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

Kui Ren, *Member*, *IEEE*, Wenjing Lou, *Member*, *IEEE*, and Yanchao Zhang, *Member*, *IEEE*

Abstract—Providing desirable data security, that is, confidentiality, authenticity, and availability, in wireless sensor networks (WSNs) is challenging, as a WSN usually consists of a large number of resource constraint sensor nodes that are generally deployed in unattended/hostile environments and, hence, are exposed to many types of severe insider attacks due to node compromise. Existing security designs mostly provide a hop-by-hop security paradigm and thus are vulnerable to such attacks. Furthermore, existing security designs are also vulnerable to many types of Denial of Service (DoS) attacks, such as report disruption attacks and selective forwarding attacks and thus put data availability at stake. In this paper, we seek to overcome these vulnerabilities for large-scale static WSNs. We come up with a location-aware end-to-end security framework in which secret keys are bound to geographic locations and each node stores a few keys based on its own location. This location-aware property effectively limits the impact of compromised nodes only to their vicinity without affecting end-to-end data security. The proposed multifunctional key management framework assures both node-to-sink and node-to-node authentication along the report forwarding routes. Moreover, the proposed data delivery approach guarantees efficient en-route bogus data filtering and is highly robust against DoS attacks. The evaluation demonstrates that the proposed design is highly resilient against an increasing number of compromised nodes and effective in energy savings.

Index Terms—Data security, wireless sensor network, end-to-end, DoS attack, false-data injection attack.

1 Introduction

Wireless sensor networks (WSNs) have attracted a lot of attention recently due to their broad applications in both military and civilian operations. WSNs usually consist of a large number of ultrasmall low-cost battery-powered devices that have limited energy resources, computation, memory, and communication capacities [1], [2], [4], [7], and according to different 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 multihop 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 cryptanalysis. Hence, this type of attack could lead to severe data confidentiality compromise in WSNs. Furthermore, the attacker may use compromised nodes to inject bogus data traffic in WSNs. In such attacks, compromised nodes pretend to have detected an event of interest within their vicinity or simply fabricate a bogus event report claiming a nonexisting event at an arbitrary location. Such *insider* attacks can severely damage network function and result in the failure of mission-critical applications. Such attacks 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 also use compromised nodes to launch a selective forwarding attack [3], in which case compromised nodes selectively drop the going-through data traffic and, thus, data availability can be severely damaged. The existence of the aforementioned attacks together with the inherent constraints of sensor nodes make it rather challenging to provide satisfying data security in WSNs with respect to all its three aspects, that is, confidentiality, authenticity, and availability [1], [2], [3], [4], [5].

Recent research has seen a growing body of work on security designs for WSNs [6], [8], [9], [10], [12], [13], [14], [15], [18], [19], [21], [36], [37], [38]. 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, particularly in a WSN consisting of a large number of nodes [7]. However, lack of the end-to-end security guarantee could make WSNs particularly vulnerable to the aforementioned attacks in many applications where node-to-sink communication is the dominant communication pattern [32], [33], [34]. This could give the attacker the advantage to obtain/manipulate its desired data using much less effort without

- *K. Ren is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, 3301 Dearborn St., Chicago, IL 60616. E-mail: kren@ece.iit.edu.*
- *W. Lou is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, 100 Institute Rd., Worcester, MA 01609. E-mail: wjlou@ece.wpi.edu.*
- *Y. Zhang is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102. E-mail: yczhang@njit.edu.*

Manuscript received 27 Dec. 2005; revised 19 Mar. 2007; accepted 15 Aug. 2007; published online 4 Sept. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0379-1205.
Digital Object Identifier no. 10.1109/TMC.2007.70753.

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 will be discussed later.

In this paper, we propose an integrated security design providing comprehensive protection over data confidentiality, authenticity, and availability. Our design establishes a location-aware end-to-end data security (LEDS) framework in WSNs. The contributions of LEDS are outlined below.

First, we propose a novel location-aware multifunctional key management framework. In LEDS, the targeted terrain is virtually divided into multiple cells using the concept of a *virtual geographic grid*. Each sensor node obtains its geographic location via a suitable localization scheme such as [11], [16], [17], and [30]. 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 efficiently 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 a security compromise of that event report, which is usually the case in existing security designs.

Third, LEDS possesses an 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. Effective en-route filtering of bogus data packets also results in significant energy savings as unnecessary forwarding is eliminated.

Last, LEDS provides high-level assurance on data availability by dealing with both report disruption attack [21] and selective forwarding attack [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 single node(s). Thus, LEDS is highly robust against selective forwarding attack as compared to the traditional one-to-one forwarding approach used by existing security designs [18], [19], [21]. In addition, LEDS also adopts a (t, T) threshold linear secret sharing scheme (LSSS) [25] so that the sink can recover the original report from any t out of T legitimate report shares. This approach, on one hand, enhances the authenticity of the event report by requiring collaborative endorsement from T different sensing nodes but, on the other hand, 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 and partly contributes to the reduction of energy waste due to the incorrect dropping of legitimate data reports.

The rest of this paper is structured as follows: Section 2 articulates the data security goals in WSNs and evaluates related work with respect to these goals. Section 3 details the proposed LEDS design. Section 4 presents the detailed security analysis of the proposed LEDS, followed by the performance analysis in Section 5. Finally, the conclusion is drawn in Section 6.

2 Data Security Requirements in WSNs and Related Work

2.1 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], [23]. 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, which may vary depending on different applications, usually appear as event reports sent by the sensing nodes from event happening area via multihop paths to the sink. As the communication range of sensor nodes is 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 follows: 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 assure data authenticity in addition to confidentiality. Since the undetected compromised node(s) can always send false reports, cryptography cannot fully prevent such attacks. However, if we require a valid report to be collectively endorsed by a number, say, $T(T > 1)$, of sensor nodes who sense the event at the same time, we can protect data authenticity to the extent that no fewer than T compromised nodes can forge a valid report.

Furthermore, by exploiting the static and location-aware nature of WSNs, we can furthermore require that a legitimate event report corresponding to certain area can only be generated by the collaborative endorsement of no less than T nodes of that area. That is, to generate a valid

report on a nonexisting event happening at a certain area, the only way is to compromise T nodes at that area, and is otherwise impossible.

Data Availability. As compromised nodes are assumed to be existing in WSNs, it is important to prevent or be tolerant to their interference as much as possible to protect data availability. In this regard, security designs should be as robust as possible in the presence of compromised nodes. In-network processing such as false data filtering is important to save scarce network resources and to prolong network lifetime. To this end, any security design in WSNs should be highly resilient against two types of DoS attacks: report disruption attack [21] and selective forwarding attack [3], in which compromised nodes purposefully drop legitimate packets to disrupt the event report service by taking advantage of the en-route-filtering policy.

2.2 End-to-End versus Hop-by-Hop Design

In the past few years, many secret key predistribution schemes have been proposed [6], [8], [9], [10], [12], [13], [14], [15], [29]. By leveraging preloaded keying materials on each sensor node, these schemes establish pairwise keys between a node and its neighbors after network deployment for every network node, respectively, and thus form a hop-by-hop security paradigm. The security strength of these schemes is analyzed in terms 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, according to key distribution information available to the attacker. Then, to compromise the whole network communication, the attacker is forced to capture at least several hundreds of sensor nodes even under selective node capture attack. Hop-by-hop security design works fine when assuming a uniform wireless communication pattern in WSNs. However, in many applications, node-to-sink communication is the dominant communication pattern in WSNs, that is, data of interest are usually generated from the event happening area and transmitted all the way to the sink. In this case, hop-by-hop security design is not sufficient anymore as it is vulnerable to communication-pattern-oriented node capture attacks. Data confidentiality can be easily compromised due to lack of end-to-end security guarantee, since compromising any intermediate node will lead to exposure of the transmitted data. At the meantime, 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. We therefore conclude that end-to-end security design is much more desirable for WSNs as compared to hop-by-hop design when node-to-sink communication is the dominant communication pattern as it can offer a much higher security resilience.

2.3 Existing Data Report Security Designs in WSNs

The general approach adopted to protect data authenticity in WSNs is given as follows: To generate a valid report, T ($T > 1$) nodes that sense the event 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 past two years, a few schemes have been proposed to design suitable key management schemes based on this approach, including Statistical En-Route Filtering (SEF) [19], Interleaved Hop-by-Hop Authentication (IHA) [18], and Location-Based Resilient Secrecy (LBRS) [21]. 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 enables 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 nonexisting event happening in a certain area, the only way is to compromise T nodes at that area, and is otherwise impossible. Therefore, there is still a gap between the protection that existing schemes can offer and the requirement of data authenticity.

In addition, all three schemes mentioned above are highly vulnerable to report disruption attack and selective forwarding attack. A single compromised node may disrupt the event report service originating in its vicinity or passing through it. Once a node in a certain area is compromised, the attacker can disrupt any event report from that area from being forwarded to the sink thereafter by simply contributing a wrong MAC to the final report. Since the en-route filtering allows intermediate nodes to drop packets with false MACs, such reports will be rejected on their way to the sink because of the presence of the wrong MAC(s). On the other hand, 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 due to failing to pass the endorsement verification. As the number of compromised nodes increases, the resulting damage will increase drastically, as discussed later in Section 5. Hence, data availability in these schemes is poorly assured. The scheme presented in [29] is a group key predistribution method that can serve as a base for designing secure event report delivery approaches.

3 LEDS: Location-Aware End-to-End Data Security Mechanism

3.1 Assumptions, Threat Model, and Design Goals

System Assumptions. In LEDS, we consider a large-scale uniformly distributed WSN that monitors a vast terrain of interest via a large number of static sensor nodes, which can be deployed via approaches such as aerial scattering. We assume that an approximate estimation on the size and shape of the terrain of interest is known a priori. Once deployed, each node is assumed to be static and can obtain its geographic location via a secure and suitable localization scheme such as [11], [16], [17], [30], and [31]. The network deployment guarantees that the established WSN is well connected and dense enough to support fine-grained collaborative sensing and be robust against node lost and failure. We assume that each event of interest can be detected by multiple sensor nodes [18], [19], [21]. 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 (for example, the Berkeley MICA motes [24]) in its computation and communication capability 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 furthermore assume that, if the node is compromised, all the information it holds will also be compromised. However, the sink is assumed to be secure as it is usually well protected and under the direct control of the network owner [19]. 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. On the other hand, we assume that there is a short bootstrapping phase right after network deployment during which no sensor node is compromised.

Design Goals. LEDS seeks to provide end-to-end data security, as well as en-route bogus data filtering in WSNs. In particular, we focus on the data such as event reports that are generated by the sensing nodes and transmitted from the sensing area to the sink. More specifically, the design of LEDS aims to achieve the following goals:

- *Provide end-to-end data confidentiality and authenticity.* Both the confidentiality and authenticity of data 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 compromised nodes.
- *Achieve a high-level of assurance on data availability.* 1) Be resilient against report disruption attacks and selective forwarding attacks and 2) be able to early detect and drop bogus reports in an effective and deterministic manner, that is, having en-route-filtering capability.
- *Realize all the security goals in a single integrated design without relying on any other security infrastructures.* Be simple and efficient while providing an end-to-end security guarantee and have low computation and communication overheads for it to be suitable in WSNs.

3.2 Notation and Terms

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

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

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, 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. For simplicity, we assume that there is only one static sink in the WSN. The size of a cell is defined by l, which is the side length of the cell. A cell is uniquely indexed by its center's location. Thereafter, when we refer to the location of a cell, we use its center's location for convenience.

Home cell, event cell. The cell that a node, say, u, is located in after network deployment is called the *home cell* of u, denoted as I_u, and $I_u = (x_1, y_1)$ when its location is (x_1, y_1). We call a cell an *event cell* when a certain event of interest happens in that cell. Each report is therefore corresponding to one particular *event cell*.

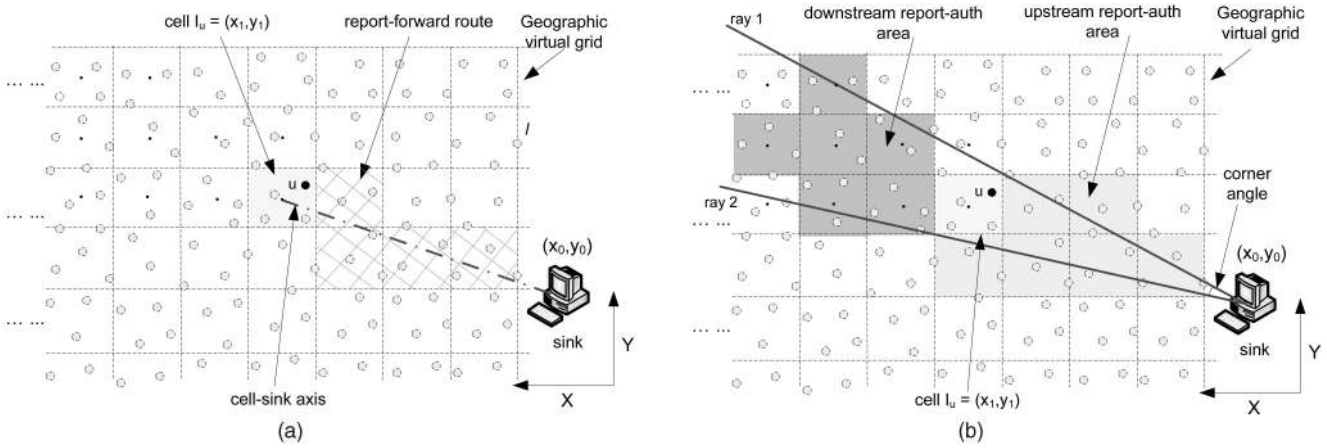


Fig. 1. Term illustration: defined for node u.

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] toward the sink. More specifically, a report is always sent from one cell to one of its four adjacent cells that is closest to the sink.[2] 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 (as shown in Fig. 1a). These cells are sequenced according to their distances to the sink. The cell that a report travels first ranks first and so on.

Report-auth area. The *report-auth area* of a node u consists of two parts, the *downstream report-auth area* and the *upstream report-auth area.* They are both defined with regard to a sector area that is bound by two rays. Each of these two rays starts from the sink (x_0, y_0) and goes through one vertex of cell I_u and the two rays form the smallest angle that contains I_u (as shown in Fig. 1b). Then, the *downstream report-auth area* of u is defined to be all the cells that are farther to the sink than I_u, and each has at least half a part located inside the sector area, whereas the *upstream report-auth area* consists of all the cells that are closer to the sink than I_u and have any part that falls into the sector area. Obviously, the *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 it belongs to u's *report-auth area*, and every 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 an *upstream report-auth cell* of u. Otherwise, it is a *downstream report-auth cell* of u.

These terms are graphically illustrated in Fig. 1.

3.3 Scheme Overview

LEDS follows the interleaved hop-by-hop forwarding and filtering approach as in [18] and [19] and adopts the cell-based report generation methodology similar to that in [21]. The proposed LEDS scheme consists of two major components: One is the underlying key management framework and the other is the corresponding end-to-end data security mechanism.

Location-aware key management framework. In LEDS, each node stores three different types of location-aware keys: 1) A *unique secret key* shared between the node and the sink that is 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. 3) A set of *authentication key*s shared with the nodes in its *report-auth cell*s that are used to provide both cell-to-cell authentication and en-route bogus data filtering. 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 seeks to protect data reports in a comprehensive and end-to-end manner. **Data confidentiality:** In LEDS, every event report is encrypted by the corresponding *cell key* of the *event cell.* As the *cell key* is solely shared among nodes of the *event cell* and the sink, the confidentiality of the report is guaranteed as long as no node in the *event cell* is compromised. **Data authenticity**: 1) Each report is endorsed by multiple sensing nodes, and the endorsements can be individually authenticated by the sink. 2) Each report is also authenticated in an interleaved cell-by-cell manner along the report-forwarding route. **Data availability**: 1) Be robust against *report disruption attack*s: The encrypted report is divided into a number of unique shares through a predefined LSSS. Each share is independently generated by a participating node using its *unique secret key* shared with sink. A predefined number of MACs are then computed over all the shares using cell-to-cell *authentication key*s as another layer of endorsements, which enables the intermediate nodes to perform en-route filtering. 2) Be robust against *selective forwarding attacks*: Using cell-to-cell *authentication key*s guarantees that each report can be verified simultaneously by multiple next-hop nodes at any point in the route. This unique feature of LEDS makes it possible for the one-to-many data forwarding approach to be used in LEDS instead of the vulnerable one-to-one approach adopted by most

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

2. In the case that two adjacent cells have the same distance to the sink, an agreement to solve the tie needs to be predefined. For example, one may pick the cell that has a smaller x-coordinate. The purpose is to guarantee that the route precomputed at the node would be the same as the actual route a report travels in a distributed cell-by-cell manner.

existing security schemes. Sink finally verifies whether the report is indeed sent by the nodes from the *event cell* as claimed through examining both the authenticity of the MACs and the uniqueness of the shares. The sink can always recover the report from a subset of the shares even if a small number of wrong shares exist due to the threshold property of the underlying LSSS.

3.4 Protocol Detail

3.4.1 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. Based on the total number of nodes in the network N, cell size l, and the average number of nodes in each cell n', the network planner further decides the values of T and t: The former is the number of endorsements included when generating a valid report, and the latter defines the minimum number of correct endorsements to validate a report. The impact of different values of these parameters will be discussed in Sections 4 and 5 when we analyze security strength and performance of LEDS. The network planner also prepares two master secret keys, K_M^I and K_M^{II}. In addition, a large prime number p is prepared, which, together with t and T, defines a (t, T) LSSS over finite field $GF(p)$.

LEDS adopts a robot-assisted network bootstrapping technique [31]. We assume that a group of mobile robots are dispatched to sweep across the whole sensor field along preplanned routes after the deployment of sensors. Mobile robots have GPS capabilities, as well as more powerful computation and communication capacities than ordinary sensors. The leading robot is also equipped with the following bootstrapping parameters:

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

The robots securely localize every sensor using the secure localization protocol given in [16] and load each of them with the corresponding location-aware keys in a cell-by-cell manner.

Specifically, the robots first determine a node u's *home cell* $I_u = (x_1, y_1)$, and then compute a *unique secret key* K_u, which u shares with the sink as

$$K_u = H(K_M^I | u | I_u),$$

where $|$ denotes concatenation operation. A *cell key* K_{I_u} is further calculated, which is shared among u and other nodes in $I_u(x_1, y_1)$, and

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

The robots load u with K_{I_u}, as well as the ID list of all the nodes in I_u.

The robots next compute a set of *authentication key*s for all the sensors in the same cell. An *authentication key* is shared among all the sensors in a given cell and its corresponding *report-auth cell*s. Supposing a *report-auth cell* of I_u has its location as (x_c, y_c), then the *authentication key* between the two cells is

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

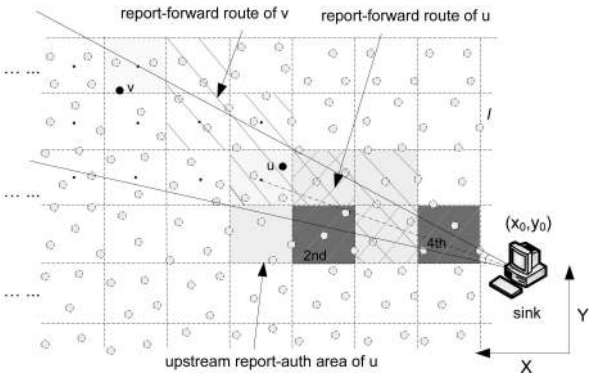


Fig. 2. Illustration of *report-auth cell*s of node u.

The *report-auth cell*s of I_u are determined according to I_u's relative location with respect to the sink. Specifically, a member of the *downstream report-auth cell*s of u is any cell in its *downstream report-auth area* that is no more than $T + 1$ cells away from I_u.[3] For example, all the gray cells shown in Fig. 1b are u's *downstream report-auth cell*s with $T = 3$. On the other hand, cell I_v is not such a member because only horizontal or vertical cell transversing is allowed in LEDS, that is, no diagonal cell transversing is allowed, and hence, I_v is five cells away from I_u. The quantitative analysis on the number of *downstream report-auth cell*s of a node will be discussed in Section 5 in the context of key storage overhead analysis.

Furthermore, the *upstream report-auth cell*s of u comprise of the following ones: The robots first randomly rank all the sensors in I_u, assigning each of them a rank between 1 and T. Supposing u is assigned a rank as $rank_u$, then the $(rank_u \bmod (T + 1))$th cell in the *report-forward route* of u is the first of such a cell. The remaining ones for u are those cells within its *upstream report-auth area* that are exactly $T + 1$ cells closer to the sink as compared to I_u. In case I_u is less than $T + 1$ cells away from the sink, the sink itself is chosen. An example is shown in Fig. 2. Supposing $T = 3$, then the second and fourth cells denoted in the figure are u's *upstream report-auth cell*s.

In fact, for any two nodes u and v, if I_v is a member of the *downstream report-auth cell*s 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 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 *upstream report-auth area* of u is a part of that of v, that is, the *report-forward route* of v falls into the *upstream report-auth area* of u after the route reaches I_u, as shown in Fig. 2.

The robots also load every sensor with $\{(t, T), p\}$. The same bootstrapping procedure is repeated for all nodes in every cell. Note that the robots may also need to relocate a small number of sensors to ensure that each cell contains no less than T nodes. The communication between the sensors and the leading robot can be easily secured using the technique introduced in [28]. We omit it here for due to space limitations. By the end of the bootstrapping phase,

3. Adjacent cells are considered one cell away.

mobile robots leave the sensor field, and the leading robot should securely erase all the keys from its memory but should report the locations of the sensors to the sink. 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. We further note that the above bootstrapping operation can also be realized through the key predistribution approach [6], [12], instead of using mobile robots. In this case, sensor nodes utilize secure localization protocols [11], [17], [31] to obtain their locations. The choice of the approaches could depend on the security conditions and their availabilities in practice. We further point out that some sensors may be dislocated during the network lifetime in many scenarios. In this case, once they are dislocated, their possessed keys should also be updated according to their new location. Such dislocation and update operations can also be fulfilled by using the mobile robots.

3.4.2 End-to-End Data Security Mechanism

LEDS requires each valid event report to be encrypted and, at the same time, attached with T endorsements from T different nodes when generated from the *event cell*. Although an event report is relayed to the sink, the intermediate nodes will drop any invalid endorsements to the report. Moreover, the report itself will be dropped when the number of valid endorsements becomes less than t. This is in contrast to the existing designs in which a report is dropped as soon as an invalid endorsement is found. The proposed design is important as it makes the system more robust in that it tolerates up to $T - t$ compromised nodes in an *event cell* colluding to launch a *report disruption attack* by contributing invalid endorsements to the legal event reports. Meanwhile, the requirement of multiple endorsements makes the system more reliable by disabling the possibility that up to $t - 1$ compromised nodes of an *event cell* or an unlimited number of compromised nodes from any other cell(s) collude to forge a report of events "appearing" at that *event cell*. The encryption prevents an unlimited number of compromised nodes not in the *event cell* from colluding to obtain the content of the reports. LEDS further adopts a one-to-many report-forwarding paradigm, which ensures that the system is being highly resilient to selective message forward attacks [3]. The detailed security mechanism is described as follows.

Report generation. Each of T participating nodes first agree on an event report M using the technique introduced in [20] based on signal strength strategy. M usually contains information such as event type, sensing location (that is, the *id* of *event cell*s), and a time stamp, etc. Note that all the related communications are protected by the *cell key* so that M is confidential against any outside node. Next, each participating node, say, u, encrypts M using the cell key K_{I_u} and obtains $C = E_{K_{I_u}}(M)$. u further computes a unique share C_u of C through the predefined (t, T) LSSS. Specifically, C_u is obtained by evaluating the following univariate polynomial of degree $t - 1$ over finite field $GF(p)$ using K_u:

$$C_u = \mathcal{F}(K_u) = \sum_{0 \le i < t} a_i K_u^i \bmod p, \tag{1}$$

where a_i $(i = [0, t-1])$ is 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. This is because the polynomial is evaluated using u's unique secret key K_u, which is only known to u and the sink. Node u then broadcasts tuple $\{u, C_u\}$ and also collects the corresponding $T - 1$ shares from other nodes. u then computes two MACs over all the T shares of C, that is, C_{share}, as another layer of endorsement to the report, which enables the intermediate nodes to perform en-route 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 u's two *upstream report-auth cell*s and both of them belong to u's *report-forward route*, in which I_o ranks $(T+1)$th. Then, the obtained MACs are $Mac_{K_{I_u,I_v}}(C_{share})$ and $Mac_{K_{I_u,I_o}}(C_{share})$. The tuple $\{u, Mac_{K_{I_u,I_v}}(C_{share}), Mac_{K_{I_u,I_o}}(C_{share})\}$ is then broadcast to complete the synthesis of the final report. Node u constructs and sends out the final report after it collects $T + 1$ different MACs and $2T$ MACs in total. The final report contains

1. an *event cell id*,
2. the *id*s of T participating nodes,
3. a 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 an order based on the node ranks (The common MAC is listed last). The report is sent by the node that completes the synthesis of the report and seizes the channel first. To avoid sending duplicate reports, each node overhears the channel and uses exactly the same random timer technique described in [21] and [27].

Interleaved cell-by-cell en-route filtering. In LEDS, data reports are relayed cell by cell and delivered following a 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 in its route-forward route. As we mentioned before, it is easy to determine the next cell on the report-forward route, which is the one that is adjacent to the sending cell and is closer 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 like back off before sending [21], [27].

In LEDS, an appropriate intermediate node authenticates a received report by checking 1) the validity of the first MAC attached in the report and 2) the number of nonzero MACs. The node verifies the first MAC attached in the report by using the corresponding *authentication key*:

- If the first MAC is zero, it deletes it and attaches another zero to the next to the end of the report.[4]
- If the first MAC is valid, it deletes it and attaches a new MAC to the next to the end of the report.
- If the first MAC is invalid, it deletes it and attaches a zero to the next to the end of the report.

4. That is, always keeps the common MAC the last.

TABLE 1
Pseudocode for Authenticating a Received Event Report

```
1   verify the 1st MAC contained in the report;
2   if (the 1st MAC is zero or invalid)
3       newMAC = 0;
4   if (the 1st MAC is valid)
5       newMAC = createMAC(C_share, key);
6       delete the 1st MAC;
7       attach newMAC to the next to the end of the report;
8   get number of different non-zero MACs;
9   if ((((j ≤ T − t)&&(Num_of_MAC < T − j + 2)) ||
10      (Num_of_MAC < t + 1))  // the event cell is j cells away
11       discard report; // not enough
12  else
13      forward report to the next cell; // enough
```

Here, the newly attached MAC is computed over C_{share} using the corresponding *authentication key* shared between the node and one of its *upstream report-auth cell*s, which is exactly $T+1$ cells closer to the sink with respect to its *report-forward route*.

The node also checks whether or not the number of nonzero MACs is enough and discards the report if the number is not enough. The number of nonzero MACs is considered not enough by an intermediate node if 1) it contains less than $t+1$ different nonzero MACs or 2) it contains less than $T-j+2$ different nonzero MACs, when an *event cell* is j cells ($j \in [1, T-t]$) away from its own. If there are enough number of nonzero MACs, the node now forwards the processed report to the next cell. Note that there is no way for a single node to launch selective forwarding attack, since each report can be verified by multiple nodes simultaneously. Every node in the same cell can be the one to forward a legal report. The pseudocode of the above authentication procedure is shown in Table 1.

Sink verification. A report is verified at the sink in two aspects to ensure its authenticity: 1) It verifies whether the report contains no less than $t+1$ valid nonzero MACs, and 2) it checks whether the report is indeed endorsed by the T nodes as claimed. Sink verifies 1) using the *authentication key*s it shares with the intermediate cells and checks and 2) by recovering the report C from C_u. To do this, it tries to recover C from any t correct shares and then decrypts the recovered C using the corresponding *cell key* of *event cell*.[5] More specifically, the recovery operation of M goes as follows: sink picks t out of T shares, using their corresponding secret keys,[6] sink solves a t-variable linear equation system to get a_i, $i = [0, t-1]$ in (1) and thus obtains C, and sink further decrypts C and gets M. At this point, if M is meaningful (that is, conforming to the predefined 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. Moreover, as long as the sink can recover the original report M, it may ascertain that all the corresponding shares are indeed generated by the nodes as claimed.

5. Based on the cell *id* contained in the report.
6. Based on the node *id* contained in the report.

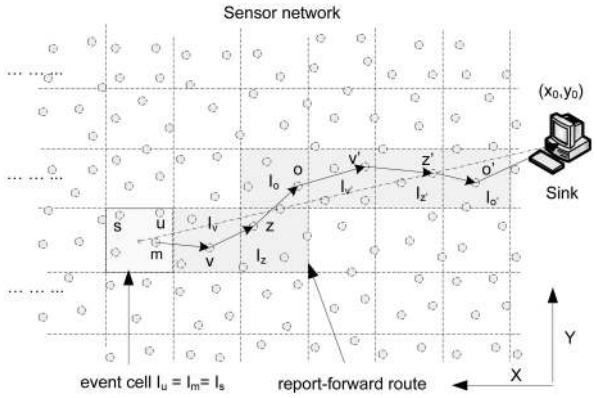


Fig. 3. An example of the proposed end-to-end data security mechanism.

3.5 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 an *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 nonzero 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. Last, node v forwards the processed report:

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

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

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

Moreover, sink receives the report as

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

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 (1), sink solves a 2-variable linear equation system and thus obtains C. Sink furthermore decrypts C using K_{I_u} and therefore obtains M. If M is meaningful, the recovery operation succeeds. Sink would

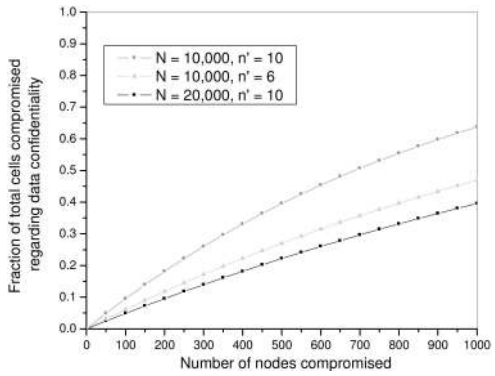


Fig. 4. Data confidentiality in LEDS under random node capture attack.

not be able to recover M if there are more than $T - t = 1$ invalid shares. Hence, as long as sink could recover the report, it accepts the report.

4 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, that is, data confidential, authenticity, and availability.

4.1 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 would 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 the total network cells?

Random node capture attack. Given 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 given cell is compromised is

$$1 - \frac{\binom{N-n'}{x}}{\binom{N}{x}}. \tag{2}$$

On the other hand, (2) also represents the fraction of total cells that are compromised given that 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 percent of the total cells, at least 5 percent 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 performs fairly well with respect to security resilience against random node capture attacks when compared with existing security designs [8], [12], [13].

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 that at least $\frac{N}{n'}$ nodes are required, that is, around 1,000 nodes, given $N = 10{,}000$ and $n' = 10$. Note that this is 10 percent of the total network nodes.

4.2 Security Strength of LEDS Regarding Data Authenticity

In addition to obtaining the content of legitimate reports, the attacker may also want to insert bogus reports to fool the sink with nonexisting 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 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 provide both node-to-sink and cell-to-cell authentication. Therefore, LEDS presents an improvement over existing security designs such as SEF, IHA, and LBRS [18], [19], [21], in which compromising any single node would result in multiple gains, that is, 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 attack, if the number of compromised nodes is x, then the probability that a cell is not affected, that is, no node in a cell is compromised, is given by

$$P_{\{0\}} = \frac{\binom{N-n'}{x}}{\binom{N}{x}}. \tag{3}$$

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, letting $P_{\{i\}}$ 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, that is, having at least T compromised nodes is

$$P_{\{\geq t\}} = \sum_{i=t}^{n'} P_{\{i\}} = \sum_{i=t}^{n'} \frac{\binom{n'}{i}\binom{N-n'}{x-i}}{\binom{N}{x}}. \tag{4}$$

This also represents the percentage of authenticity *compromised* cells. Then, the percentage of *affected* cells, that is, each of which has at least 1 and at most $t - 1$ compromised nodes, can be expressed as $1 - P_{\{0\}} - P_{\{\geq t\}}$. Fig. 5 illustrates how data authenticity is affected as the number of

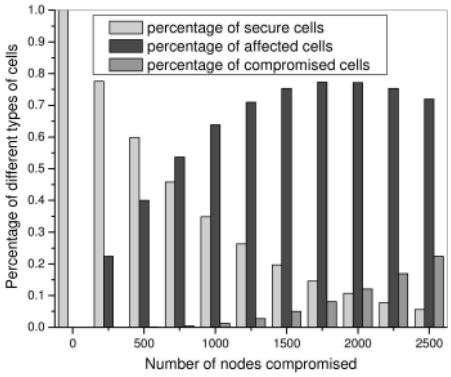


Fig. 5. Data authenticity in LEDS under random node capture attack, where $N = 10{,}000$, $n' = 10$, and $(t, T) = (4, 5)$.

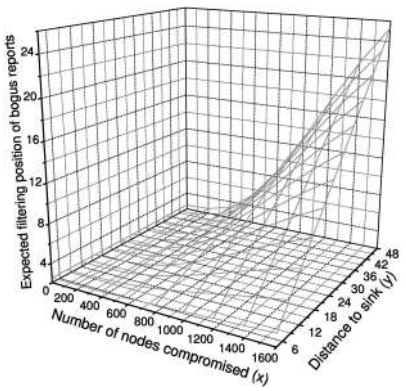


Fig. 6. Expected filtering position versus the number of compromised nodes with respect to different distance to the sink.

compromised nodes increases. It is observed that the percentage of compromised cells increases very slowly with the increase of a number of compromised nodes. Moreover, it is kept very low: Even if the compromised nodes reach 1,750, only 10 percent 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 the undetectable bogus reports. On the other hand, it is observed that the percentage of secure cells in the network deceases slowly, whereas the percentage of affected cells increases quickly as the number of compromised nodes increases. This observation tells us that it is relatively easier for the attacker to insert the bogus reports into the network; however, these bogus reports can be deterministically filtered by the intermediate nodes or the sink.

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* being affected simultaneously, assuming 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 to have enough of them. Moreover, to let pass these $t - x_2$ invalid MACs, 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. Otherwise, any bogus report from this cell will be filtered at most at Tth cell and $\frac{T}{2}$th cell on the 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 originated from this cell will be filtered out at most at the Tth cell along the route. Under a random node 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 Tth cell. Therefore, given the number of compromised nodes as x, the expected filtering position of the bogus reports from an affected cell is bounded by

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

suppose 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 $(t, T) = (4, 5)$. It is clearly shown in Fig. 6 that the bogus reports sent from the most affected cells can be efficiently filtered under random node capture attack. For example, the bogus reports from an affected cell that is 30 cells away from the sink will be filtered at less than the 10th cell in the route on the average, where the number of compromised nodes is 1,000.

On the other hand, under selective node capture attack, 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 [18], [19], [21], compromised nodes from one cell in LEDS cannot be used to compromise data authenticity of other cells. Note that, in existing security designs, the 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.

4.3 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 [18], [19], [21]. In contrast, LEDS makes a 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.

On 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

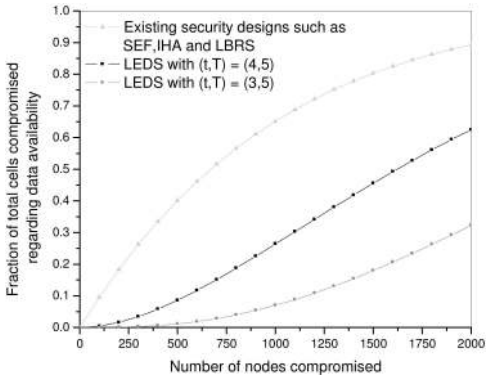


Fig. 7. Data availability in LEDS under report disruption attack.

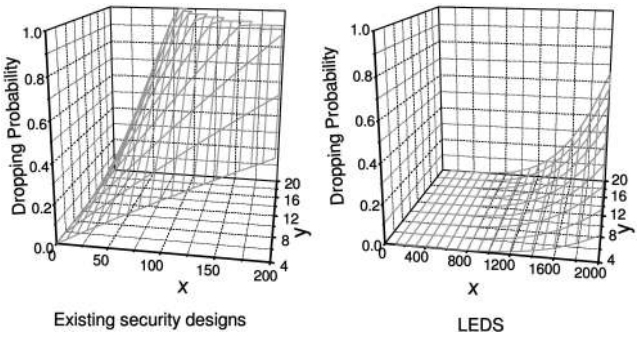


Fig. 8. Data availability in LEDS under selective forwarding attack.

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 mean time, the intermediate nodes only discard a report that contains fewer than t valid MACs. That is, if there are up to $T-t$ compromised nodes that contribute invalid MACs, the report can still be relayed to the sink. 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 a 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 $1-P_{\{0\}}$. Furthermore, the percentage of cells that have at least $T-t+1$ nodes compromised, respectively, is given by

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

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 attacker needs to compromise a lot more nodes to successfully launch report disrupt attacks in LEDS. Given $N=10{,}000$, $n'=10$, and $(t,T)=(4,5)$, to successfully launch a report disrupt attack in 10 percent of the 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, make the attack even harder, as shown in Fig. 7. Last, even under selective node capture attacks, the cost to successfully launch a 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 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 functions in 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, applying the 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

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

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

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

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

5 Performance Analysis of LEDS

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

5.1 Key Storage Overhead

In LEDS, each node stores a *unique secret key* that is 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 in addition. Furthermore, each node also stores one *authentication key* for each of its *report-auth cell*s. For a particular node, say, u, the number of its *report-auth cell*s is decided by u's relative position with respect to the sink.

More specifically, the number of *downstream report-auth cell*s of u is bounded by $\frac{(T+1)(T+2)}{2}$, when *home cell* I_u is right next to the sink, as shown in Fig. 9a. 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. 9b. Obviously, in a two-cell-wide band area,

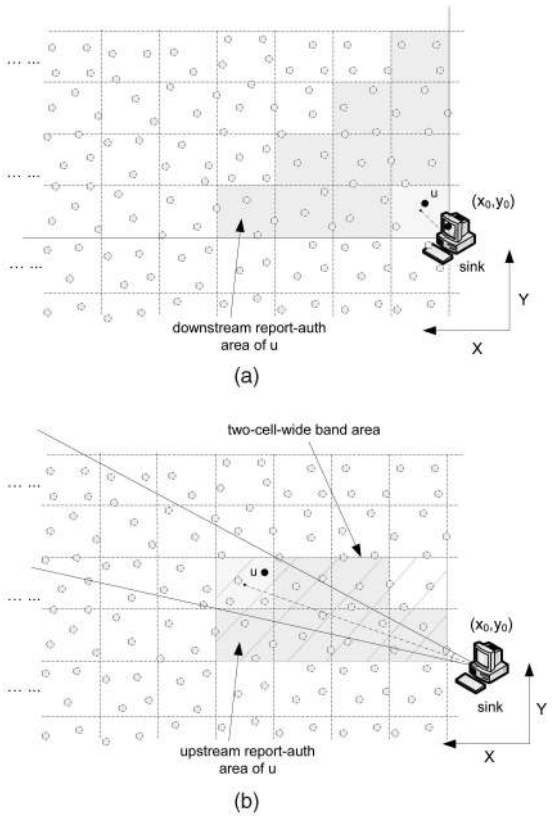


Fig. 9. The upper bound of the number of *report-auth cell*s.

all the possible routes monotonically toward the sink[7] 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 two different choices. Hence, the number of *upstream report-auth cell*s of any node is bounded by 3, and the total number of keys stored by each node in LEDS is bounded by

$$\frac{(T+1)(T+2)}{2}+5. \tag{9}$$

Therefore, LEDS only requires the nodes to store a small number of keys, which can be as low as 20, given $T=5$. Moreover, the number of keys is independent of3 the network size, which makes LEDS highly suitable in large-scale WSNs. Furthermore, the sink also stores very few keys in LEDS, that is, 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 (that is, *cell id*) contained in the received data reports.

5.2 Computation and Communication Overheads

In LEDS, key establishment only involves efficient hash operations during the bootstrapping period. Moreover, since the authentication keys are shared in a cell-to-cell manner, they can be reused for en-route filtering during whole network life. This feature saves a lot of unnecessary computation due to key reestablishment. On the contrary, whenever forward route changes, all the authentication keys should be reestablished to enable en-route filtering as in IHA [18] 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 [25]. 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 highly efficient. In addition, to judge whether a node belongs to a particular *report-forward route*, only simple geometry computation is involved based on a *geographic virtual grid*.

The communication overhead of our scheme arises 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 trade-off 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 bytes (288 bits) long as in TinyOS [22] and $(t,T)=(4,5)$, then each share will be 9 bytes in length and there will be five shares in total according to the underlying LSSS. Hence, the size of an additional message overhead is only $1/4$ of the original message length, that is, 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 predefined 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 [18], [19], [21]. The elimination of unnecessary routing overheads also helps LEDS be communication efficient.

5.3 Energy Savings

Existing security designs only aim to save the energy of intermediate nodes along the forwarding path to the sink through early detection and dropping of bogus data reports inserted by compromised nodes. However, compromised nodes may also intentionally drop legitimate reports and thus cause futile energy consumption, which implies extra energy waste. To address this problem, LEDS aims to reduce the energy waste resulting from both bogus data report insertion and legitimate report dropping. On the other hand, in doing so, the introduced message overhead and en-route-filtering operations inevitably incur extra energy consumption in both communication and computation.

In the following, we employ a similar model to that in [19] to analyze the energy savings caused by the proposed LEDS. We denote by E_{tr} the energy consumption for transmitting and receiving one bit by L_n, the bit-length of an original report without using LEDS, by L_a, the bit-length of

7. This involves horizontal and vertical cell transverse only.

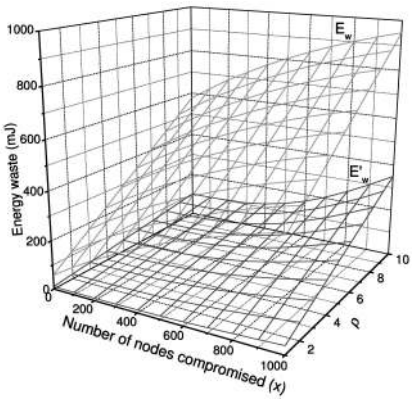


Fig. 10. Energy waste due to node compromise under different bogus traffic ratio.

a report with LEDS, and by h, the average number of cells a report travels. We further assume that the ratio of legitimate data traffic to bogus data traffic is $1:\rho$ and a uniform traffic pattern (that is, nodes from each cell generate the same number of data reports). Then, the normalized energy waste in delivering all the traffic, denoted by E_w without LEDS and E'_w with LEDS, will be

$$\begin{aligned} E_w &= L_n \rho E_{tr} \frac{hl}{r} + L_n \frac{hl}{r}(1 - P_{\{0\}}) E_{tr} \frac{hl}{2r} \\ &= L_n E_{tr} \frac{hl}{r}\left(\rho + (1 - P_{\{0\}})\frac{hl}{2r}\right), \end{aligned} \tag{10}$$

$$\begin{aligned} E'_w = (L_n + L_a)\rho E_{tr} \Bigg(& h \sum_{i=1}^{t-1} P_{\{i\}} (1 - P_{\{0\}})^{t-i} \\ & + \frac{T}{2}\left(1 - \sum_{i=1}^{t-1} P_{\{i\}}(1 - P_{\{0\}})^{t-i}\right)\Bigg) \\ & + (L_n + L_a) h \left(1 - \sum_{i=0}^{\lfloor \frac{n'(r-l)}{l} \rfloor} P_{\{i\}}\right) E_{tr} \frac{h}{2}. \end{aligned} \tag{11}$$

It was reported in [26] that Rockwell Science Center's WINS sensor node consumes about $E_{tr} = 10\mu J$ for the hop-wise transmission and reception of one bit. Using this exemplary value, Fig. 10 plots the comparison of E_w and E'_w as a function of the bogus traffic ratio ρ and the number of compromised nodes x, when $L_n = 288$ bits, $L_a = 112$ bits, $(t, T) = (4, 5)$, $(N, n') = (10{,}000, 10)$, $l = \frac{2r}{3}$, and $h = 30$.

We can see that E_w increases dramatically with the increase of injected bogus data reports, whereas E'_w always maintains a rather stable level because 1) most bogus reports can be detected and dropped during their early transmission stages with LEDS in place and 2) it is much harder to drop the legitimate reports with LEDS in place. Furthermore, LEDS shows remarkable energy savings in contrast to the case without using LEDS. For example, when $x = 300$ and $\rho = 5$, LEDS saves more than 85 percent energy, that is, 385 mJ.

6 Conclusion

In this paper, through 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 few 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 multifunctional key management framework assures 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 effectiveness in energy savings, that is, achieving 85 percent or more energy savings in contrast to the case without using our design when appropriate parameters are chosen.

Acknowledgments

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

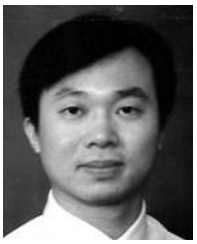
References

[1] D. Carman, P. Kruus, and B. Matt, "Constraints and Approaches for Distributed Sensor Network Security," Technical Report 00-010, NAI Labs, 2000.
[2] A. Wood and J. Stankovic, "Denial of Service in Sensor Networks," *Computer*, Oct. 2002.
[3] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Ad Hoc Networks*, vol. 1, no. 2, 2003.
[4] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, "SPINS: Security Protocols for Sensor Networks," *Proc. MobiCom*, July 2001.
[5] E. Shi and A. Perrig, "Designing Secure Sensor Networks," *Wireless Comm. Magazine*, vol. 11, no. 6, Dec. 2004.
[6] L. Eschenauer and V. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *Proc. Ninth ACM Conf. Computer and Comm. Security (CCS '02)*, 2002.
[7] H. Chan and A. Perrig, "Security and Privacy in Sensor Networks," *Computer*, pp. 103-105, Oct. 2003.
[8] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," *Proc. IEEE Symp. Research in Security and Privacy*, 2003.
[9] D. Liu and P. Ning, "Location-Based Pairwise Key Establishments for Relatively Static Sensor Networks," *Proc. ACM Workshop Security of Ad Hoc and Sensor Networks (SASN '03)*, Oct. 2003.
[10] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," *Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03)*, Oct. 2003.
[11] L. Lazos and R. Poovendran, "Serloc: Secure Range-Independent Localization for Wireless Sensor Networks," *Proc. ACM Int'l Conf. Mobile Computing and Networking (WiSe '04)*, Oct. 2004.
[12] W. Du, J. Deng, Y. Han, and P. Varshney, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks," *ACM Trans. Information and System Security*, vol. 8, no. 2, pp. 228-258, May 2005.
[13] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge," *IEEE Trans. Dependable and Secure Computing*, vol. 3, no. 2, pp. 62-77, Jan.-Mar. 2006.
[14] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing Pair-Wise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic

Approach," *Proc. IEEE Int'l Conf. Network Protocols (ICNP '03),* Nov. 2003.
[15] H. Chan and A. Perrig, "PIKE: Peer Intermediaries for Key Establishment," *Proc. IEEE INFOCOM,* Mar. 2005.
[16] S. Capkun and J.P. Hubaux, "Secure Positioning in Wireless Networks," *IEEE J. Selected Areas in Comm.,* Feb. 2006.
[17] S. Capkun and J. Hubaux, "Secure Positioning of Wireless Devices with Application to Sensor Networks," *Proc. IEEE INFOCOM,* Mar. 2005.
[18] 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," *Proc. IEEE Symp. Security and Privacy,* May 2004.
[19] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical En-Route Filtering of Injected False Data in Sensor Networks," *Proc. IEEE INFOCOM,* Mar. 2004.
[20] F. Ye, S. Lu, and L. Zhang, *Gradient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks,* ACM/Baltzer J. Wireless Networks, Mar. 2005.
[21] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward Resilient Security in Wireless Sensor Networks," *Proc. ACM MobiHoc,* 2005.
[22] *TinyOS Operation System,* http://millennium.berkeley.edu, 2005.
[23] H. Vogt, "Exploring Message Authentication in Sensor Networks," *Proc. European Workshop Security in Ad Hoc and Sensor Networks (ESAS '04),* Aug. 2004.
[24] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS '00),* 2000.
[25] A. Shamir, "How to Share a Secret," *Comm. ACM,* vol. 22, no. 11, pp. 612-613, Nov. 1979.
[26] D. Estrin, A. Sayeed, and M. Srivastava, "Wireless Sensor Networks," *Proc. MobiCom,* tutorial, 2002.
[27] J. Jung, T. Park, and C. Kim, "A Forwarding Scheme for Reliable and Energy-Efficient Data Delivery in Cluster-Based Sensor Networks," *IEEE Comm. Letters,* vol. 9, no. 2, pp. 112-114, Feb. 2005.
[28] W. Zhang, H. Song, S. Zhu, and G. Cao, "Least Privilege and Privilege Deprivation: Towards Tolerating Mobile Sink Compromises in Wireless Sensor Networks," *Proc. ACM MobiHoc,* May 2005.
[29] W. Zhang and G. Cao, "Group Rekeying for Filtering False Data in Sensor Networks: A Predistribution and Local Collaboration-Based Approach," *Proc. IEEE INFOCOM,* 2005.
[30] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks," *IEEE J. Selected Areas in Comm.,* vol. 24, no. 2, pp. 247-260, Feb. 2006.
[31] Y. Zhang, W. Liu, Y. Fang, and D. Wu, "Secure Localization and Authentication in Ultra-Wideband Sensor Networks," *IEEE J. Selected Areas in Comm.,* vol. 24, no. 4, pp. 829-835, Apr. 2006.
[32] J. Deng, R. Han, and S. Mishra, "Intrusion Tolerance and Anti-Traffic Analysis Strategies for Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Dependable Systems and Networks (DSN '04),* June 2004.
[33] W. Conner, T. Abdelzaher, and K. Nahrstedt, "Using Data Aggregation to Prevent Traffic Analysis in Wireless Sensor Networks," *Proc. Int'l Conf. Distributed Computing in Sensor Systems (DCOSS '06),* 2006.
[34] *CENS Research: Systems Infrastructure,* http://research.cens.ucla.edu/areas/2006/Systems_Infrastructure/default.htm, 2008.
[35] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing Location-Aware End-to-End Data Security in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* Apr. 2006.
[36] K. Ren, K. Zeng, and W. Lou, "A New Approach for Random Key Pre-Distribution in Large Scale Wireless Sensor Networks," *Wiley J. Wireless Comm. and Mobile Computing,* vol. 6, no. 3, pp. 307-318, 2006.
[37] K. Ren, K. Zeng, and W. Lou, "Secure and Fault-Tolerant Event Boundary Detection in Wireless Sensor Networks," *IEEE Trans. Wireless Comm.,* vol. 7, no. 1, Jan. 2008.
[38] K. Ren, W. Lou, K. Zeng, and P. Moran, "On Broadcast Authentication in Wireless Sensor Networks," *IEEE Trans. Wireless Comm.,* vol. 6, no. 11, pp. 4136-4144, Nov. 2007.

Kui Ren received the BEng and MEng degrees from Zhejiang University, China, in 1998 and 2001, respectively, and the PhD degree in electrical and computer engineering from Worcester Polytechnic Institute in 2007. He worked as a research assistant at the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, from March 2001 to January 2003, at the Institute for Infocomm Research, Singapore, from January 2003 to August 2003, and at the Information and Communications University, South Korea, from September 2003 to June 2004. He is an assistant professor in the Electrical and Computer Engineering Department, Illinois Institute of Technology. His research interests include ad hoc/sensor network security, wireless mesh network security, Internet security, and security and privacy in networks and systems. He is a member of the IEEE and the ACM.

Wenjing Lou received the BE and ME degrees in computer science and engineering from Xi'an Jiaotong University, China, in 1993 and 1996, respectively, the MASc degree from Nanyang Technological University, Singapore, in 1998, and the PhD degree in electrical and computer engineering from the University of Florida in 2003. From December 1997 to July 1999, she worked as a research engineer in the Network Technology Research Center, Nanyang Technological University. She is an assistant professor in the Electrical and Computer Engineering Department, Worcester Polytechnic Institute. Her current research interests include wireless ad hoc, sensor, and mesh networks, with emphases on network security and routing issues. She is an editor of the *IEEE Transactions on Wireless Communications.* She served as a TPC cochair for the general symposium, Globecom 2007. She served as a TPC vice cochair for the International Conference on Embedded Software and Systems (ICESS '05). She has served as a TPC member in many conferences, including IEEE INFOCOM 2005, 2007, and 2008. She is a member of the IEEE.



Yanchao Zhang received the BE degree in computer communications from the Nanjing University of Posts and Telecommunications, Nanjing, China, in July 1999, the ME degree in computer applications from the Beijing University of Posts and Telecommunications, Beijing, in April 2002, and the PhD degree in electrical and computer engineering from the University of Florida, Gainesville, in August 2006. He is currently an assistant professor in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark. His research interests include network and distributed system security, wireless networking, and mobile computing. He is a member of the IEEE and the ACM.