-existing events and event boundaries. They may claim such boundaries appearing at any location of the network as desired by the attacker, not necessarily at their own actual locations.

B. Assumptions, network model and design goal

We assume that sensor nodes are uniformly deployed in a two-dimensional territory, i.e., a *sensor field*, and they are dense enough to support fine-grained collaborative sensing. Topology control mechanisms for such purpose have been

[1]When it does not lie, it does not need to be treated.

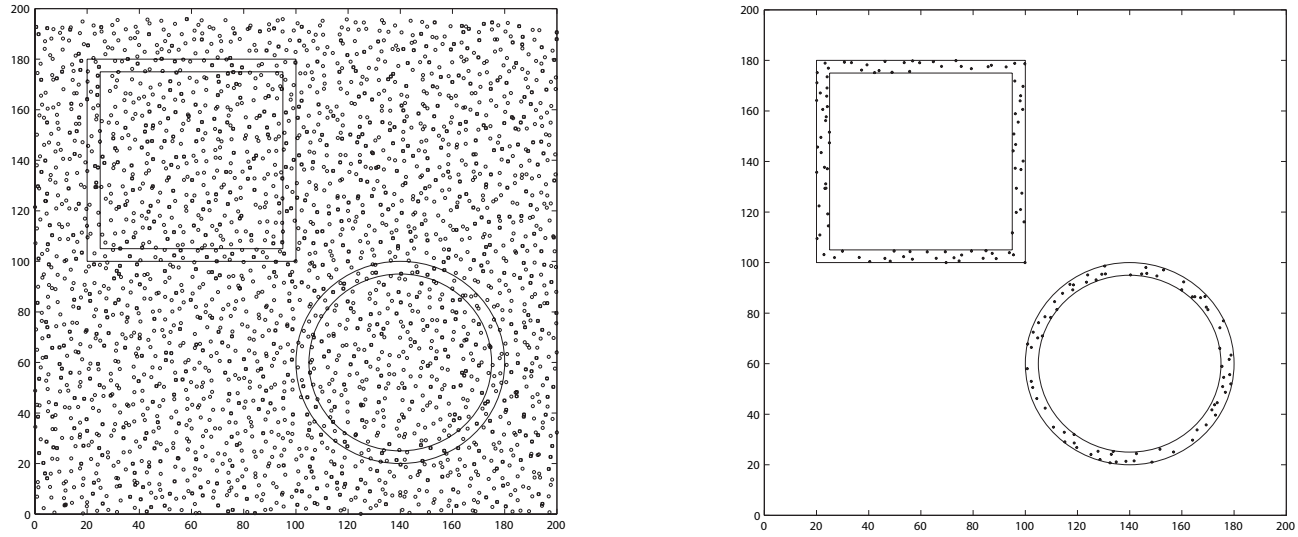


Fig. 1. The left figure denotes a sensor field, where the event area is located inside the outer square and circle. In the figure, normal nodes are denoted as 'o', compromised and faulty nodes are denoted as '□'. The right figure is an illustration of boundary $\mathbb{B}$ with $r = \frac{R}{2}$. By definition, the nodes denoted as '*' are the boundary nodes.

studied in [17]. We assume that sensor nodes are similar to the current generation of sensor nodes (e.g., the Berkeley MICA motes [18]) in their computation and communication capability and power resource and they are loosely synchronized. We assume that even if sensor nodes may execute certain sleeping strategy for energy conservation, they can still wake up periodically or be woken up by ceratin events to work collaboratively, according to certain wakeup mechanism [19]. The term *sensor field*, denoted as $\mathbb{S}$, is referred to both the geographical region covered by the WSN, and the set of sensors within the region. A sensor S_i and its location will be used interchangeably, that is, $S_i = (x_i, y_i)$. We assume that sensor nodes can make random measurement errors, and such nodes are called *faulty* nodes. (However, we do not address location information measurement errors, since our focus in this paper is how to securely detect event boundary given correct location information). Furthermore, we assume that sensor nodes can be compromised and controlled by the adversary, whose purposes are to 1) prevent event boundary from being correctly detected; 2) collude to fabricate non-existing events and event boundaries.

We use a refined event boundary model as compared to the ones in [4], [6]. Consider a phenomenon (i.e., event $\mathcal{E}$) that spans some arbitrarily shaped sub-region of $\mathbb{S}$. Each sensor can, based on locally collected measurements, determine whether it belongs to the sub-region covered by the phenomenon or not. Ideally, a boundary node, say S_i, is such a node that every closed disc centered at S_i contains both points in $\mathcal{E}$ and $\overline{\mathcal{E}}$ (that is, the boundary node should be right on the *real event boundary*, denoted as $\mathcal{B}_R$.), where $\mathcal{E}$ is the ground truth of the event covering sub-region in $\mathbb{S}$, and $\overline{\mathcal{E}}$ represents the remaining region, i.e., $\overline{\mathcal{E}} = \mathbb{S} - \mathcal{E}$. Hence, an event boundary, denoted as $\mathbb{B}$, when represented by sensor nodes, is simply a collection of such boundary nodes. However, due to the actual node density in practice, an event boundary found in this case constitutes only a very restrictive node set, which is far from enough to approximate/reveal $\mathcal{B}_R$ [6]. For this reason, the notion of *boundary width* is introduced with its value $0 < r < R$ in SEBD, where R is the communication radius of sensor nodes. In SEBD, we define a sensor node, S_i, as a boundary node, $\mathbb{B} = \bigcup_i S_i,\ \forall\ i : |S_i \perp \mathcal{B}_R| \leq r, and\ S_i \in \mathcal{E}$, where $|S_i \perp \mathcal{B}_R|$ denotes the distance between S_i and $\mathcal{B}_R$. The definition is illustrated in Fig.1.

Naturally, the design goal of SEBD is then to securely identify as many nodes as possible in $\mathbb{B}$ (bounded by the underlying distributed statistical model) under adversarial environments. In other words, SEBD should have a strong security strength to prevent compromised nodes from successfully claiming themselves as boundary nodes when they are actually not. Furthermore, even if a compromised non-boundary node succeeds in claiming itself as a boundary node, it should not be able to claim the boundary at locations other than where it is. That is, the damage caused by compromised nodes should be limited to their vicinity only.

C. Overview of SEBD

The proposed SEBD is designed to be robust against node compromise and random fault. SEBD consists of two key components: the underlying location-aware key management framework, and the corresponding distributed statistic boundary detection model that is seamlessly built upon the former.

Key management framework in SEBD exploits the static and location-aware nature of WSNs. By leveraging robot-assisted secure bootstrapping technique, a secure location-aware key management is efficiently realized through embedding location information into the keys. In SEBD, each node possesses two different types of location-aware symmetric keys: 1) a *unique secret key* shared between the node and the sink that is used to provide node-to-sink authentication and data confidentiality; 2) a set of *neighbor pairwise key*s shared with each of the neighbor nodes respectively for node-to-node authentication and data confidentiality.

In our design, a sensor, after having detected an event of interest, proceeds to find out whether or not it is a boundary

sensor. To do so, it first shares its sensing result within the neighborhood, and then makes use of the collective sensing result information for 1) suppressing its own potential sensing error; 2) judging whether or not a neighbor sensor/itself is a boundary node. If a sensor recognizes itself as boundary node, it further proceeds to request endorsements from its neighbors. Every neighbor sensor chooses to or not to endorse such requests independently. The judgement is based on 1) the collective sensing results in the neighborhood; 2) the behavior of the sensor that seeks endorsements. In this way, the boundary sensors are detected statistically, and, at the same time, the illegal attempts of claiming a non-boundary node as a boundary node are effectively suppressed. More specifically, SEBD detects an event boundary in three essential stages: In 1) *local sensing and measurement adjusting* stage, each node exchanges its event measurement in the neighborhood. Then, every node adjusts its own measurement result according to *the majority rule*. Next, in 2) *distributed boundary detection* stage, each node independently determines whether or not it is a boundary node according to the updated measurements distribution in its neighborhood and the predefined statistic model. Once a node judges itself as a boundary node, it makes a boundary claim and seeks endorsements from its neighbors. Then, a neighbor node that receives a boundary claim will carry out a *consistency check* based on knowledge it learned directly from its neighbors and will follow the same statistical model to judge whether or not the sender is a boundary node or not. Upon getting a positive result, the receiving node endorses the boundary claim using the *unique secret key* it has. Lastly, in 3) *final message composition* stage, a boundary node constructs an overall synthesized endorsement from the individual ones it collected from its neighbors. The sink only accepts a boundary claim with a valid overall synthesized endorsement.

D. The Enhanced Statistical Boundary Detection Model

The observation behind our statistical boundary detection model is two-fold: 1) The original event measurements collected from neighbor nodes usually contain faulty measurements due to node measurement error and compromise; if faulty measurements can be corrected, boundary detection can certainly be more accurate and thus more tolerant against node fault and compromise. 2) Statistically, a boundary node can be determined by comparing the event measurements among its neighbor nodes by assuming that the neighbor area of a sensor node is so "small" in comparison to the area covered by the entire event that the ground truth boundary can be approximated by a straight line in this area. In Particular, a boundary node (i.e., belonging to $\mathbb{B}$) will always have the difference between the numbers of '0' and '1' measurements in its neighborhood limited by a certain threshold and its value is determined by *boundary width* r given a uniform node distribution.

In SEBD, the following specific rules are designed to reflect the above observation:

Majority rule: A node maintains its own measurement only when this result is the majority result within its neighborhood. Statistically, this rule could lead to error correction, as long as sensor fault probability is less than 50%. Note that the *majority rule* has been proved to be optimal based on an Bayesian fault recognition model in detecting node random measurement errors [8]. We refer the interested reader to [8] for the detailed proof.

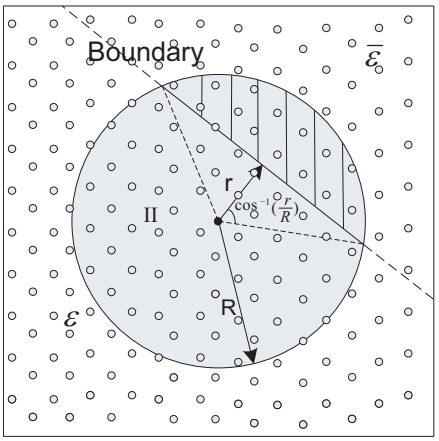


Fig. 2. An illustration of areas I and II.

Consistency rule: Since compromised nodes may lie about its measurements, we further require that, if a node does not follow the *majority rule*, its result will be ignored by its neighbors. This *consistency rule* is enforced in SEBD by consistency verification.

Determination rule: A node recognizes itself as a boundary node only when

$$1 - \frac{n_+ - n_-}{n_u} \geq \gamma, \quad (1)$$

where n_+ is the number of '1' measurements in a node u's neighborhood, n_- is the number of '0's, and $n_u = n_+ + n_-$ is the actual neighborhood size. Furthermore, γ is a preset system parameter, called *normalized acceptance threshold*. In contrast to the previous schemes, the optimal choice of γ in SEBD does not rely on the sensor fault probability. In fact, we set $\gamma = 1 - \frac{II(r) - I(r)}{\pi R^2}$, where the areas of II and I are illustrated in Fig.2. This selection of γ is based on uniform node distribution, since the area size is proportional to the number of nodes located inside the area.

E. Scheme Details

1) Network initialization phase: SEBD adopts a robot-assisted bootstrapping technique, which securely initializes each sensor node with the required scheme parameters and the secret symmetric keys. Specifically, we assume that a group of mobile robots are dispatched to sweep across the whole sensor field along pre-planned routes. Mobile robots have GPS capabilities as well as more powerful computation and communication capacities than ordinary nodes. The leading robot is also equipped with the network master secret keys K_M^I and K_M^{II}, and γ. To localize a node, say S_{i_u}, mobile robots run the secure range-based localization protocol given in [20], [21] to first measure their respective absolute distance to node S_{i_u} and then co-determine its location (x_{i_u}, y_{i_u}). Subsequently, the leading robot computes the *unique secret key* that is only shared between the sink and S_{i_u} after bootstrapping:

$$K_{S_{i_u}} = H(K_M^I | S_{i_u}), \quad (2)$$

where '|' denotes concatenation operation. It also generates a complete list of neighbor nodes, denoted as $\mathbb{N}_{S_{i_u}}$ for a node S_{i_u}[2], and computes a set of *neighbor pairwise key*s: suppose $S_{i_v} = (x_{i_v}, y_{i_v})$ is a neighbor of S_{i_u}, then the *neighbor pairwise key* between the two is

$$K_{S_{i_u},S_{i_v}} = H(K_M^{II}|S_{i_u}|S_{i_v}), \quad (3)$$

where $S_{i_u} < S_{i_v}$ in their binary representations. The leading robot repeats the calculation for all nodes in $\mathbb{N}_{S_{i_u}}$ and sends all the generated keys plus γ to S_{i_u}. Note that the authentication between the sensor nodes and the leading robot can be easily achieved using the technique introduced in [22]. We omit it here for the space limit. Following this process, all the nodes can be furnished with their respective location and the required keys. After that, mobile robots leave the sensor field and the leading robot should securely erase all the keys from its memory. The assumption underlying this approach is that adversaries do not launch active and explicit pinpoint attacks on mobile robots at this stage which usually does not last too long. That is, the robots are not likely subject to compromise. However, the adversaries may still perform relatively passive attacks such as message eavesdropping or strategic channel inference to disturb the localization process [20]. This assumption is reasonable in that mobile robots are much fewer than ordinary sensor nodes and hence we can spend more on them by enclosing them in high-quality tamper-proof hardware and putting them under super monitoring.

2) Boundary detection phase: For each sensor node S_{i_u}, the following data structures are defined: $list_{S_{i_u}}$ is a table used to store the measurements from nodes in $\mathbb{N}_{S_{i_u}}$. n_u is the actual number of the one-hop neighbors of node u, and n' is the expected node degree. $tt_{S_{i_u}}$ is used to store current operation time. n_+ is the number of '1' measurements in $\mathbb{N}_{S_{i_u}}$, while n_- is the number of '0's. $\mathbb{EN}$ is a list used to store the *id*s of nodes having endorsed on S_{i_u}'s *boundary claim* message if any. Lastly, Mac is for final overall endorsement constructed. Note that all of them are initialized to zero or $\emptyset$.

Upon receiving a boundary extraction request on event type e_{id}, from the sink or a pre-defined periodical time-out, a node S_{i_u} performs the measurement, notifies its neighbors of the result, and adjusts the result according to others' notifications if necessary:

- S_{i_u} prepares $\mathsf{MR}_{S_{i_u}} := \{e_{id}, S_{i_u}, m_0, 0\}$, and broadcasts $\mathsf{MR}_{S_{i_u}}$ to its neighborhood.
- S_{i_u} collects $\mathsf{MR}_{S_{i_j}}$ for all $S_{i_j} \in \mathbb{N}_{S_{i_u}}$ and updates $list_{S_{i_u}}$, i.e., upon receiving message $\mathsf{MR}_{S_{i_j}} = \{e_{id}, S_{i_j}, m_0, 0\}$, S_{i_u} updates $list_{S_{i_u}}$ by including an entry (S_{i_j}, m_0).
- Once having received $\mathsf{MR}_{S_{i_j}}$ from all $S_{i_j} \in \mathbb{N}_{S_{i_u}}$, S_{i_u} calculates the number of '1' measurements n_+ and the number of '0' measurements n_- from $list_{S_{i_u}}$; S_{i_u} adjusts its own measurement m as follows: if $m_0 == 1$ and $n_+ < \lfloor \frac{n_u - 1}{2} \rfloor$, S_{i_u} reverses its measurement to $m_1 := 0$; if $m_0 == 0$ and $n_- < \lfloor \frac{n_u - 1}{2} \rfloor$, S_{i_u} reverses its measurement to $m_1 := 1$, otherwise, the original measurement is retained, $m_1 := m_0$.
- S_{i_u} then broadcasts the updated $\mathsf{MR}_{S_{i_u}} := \{e_{id}, S_{i_u}, m_1, 1\}$ together with $list_{S_{i_u}}$ in its neighborhood.

[2]$\#\{\mathbb{N}_{S_{i_u}}\} = n'$ on average, where n' is the expected number of sensor nodes in πR^2 area.

Here, a *measurement report* message $\mathsf{MR}_{S_{i_u}}$ consists of four fields: i) an event *id*, e_{id}, ii) a node *id*, iii) m, a logic value '0/1', representing whether event e_{id} is detected or not, and iv) a '0/1' valued indicator, indicating the message is either an original measurement report or an updated report after local adjustment for random error correction. Next, if a node S_{i_u}'s updated measurement is '1', it proceeds to check whether or not it is a boundary node based on the information it received. Note that, if the measurement of S_{i_u} is now a '0', then no further operation is needed. The following operations are sequentially executed before reaching the decision:

- For every node S_{i_j} in $\mathbb{N}_{S_{i_u}}$, S_{i_u} does the following consistency check and updates $list_{S_{i_u}}$ accordingly: 1) it verifies that the measurements in the common entries in $list_{S_{i_j}}$ and $list_{S_{i_u}}$ are consistent; and 2) it verifies that node S_{i_j}'s self-adjusted value, i.e., m_1 in $\mathsf{MR}_{S_{i_j}} = \{e_{id}, S_{i_u}, m_1, 1\}$, conforms to the majority measurements in $list_{S_{i_j}}$.
- S_{i_u} then calculates n_+ and n_- for the updated $list_{S_{i_u}}$ and further calculates $1 - \frac{|n_+ - n_-|}{n_u}$. If $1 - \frac{|n_+ - n_-|}{n_u} \geq \gamma$, S_{i_u} considers itself a boundary node and prepares a *boundary claim* message, $\mathsf{BC}_{S_{i_u}} := \{e_{id}, S_{i_u}, 1, tt_{S_{i_u}}\}$, where $tt_{S_{i_u}}$ is a time stamp. S_{i_u} then broadcasts $\{list_{S_{i_u}}, \mathsf{BC}_{S_{i_u}}\}$ to the neighbors to seek their endorsements. $list_{S_{i_u}}$ is attached for consistency verification.

Now assume that a neighbor node, say S_{i_j}, receives S_{i_u}'s boundary claim $\{list_{S_{i_u}}, \mathsf{BC}_{S_{i_u}}\}$. S_{i_j} proceeds as follows to endorse the BC [3].

- Consistency check: 1) it verifies if the time stamp is fresh, i.e., within the allowed delay interval; 2) it checks the measurement consistency of the common entries contained in both $list_{S_{i_u}}$ and its own $list_{S_{i_j}}$; 3) it carries out the same procedure to determine if node S_{i_u} is a boundary node and verifies that the result conforms to S_{i_u}'s claim.
- Upon successful checking, S_{i_j} endorses $\mathsf{BC}_{S_{i_u}} := \{e_{id}, S_{i_u}, 1, tt_{S_{i_u}}\}$ by calculating $\overline{MAC}_{S_{i_j}} := MAC(\mathsf{BC}_{S_{i_u}}, K_{S_{i_j}})$, generating $\mathsf{ER}_{S_{i_j}} := \{S_{i_j}, S_{i_u}, e_{id}, tt_{S_{i_j}}, E(\overline{MAC}_{S_{i_j}}, K_{S_{i_j},S_{i_u}})\}$. It further calculates $\overline{MAC}_{S_{i_j},S_{i_u}} := MAC(\mathsf{ER}_{S_{i_j}}, K_{S_{i_j},S_{i_u}})$, and sends $\{\mathsf{ER}_{S_{i_j}}, \overline{MAC}_{S_{i_j},S_{i_u}}\}$ back to S_{i_u}.

Here, the endorsement to $\mathsf{BC}_{S_{i_u}}$ from S_{i_j}, i.e., $\overline{MAC}_{S_{i_j}}$, is a unique MAC generated over message $\{e_{id}|S_{i_u}|1|tt_{S_{i_u}}\}$ using the *unique secret key* $K_{S_{i_j}}$ shared between S_{i_j} and the sink. Hence, no node could forge such a MAC on behalf of others. SEBD also ensures that only the claimed sender can get the endorsement from the receiver/endorser: $\overline{MAC}_{S_{i_j}}$ is sent after encryption using the *neighbor pairwise key* shared between the sender and receiver. Meanwhile, $\overline{MAC}_{S_{i_j},S_{i_u}}$ is computed over $\mathsf{ER}_{S_{i_j}}$ using the same *neighbor pairwise key* shared between the sender and receiver, which authenticates the message sender to the receiver. The intended receiver could therefore be assured that the endorsement is indeed from the claimed endorser. Note that $\overline{MAC}_{S_{i_u},S_{i_j}} \neq \overline{MAC}_{S_{i_j},S_{i_u}}$.

Lastly, node S_{i_u} collects all the ERs replied by its neighbors after sending $\mathsf{BC}_{S_{i_u}}$. It then constructs a final synthesized

[3]Below $MAC(M, K)$ denotes the message authentication code generated over message M using symmetric key K and $E(M, K)$ denotes an encryption operation over message M using K.

boundary report with appropriate endorsements from its neighbors, and sends it to the sink.

– Upon receiving $\{\mathsf{ER}_{S_{i_j}}, \overline{MAC}_{S_{i_j},S_{i_u}}\}$ from its neighbor S_{i_j}, S_{i_u} first checks the time stamp included in ER to make sure the freshness of the message. It further verifies $\overline{MAC}_{S_{i_j},S_{i_u}}$.
– Upon successful verification, S_{i_u} then includes S_{i_j} into $\mathbb{EN}$, i.e., $\mathbb{EN} := \mathbb{EN} \cup S_{i_j}$, and recovers the unique MAC generated by S_{i_j}, which is further combined to the synthesized MAC, i.e., $Mac := Mac \oplus D(E(\overline{MAC}_{S_{i_j}}, K_{S_{i_j},S_{i_u}}), K_{S_{i_u},S_{i_j}})$, where $D(M,K)$ denotes a decryption operation over message M using symmetric key K and '$\oplus$' denotes exclusive or operation.
– Upon $\#\{\mathbb{EN}\} \geq \lfloor \frac{n'-1}{2} \rfloor$, S_{i_u} forms a *boundary report* message $\mathsf{BR}_{S_{i_u}} := \{\mathsf{BC}_{S_{i_u}}, Mac, \mathbb{EN}\}$, and forwards $\{\mathsf{BR}_{S_{i_u}}, \overline{MAC}_{S_{i_u},sink}\}$ to the sink, where $\overline{MAC}_{S_{i_u},sink} := MAC(\mathsf{BR}_{S_{i_u}}, K_{S_{i_u}})$.

A $\mathsf{BR}_{S_{i_u}}$ is accepted by the sink if and only if i) $\overline{MAC}_{S_{i_u},sink}$ is authentic; and ii) $t \geq \lfloor \frac{n'-1}{2} \rfloor$, where t is the number of members in $\mathbb{EN}$; and iii) all nodes in $\mathbb{EN}$ are indeed the neighbors of S_{i_u}[4]; and iv) Mac is authentic, which, in other words, means that all the t individual $\overline{MAC}_{S_{i_j}}$s are authentic; Note that, in this paper, we focus on the compromise-tolerant event boundary detection mechanism, thus we simply assume all BRs are directly forwarded to the sink [5].

III. Security Analysis

A. Qualitative Analysis

The proposed SEBD presents several nice security features as below, which greatly mitigate the security threat caused by compromised nodes. Firstly, in SEBD, to successfully claim itself as a boundary node, a node has to collect enough endorsements from its neighbors so that its claim can be accepted by the sink. Meanwhile, each sensor node independently makes endorsement decisions by itself based on the information it collected directly from its neighborhood. Therefore, in contrast to existing boundary detection schemes, before compromised nodes are able to make such claims, a required number of endorsements have to be collected. Secondly, even if a compromised node has collected enough endorsements, it can only claim the boundary at its own location. That is, the damage caused by compromised nodes is limited to their vicinity only. This is because the location information has been embedded into the *unique secret key* shared between each node and the sink, and any claim other than a node's actual location will be rejected by the sink due to lack of the corresponding *unique secret key*.

SEBD also withstands the following attacks:

Cheating attack: Under this attack, a compromised node lies about some of its neighbors' measurements in a *boundary claim* message in order to deceive its neighbors and obtain the endorsements. However, this attack will not likely succeed in SEBD. Although a broadcast BC in plaintext cannot be verified through cryptographic means like MAC, it is indeed verified through consistency check, which relies on the local and direct knowledge each node learned from its neighborhood. This is so designed because, in order to provide a cryptographically authentic BC, the sender has to attach up to n_u different MACs from its neighbors. This, however, will unnecessarily waste large amounts of the precious energy and bandwidth, and greatly decrease the protocol efficiency.

Consistency check, on the other hand, is more efficient, and effectively prevents cheating attack because every node in SEBD is required to maintain a table storing the measurement information of its neighbors, which serves as the information source for boundary nodes self-determination. This table, at the same time, is also conveniently used for consistency check: each node u independently verifies the authenticity of a BC using its own local knowledge stored in $list_{S_{i_u}}$. Although each node may not be able to verify the whole information contained in a received BC, a lie about one node's measurement will always be detected by some corresponding neighbor nodes, since the messages are always broadcast. It is very hard for a compromised node to lie about a number of nodes' measurements simultaneously to be falsely recognized as a boundary node (or reverse) and still get enough endorsements without being detected. If a complete consistency check is necessary, we may allow each node to increase its transmission range to $2R$ only when broadcasting its event measurements. Then, a $list_{S_{i_u}}$ can contain the whole measurement information of all two-hop neighbors. And, in this case, any individual node can detect a lie in $list_{S_{i_j}}$ from its immediate neighbors. Thus, the cheating attack can be completely prevented.

Impersonating and colluding attack: Under this attack, a compromised node may try to impersonate another node at a different location. And compromised nodes at different locations may collude and endorse each others' *boundary claim* message. However, this attack is not possible in SEBD because the sink only accepts those endorsements obtained from neighbor nodes. Since the location information is embedded into the *unique secret key* shared between a node and the sink, and the communication of any two neighbor nodes is protected by the corresponding *neighborhood pairwise key*, there's no way for a node to impersonate another node or generate a valid endorsement for a colluding node which is not in its neighborhood.

Replay attack: Under this attack, a compromised node may replay old messages in response to a new boundary query. This attack is prevented in SEBD through embedding time information in the BR messages. Any boundary report message that is out of the pre-specified delay tolerance will be automatically rejected.

Node Relocation and Replication Attack: Under this type of attacks, the adversary may 1) compromise and relocate some sensor nodes to other positions in the sensor field; 2) replicate compromised sensors and place them to the positions of the adversary's interest. The goal of the adversary is to have the compromised nodes outnumber the normal nodes at certain areas, and hence try to cheat the sink with bogus boundary claims. However, this type of attacks are also not possible in SEBD because the location information is always embedded into all the keys used to endorse the boundary claims. Similar to the analysis for impersonating and colluding attacks, we can easily find that the relocated sensors can never obtain the legitimate *unique secret key* and the *neighborhood pairwise key*s.

[4]This is achieved by extracting node's location information from its id, and ensuring that the distance between two nodes is no farther than R.

[5]The sink is always assumed trustworthy and well protected.

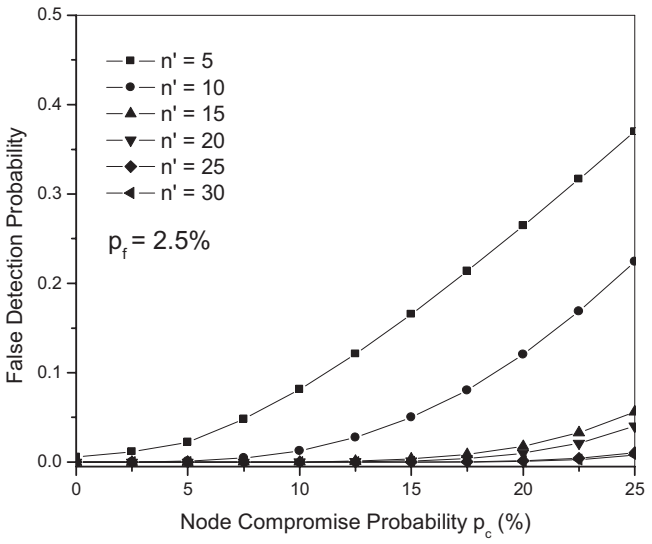


Fig. 3. False detection probability vs. node compromise probability.

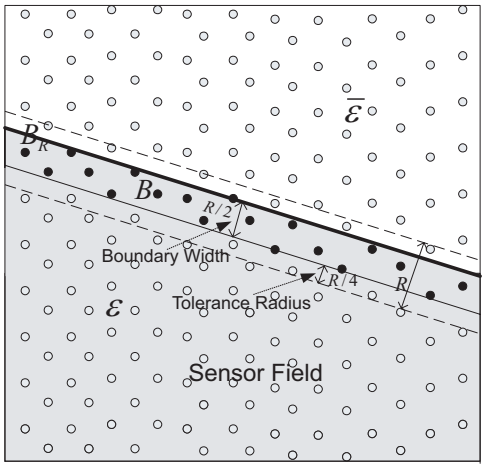


Fig. 4. An illustration of boundary model with $r = R/4$.

B. Quantitative Analysis

In SEBD, compromised and faulty nodes could still possibly result in false recognition of non-boundary nodes (including compromised and faulty nodes themselves) as boundary nodes. This is because the sink accepts a boundary claim as long as such a message is endorsed by $\lfloor \frac{n'-1}{2} \rfloor$ nodes, and n' is the expected neighborhood size. That is, a compromised/faulty node could pass its boundary claim, as long as there are no fewer than $\lfloor \frac{n'-1}{2} \rfloor$ faulty/compromised nodes in its neighborhood. However, the attack is actually very hard to succeed as the following analysis shows.

Assume a WSN consisting of N nodes, node compromise and fault probability p_c and p_f respectively, the expected number of compromised nodes is then $N(p_c+p_f)$. Therefore, the probability that exactly i nodes are compromised/faulty in a neighborhood is $P_{\{i\}} = \frac{\binom{n'}{i}\binom{N-n'}{N(p_c+p_f)-i}}{\binom{N}{N(p_c+p_f)}}$, assuming compromised and faulty nodes are uniformly distributed in the sensor field. Hence, the expected probability that a non-boundary node is falsely detected as a boundary node, is

$$P_{\{\geq \lfloor \frac{n'-1}{2} \rfloor\}} = \sum_{i=\lfloor \frac{n'-1}{2} \rfloor}^{n'} P_{\{i\}} \quad (4)$$

Obviously, $P_{\{\geq \lfloor \frac{n'-1}{2} \rfloor\}}$ also represents the fraction of nodes that are falsely detected as boundary nodes when there are no events going on in a WSN.

Fig. 3 shows how the value of $P_{\{\geq \lfloor \frac{n'-1}{2} \rfloor\}}$ changes as node compromise probability changes under different neighborhood size n'. It is clearly shown that as long as n' is reasonably large ($\geq$ 15), the value of $P_{\{\geq \lfloor \frac{n'-1}{2} \rfloor\}}$ keeps below 0.8%, given $p_c \leq 15\%$. Even when p_c reaches as high as 20%, we still have $P_{\{\geq \lfloor \frac{n'-1}{2} \rfloor\}} < 3\%$ with $n' = 30$. Furthermore, the number of nodes in a single area can be modelled as poisson random variable as we assume a uniform node distribution in WSN [6]. This implies that, given the expected neighborhood size (i.e., node degree) $n' \leq 50$, the probability that the number of nodes in a neighborhood is less than $n' - 4$ is very small. That is, given an expected n', the size of a neighborhood in WSN has an overwhelming probability to be larger than $n'-4$. Therefore, in SEBD, it is very hard for compromised nodes to fabricate non-existing events and event boundaries.

IV. Simulation Studies

A. Metrics for Performance and Security Strength

The following three metrics are used to evaluate the performance of SEBD. Let $\mathbb{B}'$ be the set of boundary nodes detected by SEBD. Let $\mathbb{B}$ be the set of actual boundary nodes as defined in Section III.A.

Hit Rate e_f: e_f represents the fraction of sensors in $\mathbb{B}$ that are detected by SEBD, with respect to the size of $\mathbb{B}$:

$$e_f = \frac{\#\{\mathbb{B} \cap \mathbb{B}'\}}{\#\{\mathbb{B}\}} \quad (5)$$

On top of *boundary width* r, we futher introduce the notion of *tolerance radius* to characterize the distribution of the boundary nodes detected by SEBD. In particular, any falsely detected boundary node that has its distance to real boundary $\mathbb{B}$ no more than $\frac{R-r}{2}$ is said to be within *tolerance radius*. We are more interested in the fraction of falsely detected boundary nodes that are far from the event boundary $\mathbb{B}$. An illustration of this definition is shown in Fig. 4. Based on the definition of *tolerance radius*, *False Detection Rate* is defined.

False Detection Rate e_d: e_d represents the ratio of falsely detected sensors with respect to the size of $\mathbb{B}$. Here, only those falsely detected sensors whose distance to the boundary are at least $\frac{R-r}{2}$ are counted. Let $\mathbb{A}$ denote the set of falsely detected nodes whose distance to the boundary is larger than $\frac{R-r}{2}$.

$$e_d = \frac{\#\{\mathbb{A}\}}{\#\{\mathbb{B}\}} \quad (6)$$

Furthermore, we denote the mean distance of the nodes in $\mathbb{B}'$ to $\mathbb{B}_R$ as $d_{\mathbb{B}'}$.

Normalized Mean Distance e_w: e_w represents the normalized mean distance of $\mathbb{B}'$ regarding boundary width:

$$e_w = \frac{d_{\mathbb{B}'}}{r} \quad (7)$$

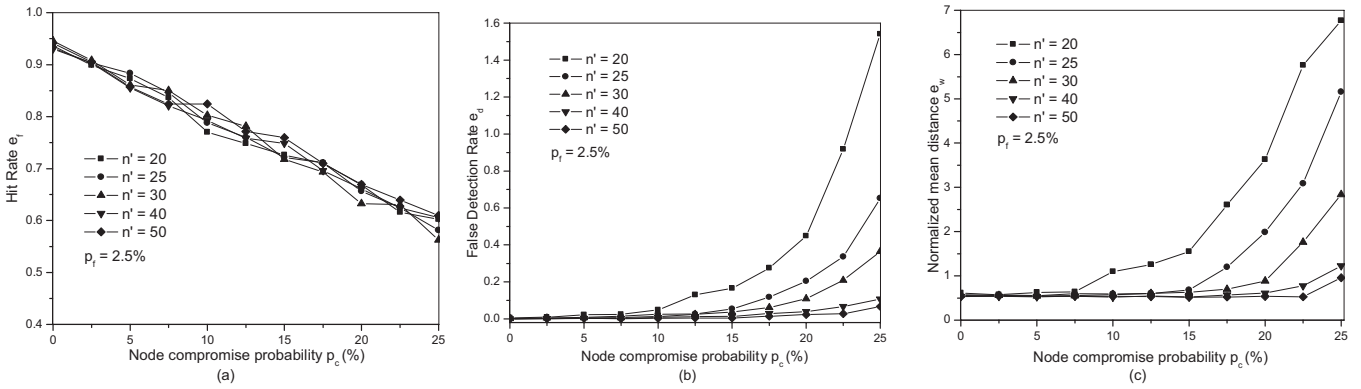


Fig. 5. Simulation results with respect to three evaluation metrics

B. Simulation setup

In all simulations, sensor nodes are located in a 200m by 200m area, their locations drawn from a uniform distribution over the area. The radio range of all the sensors is 10m and assumed omni-directional. In all simulations, we arbitrarily choose the *boundary width* $r = R/2$. And $\gamma = 1 - \frac{II(r=\frac{R}{2}) - I(r=\frac{R}{2})}{\pi R^2} = \frac{2\pi - 3\sqrt{3}}{3\pi} \approx 0.12$; the areas of II and I are illustrated in Fig. 2. Note that our γ value is independent of node compromise probability. We report the results for event regions with ellipses or straight lines as the boundaries. And our simulation produces similar results for event regions with other boundary shapes. The simulation runs under different densities and node compromise probabilities. For each given density and node compromise probability, the results for the three security and performance metrics are averaged over 50 simulation runs.

C. Simulation results

In this subsection, the simulation results are reported in details. The three performance and security strength evaluation metrics defined in section V.A with regard to network density and node compromise probability are reported in Fig. 5 (a), (b) and (c), respectively. In contrast to the previous schemes, we did not change any setting on parameters as node compromise probability increases from 0% to 25%. That is, our simulation results do not rely on the pre-knowledge of node compromise/fault probability, which, in fact, may not be available as a priori in many practical applications.

Firstly, we observe that the proposed SEBD performs very well, when node compromise probability equals to zero. In this case, the hit rate e_f is always as high as 93%, no matter what the network density n' is. Note that we still have $p_f = 2.5\%$ in this case. In fact, when both p_c and p_f equal to zero, e_f is always around 95% in SEBD. In the previous schemes [4], [6], e_f is generally no more than 85%, even if all the compromised nodes can be assumed as random faulty ones. Hence, SEBD has the highest hit rate in the ideal situation as compared to the previous schemes.

Secondly, Fig. 5 (a) shows that 1) the hit rate e_f in SEBD does not rely on network density; this is because we intentionally used normalized threshold value in boundary node detection process. 2) SEBD is very good at detecting boundary nodes: e_f remains to be larger than 55%, when p_c reaches as high as 25% plus 2.5% node fault probability. This result significantly outperforms any of the previous schemes [3], [4], [6].

Thirdly, SEBD presents a high security strength as shown in Fig. 5 (b). When n' is as low as 20, the false detection rate e_d is still less than 5%, given $p_c = 10\%$. And for the same p_c, e_d can be kept as low as 30% when $n' = 50$. Furthermore, given $n' = 50$, e_d increases very slowly as p_c increases; e_d equals to only 67%, when p_c reaches to 25% plus 2.5% node fault probability.

Fourthly, Fig. 5 (c) shows that the detected boundary nodes by SEBD are very close to the defined boundary $\mathbb{B}$. It is shown that as long as $n' \geq 25$, the normalized mean distances of the detected boundary nodes are always kept to be around the ideal value 0.5, given $p_c \leq 15\%$.

In summary, the simulation results shown in Fig. 5 indicate that 1) SEBD performs well until p_c is up to 10%, even when n' is as low as 20; 2) SEBD keeps presenting a very good performance and security strength even when p_c goes up as high as 20%, given a reasonable high n'; 3) SEBD significantly outperforms the previous schemes in all the three metrics [4], [6].

Fig. 6 gives several visualized results to illustrate the performance of SEBD. The left figure gives the performance of SEBD at low node compromise probability. Clearly, when $p_c = 5\%$ and $p_f = 2.5\%$, SEBD has a very high hit rate: $e_f = 85\%$. The middle figure gives the performance of SEBD at medium node compromise probability: $e_f = 79\%$, when $p_c = 12.5\%$ and $p_f = 2.5\%$. Obviously, the detected event boundaries in both left and right figure are very good approximations of the real event boundaries as defined in Fig. 1. In the right figure, we can find that as node compromise probability continues to be higher, the detected boundary presents a larger false detection rate as compared to the previous ones. But still, we have $e_f = 60\%$, given p_c as high as 25% and $p_f = 2.5\%$.

V. Efficiency Evaluation of SEBD

Communication Overhead: One can easily see that the performance of SEBD improves as we require each node to collect more event measurements from more sensor nodes in its neighborhood. This is because each node can get more samples from both the interior and exterior of the event,

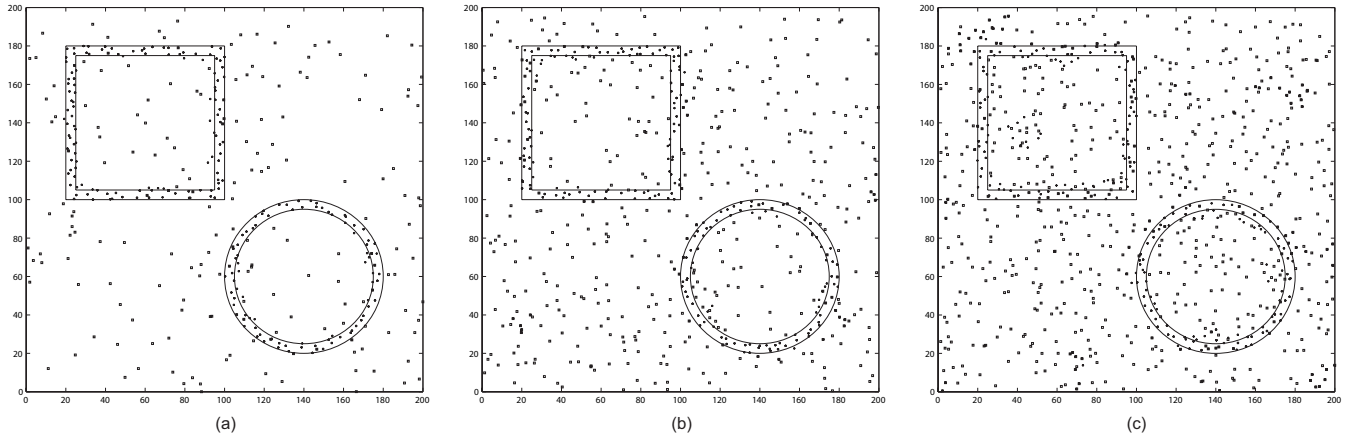


Fig. 6. Simulation results: left): $e_f = 85\%$ with $p_c = 5\%$; middle) $e_f = 79\%$ with $p_c = 12.5\%$; right) $e_f = 60\%$ with $p_c = 25\%$. And $p_f = 2.5\%$ in all three cases. '*' denotes the detected boundary nodes, and '□' denotes compromised and faulty nodes.

and makes more accurate estimate in the presence of node compromise and fault. However, collecting more measurements from more nodes other than the immediate neighbors incurs much higher communication overhead. As mentioned in [6], communication overhead increases roughly quadratically as the neighbor range increases. This will result in a much higher energy consumption. In SEBD, we assume that the underlying network is well connected, that is, neighborhood size is reasonably large to support fine grained collaborative event detection. Hence, a good energy-accuracy tradeoff is achieved by letting each node collect the measurements from their immediate neighbors only. As we have shown in Fig. 5(a), e_f is larger than 80% with n' as low as 20, given p_c up to 10%.

Computation Overhead: SEBD uses very simple arithmetic computations to obtain the measurement statistics. At the same time, SEBD also involves some security related computations: endorsement operation, message authentication operation, and overall endorsement synthesization operation. SEBD exploits highly efficient security primitives to construct these operations: the first two are both realized through highly efficient MAC algorithm, while the last requires "exclusive or" operation only. More specifically, to accomplish a boundary node detection and authentication process, there are up to $2n'$ MAC operations required in total. Hence, the computation costs incurred by the security related operations in SEBD is light-weight.

VI. Concluding Remarks

In this paper, we have studied a special instance of fault-tolerant collaborative in-network processing tasks in WSNs, i.e., distributed event boundary detection. We first introduced the problem of securing event boundary detection in WSNs for the applications related to large-scale phenomena monitoring, and showed how existing boundary detection schemes fail in adversarial environments. Then, we presented the SEBD scheme, which withstands many different types of attacks. To the best knowledge of the authors, SEBD is the first protocol of its kind to secure event boundary detection in WSNs. Along with SEBD, we also proposed an enhanced nonparametric statistic model for localized event boundary detection, which allows faulty measurement correction and thus achieves higher performance. The security strength and performance of SEBD are justified by our extensive analysis and simulations.

Acknowledgement

This work was supported in part by the US National Science Foundation under grants CNS-0626601 and CNS-0716306.

References

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: Research challenges," *Ad Hoc Networks J. (Elsevier)*, vol. 2, no. 4, pp. 351–367, Oct. 2004.

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM WSNA'02*, Sept. 2002.

[3] B. Krishnamachari and S. Iyengar, "Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Trans. Computers*, vol. 53, no. 3, pp. 241–250, Mar. 2004.

[4] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized event detection in sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2005.

[5] S. Madden, M. J. Franklin, J. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," in *OSDI*, 2002.

[6] K. K. Chintalapudi and R. Govindan, "Localized edge detection in sensor fields," *Ad Hoc Networks J.*, pp. 59–70, 2003.

[7] R. Nowak and U. Mitra, "Boundary estimation in sensor networks: Theory and methods," in *Proc. IPSN 2003*, vol. LNCS 2634, pp. 80–95.

[8] B. Krishnamachari and S. Iyengar, "Efficient and fault-tolerant feature extraction in wireless sensor networks," in *Proc. IPSN 2003*, vol. LNCS 2634, pp. 488–501.

[9] T. Clouqueur, K. Saluja, and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection," *IEEE Trans. Computers*, vol. 53, no. 3, pp. 320–333, 2004.

[10] F. Ye, H. Luo, S. Lu, and L. Zhang, "Stastical en-route filtering of injected false data in sensor networks," in *Proc. IEEE INFOCOM'04*, Mar. 2004.

[11] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proc. IEEE Symp. Security Privacy*, May 2004.

[12] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward resilient security in wireless sensor networks," May 2005.

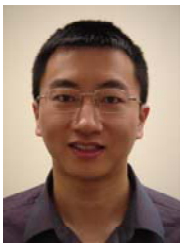
[13] K. Ren, W. Lou, and Y. Zhang, "Leds: Providing location-aware end-to-end data security in wireless sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2006.

[14] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based security mechanisms in wireless sensor networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 2, pp. 247–260, Feb. 2006.

[15] K. Ren, W. Lou, and Y. Zhang, "LEDS: providing location-aware end-to-end data security in wireless sensor networks," to appear, *IEEE Trans. Mobile Computing*.

[16] K. Ren, K. Zeng, and W. Lou, "A new approach for random key pre-distribution in large-scale wireless sensor networks," *J. Wireless Commun. and Mobile Computing* (special issue on wireless networks security), vol. 6, no. 3, pp. 307–318, 2006.
[17] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *IEEE Trans. Mobile Computing*, vol. 5, no. 6, pp. 640–652, June 2006.
[18] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proc. ASPLOS IX*, 2000.
[19] N. H. Vaidya and M. J. Miller, "A mac protocol to reduce sensor network energy consumption using a wakeup radio," *IEEE Trans. Mobile Computing*, vol. 4, no. 3, pp. 228–242, 2005.
[20] S. Čapkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. IEEE INFOCOM'05*, Mar. 2005.
[21] Y. Zhang, W. Liu, Y. Fang, and D. Wu, "Secure localization and authentication in ultra-wideband sensor networks," *IEEE J. Select. Areas Commun.* (special issue on UWB wireless communications - theory and applications), vol. 24, no. 4, pp. 829–835, April 2006.
[22] W. Zhang, H. Song, S. Zhu, and G. Cao, "Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks," in *Proc. ACM MOBIHOC*, May 2005.

Kui Ren is an assistant professor in the Electrical and Computer Engineering department at Illinois Institute of Technology. He obtained his Ph.D degree in Electrical and Computer Engineering from Worcester Polytechnic Institute in 2007. He received his B.Eng and M.Eng both from Zhejiang University, China, in 1998 and 2001, respectively. He worked as a research assistant at Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, from March 2001 to January 2003, at Institute for Infocomm Research, Singapore, from January 2003 to August 2003, and at Information and Communications University, South Korea from September 2003 to June 2004. His research interests include ad hoc/sensor network security, wireless mesh network security, Internet security, and security and privacy in networks and systems.



Kai Zeng received his B.E degree in Communication Engineering and M.E degree in Communication and Information System both from Huazhong University of Science and Technology, China, in 2001 and 2004, respectively. He is currently a Ph.D. student in the Electrical and Computer Engineering department at Worcester Polytechnic Institute. His research interests are in the areas of wireless ad hoc and sensor networks with emphases on energy-efficient protocol, cross-layer design, routing, and network security.

Wenjing Lou is an assistant professor in the Electrical and Computer Engineering department at Worcester Polytechnic Institute. She obtained her Ph.D degree in Electrical and Computer Engineering from University of Florida in 2003. She received the M.A.Sc degree from Nanyang Technological University, Singapore, in 1998, the M.E degree and the B.E degree in Computer Science and Engineering from Xi'an Jiaotong University, China, in 1996 and 1993 respectively. From December 1997 to July 1999, she worked as a Research Engineer in Network Technology Research Center, Nanyang Technological University. Her current research interests are in the areas of ad hoc and sensor networks, with emphases on network and system security and routing.