LEDS: Providing Location-aware End-to-end Data Security in Wireless Sensor Networks

Kui Ren
Worcester Polytechnic Institute,
Worcester, MA 01609
kren@ece.wpi.edu

Wenjing Lou
Worcester Polytechnic Institute,
Worcester, MA 01609
wjlou@ece.wpi.edu

Yanchao Zhang
University of Florida,
Gainesville, FL 32611
yczhang@ufl.edu

Abstract— **Providing end-to-end data security, i.e., data confidentiality, authenticity, and availability, in wireless sensor networks (WSNs) is a non-trivial task. In addition to the large number and severe resource constraint of sensor nodes, a particular challenge comes from potential insider attacks due to possible node compromise, since a WSN is usually deployed in unattended/hostile environments. Existing security designs provide a hop-by-hop security paradigm only, which leaves the end-to-end data security at high stake. Data confidentiality and authenticity is highly vulnerable to insider attacks, and the multi-hop transmission of messages aggravates the situation. Moreover, data availability is not sufficiently addressed in existing security designs, many of which are highly vulnerable to many types of Denial of Service (DoS) attacks, such as report disruption attacks, selective forwarding attacks, etc. In this paper, we seek feasible solutions to overcome these vulnerabilities. Through exploiting the static and location-aware nature of WSNs, we come up with a location-aware end-to-end security framework in which each node only stores a few secret keys and those secret keys are bound to the node's geographic location. The property of the location-aware keys successfully limits the impact of compromised nodes to their vicinity. We also propose a multi-functional key management framework which ensures both node-to-sink and node-to-node authentication along report forwarding routes. Moreover, our novel one-to-many data delivery approach guarantees efficient en-route bogus data filtering and is highly robust against many known DoS attacks. We evaluate our design through extensive analysis, which demonstrates a high security resilience against an increasing number of compromised nodes at the cost of a moderate protocol overhead.**

I. Introduction

Wireless sensor networks (WSNs) have drawn a lot of attention recently due to their broad applications in both military and civilian operations. A WSN usually consists of a large number of ultra-small, low-cost devices that have limited energy resources, computation, memory, and communication capacities [1], [2], [4], [7], and for the applications such as battlefield reconnaissance and homeland security monitoring. WSNs are often deployed in a vast terrain to detect events of interest and deliver data reports over multi-hop wireless paths to the sink. Data security is essential for these mission-critical applications to work in unattended and even hostile environments.

One of the most severe security threats in WSNs is security compromise of sensor nodes due to their lack of tamper resistance [7]. In WSNs, the attacker could compromise multiple nodes to obtain their carried keying materials and control them, and thus is able to intercept data transmitted through these nodes thereafter. As the number of compromised nodes grows, communication links between uncompromised nodes might also be compromised through malicious cryptoanalysis. Hence, this type of attacks could lead to severe data confidentiality compromise in WSNs. Furthermore, the attacker may use compromised nodes to inject bogus data traffic into WSNs. In this attack, compromised nodes pretend to have detected an event of interest within their vicinity, or simply fabricate an bogus event report claiming a non-existing event at an arbitrary location. Such *insider* attacks can severely damage network function and result in the failure of mission-critical applications. They may also induce network congestion and wireless contention, and waste the scarce network resources such as energy and bandwidth, hence, severely affecting both data authenticity and availability. Lastly, the attacker could use compromised nodes to launch selective forwarding attacks [3], in which compromised nodes selectively drop the going-through data traffic and thus to severely jeopardize data availability. The existence of the aforementioned attacks together with the inherent constraints of sensor nodes, make it rather challenging to provide satisfactory data security in WSNs with respect to all its three aspects, i.e., confidentiality, authenticity and availability [1]–[5].

Recent research has seen a growing body of work on security designs for WSNs [9]–[17]. Due to the resource constraint, most of the proposals are based on symmetric cryptography and only provide data authenticity and/or confidentiality in a hop-by-hop manner. End-to-end encryption/authentication is considered less feasible, particular in a WSN consisting of a large number of nodes [7]. However, the lack of the end-to-end security guarantee makes the WSN particulary vulnerable to the aforementioned attacks. The unique node-to-sink communication pattern in WSNs and the multi-hop communication paths aggravate the situation. The attacker could, therefore, make much less effort to obtain/manipulate its desired data without having to compromise a large number of nodes. To make things worse, existing security designs are highly vulnerable to many types of Denial of Service (DoS) attacks, such as report disruption attacks and selective forwarding attacks as discussed later. Data availability is far from sufficiently addressed in existing security designs.

In this paper, we propose an integrated security design providing comprehensive protection over data confidentiality, authenticity, and availability. Our design overcomes the limitations of the existing hop-by-hop security paradigm and achieves an efficient and effective end-to-end security paradigm in WSNs. We exploit the static and location-aware nature of WSNs, and propose a novel location-aware security approach through two seamlessly integrated building blocks: a location-aware key management framework and an end-to-end data security mechanism. In this approach, each sensor node is equipped with several types of symmetric secret keys, some of which aim to provide end-to-end data confidentiality, and others aim to provide both end-to-end data authenticity and hop-by-hop authentication. All the keys are computed at each sensor node independently from keying materials preloaded before network deployment and the location information obtained after network deployment, without inducing extra communication overhead for shared key establishment. Our Location-aware End-to-end Data Security design (LEDS) then provides a secure and reliable data delivery mechanism, which is highly resilient to even a large number of compromised nodes. The features of LEDS and the contributions of the paper are outlined as follows:

First, we propose a novel location-aware multi-functional key management framework. In LEDS, the targeted terrain is virtually divided into multiple cells using a concept called *virtual geographic grid*. LEDS then efficiently binds the location (cell) information of each sensor into all types of symmetric secret keys owned by that node. By this means, the impact of compromised nodes can be effectively confined to their vicinity, which is a nice property absent in most existing security designs. What the attacker can do is to misbehave only at the locations of compromised nodes, by which they will run a high risk of being detected by legitimate nodes if effective misbehavior detection mechanisms are implemented.

Second, LEDS provides end-to-end security guarantee. Every legitimate event report in LEDS is endorsed by multiple sensing nodes and is encrypted with a unique secret key shared between the event sensing nodes and the sink. Furthermore, the authenticity of the corresponding event sensing nodes can be individually verified by the sink. This novel setting successfully eliminates the possibility that the compromise of nodes other than the sensing nodes of an event report may result in security compromise of that event report, which is usually the case in existing security designs.

Third, LEDS possesses efficient en-route false data filtering capability to deal with the infamous bogus data injection attack. As long as there are no more than t compromised nodes in each single area of interest, LEDS guarantees that a bogus data report from that cell can be filtered by legitimate intermediate nodes or the sink deterministically.

Last, LEDS provides high level assurance on data availability by counteracting both report disruption [16] and selective forwarding attacks [3], simultaneously. By taking advantage of the broadcast nature of wireless links, LEDS adopts a one-to-many data forwarding approach, which is fully compatible with the proposed security framework. That is, all reports in LEDS can be authenticated by multiple next-hop nodes independently so that no reports could be dropped by a single node(s). Thus, LEDS is highly robust against selective forwarding attacks as compared to the traditional one-to-one forwarding approach used by existing security designs [14]–[16]. In addition, LEDS adopts a (t, T) threshold linear secret sharing scheme (LSSS) [24] so that the sink can recover the original report from any t out of T legitimate report shares. Not only this approach enhances the event report authenticity by requiring T sensing nodes to collaboratively endorsement the report, but also makes LEDS resilient to the interference from up to $T-t$ compromised nodes in the event area. Detailed analysis shows that the proposed LEDS is highly resilient to both types of attacks.

The rest of this paper is structured as follows. Section II articulates the data security goals in WSNs and evaluates related work with respect to these goals. Section III details the proposed LEDS design. Section IV presents the detailed security analysis of the proposed LEDS, followed by the performance analysis in Section V. Finally, the conclusion is drawn in Section VI.

II. Data Security Requirements in WSNs and Related Work

A. Data Security Requirements in WSNs

The requirements of data security in WSNs are basically the same as those well defined in the traditional networks, that is, data confidentiality, authenticity and availability [5], [19] - Data should be accessible only to authorized entities (usually the sink in WSNs), should be genuine, and should be always available upon request to the authorized entities. More specifically, the above three requirements can be further elaborated in WSNs as follows:

Data Confidentiality: In WSNs, data of interest usually appear as event reports sent by the sensing nodes from the area of occurrence via multihop paths to the sink. As the communication range of sensor nodes are limited, the reports will be relayed by the intermediate nodes before finally reaching the sink. Hence, the requirement on data confidentiality in WSNs is naturally: as long as the event sensing nodes are not compromised, the confidentiality of the corresponding data report should not be compromised due to any other nodes' compromise including the intermediate nodes along the report forwarding route.

Data Authenticity: Data reports collected by WSNs are usually sensitive and even critical such as in military applications, and hence, it is important to ensure data authenticity in addition to confidentiality. Since the undetected compromised node(s) can always send false reports, cryptography alone can not fully prevent such attacks. However, if we require that a valid report be collectively endorsed by a number, say T ($T > 1$), of sensor nodes which sense the event at the same time, we can protect data authenticity to the extent that no less than T compromised nodes can forge a valid report. Furthermore, by exploiting the static and location-aware nature of WSNs, we can require that a legitimate event report corresponding to certain area be only generated by the collaborative endorsement of no less than T nodes of that area. That is, to generate a valid report on a non-existing event happening in a certain area, the only way is to compromise T nodes in that area.

Data Availability: Since node compromise is usually inevitable in large-scale WSNs, it is rather important to prevent or be tolerant of the interference from compromised nodes as much as possible to ensure data availability. Therefore, security designs should be highly resilient to node compromise and the resulting attacks such as report disruption [16] and selective forwarding attacks [3]. In-network security-related processing such as false data filtering is vital to save scarce network resources and to prolong network lifetime.

B. Evaluation of Existing Security Designs in WSNs

In this section, we review existing security designs in the literature and evaluate them according to the above mentioned three data security requirements. We show that due to lack of end-to-end security guarantee, existing security designs fail to provide satisfactory security strength and are vulnerable to many types of attacks.

1) Limitations of existing key management schemes: Symmetric secret key pre-distribution is viewed as the most practical approach for establishing secure channels among sensor nodes because of the resource limitations in WSNs [6], [8], [10]. In the past few years, many secret key pre-distribution schemes have been proposed [6], [8]–[13]. By leveraging preloaded keying materials on each sensor node, these schemes establish pairwise keys between every two neighbor nodes after network deployment, and thus realize a hop-by-hop security paradigm. The security strength of these schemes is analyzed in term of the ratio of compromised communication links over total network communication links due to node compromise. Two types of node compromise are considered: random node capture and selective node capture, which differ in the key distribution information available to the attacker. Then to compromise the whole network communication, the attacker has to capture at least several hundreds of sensor nodes even under selective node capture attacks.

However, all these schemes assume a uniform wireless communication pattern in WSNs. Therefore, they are highly vulnerable to communication pattern oriented node capture attacks, because data of interest in WSNs are usually generated from the event happening area and transmitted all the way to the sink. Data confidentiality can be easily compromised due to lack of end-to-end security guarantee, since compromising any intermediate node will lead to the disclosure of the transmitted data. Therefore, the attacker only needs to compromise a relatively very small number of nodes to be able to obtain all the data transmitted in the whole network according to the observed communication pattern and network topology. The inherent reason is that the hop-by-hop security paradigm can only protect local communications but fails to provide strong protection to the most valuable node-to-sink data, which is of more interest to the attacker. At the same time, as the attacker could decrypt the intercepted data, it could, therefore, freely manipulate them to deceive the sink and hence severely affect data availability. The lack of end-to-end security association also makes it hard, if not impossible, to enforce data authenticity.

2) False data filtering schemes and their analysis: The general approach adopted to protect data authenticity in WSNs is: to generate a valid report, T ($T > 1$) nodes that sense the event simultaneously should first agree on the content of the event report, and in order to be forwarded by intermediate nodes and accepted by the sink, a valid report should be collaboratively endorsed (usually through Message Authentication Codes (MACs)) by these T nodes. Reports that are not properly endorsed will be filtered out by the intermediate nodes or the sink. Here the assumption is that every event of interest can be detected by at least T nodes simultaneously and the value of T is a system parameter. In the recent years, a few schemes have been proposed to design suitable key management schemes based on this approach, including Statistical En-route Filtering (SEF) [15], Interleaved Hop-by-hop Authentication (IHA) [14], Location-Based Resilient Secrecy (LBRS) [16], and our Location-Based Compromise-Tolerant (LBCT) [17]. LBRS is the most recently proposed scheme, which aims to solve the problems identified in the two previous schemes (SEF and IHA), and is a major improvement over these two schemes. In both SEF and IHA, compromising T nodes could break down the whole scheme. That is to say, after compromising T nodes, the attacker can then freely forge events "appearing" at arbitrary locations without being detected. In LBRS, the damage caused by node compromise is reduced due to the adopted location-key binding mechanism. Compromising T nodes now enable the attacker to fabricate events "appearing" at certain areas without being detected. However, it is still far from achieving the data authenticity requirement as stated above: to generate a valid report on a non-existing event happening in a certain area, the only way is to compromise T nodes in that area, and otherwise impossible. Therefore, there is still a big gap between the protection that existing schemes can offer and the requirement of data authenticity.

In addition, all the three schemes mentioned above are highly vulnerable to report disruption attack and selective forwarding attack. A single compromised node may prevent any event report in that area from being sent to the sink by simply offering a wrong MAC. Since the en-route filtering allows intermediate nodes to drop packets with false MACs, such reports will be rejected on its way to the sink because of the presence of the wrong MAC(s). In addition, with the common one-to-one forwarding approach, a compromised node can also drop any data report sent by its downstream nodes. Since the received report can only be verified by the compromised node at that point, there is no way for other nodes in its vicinity to distinguish such malicious dropping from legal dropping caused by the failure of endorsement verification. As the number of compromised nodes increases, the resulting damage will increase dramatically as discussed later in Section V. Therefore, these schemes do not provide adequate data availability.

III. LEDS: Location-aware End-to-end Data Security Mechanism

A. Assumptions, Threat Model and Design Goals

System Assumptions: In LEDS, we consider a large-scale uniformly distributed WSN that monitors a vast terrain via a large number of static sensor nodes, which can be deployed through approaches such as aerial scattering. We assume that an approximate estimation on the size and shape of the terrain is known a priori. Once deployed, each node can obtain its geographic location via a localization scheme [21]–[23]. We also assume that the WSN is well connected and densely deployed to support fine-grained collaborative sensing and be robust against node loss and failure. We assume that every event of interest can be detected by multiple sensor nodes. Once an event happens, the sensing nodes agree on a synthesized report, which is then forwarded toward the sink, typically traversing a large number of hops. The sink is a data collection center equipped with sufficient computation and storage capabilities. We assume that every sensor node has a unique *id* and is similar to the current generation of sensor nodes (e.g., the Berkeley MICA motes [20]) in its computation and communication capabilities and power resource. We also assume that sensor nodes are not tamper-resistant.

Threat Model: We assume that the attacker could compromise multiple nodes chosen arbitrarily and that if a node is compromised, all the information it holds will also be compromised. However, the sink is assumed to be secure because it is usually well protected and under the direct control of the network owner [15]. We also assume that the attacker can eavesdrop on all traffic, inject packets, and replay older packets. The attacker can take full control of compromised nodes and thus can manipulate compromised nodes to drop or alter messages going through them. We, however, do assume that there is a short bootstrapping phase right after network deployment during which no sensor nodes are compromised.

Design Goals: LEDS seeks to provide end-to-end data security for event reports, as well as en-route bogus report filtering in WSNs. In particular, it is designed to achieve the following goals:

- Provide end-to-end data confidentiality and authenticity: both confidentiality and authenticity of event reports should be guaranteed as long as the sending nodes themselves are not compromised. Moreover, the impact of compromised nodes (if any) should be confined to their vicinity. In other words, the attacker cannot utilize the cryptographic materials obtained from compromised nodes to launch attacks at places other than the locations of the compromised nodes.
- Achieve high-level of assurance on data availability: 1) being resilient against report disruption attacks and selective forwarding attacks; 2) being able to early detect and drop bogus reports in an effective and deterministic manner.

B. Notation and Terms

For ease of description, we use the following notation and terms:

N	network size
n'	number of nodes within one cell
u, v, z, m	unique *id*s of sensor nodes
I_u	index of node u's home cell
l	side length of a cell
K_M^I, K_M^{II}	two master secret keys
K_u^1, K_u^2	two unique secret keys shared between u and sink
SK_u	keying material for shared key establishment between u and dynamically added nodes
K_{I_u}	the *cell key* shared among the nodes in the same cell I_u
K_{I_u,I_v}	the *authentication key* shared between nodes in cell I_u and nodes in cell I_v
H	hash function
M	the event report to be protected.
C	encrypted report
C_u	a share of C computed through a LSSS, contributed by a node u
C_{share}	a set of shares with $\lvert C_{share}\rvert = T$
$E_\bullet(M)$	encryption of M using key "$\bullet$"
$Mac_\bullet(M)$	the message authentication code (MAC) computed over M using key "$\bullet$"
T	the number of endorsements included in an event report
t	the minimum number of endorsements to validate an event report
r $(r > l)$	communication radius of sensor nodes
p	a large prime

geographic virtual grid: A geographic virtual grid is a virtual geographic partition of the target terrain, which divides the terrain into multiple square cells. The parameters of a geographic virtual grid consist of a reference point and the cell size. For convenience only, we assume that there is only one static sink in the WSN, and the reference point, referred to as (x_0, y_0), is set to be the location of the sink, which is known before network deployment. The cell size is defined by l, which is the side length of the cell, and a cell is uniquely indexed by its center location. And hereafter, we refer to a cell by its center location.

home cell, event cell: The cell that a node, say u, is located in after network deployment, is called *home cell* of u, denoted as I_u. We call a cell an *event cell*, when a certain event of interest happens in that cell. Each report thus corresponds to one particular *event cell*.

report-forward route: In LEDS, an event report is relayed from the event cell to the sink in a cell-by-cell basis along its *report-forward route*. A report is always relayed between adjacent cells[1] towards the sink. More specifically, a report is always sent from one cell to one of its four adjacent cells

[1]Two cells are adjacent if they share a common side.

that is closest to the sink. The *report-forward route* of node u therefore consists of all the cells that are intersected by the line segment that connects the center of I_u and the sink. These cells are sequenced according to their distances to the sink: the cell that an report travels first ranks first and so on.

report-auth area: The *report-auth area* of node u comprises two parts, namely, the *downstream report-auth area* and the *upstream report-auth area*, depending on their location related to node u. Both areas are defined with regard to a sector area that is bound by two rays, both extending from the sink (x_0, y_0) and through a vertex of cell I_u. The two rays form the largest acute angle which contains the center of I_u as shown in Fig. 1. Then the *downstream report-auth area* of u is defined as all the cells that are farther to the sink than I_u and have their centers located inside the sector area, while the *upstream report-auth area* consists of all the cells that are closer to the sink than I_u and have any part of them falls into the sector area. Obviously, *report-forward route* of node u is always a part of its *upstream report-auth area*.

report-auth cell: A cell is called a *report-auth cell* of node u, if this cell belongs to u's *report-auth area* and at least one node in this cell shares an *authentication key* with u. Furthermore, if a *report-auth cell* of u is located in the *upstream report-auth area* of u, it is a *upstream report-auth cell* of u. Otherwise, it is a *downstream report-auth cell* of u.

These terms are graphically illustrated in Fig. 1.

C. Scheme Overview

The proposed LEDS scheme consists of two major components: the underlying key management framework and the end-to-end data security mechanism seamlessly built upon the former.

Location-aware key management framework: Key management in LEDS exploits the static and location-aware nature of WSNs. By leveraging pre-loaded keying knowledge among sensor nodes, a light-weight and robust location-aware key management framework is efficiently realized through embedding location information into the keys. In LEDS, each node computes three different types of location-aware keys: 1) two *unique secret key*s shared between the node and the sink and used to provide node-to-sink authentication; 2) a *cell key* shared with other nodes in the same cell that is used to provide data confidentiality; and 3) a set of *authentication key*s shared with the nodes in its *report-auth cell*s and used to provide cell-to-cell authentication and en-route bogus data filtering. All these keys are computed by each node locally and independently. Together with a predefined threshold secret sharing scheme, the key management framework serves as the basis for the upper layer end-to-end data security mechanism.

End-to-end data security mechanism: LEDS aims to protect data reports in a comprehensive and end-to-end manner. It provides data confidentiality by encrypting each event report with the cell key of the corresponding event cell. Since the cell key is merely shared among nodes of the event cell and the sink, the report confidentiality is guaranteed as long as no node in the event cell is compromised. Moreover, LEDS ensures

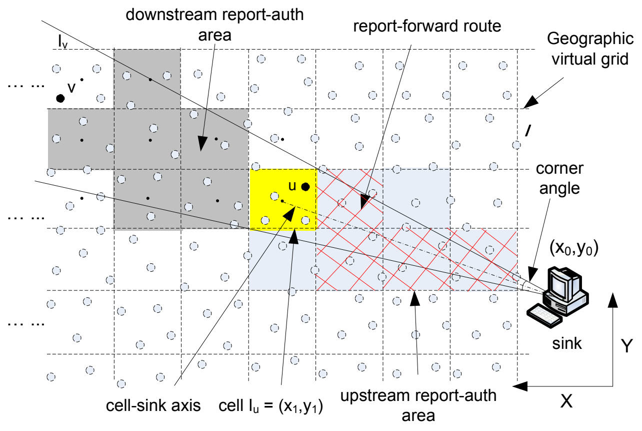


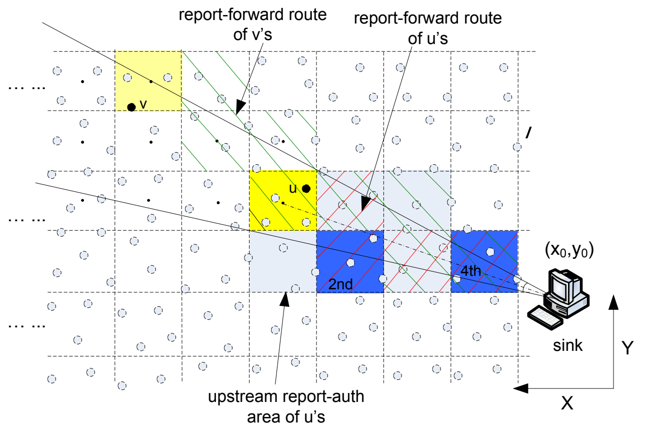
Fig. 1. Term illustration: defined for node u

data availability by offering strong protection from both report disruption and selective forwarding attacks. In particular, it deals with the report disruption attack by dividing the encrypted report into a number of unique shares via a pre-defined linear secret sharing scheme (LSSS). Due to the threshold property of LSSS, the sink can always recover the report from a subset of shares despite the presence of some wrong shares. To defend against the selective forwarding attack, LEDS uses cell-to-cell authentication keys to guarantee that each report can be verified by multiple next-hop nodes at any point on the report-forward route. This unique design makes it possible to use the one-to-many data forwarding approach instead of the vulnerable one-to-one approach adopted by most existing security schemes. Furthermore, LEDS ensures data authenticity by enforcing both en-route filtering at the intermediate nodes and end-to-end verification at the sink. The intermediate nodes can perform en-route bogus report filtering through verifying the attached MACs. And the sink can finally verify whether the report was indeed sent by the claiming nodes through examining both the authenticity of the attached MACs and the uniqueness of the shares.

D. Protocol Detail

In what follows, we present the detailed design of the proposed LEDS.

1) ***Location-aware key management framework:*** Before network deployment, the network planner prepares a *geographic virtual grid* of the targeted terrain with reference point (x_0, y_0) and cell size l. Let N and n' be the total number of network nodes and the average number of nodes in each cell, respectively. Based on N, n', and l, the network planner further decides two parameters T and t, of which the former is the number of endorsements to be included in a valid report and the latter refers to the minimum number of correct endorsements required to validate a report. The impact of these parameters on the security strength and performance of LEDS will be discussed in Sections IV and V, respectively. The network planner also selects two master secret keys, K_M^I and K_M^{II}, and a large prime p. The three parameters T, t,


report-forward route of v's
report-forward route of u's
v
u
2nd
4th
(x0,y0)
sink
Y
X
l
upstream report-auth area of u's
Fig. 2. Illustration of *report-auth cell*s of node u

and p defines a (t, T) LSSS over finite field $GF(p)$. Finally, each sensor node is preloaded with the following bootstrapping parameters before network deployment:

$$\{K_M^I, K_M^{II}, l, (x_0, y_0), (t, T), p\}.$$

In this paper, we assume that there is a short safe period right after network deployment, during which some sensor self-positioning algorithm is executed so that each sensor node obtains its location information. Key establishment is performed by each sensor node independently upon the availability of its location information. In what follows, we take node u as an example to illustrate the key establishment process:

Node u first determines its *home cell* $I_u = (x_1, y_1)$, then computes two *unique secret key*s it shares with the sink as

$$K_u^1 = H(K_M^I|u|I_u|0), \quad K_u^2 = H(K_M^I|u|I_u|1)$$

where | denotes message concatenation. u also computes a *cell key* K_{I_u} it shares with other nodes in its *home cell* as

$$K_{I_u} = H(K_M^I|I_u).$$

After that, u declares its existence to the neighbors by broadcasting tuple $\{u, I_u\}$, which is encrypted with K_{I_u}.

A node u proceeds to compute the *authentication key*s shared with its *report-auth cell*s. Consider one such cell with location (x_c, y_c) as an example. The corresponding authentication key is derived as

$$H(K_M^{II}|(x_1, y_1)|(x_c, y_c)).$$

The report-auth cells are determined according to cell I_u's relative location to the sink. Specifically, a cell is a *downstream report-auth cell* of node u, if it is in u's *downstream report-auth area* and is no more than $T+1$ cells away from I_u[2]. As an example, all the grey cells shown in Fig. 1 are u's *downstream report-auth cell*s, where $T = 3$. It is worth noting that only horizontal or vertical cell transverse is allowed in LEDS, that is, no diagonal cell transverse is allowed. Therefore, cell I_v is not a downstream report-auth cell of node u because it is five cells away from I_u. The quantitative analysis on the number of *downstream report-auth cell*s a node has will be discussed in Section V in the context of key storage overhead analysis.

[2]Adjacent cells are considered one cell away.

To determine its *upstream report-auth cell*s, node u first determines its own rank $rank_u$ in I_u according to its *id*, and calculates $rank_u = (rank_u \ mod \ T) + 1$. Then the $rank_u$-th cell in the report-auth route of u is one of its report-auth cells. The remaining ones are those cells within its *upstream report-auth area* that are exactly $T + 1$ cells closer to the sink than I_u. In case that I_u is less than $(rank_u + 1)$ or $(T + 1)$ cells away from the sink, the sink can be chosen. Consider node u with $rank_u = 1$ in Fig. 2 as an example. Assuming $T = 3$, the second and fourth cells denoted in Fig. 2 are u's *upstream report-auth cell*s.

To summarize, for any two nodes u and v, if I_v is a *downstream report-auth cell* of u, then

- every node in I_u shares the *authentication key* $K_{I_u,I_v} = H(K_M^{II}|I_u|I_v)$ with at least one node in I_v. Furthermore, if the two cells are exactly $T + 1$ cells away from each other in the *report-forward route* of v, then every node in I_u shares K_{I_u,I_v} with every node in I_v.
- the *report-forward route* of v falls into the *upstream report-auth area* of u after the route reaches I_u (see Fig. 2).

At last, a node u deletes the two system secrets K_M^I and K_M^{II} but keeps $SK_u = H(K_M^{II}|I_u)$. This operation is for dual purposes: 1) it ensures that adversaries will not be able to derive other nodes' keying information by harnessing compromised nodes; 2) it allows future shared key establishment with dynamically deployed nodes: for a newly added node w, $K_{I_u,I_w} = H(H(K_M^{II}|I_u)|I_w)$, and for node u, $K_{I_u,I_w} = H(SK_u|I_w)$.

*2) **End-to-end data security mechanism**: Report generation*: We assume that each event of interest is simultaneously detected by at least T nodes in each cell and exactly T of them participate in report generation. Consider cell I_u as an example. Each of the T participating nodes first agree on an event report M, which usually contains information such as event type, sensing location (i.e., *id* of the *event cell*), and a timestamp. Note that all the related communications are protected by the *cell key* so that M is confidential against any outside nodes. Next, each participating node, say u, encrypts M using the cell key K_{I_u} and gets $C = E_{K_{I_u}}(M)$. Then it computes a unique share C_u of C through the predefined (t, T) LSSS. Specifically, C_u is obtained by evaluating the following bivariate polynomial of degree t over finite field $GF(p)$ using K_u^1 and K_u^2:

$$C_u = \mathcal{F}(K_u^1, K_u^2) = \sum_{0 \le i \le t-2} a_i (K_u^1)^{i+1} + a_{t-1}(K_u^2)^t \text{ mod } p, \quad (1)$$

where a_i $(i = [0, t-1])$ are a full partition of C, and both p and t are the two preloaded parameters. Note that C_u is uniquely generated by u and therefore can be viewed as an endorsement to be verified by the sink, because the polynomial is evaluated using u's two *unique secret key*s, which are only known to u and the sink. Node u then broadcasts tuple $\{u, C_u\}$ and collecting other $T-1$ shares at the same time. Next, u computes two MACs over all the T shares of C, i.e., C_{share}, as another layer of endorsement to the report, which enable the intermediate nodes to perform en-route bogus report filtering. The two MACs are computed using the authentication keys that u shares with two of its *upstream report-auth cell*s. Suppose I_v and I_o are two *upstream report-auth cell*s of u, and I_o ranks $T+1$ -th with respect to u's *report-forward route*. Then the obtained MACs are $Mac_{K_{I_u,I_v}}(C_{share})$ and $Mac_{K_{I_u,I_o}}(C_{share})$. Tuple $\{u, Mac_{K_{I_u,I_v}}(C_{share}), Mac_{K_{I_u,I_o}}(C_{share})\}$ is then broadcasted to finish the synthesization of the final report. Node u constructs and sends out the final report after collecting $T+1$ different MACs ($2T$ MACs in total). The final report contains: 1) *event cell id*; 2) *id*s of T participating nodes'; 3) C_{share}; and 4) $T+1$ MACs. Note that both the *id*s of the participating nodes and the $T+1$ MACs are listed in the final report in order based on the node ranks (The common MAC $Mac_{K_{I_u,I_o}}(C_{share})$ is listed lastly). The report is sent by the node which completes the synthesis of the report and seizes the channel first. To avoid sending duplicate reports, each node overhears the channel and uses the techniques described in [16], [26].

Interleaved cell-by-cell en-route filtering: In LEDS, data reports are relayed cell by cell and delivered following the robust one-to-many, instead of existing failure-prone one-to-one, forwarding paradigm. A sending/intermediate node locally broadcasts a data report to the next cell on the report-forward route. As we mentioned before, it is easy to determine the next cell on the report-forward route: the one that is adjacent to the sending cell and is closest to the sink. Nodes in the receiving cell verify the report and, upon successful verification and processing, one of them rebroadcasts the report further to the next cell. Again, duplicate reports are suppressed by using the techniques such as back-off before sending [16], [26].

In LEDS, a legitimate intermediate node performs the following verification to a received report. It first verifies the first MAC in the report using the corresponding shared *authentication key*:

- if zero, deletes it and attaches another zero to the end;
- If valid, deletes it and attaches a new MAC to the end;
- If invalid, deletes it and attaches a zero to the end.

It then checks whether the number of non-zero MACs is enough or not: if the number is enough, it forwards the processed report; otherwise, it discards it. Note that there is no way for a single node to launch the *selective forwarding attack*, since each report can be verified by multiple nodes simultaneously. Every other node from the same cell is ready to forward a legal report.

The number of non-zero MACs is considered not enough by an intermediate node if 1) it contains less than $t+1$ different non-zero MACs or 2) it contains less than $T-j+2$ different non-zero MACs, when *event cell* is j cells ($j \in [1, T-t]$) away from its own. And the new MAC is computed over C_{share} using the corresponding *authentication key* shared between the intermediate node and one of its *upstream report-auth cell*s that is exactly $T+1$ cells away from its own *home cell* with respect to the *report-forward route* of I_u.

Sink verification: A report is verified at the sink in two steps to ensure its authenticity: 1) the sink verifies whether the report contains no less than $t+1$ valid non-zero MACs; 2) the sink checks whether the report is indeed endorsed by the T nodes as claimed. The sink fulfills the first step via the *authentication key*s it shares with the intermediate cells, and the second step by recovering the report C from C_u. To do so, it tries to recover C from any t correct shares, and then decrypts the recovered C using the corresponding *cell key* of the *event cell*[3]. More specifically, the recovery operation of M goes as follows: the sink picks t out of T shares, and based on their corresponding secret keys[4], solves a t-variable linear equation system to get $a_i, i = [0, t-1]$ in Equ. (1). The sink thus obtains C. The sink further decrypts C and gets M. At this point, if M is meaningful (i.e., conforming to the pre-defined report format), the recovery operation succeeds. Otherwise, sink tries another combination of t shares. Note that as long as there are no more than $T-t$ invalid shares, sink is always able to recover the original report due to the nice threshold property of the adopted (t, T) LSSS. And as long as the sink can recover the original report M, it can ascertain that all the corresponding shares are indeed generated by the nodes as claimed.

E. An example

In Fig. 3, we show how the proposed data security framework works through a simple example. For brevity, we show the corresponding security operations only. Suppose $T=3$, $t=2$ and nodes m, s and u ($m < s < u$) are three nodes from the *event cell*. Hence, a report can be:

$$\{I_u, m, s, u, C_m, C_s, C_u, Mac_{K_{I_u,I_v}}(C_m|C_s|C_u),$$
$$Mac_{K_{I_u,I_z}}(C_m|C_s|C_u), Mac_{K_{I_u,I_o}}(C_m|C_s|C_u),$$
$$Mac_{K_{I_u,I_{v'}}}(C_m|C_s|C_u)\}.$$

Then a successful protocol run goes as follows: when node v receives the report, it checks that the report contains four non-zero MACs. Next, v verifies the first MAC in the report using K_{I_u,I_v}. Then v removes this MAC and attaches a new one to the end, which is also computed over C_{share} but with $K_{I_v,I_{z'}}$, because $I_{z'}$ is four cells closer to the sink with respect to the *report forwarding route* of I_u. Lastly, node v forwards the processed report:

$$\{I_u, m, s, u, C_m, C_s, C_u, Mac_{K_{I_u,I_z}}(C_m|C_s|C_u),$$
$$Mac_{K_{I_u,I_o}}(C_m|C_s|C_u), Mac_{K_{I_u,I_{v'}}}(C_m|C_s|C_u),$$
$$Mac_{K_{I_v,I_{z'}}}(C_m|C_s|C_u)\}.$$

[3]Based on the cell *id* contained in the report.
[4]Based on the node *id* contained in the report.

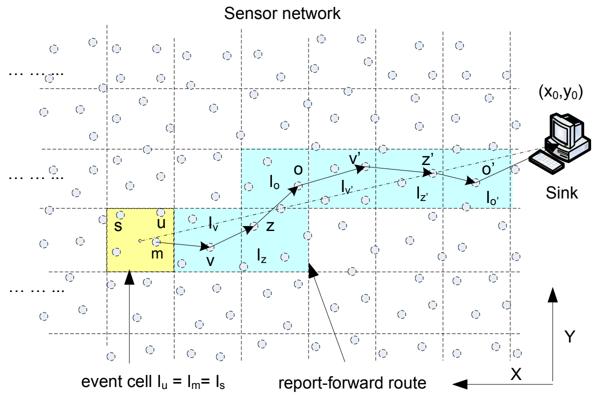


Fig. 3. An example of the proposed end-to-end data security mechanism

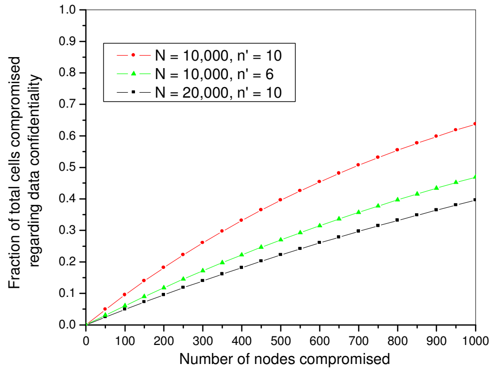


Fig. 4. Data confidentiality in LEDS under random node capture attacks

As the report is forwarded along the route, it is further verified and processed by the intermediate nodes accordingly. Therefore, node z' receives the report as

$$\{I_u, m, s, u, C_m, C_s, C_u, Mac_{K_{I_v,I_{z'}}}(C_m|C_s|C_u),$$
$$Mac_{K_{I_z,I_{o'}}}(C_m|C_s|C_u), Mac_{K_{I_o,sink}}(C_m|C_s|C_u),$$
$$Mac_{K_{I_{v'},sink}}(C_m|C_s|C_u)\}.$$

And the report reaching the sink is of format

$$\{I_u, m, s, u, C_m, C_s, C_u, Mac_{K_{I_o,sink}}(C_m|C_s|C_u),$$
$$Mac_{K_{I_{v'},sink}}(C_m|C_s|C_u), Mac_{K_{I_{z'},sink}}(C_m|C_s|C_u)$$
$$Mac_{K_{I_{o'},sink}}(C_m|C_s|C_u)\}.$$

The sink first verifies all the four MACs and then recovers the original C from any two of C_m, C_u, and C_s. From the *id* information in the report and Equ. 1, the sink solves a 2-variable linear equation system and thus obtains C. Sink further decrypts C using K_{I_u}, and therefore obtains M. If M is meaningful, the recovery operation succeeds. Note that the sink will always be able to recover M, if there are no more than $T - t = 1$ invalid shares. In contrast, in existing schemes [14]–[16], the authenticity of M is verified at the sink through verification on a set of end-to-end MACs; any invalid MAC results in discarding the report.

IV. Security Analysis of LEDS

In this section, the security strength of the proposed LEDS is analyzed with respect to the three aspects as mentioned in design goals, i.e., data confidential, authenticity and availability.

A. Security Strength of LEDS Regarding Data Confidentiality

In LEDS, every report is encrypted by the corresponding *cell key* and therefore, no nodes out of the *event cell* could obtain its content. Compromising any number of intermediate nodes will not break the confidentiality of the report. Only when a node from the *event cell* is compromised could the attacker obtain the contents of the corresponding reports. We say that a cell is compromised with regard to data confidentiality in this case. Our concern here is how compromised nodes under both random and selective node capture attacks affect the confidentiality of the communications from different cells? That is, given the number of compromised nodes, what is the fraction of the compromised cells with respect to total network cells? *Random node capture attack*: Given the network size N and the average number of nodes in each cell n', there are altogether $\frac{N}{n'}$ cells in a *geographic virtual grid*, assuming n' divides N. Therefore, if x nodes are compromised under random node capture attack, the probability that a cell is compromised is

$$1 - \frac{\binom{N-n'}{x}}{\binom{N}{x}} \quad (2)$$

On the other hand, Equ. (2) also represents the fraction of total cells that are compromised given x nodes are compromised. In Fig. 4, we show how the number of compromised nodes affects data confidentiality in LEDS. It is clear that to compromise 40% of the total cells, at least 5% of the total nodes have to be compromised. This means at least 500 nodes, given $N = 10,000$ and $n' = 10$. Furthermore, the security resilience increases as n' decreases as shown in Fig. 4. Therefore, LEDS has a much higher level of resilience against random node capture attacks than existing security designs [8], [10], in which compromising a few hundred nodes usually compromises even all the network communications, given the same network size.

Selective node capture attack: In this case, to compromise the whole network, the attacker has to selectively capture at least one node from each cell. This implies at least $\frac{N}{n'}$ nodes are required, that is, around 1000 nodes, given $N = 10,000$ and $n' = 10$. Note that this is 10% of the total network nodes. In LEDS the damages caused by the compromised nodes are confined due to the location-aware nature of the *cell keys*. Compromised nodes at one area cannot be used to compromise communications originating from other areas, since they do not have any information on *cell keys* of other cells. In contrast, it is much easier to compromise the total network communications under the existing hop-by-hop security paradigm, in which compromising one node will compromise all the communications going through it. Therefore, to compromise the whole network communications, the attacker may merely need to compromise a very small number of nodes on the order of as low as tens[5].

B. Security Strength of LEDS Regarding Data Authenticity

In addition to obtaining the content of legitimate reports, the attacker may want to insert bogus reports to fool the sink with non-existing events. In LEDS, in order for a bogus report to successfully pass both en-route filtering and sink verification, the attacker has to compromise at least t nodes in the corresponding *event cell*. We say that a cell is compromised with regard to data authenticity in this case. Notice that under this worst-case scenario, namely, t or more nodes in a single cell have been compromised, only events "appearing" in that cell can be forged, due to the location-aware property of the underlying endorsement keys that provides both node-to-sink and cell-to-cell authentication. Therefore, LEDS presents an improvement over existing security designs such as SEF, IHA, and LBRS [14]–[16], in which compromising any single node would result in multiple gains, i.e., helping the attacker compromise the authenticity of both its own home cell/cluster and any of its downstream cells/clusters.

Therefore, our first concern is that given the number of compromised nodes, what fraction of the total cells are affected with respect to data authenticity? Under random node capture attacks, if the number of compromised nodes is x, then the probability that a cell is not affected, i.e., no node in a cell is compromised, is given by

$$P_{\{0\}} = \frac{\binom{N-n'}{x}}{\binom{N}{x}} \quad (3)$$

This also represents the percentage of cells that are *secure*. Accordingly, the percentage of cells that have at least one node compromised, respectively, is given by $1-P_{\{0\}}$. Furthermore, let $P_{\{i\}}$ represent the probability that exactly i nodes are compromised in a cell, we have

$$P_{\{i\}} = \frac{\binom{n'}{i}\binom{N-n'}{x-i}}{\binom{N}{x}}$$

Then the probability that the authenticity of a cell is compromised, i.e., having at least t compromised nodes is

$$P_{\{\geq t\}} = \sum_{i=t}^{n'} P_{\{i\}} = \sum_{i=t}^{n'} \frac{\binom{n'}{i}\binom{N-n'}{x-i}}{\binom{N}{x}} \quad (4)$$

This also represents the percentage of authenticity-compromised cells. Then the percentage of *affected* cells, i.e., each of which has at least 1 and at most $t-1$ compromised nodes, can be expressed as $1 - P_{\{0\}} - P_{\{\geq t\}}$. Obviously, the larger t is, the harder to compromise the authenticity of a cell. Fig. 5 illustrates how data authenticity is affected as the number of compromised nodes increases. We can observe that the percentage of compromised cells increases very slowly with the increase of number of compromised nodes. And it keeps very low: even if the number of compromised nodes reaches 1750, only 10% of cells are compromised. This indicates that under random node capture attacks, it is very hard for the attacker to compromise a cell and thus fool the sink with bogus reports. On the other hand, it is observed that the percentage of secure cells in the network deceases slowly while the percentage of affected cells increases quickly as the number of compromised nodes increases. This observation tells us that, although it is relatively easier for the attacker to insert the bogus reports into the network, these bogus reports can be deterministically filtered by the intermediate nodes or the sink.

[5]Such as a cut set close to the sink, which, when cut, isolates the sink from other part of the network.

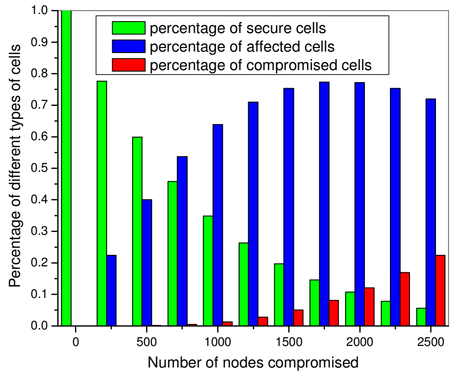


Fig. 5. Data authenticity in LEDS under random node capture attacks, where $N = 10,000$, $n' = 10$ and $(t, T) = (4, 5)$.

Hence, our next concern is that given the number of compromised nodes, what is the expected filtering position of a bogus report sent from an affected cell? In LEDS, in order for a bogus report from an affected cell to reach the sink (but be rejected by the sink), there should be at least $t-x_2$ of the first T cells in its *report-forward route* affected simultaneously, assuming that the number of compromised nodes in this affected cell is x_2 ($1 \leq x_2 \leq T-1$). This is because, to insert a bogus report, the compromised nodes in this affected cell have to forge at least $t-x_2$ MACs. In addition, to let these $t-x_2$ invalid MACs pass through the enroute filtering, there should be at least $t-x_2$ affected cells of the first T cells in its *report-forward route*: compromised node(s) from each affected cell could therefore let pass one corresponding invalid MAC and attach a new one as defined in LEDS. Therefore, there is no way for the intermediate nodes to check the authenticity of the received report after T cells, since now all the contained MACs in the report are indeed valid ones. In this case, the filtering position of the bogus reports from this affected cell should be its distance to the sink (that is, filtered by the sink). Otherwise, any bogus report from this cell will be filtered at most at the T-th cell and $\frac{T}{2}$-th cell on average. Assuming there are less than $t-x_2$ affected cells of the first T cells in its *report-forward route*, then at least one invalid MAC will be detected by nodes from the remaining secure cells. Now the bogus report originating from this cell will be filtered out at most at the T-th cell along the route. Under random node

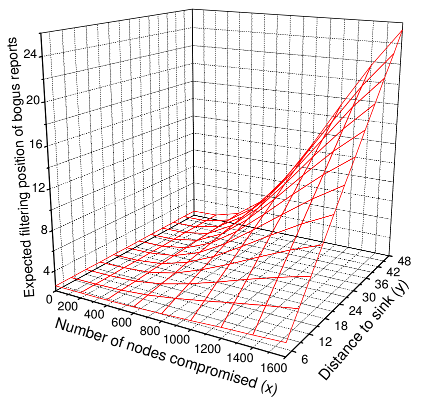


Fig. 6. Expected filtering position vs. number of compromised nodes with respect to different distances to the sink

capture attack, the average filtering position will be bounded by $\frac{T}{2}$, since the invalid MAC can be detected at any position between the first and T-th cells. Therefore, given the number of compromised nodes x, the expected filtering position of the bogus reports from an affected cell is bounded by

$$y\sum_{i=1}^{t-1}P_{\{i\}}(1-P_{\{0\}})^{t-i}+\frac{T}{2}(1-\sum_{i=1}^{t-1}P_{\{i\}}(1-P_{\{0\}})^{t-i}), \quad (5)$$

where this affected cell is y cells away from the sink with respect to its *report-forward route*. Fig. 6 illustrates how the filtering position varies as the number of compromised nodes increases, when $N = 10,000$, $n' = 10$, and T as low as 5. It is clearly shown in Fig. 6 that the bogus reports sent from most affected cells can be efficiently filtered under random node capture attacks. For example, the bogus reports from an affected cell that is 30 cells away from the sink will be filtered at no farther than the 10-th cell in the route on average, where the number of compromised nodes is 1000.

On the other hand, under selective node capture attacks, the attacker can choose as low as t nodes from one particular cell to compromise data authenticity of that cell. As discussed above, unlike existing security designs [14]–[16], compromised nodes from one cell in LEDS can not be used to compromise data authenticity of other cells. Note that in existing security designs, data authenticity of one cell can always be compromised because of the compromise of nodes from other cells. Hence, this feature of LEDS greatly increases the attacker's cost to launch such attacks.

C. Security Strength of LEDS Regarding Data Availability

As discussed before, there are two possible attacks that could severely affect data availability in WSN, namely, report disruption attack and selective forwarding attack. Existing security designs are highly vulnerable to these attacks [14]–[16]. In contrast, LEDS makes significant improvement in terms of data availability by being more resilient to such attacks. The strength of LEDS comes from both its report endorsement mechanism and its forwarding mechanism.

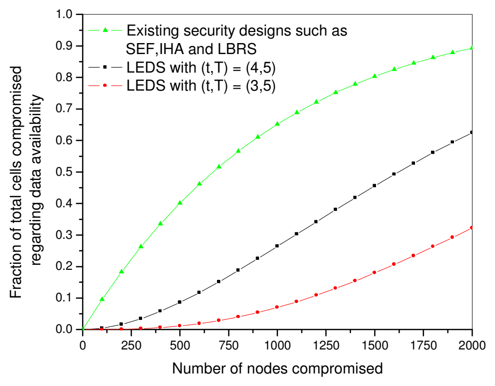


Fig. 7. Data availability in LEDS under report disruption attack

On the one hand, in LEDS, each node only contributes one share of the report following a (t, T) threshold LSSS. Therefore, the sink can always recover the original report even if there are up to $T - t$ compromised nodes from the corresponding *event cell* that contribute wrong shares to prevent the sink from obtaining the report. At the same time, the intermediate nodes only discard a report which contains less than t valid MACs. That is, if there are up to $T - t$ compromised nodes that contributes invalid MACs, the report can still be relayed to the sink. While in existing security designs, a single compromised node could prevent the sink from obtaining any report from that cell. Simply by contributing an invalid MAC to any report sent from that cell, the compromised node can always make the report to be discarded by the intermediate nodes. Under random node capture attack, given the number of compromised nodes x, the percentage of cells that have at least one node compromised, respectively, is given by $1 - P_{\{0\}}$; further the percentage of cells that have at least $T - t + 1$ nodes compromised, respectively, is given by

$$1-\sum_{i=0}^{T-t}P_{\{i\}} \quad (6).$$

Fig. 7 compares the data availability protection of LEDS with other existing security designs. It clearly shows that LEDS is much more resilient to the report disrupt attacks. In other words, an attacher needs to compromise a lot more nodes to successfully launch report disrupt attacks in LEDS. Given $N = 10,000$, $n' = 10$ and T as low as 5, to successfully launch report disrupt attack in 10% of total cells, around 100 nodes have to be compromised in existing security designs, while this number has to be no less than 600 in LEDS. Furthermore, by increasing $T - t$, LEDS can increase the resilience even more, or in other words, making the attack even harder, as shown in Fig. 7. Lastly, even under selective node capture attacks, the cost to successfully launch report disrupt attack in the same number of cells in existing security designs, will still be $T - t$ times higher than in LEDS.

On the other hand, a compromised node can always drop all the reports going through itself in existing security designs due

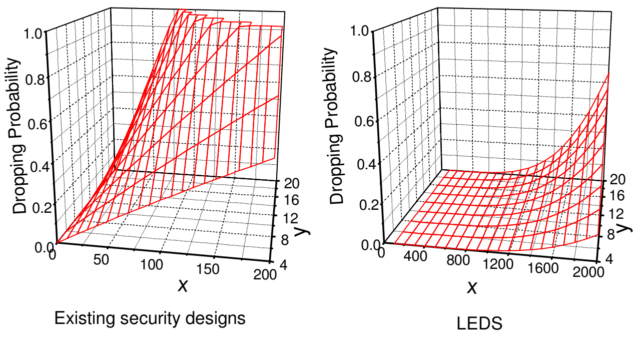


Fig. 8. Data availability in LEDS under selective forwarding attack

to the failure-prone nature of one-to-one forwarding paradigm. Compromising any intermediate node from the report-forward route would be sufficient enough for the attacker to successfully drop the message without being detected, since other nodes have no appropriate keys to verify the authenticity of the report. However, in LEDS it is impossible for a compromised node to prevent the report from being forwarded. This is because every report in LEDS is forwarded to all nodes in the next cell and each of them function the same way. Therefore, as long as not all the nodes that hear the report are compromised, the report can always be forwarded to the next cell. Hence, the proposed one-to-many forwarding approach in LEDS greatly enhances data availability in WSNs.

More precisely, suppose a cell is y cells away from the sink. Then, if we apply one-to-one forwarding approach as in existing security designs, the probability that the corresponding report sent from this cell is dropped by a compromised intermediate node can be estimated by

$$\frac{yl}{r}(1 - P_{\{0\}}), \qquad (7)$$

under random node capture attack, while in LEDS this probability is bounded by

$$y\left(1 - \sum_{i=0}^{\lfloor \frac{n'(r-l)}{l} \rfloor} P_{\{i\}}\right), \qquad (8)$$

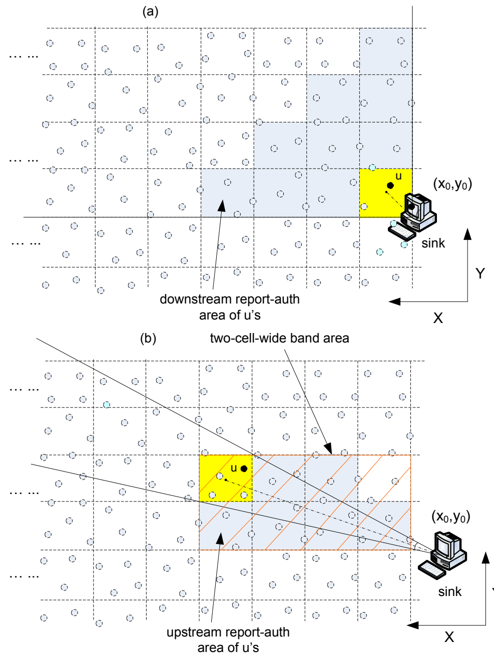
assuming $l \leq r \leq 2l$. Fig. 8 clearly illustrates the huge improvement on data availability provided by LEDS.

V. Performance Analysis of LEDS

In this section, we evaluate the performance of the proposed LEDS in terms of storage overhead and computation and communication overheads.

A. Key Storage Overhead

In LEDS, each node stores two *unique secret keys* which are only known to itself, and one *cell key* shared with all other nodes in its *home cell*. Of course, both keys are also known by the sink. Furthermore, each node stores one *authentication key* for each of its *report-auth cells*. For a particular node, say u, the number of its *report-auth cells* is decided by u's relative position with respect to the sink.



(a)
… …
u
(x₀,y₀)
sink
Y
X
downstream report-auth area of u's
(b)
two-cell-wide band area
u
(x₀,y₀)
sink
Y
X
upstream report-auth area of u's

Fig. 9. Data availability against selective forwarding attack

More specifically, the number of *downstream report-auth cells* of u is bounded by $\frac{(T+1)(T+2)}{2}$, when the *home cell* I_u is right next to the sink as shown in Fig. 9(a). On the other hand, from its definition, we know that any node's *upstream report-auth area* is a subset of the two-cell-wide band area as shown in Fig. 9(b). Obviously, in a two-cell-wide band area, all the possible routes, extending monotonically toward the sink[6] have at most two different choices at each step. Therefore, the cells that are exactly $T+1$ cells closer to the sink as compared to I_u also have at most 2 different choices. Hence, the number of *upstream report-auth cells* of any node is bounded by 3, and the total number of keys stored by each node in LEDS is bounded by

$$\frac{(T+1)(T+2)}{2} + 6. \qquad (9)$$

Therefore, despite of its strong filtering capability and end-to-end security guarantee, LEDS only requires the nodes to storage a small number of keys, which can be as low as 21 given $T = 5$. Moreover, the number of keys is independent of the network size, which makes LEDS highly suitable in large scale WSNs. Furthermore, the sink stores very few keys in LEDS, i.e., two master keys K_M^I and K_M^{II} only. All the other keys can be derived on-the-fly from the *id* and location information (i.e., *cell id*) contained in the received data reports.

B. Computation and Communication Overheads

In LEDS, key establishment only involves efficient hash operations during the bootstrapping period. And since the authentication keys are shared in a cell-to-cell manner, they can be reused for en-route filtering during the whole network

[6]horizontal and vertical cell transverse only

life. This feature saves a lot of unnecessary computation due to key reestablishment. In contrast, whenever a forward route changes, all the authentication keys in IHA [14] should be reestablished to enable en-route filtering due to the weakness of the one-to-one forwarding approach. On the other hand, to generate an authentic report, each node needs to compute two MACs and execute one LSSS operation, which can be performed using efficient $\mathcal{O}(|p|\log^2|p|)$ algorithms [24]. Furthermore, to forward a report, each node needs to verify one MAC and compute another MAC. Since the energy for computing a MAC is about the same as that for transmitting one byte, the computation cost involved by LEDS is very low. In addition, to judge whether a node belongs to a particular *report-forward route*, only simple geometry computation is involved based on the *geographic virtual grid*.

The communication overhead of our scheme results from two sources as compared to the original report. First, every authentic report contains $T+1$ MACs. Since the size of these MACs only impacts the capability of en-route filtering, in practice it can be made smaller as a tradeoff between performance and security. For example, if we use 6 bytes for all the MACs, and $T = 5$, the size of a MAC will be 1 byte. Therefore, the introduced additional message overhead is only 6 bytes in this example. Second, since the encrypted report is divided into a set of unique shares as node-to-sink endorsements, this would result in possible message size enlargement. For example, assuming M is 36-byte (288-bit) long as in TinyOS [18] and $(t,T) = (4,5)$, then each share will be 9 bytes in length and there will be 5 shares in total according to the underlying LSSS. Hence, the size of additional message overhead is only one-fourth of the original message length, i.e., 9 bytes. Note that these additional message overheads provide much stronger security strength and resilience. Also note that the choice of T should be based on both security and node density. A large T makes it more difficult for the adversary to launch a false data injection attack, but it also requires more nodes to form a cell. Moreover, report delivery in LEDS follows a pre-defined route in a cell-by-cell manner. Hence, it is highly robust and resilient against node failures and other possible routing changes as compared to the one-to-one forwarding paradigm in existing security designs [14]–[16]. The elimination of unnecessary routing overhead also helps LEDS achieve communication efficiency.

VI. Conclusion

In this paper, by exploiting the static and location-aware nature of WSNs we came up with a location-aware end-to-end security framework to address the vulnerabilities in existing security designs. In our design, the secret keys are bound to geographic locations, and each node stores a small number of keys based on its own location. This location-aware property successfully limits the impact of compromised nodes only to their vicinity without affecting end-to-end data security. Furthermore, the proposed multi-functional key management framework ensures both node-to-sink and node-to-node authentication along report forwarding routes. Moreover, our data delivery approach guarantees efficient en-route bogus data filtering, and is highly robust against DoS attacks. We evaluate our design through extensive analysis, which demonstrates its high resilience against an increasing number of compromised nodes and its efficiency in terms of protocol overheads.

References

[1] D. Carman, P. Kruus, B. Matt, “Constraints and approaches for distributed sensor network security,” NAI Labs Tech. Report #00-010, 2000

[2] A. Wood and J. Stankovic, “Denial of Service in Sensor Networks,” IEEE Computer, Oct. 2002.

[3] C. Karlof and D. Wagner, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures”, Ad Hoc Networks, 1(2), 2003.

[4] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, “SPINS: Security Protocols for Sensor Networks,” In Proc. of Mobicom 2001.

[5] Elaine Shi and A. perrig, “Designing Secure Sensor Networks,” Wireless Communication Magazine, 11(6), December 2004.

[6] L. Eschenauer and V. Gligor, “A key-management scheme for distributed sensor networks,” In Proc. of the 9th ACM CCS, Washington, 2002

[7] H. Chan, A. Perrig, “Security and privacy in sensor networks,” IEEE Computer, pp. 103-105, Oct 2003

[8] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks,” In Proc. of IEEE S&P, 2003

[9] D. Liu and P. Ning, “Establishing Pairwise Keys in Distributed Sensor Networks,” In Proc. of the 10th ACM CCS, Washington, Oct., 2003

[10] W. Du, J. Deng, Y. Han, and P. Varshney, “A Pairwise Key Predistribution Scheme for Wireless Sensor Networks,” In Proc. of the 10th ACM CCS, Washington, Oct., 2003.

[11] W. Du, J. Deng, Y. Han, S. Chen and P. Varshney, “A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge,” IEEE INFOCOM04, Hongkong, 2004.

[12] S. Zhu, S. Xu, S. Setia, and S. Jajodia, “Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach,” In Proc. of ICNP03, Atlanta, Nov., 2003.

[13] K. Ren, K. Zeng and W. Lou, “A new approach for random key predistribution in large scale wireless sensor networks”, to appear in Journal of Wireless Communication and Mobile Computing.

[14] 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,” IEEE S&P, Oakland, CA, May 2004.

[15] F. Ye, H. Luo, S. Lu, and L. Zhang, “Statiscal Enroute Filtering of Injected False Data in Sensor Networks,” IEEE Infocom, Mar. 2004.

[16] H. Yang, F. Ye, Y. Yuan, S. Lu and W. Arbaugh, “Toward Resilient Security in Wireless Sensor Networks,” In Proc. of ACM MOBIHOC’05, 2005.

[17] Y. Zhang, W. Liu, W. Lou, and Y. Fang, “Location based security mechanisms in wireless sensor networks,” to appear in IEEE JSAC, Special Issue on Security in Wireless Ad Hoc Networks.

[18] “TinyOS Operation System,” http://millennium.berkeley.edu.

[19] Harald Vogt, “Exploring Message Authentication in Sensor Networks‘,” Proceedings of ESAS 2004, Springer-Verlag, Aug. 2004.

[20] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors,” In Proc. of ASPLOS IX, 2000.

[21] X. Cheng, A. Thaeler, G. Xue, D. Chen, “TPS: A Time-Based Positioning Scheme for Outdoor Wireless Sensor Networks,” IEEE INFOCOM, 2004.

[22] L. Hu, D. Evans, “Localization for Mobile Sensor Networks,” In Proc. of ACM MOBICOM, 2004.

[23] Y. Zhang, W. Liu, Y. Fang, and D. Wu, “Secure localization and authentication in ultra-wideband sensor networks,” to appear in IEEE JSAC, Special Issue on UWB Wireless Communications - Theory and Applications.

[24] A. Shamir, How to Share a Secret, Communications of the ACM, 22(11):612-613, Nov 1979

[25] D. Estrin, A. Sayeed and M. Srivastava, “Wireless Sensor Networks,” MobiCom 2002 tutorial.

[26] J. Jung, T. Park and C. Kim, “A Forwarding Scheme for Reliable and Energy-efficient Data Delivery in Cluster-based Sensor Networks,” IEEE Communication Letters, Vol.9, No.2: 112-114, Feb. 2005.