Location-Based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks

Yanchao Zhang, *Student Member, IEEE*, Wei Liu, Wenjing Lou, *Member, IEEE*, and Yuguang Fang, *Senior Member, IEEE*

***Abstract*—Node compromise is a serious threat to wireless sensor networks deployed in unattended and hostile environments. To mitigate the impact of compromised nodes, we propose a suite of location-based compromise-tolerant security mechanisms. Based on a new cryptographic concept called *pairing*, we propose the notion of location-based keys (LBKs) by binding private keys of individual nodes to both their IDs and geographic locations. We then develop an LBK-based neighborhood authentication scheme to localize the impact of compromised nodes to their vicinity. We also present efficient approaches to establish a shared key between any two network nodes. In contrast to previous key establishment solutions, our approaches feature nearly perfect resilience to node compromise, low communication and computation overhead, low memory requirements, and high network scalability. Moreover, we demonstrate the efficacy of LBKs in counteracting several notorious attacks against sensor networks such as the Sybil attack, the identity replication attack, and wormhole and sinkhole attacks. Finally, we propose a location-based threshold-endorsement scheme, called LTE, to thwart the infamous *bogus data injection* attack, in which adversaries inject lots of bogus data into the network. The utility of LTE in achieving remarkable energy savings is validated by detailed performance evaluation.**

***Index Terms*—Compromise tolerance, location, pairing, security, wireless sensor networks.**

I. Introduction

WIRELESS SENSOR NETWORKs (WSNs) have attracted a lot of attention recently due to their broad applications in both military and civilian operations. Many WSNs are deployed in unattended and often hostile environments such as military and homeland security operations. Therefore, security mechanisms providing confidentiality, authentication, data integrity, and nonrepudiation, among other security objectives, are vital to ensure proper network operations.

A future WSN is expected to consist of hundreds or even thousands of sensor nodes. This renders it impractical to monitor and protect each individual node from either physical or logical attack. It is also unrealistic and uneconomical to enclose each node in tamper-resistant hardware. Thus, each node represents a potential point of compromise. Once compromising certain nodes and acquiring their keying material, adversaries can launch various *insider* attacks. For example, they might spoof, alter or replay routing information to interrupt the network routing [1]. They may also launch the *Sybil* attack [2], [3] where a single node presents multiple identities to other nodes, or the *identity replication* attack, in which clones of a compromised node are put into multiple network places [3]. Moreover, adversaries may inject bogus data into the network to consume the scarce network resources [4], [5]. This situation poses the demand for *compromise-tolerant* security design. That is, the network should remain highly secure even when a number of nodes are compromised. Although a lot of solutions such as [6]–[14] have been proposed for securing WSNs, most of them do not provide adequate resilience to node compromise and the resulting attacks.

Many WSNs have an intrinsic property that sensor nodes are stationary, i.e., fixed at where they were deployed. This property has played an important role in many WSN applications such as target tracking [15] and geographic routing [16]. By contrast, its great potential in securing WSNs has so far drawn little attention. Based on this observation, we propose a suite of location-based compromise-tolerant security mechanisms for WSNs in this paper. Our main contributions are summarized as follows.

First, we propose the novel notion of *location-based keys* (LBKs) based on a new cryptographic concept called *pairing* (cf. Section II-A). In our scheme, each node holds a private key bound to both its ID and geographic location rather than merely its ID as in conventional schemes. To the best of our knowledge, this is the first such effort in the context of WSNs.

Second, we design a novel node-to-node neighborhood authentication protocol based on LBKs. It helps achieve the desirable goal of localizing the impact of compromise nodes (if any) to their vicinity, which is a nice property absent in most previous proposals.

Third, we present efficient approaches to establish pairwise shared keys between any two nodes that are either immediate neighbors or multihop away. Such keys are fundamental in providing security support for WSNs [7]–[14]. In contrast to previous proposals, our approaches feature low communication and computation overhead, low memory requirements, and good network scalability. More important, our approaches show perfect resistance to node compromise in that pairwise shared keys between noncompromised nodes always remain secure, no matter how many nodes are compromised.

Manuscript received October 1, 2004; revised August 10, 2005. This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464 and the U.S. National Science Foundation under Grant ANI-0093241 (CAREER Award).

Y. Zhang and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: yczhang@ufl.edu; fang@ece.ufl.edu).

W. Liu is with Scalable Network Technologies, Los Angeles, CA 90045 USA (e-mail: liuw@ufl.edu).

W. Lou is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA (e-mail: wjlou@ece.wpi.edu).

Digital Object Identifier 10.1109/JSAC.2005.861382

Fourth, we demonstrate how LBKs can act as efficient countermeasures against some notorious attacks against WSNs. These include the Sybil attack [1], [3], the identity replication attack [3], wormhole and sinkhole attacks [1], and so on.

Finally, we develop a location-based threshold-endorsement scheme (LTE) to thwart the aforementioned *bogus data injection* attack [4], [5]. Detailed performance evaluation shows that LTE can achieve remarkable energy savings by detecting and dropping bogus traffic at their early transmission stages. Moreover, our LTE has a much higher level of compromise tolerance than previous work [4], [5].

The rest of this paper is structured as follows. Section II introduces the cryptographic basis, the adversary model, and the security objectives of this paper. Next, we detail a LBK management scheme, including key generation, authentication, and shared-key establishment. This is followed by a detailed illustration of using LBKs in combating various attacks. Section V presents the LTE scheme and evaluates its performance. We then survey related work in Section VI, discuss the use of symmetric-key versus public-key cryptography in Section VII, and end with conclusions and future work in Section VIII.

II. Preliminaries

A. Pairing Concept

Identity-based cryptography (IBC) is receiving extensive attention as a powerful alternative to traditional certificate-based cryptography (CBC). Its main idea is to make an entity's public key directly derivable from its publicly known identity information such as the e-mail address. Eliminating the need for public-key certificates and their distribution makes IBC much more appealing for securing WSNs, where the need to transmit and check certificates has been identified as a significant limitation. For example, wireless transmission of a bit can require over 1000 times more energy than a single 32-bit computation, as shown in [17]. For this reason, we adopt IBC as the cryptographic foundation of this paper. Although the idea of IBC dates back to 1984 [18], only recently has its rapid development taken place due to the application of the *pairing* technique outlined below.

Let p, q be two large primes and $\mathbb{E}/\mathbb{F}_p$ indicate an elliptic curve $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$. We denote by $\mathbb{G}_1$ a q-order subgroup of the additive group of points of $\mathbb{E}/\mathbb{F}_p$, and by $\mathbb{G}_2$ a q-order subgroup of the multiplicative group of the finite field $\mathbb{F}^*_{p^2}$. The Discrete Logarithm Problem (DLP) is required to be hard[1] in both $\mathbb{G}_1$ and $\mathbb{G}_2$. For us, a pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with the following properties.

1) *Bilinear*: For $\forall\ P, Q, R, S \in \mathbb{G}_1$

$$\hat{e}(P+Q, R+S) = \hat{e}(P,R)\hat{e}(P,S)\hat{e}(Q,R)\hat{e}(Q,S). \quad (1)$$

Consequently, for $\forall\ c, d \in \mathbb{Z}^*_q$, we have

$$\hat{e}(cP, dQ) = \hat{e}(cP,Q)^d = \hat{e}(P,dQ)^c = \hat{e}(P,Q)^{cd} \text{ etc.}$$

2) *Nondegenerate*: If P is a generator of $\mathbb{G}_1$, then $\hat{e}(P,P) \in \mathbb{F}^*_{p^2}$ is a generator of $\mathbb{G}_2$.

3) *Computable*: There is an efficient algorithm to compute $\hat{e}(P,Q)$ for all $P, Q \in \mathbb{G}_1$.

Note that $\hat{e}$ is also *symmetric*, i.e., $\hat{e}(P,Q) = \hat{e}(Q,P)$, for all $P, Q \in \mathbb{G}_1$, which follows immediately from the bilinearity and the fact that $\mathbb{G}_1$ is a cyclic group. Modified Weil [19] and Tate [20] pairings are examples of such bilinear maps for which the *Bilinear Diffie–Hellman Problem* (BDHP) is believed to be hard.[2] We refer to [19] and [20] for a more comprehensive description of how these pairing parameters should be selected in practice for efficiency and security.

B. Adversary Model

Adversaries in WSNs can be classified as either *external* or *internal* adversaries. The former do not have authentic keying material whereby to participate in network operations as legitimate nodes. They might just passively eavesdrop on radio transmissions or actively inject bogus data or routing messages into the network to consume the network resources. Once in full control of certain nodes, external adversaries can become internal ones to be able to launch more subtle attacks like those mentioned in Section I. Internal adversaries are generally more difficult to defend against than external ones for their possession of authentic keying material. We further assume that adversaries have much more powerful resources regarding energy, communication, and communication capacities than ordinary sensor nodes. They might also communicate and collaborate over a high-bandwidth and low-latency channel invisible to legitimate sensor nodes. However, we do assume that adversaries cannot compromise an unlimited number of sensor nodes. Neither can they break any cryptographic primitive on which we base our design. Otherwise, there is unlikely to be any feasible security solution.

C. Security Objectives

We aim to provide confidentiality, authentication, data integrity, and nonrepudiation, four essential security objectives. We also intend to offer both *link-layer* and *end-to-end* security guarantees, both of which are indispensable for security-sensitive WSNs [1]. By definition, link-layer security indicates the security of radio links between neighboring nodes. It is a prerequisite to prevent external adversaries from accessing or modifying or faking radio transmissions. In contrast, end-to-end security refers to the communication security between a pair of source and destination nodes, e.g., a data aggregation point (AP) to a higher level AP or the sink [1]. We achieve link-layer security by *immediate pairwise keys* shared between neighboring nodes and end-to-end security by *multihop pairwise keys* shared between end-to-end sources and destinations.

III. Location-Based Key (LBK) Management Scheme

This section presents an LBK management scheme for WSNs, including the generation and distribution of LBKs, a secure LBK-based neighborhood authentication scheme, and methods for establishing both immediate and multihop pairwise shared keys.

[1]It is computationally infeasible to extract the integer $x \in \mathbb{Z}^*_q = \{a | 1 \le a \le q-1\}$, given $P, Q \in \mathbb{G}_1$ (respectively, $P, Q \in \mathbb{G}_2$) such that $Q = xP$ (respectively, $Q = P^x$).

[2]It is believed that, given $\langle P, xP, yP, zP \rangle$ for random x, y, $z \in \mathbb{Z}^*_q$ and $P \in \mathbb{G}_1$, there is no algorithm running in expected polynomial time, which can compute $\hat{e}(P,P)^{xyz} \in \mathbb{G}_2$ with nonnegligible probability.

A. Predeployment Phase

We examine a large-scale WSN consisting of hundreds or even thousands of sensor nodes. We assume that all the nodes have the same transmission range $\mathcal{R}$ and communicate via bidirectional wireless links. Nodes perform a collaborative monitoring of the designated sensor field and report the sensed events to the distant sink, which is a data collection center with sufficiently powerful processing capabilities and resources. We further assume that each node A has a unique, integer-valued and nonzero ID, denoted by ID_A. In view of the cost constraints, nodes are assumed to be not tamper-resistant in the sense that adversaries can extract all the keying material and data stored on a compromised node. However, we postulate that the sink is trustworthy and unassailable, as is commonly assumed in the literature [7]–[14].

Prior to network deployment, we assume that a trusted authority (TA) does the following operations.

1) Generate the pairing parameters $(p, q, \mathbb{E}/\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \hat{e})$, as described in Section II-A. Select an arbitrary generator W of $\mathbb{G}_1$.
2) Choose two cryptographic hash functions: H, mapping strings to nonzero elements in $\mathbb{G}_1$, and h, mapping arbitrary inputs to fixed-length outputs, e.g., SHA-1 [21].
3) Pick a random $\kappa \in \mathbb{Z}_q^*$ as the network master secret and set $W_{\mathrm{pub}} = \kappa W$.
4) Calculate for each node A an ID-based key (IBK for short), $\mathrm{IK}_A = \kappa H(\mathrm{ID}_A) \in \mathbb{G}_1$.

Each node A is preloaded with the public system parameters $(p, q, \mathbb{E}/\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H, h, W, W_{\mathrm{pub}})$ and its private IK_A. It is important to note that it is computationally infeasible to deduce κ from either (W, W_{pub}) or any (ID, IBK) pair like (ID_A, IK_A), due to the difficulty of solving the DLP in $\mathbb{G}_1$ (cf. Section II-A). Therefore, even after compromising an arbitrary number of nodes and their IBKs, adversaries are still unable to calculate the IBKs of noncompromised nodes.

B. Sensor Deployment and Localization

After loaded with the keying material, sensor nodes can be deployed in various ways such as physical installation or random aerial scattering. There are also many methods to localize each node, i.e., furnishing each node with its geographic location. We consider the following two sensor localization techniques, which accordingly differ in their ways of generating LBKs for individual nodes. The final outcome of either approach is that each node A possesses its location denoted by l_A and an LBK $\mathrm{LK}_A = \kappa H(\mathrm{ID}_A \parallel l_A)$, where $\parallel$ denotes message concatenation.

1) Range-Based Localization: In this approach, we assume that a group of mobile robots are dispatched to sweep across the whole sensor field along preplanned routes. Mobile robots have GPS capabilities, as well as more powerful computation and communication capacities than ordinary nodes. The leading robot is also equipped with the network master secret κ. To localize a node, say A, mobile robots run the secure range-based localization protocol given in [22] or [23] to first measure their respective absolute distance to node A and then co-determine l_A, the location of A. Subsequently, the leading robot calculates $\mathrm{LK}_A = \kappa H(\mathrm{ID}_A \parallel l_A)$. It then generates $\mathrm{IK}_A = \kappa H(\mathrm{ID}_A)$ and sends $\langle \{\mathrm{LK}_A \parallel l_A\}_{\mathrm{IK}_A}, h_{\mathrm{IK}_A}(\mathrm{LK}_A \parallel l_A) \rangle$ to A. Henceforth, $\{M\}_k$ means encrypting message M with key k, and $h_k(M)$ refers to the message integrity code (MIC) of message M under key k.

Upon receipt of the message, node A first uses its preloaded IBK IK_A to decrypt LK_A and l_A and then regenerates the MIC. If the result matches with what the robot sent, A saves LK_A and l_A for subsequent use. Following this process, all the nodes can be furnished with their respective location and LBK. After that, mobile robots leave the sensor field and the leading robot should securely erase κ from its memory. During subsequent network operations, node addition may be necessary to maintain good network connectivity. The localization of new nodes can be done in the same manner.

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. However, they may still perform relatively passive attacks such as message eavesdropping or strategic channel inference to disturb the localization process [22], [23]. This assumption is reasonable in that mobile robots are much fewer than ordinary sensor nodes and, hence, we can spend more on them by enclosing them in high-quality tamper-proof hardware and putting them under super monitoring. Adversaries may also want to temporarily avoid active and explicit attacks that may easily expose themselves. After the localization phase, adversaries are free to launch all kinds of attacks.

2) Range-Free Localization: By contrast, the range-free localization approach does not rely on exact distance or range measurements. Instead, we assume that there are some special nodes called anchors knowing their own locations. All the nonanchor nodes autonomously derive their locations based on information from the anchors and neighboring nodes via secure range-free localization techniques such as [24]–[26].

The LBKs are also generated on the nodes' own. To enable this, each node A is preloaded with the network master secret κ whereby to generate its LBK $\mathrm{LK}_A = \kappa H(\mathrm{ID}_A \parallel l_A)$. As LEAP [27], this approach takes advantage of the fact that sensor nodes deployed in security-sensitive environments are usually designed to withstand break-in attacks at least for a short interval when captured by adversaries. Specifically, we assume that an adversary needs a time interval at least $T_{\min}$ to successfully compromise a node, and each node takes some time less than $T_{\min}$ to finish localization and generation of its LBK. In addition, each node should be programmed to securely erase κ from its memory after $T_{\min}$ of its deployment. In the case of subsequent node addition, new nodes can get their locations and LBKs in the same way.

C. Location-Based Neighborhood Authentication

By definition, neighborhood authentication means the process that any two neighboring nodes validate each other's network membership. This process is fundamental in supporting many security services in WSNs. For example, a node should only accept messages from and forward messages to authenticated neighbors. Otherwise, external adversaries can easily inject bogus broadcast messages into the network or swindle network secret information from legitimate nodes.

During the post-deployment phase, each node is required to discover and perform mutual authentication with neighboring nodes, which is a normal process in many existing security solutions for sensor networks. In our scheme, each node will think of another node as an authentic neighbor if and only that node is within its transmission range $\mathcal{R}$ and also holds the correct corresponding LBK. We take the following concrete example to explain the neighborhood authentication process.

1) $A \rightarrow * \ : \ \mathrm{ID}_A, l_A, n_A.$
2) $B \rightarrow A \ : \ \mathrm{ID}_B, l_B, n_B, h_{K_{B,A}}(n_A \| n_B \| 1).$
3) $A \rightarrow B \ : \ h_{K_{A,B}}(n_A \| n_B \| 2).$

Suppose node A wishes to discover and authenticate neighboring nodes once having its location and LBK. To do so, A locally broadcasts an authentication request including its ID ID_A, location l_A and a random nonce n_A. Upon receipt of such a request, node B first needs to ascertain that the claimed location l_A is in its transmission range by verifying if the Euclidean distance $\|l_A - l_B\| \leq \mathcal{R}$. This check is the baseline defense against the attack that adversaries surreptitiously tunnel authentication messages between B and a virtually nonneighboring node. Without the location check, B and that victim will falsely believe that they are neighbors because both possess an authentic LBK whereby to successfully finish the following authentication process.

If the inequality does not hold, node B simply discards the authentication request. Otherwise, B calculates a shared key as $K_{B,A} = \hat{e}(\mathrm{LK}_B, H(\mathrm{ID}_A \ \| \ l_A))$. It then unicasts a reply to node A including its ID and location, a random nonce n_B, and a MIC computed as $h_{K_{B,A}}(n_A \ \| \ n_B \ \| \ 1)$. Upon receiving the reply, node A also first checks if the inequality $\|l_A - l_B\| \leq \mathcal{R}$ holds. If so, it proceeds to derive a shared key as $K_{A,B} = \hat{e}(\mathrm{LK}_A, H(\mathrm{ID}_B \ \| \ l_B))$ whereby to recompute the MIC. If the result is equal to what B sent, node A considers B an authentic neighbor. Subsequently, A returns to node B a new MIC computed as $h_{K_{A,B}}(n_A \ \| \ n_B \ \| \ 2)$. Upon receipt of it, B uses $K_{B,A}$ to regenerate the MIC and compares the result with what it just received. If they are equal, B regards node A as an authentic neighbor as well.

The above process is valid because, if and only if both A and B have a correct LBK, $K_{A,B}$ is equal to $K_{B,A}$ due to the following equations:

$$\begin{aligned} K_{A,B} &= \hat{e}(\mathrm{LK}_A, H(\mathrm{ID}_B \ \| \ l_B)) \\ &= \hat{e}(\kappa H(\mathrm{ID}_A \ \| \ l_A), H(\mathrm{ID}_B \ \| \ l_B)) \\ &= \hat{e}(H(\mathrm{ID}_A \ \| \ l_A), \kappa H(\mathrm{ID}_B \ \| \ l_B)) \\ &= \hat{e}(\kappa H(\mathrm{ID}_B \ \| \ l_B), H(\mathrm{ID}_A \ \| \ l_A)) \\ &= \hat{e}(\mathrm{LK}_B, H(\mathrm{ID}_A \ \| \ l_A)) = K_{B,A}. \end{aligned} \tag{2}$$

The second and third lines hold for the bilinearity of $\hat{e}$ and the fourth line holds by the symmetry of $\hat{e}$ (cf. Section II-A).

Using the above three-way handshake, all the nodes can achieve mutual authentication with neighboring nodes. Note that if multiple nodes simultaneously respond to the same authentication request, possible MAC-layer collision may happen. We resort to effective MAC-layer mechanisms to resolve this issue. For example, it can be alleviated through MAC-layer retransmission or by using a random jitter delay for which each node has to wait before answering an authentication request.

In our scheme, new nodes can be added freely to maintain necessary network connectivity, especially when some existing nodes die out because of power shortage or other reasons. A new node is also required to execute the authentication protocol once localized properly.

Security Analysis: Our location-based authentication scheme is secure against various malicious attacks. For example, in a *location forgery* attack, an adversary might send an authentication request with a forged location within node B's range. Since the adversary does not hold the LBK corresponding to the forged location, he or she cannot successfully finish the authentication procedure and, thus, deceive B into believing that he or she is an authentic neighbor. Adversaries might as well launch the *tunnelling of authentication messages* attack by tunnelling authentication messages received at one location of the network over an invisible, out-of-band and low-latency channel to another network location which is typically multihop away. By doing so, they attempt to make two victim nodes far away from each other believe that they are authentic neighbors. This attack is infeasible with our scheme in that each node will simply deny authentication requests from nodes that are not physically within its transmission range. In addition, an adversary might put into the vicinity of a legitimate node, say B, a replica of one compromised node at other distant locations. Most purely ID-based authentication schemes are vulnerable to this attack because, without dependence on any central authority [3], [8] the victim B has great difficulty in differentiating between legitimate authentication requests and malicious ones from replicas of a compromised node. With our scheme in place, node B will simply ignore the replica's authentication request because the replica should not appear in its transmission range.

It is worth pointing out that, as any other security solution, our scheme itself cannot prevent a compromised node or its replicas from achieving mutual authentication with its legitimate neighbors. However, it can guarantee that the compromised node or its replicas receive nothing more than some random numbers, public IDs and locations from legitimate nodes. This ensures that the compromised node cannot impersonate its legitimate neighbors to other nodes. Therefore, our location-based authentication scheme can reduce the impact of a compromised node from the otherwise network-wide scale to its vicinity, more specifically, within a circle with radius $2\mathcal{R}$ centered at its current location. This makes it far more easier to devise efficient localized intrusion detection mechanisms.

One may worry that adversaries might mount the denial-of-service attack by continuously sending bogus authentication requests or replies to allure legitimate nodes into endless processing of such messages. In our opinion, this attack is in fact less worrisome. The reason is that the number of neighbors of any node is limited in reality. Therefore, abnormally many authentication requests or replies are highly likely an indicator of malicious attacks. Under such situations, we assume that there are efficient mechanisms available for legitimate nodes to report such an abnormality to the sink.

D. Immediate Pairwise Key (IPK) Establishment

Link-layer security schemes demand an efficient method to establish pairwise shared keys between neighboring nodes. Henceforth, we refer to such keys as *immediate pairwise key*s (or IPKs for short). With IPKs, messages exchanged between neighboring nodes can be encrypted and authenticated via efficient symmetric-key algorithms.

Note that after a successful three-way handshake, two neighboring nodes, say A and B, have established a shared key $K_{A,B} = K_{B,A}$. Adversaries, be they external or internal, may overhear the authentication messages, but cannot deduce the shared key for the lack of the LBKs of A and B. From $K_{A,B}$, A and B can derive various shared session keys for different security purposes by feeding $K_{A,B}$ into the hash function h. For example, they can use $k_0 = h(K_{A,B} \parallel 0)$ for message encryption and $k_1 = h(K_{A,B} \parallel 1)$ for message authentication. In the similar way, each node can establish IPKs with all its legitimate neighbors after the neighbor discovery and authentication phase.

Since the IPKs are by-products of the neighborhood authentication process, there is no extra key-establishment communication and computation overhead. In addition, our IPK establishment method has perfect resistance to node compromise because the IPKs are built upon the private LBKs of individual nodes. No matter how many nodes are compromised, the LBKs of noncompromised nodes always remain secure, and so do the IPKs established between them.

E. Multihop Pairwise Key (MPK) Establishment

In addition to the IPKs, a node may need to establish pairwise shared keys with other nodes that are multihop away. We call such keys as *multihop pairwise keys* (or MPKs for short) that are required for securing end-to-end traffic.

Assume that nodes U and V are multihop apart and the routing path between them has been established using the underlying routing protocol. To establish an MPK, U and V execute the following protocol.:

$$1)\ U \rightarrow V\ :\ \mathrm{ID}_U, l_U, n_U H(\mathrm{ID}_U \parallel l_U).$$
$$2)\ V \rightarrow U\ :\ \mathrm{ID}_V, l_V, n_V H(\mathrm{ID}_V \parallel l_V).$$

Here, n_U, $n_V \in \mathbb{Z}_q^*$ are random private numbers chosen by nodes U and V, respectively. At the conclusion of the protocol, node V calculates

$$\begin{aligned} K_{V,U} &= \hat{e}\left(\mathrm{LK}_V, n_V H(\mathrm{ID}_U \parallel l_U) + n_U H(\mathrm{ID}_U \parallel l_U)\right) \\ &= \hat{e}\left(\kappa H(\mathrm{ID}_V \parallel l_V), (n_V + n_U) H(\mathrm{ID}_U \parallel l_U)\right). \end{aligned}$$

Likewise, node U computes

$$\begin{aligned} K_{U,V} &= \hat{e}\left(\mathrm{LK}_U, n_U H(\mathrm{ID}_V \parallel l_V) + n_V H(\mathrm{ID}_V \parallel l_V)\right) \\ &= \hat{e}\left(\kappa H(\mathrm{ID}_U \parallel l_U), (n_U + n_V) H(\mathrm{ID}_V \parallel l_V)\right). \end{aligned}$$

If both nodes are legitimate and have followed the protocol correctly, by the bilinearity and symmetry of $\hat{e}$:

$$K_{U,V} = K_{V,U} = \hat{e}\left(H(\mathrm{ID}_U \parallel l_U), H(\mathrm{ID}_V \parallel l_V)\right)^{(n_U + n_V)\kappa}.$$

Based on the MPK $K_{U,V}$, nodes U and V can derive various shared session keys for different security purposes as before.

Discussion: If possible, the two protocol messages can piggyback on the routing messages used to establish the routing path between U and V. In doing so, the related communication overhead can be much reduced. In addition, there is no need for U and V to further exchange messages to prove to the other the knowledge of the MPK. Any future messages encrypted and authenticated with the MPK or the derivative session keys can implicitly achieve the same effect.

Our MPK establishment protocol is a simple adaptation of the provably secure ID-based key agreement protocol [28]. Any third party may overhear the plaintext messages exchanged between U and V, but cannot derive the MPK $K_{U,V}$ without knowing the LBKs of U or V. This protocol also has perfect resilience against node compromise because of the dependence of the MPKs on the nodes' private LBKs.

IV. Efficacy of LBKs in Attack Mitigation

In this section, we show how the proposed LBKs can act as effective and efficient countermeasures against several notorious attacks against WSNs.

A. Spoofing, Altering, or Replaying Routing Information

Without precaution, external adversaries are able to spoof, alter or replay routing messages. By doing so, they attempt to create routing loops, cause network partitions, incur false error messages, and so on [1].

As mentioned before, neighboring nodes are required to perform mutual authentication based on their private LBKs. Since each node only processes routing messages from authenticated neighbors, external adversaries can be prevented from entering the network and distributing phony routing messages. The remaining problem is how to defend against internal adversaries or compromised nodes in possession of authentic keying material. It is believed that there is no cryptographic way that can prevent them from manipulating routing information. However, our location-based neighborhood authentication scheme can constrain the impact of compromised nodes to a small range centered at their original locations. In other words, internal adversaries cannot utilize the acquired keying material at one place to launch routing attacks at another distant place. What they can only possibly do is to continue misbehaving at "the scene of the crime," i.e., a small range around the location of the compromised node. If doing so, they might run a high risk of being detected by legitimate nodes if effective localized misbehavior detection mechanisms are available.

B. Sybil Attack

The Sybil attack happens when a malicious node behaves as if it were a large number of nodes, e.g., by impersonating other nodes or simply claiming multiple forged IDs and/or locations. As pointed out in [1] and [3], this attack is extremely detrimental to many important WSN functions, such as routing, fair resource allocation, misbehavior detection, data aggregation, and distributed storage.

With our scheme in place, when a malicious node intends to impersonate a legitimate node, it does not have the authentic LBK and, thus, cannot successfully finish mutual authentication with other legitimate nodes. For the same reason, a malicious node cannot claim forged IDs and/or locations without being detected. Therefore, the Sybil attack is effectively defeated.

C. Identity Replication Attack

The identity replication attack [3] takes place when adversaries put multiple replicas of a compromised node in different geographic locations. It may lead to the inconsistence of the network routing information, as well as jeopardizing other important network functions. Conventional defenses often involve a central authority, e.g., the sink, that either keeps a record of each node's location [3], or centrally counts the number of connections a node has and revokes those with too many connections [8]. These solutions require node-to-node authentication and pairwise key establishment to be performed through the central authority, thereby causing significant communication overhead and the lack of scalability.

This attack is no longer feasible when our location-based neighborhood authentication scheme is applied. The replicas of a compromised node will be prevented from entering the network by legitimate nodes at locations other than the neighborhood of the compromised node. Our countermeasure is totally self-organizing and does not involve any central authority, hence, it is rather lightweight and highly scalable in contrast to previous solutions.

D. Wormhole and Sinkhole Attacks

Wormhole [1], [29] and sinkhole [1] attacks are two notorious attacks against WSN routing protocols that are difficult to withstand, especially when the two are used in combination.

In the wormhole attack, instead of compromising any node, collaborative adversaries first create a *wormhole link*, essentially an out-of-band and low-latency channel, between two distant network locations. They then tunnel routing messages recorded at one location via the wormhole link to the other, leading to the chaos of the routing operations. Hu *et al.* [29] presented a technique called *packet leashes* to withstand the wormhole attack. It requires extremely tight time synchronization and is, thus, infeasible for most WSNs, as noted in [1]. In contrast, each node in our scheme only accepts routing messages from authenticated neighbors and will discard those tunnelled from distant locations. Therefore, the wormhole attack is effectively and efficiently thwarted.

In the sinkhole attack, compromised nodes attempt to attract all the traffic from their surrounding nodes by announcing a high-quality route to the sink or some other destinations. For example, adversaries create an invisible and fast channel between two compromised nodes A and B residing in distant network regions. Node A claims that it is one hop or a few hops away from B or other nodes close to B. By doing so, A aims to be selected by legitimate surrounding nodes as a packet relay to B or other nodes in that region. Fortunately, our scheme can withstand such sinkhole attacks against minimum-hop routing protocols. For instance, upon seeing A's advertisement of a single-hop path to node B, a legitimate node can immediately find out that A is malicious by noting that the distance between A and B is far more larger than the normal transmission range $\mathcal{R}$. In addition, geographic routing protocols such as [16] have been identified in [1] as promising solutions resistant to sinkhole and wormhole attacks. The reason is that they construct the routing topology on demand using only localized interactions and geographic information. To apply such schemes, however, the location information advertised from neighboring nodes must be authenticated. We provide such a guarantee by the LBKs and the location-based neighborhood authentication scheme.

We note that our scheme itself cannot prevent the sinkhole attacks against routing protocols with routing metrics such as remaining energy or end-to-end reliability. The major reason is that the authenticity of these information is very difficult to verify by cryptographic means alone. As far as we know, the related countermeasure, thus, far remains an open challenging issue, and is an interesting topic worthy of further study.

V. Location-Based Filtering of Bogus Data

In this section, we first describe the bogus data injection attack. We then present a LTE scheme as the countermeasure. At last, we evaluate the performance of LTE in terms of energy savings.

A. Bogus Data Injection Attack

As mentioned before, neighborhood mutual authentication is sufficient to prevent external adversaries from injecting bogus data into the network, but will fail in the presence of internal adversaries. By a single compromised node, internal adversaries can induce arbitrary and seemingly authentic data reports into the network. Without precaution, this kind of attack may do a lot of damage to the network, e.g., causing false alarms or network traffic congestion. Even worse, it can deplete the precious energy of relaying nodes on any forwarding path to the sink, which is often tens or even hundreds of hops away from the sources of data reports. It is, therefore, important to design effective and efficient countermeasures against this attack.

Since there is no way of hindering internal adversaries from injecting bogus data, we attempt to figure out ways to mitigate their impact. Our first goal is to filter bogus data reports as early as possible before they reach the sink. Our second goal is to detain adversaries from freely fabricating the originating locations of injected bogus data reports.

We achieve the first goal by a threshold-endorsement method. That is, a data report should be co-signed by t nodes for it to be considered authentic. A report without a correct endorsement will be regarded as a fake one and discarded by any legitimate node after verifying it. Our method is motivated by the observation that every point in the sensor field should be covered by at least t nodes, known as the t-coverage problem [30]. The t-coverage property is required by many security-sensitive WSN applications such as intrusion detection to facilitate fine-grained surveillance. In our case, adversaries will have much greater difficulty in injecting seemingly authentic yet bogus data reports, as they now have to compromise at least t nodes instead of only one as before.

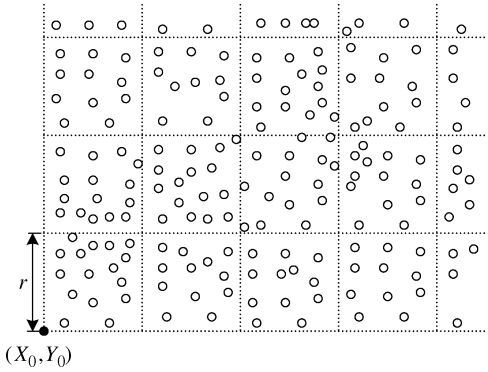


Fig. 1. Node deployment model.

We fulfill the second objective by embedding the location information of a data report's originating area in the joint endorsement it carries. To inject a bogus data report that originates from a certain area and can survive the filtering by legitimate intermediate nodes, adversaries must actually compromise at least t nodes holding keying material of that area. Even so, they cannot utilize the acquired keying material to fake data reports that seem to originate from other areas. Another benefit is that, once determining that some arriving reports are unfiltered bogus ones, the sink can pinpoint their originating areas, and then take specific remedy actions.

Below, we detail how to actually realize the above ideas.

B. Generation and Distribution of Cell Keys

To enable LTE, we propose the notion of *cell keys*. For the sake of simplicity, we assume that the sensor field is a $Mr \times Nr$ rectangle whose lower-left corner is at location (X_0, Y_0). The sensor field is divided into MN square cells of equal side length r. Each cell is labeled with a pair of integers $\langle m, n\rangle$, for $1 \le m \le M$ and $1 \le n \le N$. Prior to deployment, (X_0, Y_0) and r are preloaded to each node. Also, note that our LTE can be easily extended for use with any other node deployment model (Fig. 1).

We define the cell key of cell $\langle m, n\rangle$ as $\mathcal{K}_{m,n} = \kappa H(m \parallel n)$, which shall be used to endorse any report originating from that cell. The next question is how to distribute $\mathcal{K}_{m,n}$ to nodes in cell $\langle m, n\rangle$. Let $\mathrm{ID}^i_{m,n}$ denote the ith node with location $l^i_{m,n}$ in cell $\langle m, n\rangle$. The naive method of letting each $\mathrm{ID}^i_{m,n}$ hold one copy of $\mathcal{K}_{m,n}$ obviously suffers from single node compromise. Instead, we propose to utilize the secret-sharing technique [31] to assign a share of $\mathcal{K}_{m,n}$ to each $\mathrm{ID}^i_{m,n}$. The purpose is to make $\mathcal{K}_{m,n}$ reconstructible by any t nodes in cell $\langle m, n\rangle$, while irrecoverable by any less than t of them. To do this, prior to network deployment, the TA additionally generates a $(t-1)$-degree polynomial, $\mathcal{F}(x) = \sum_{j=1}^{t-1} F_j x^j \in \mathbb{G}_1$, with coefficients F_j randomly selected from $\mathbb{G}_1^*$.[3] It also selects another system parameter $c \le r$ whose use is explained shortly. We consider the following two cases of cell-key share distribution, depending on whether node localization is range-based (cf. Section III-B1) or range-free (cf. Section III-B2).

1) Range-Based Cell-Key Distribution: In this approach, the leading robot is preloaded with the polynomial $\mathcal{F}(x)$. In addition to determining a node's location, it decides that node's present cell by simple geometric calculations. Consider node $\mathrm{ID}^i_{m,n}$ as an example. Its location $l^i_{m,n}$, i.e., $(X^i_{m,n}, Y^i_{m,n})$, will satisfy $(m-1)r \le X^i_{m,n} - X_0 < mr$ and $(n-1)r \le Y^i_{m,n} - Y_0 < nr$. Then, the leading robot derives $\mathcal{K}_{m,n} = \kappa H(m \parallel n)$ and a set of authenticators $\vec{V}_{m,n} = \{v^{(j)}_{m,n} | 0 \le j \le t-1\}$, where $v^{(j)}_{m,n} = \hat{e}(F_j, W)$ and $F_0 = \mathcal{K}_{m,n}$. Note that it just needs to do these computations once for each cell. Next, the leading robot calculates $\mathcal{K}^i_{m,n} = \mathcal{F}(\mathrm{ID}^i_{m,n} \parallel l^i_{m,n}) + \mathcal{K}_{m,n} \in \mathbb{G}_1$, referred to as node $\mathrm{ID}^i_{m,n}$'s share of $\mathcal{K}_{m,n}$. Finally, $\mathcal{K}^i_{m,n}$ and $\vec{V}_{m,n}$ are securely sent to node $\mathrm{ID}^i_{m,n}$ along with $l^i_{m,n}$ and its LBK (cf. Section III-B1).

$\mathcal{K}_{m,n}$ can be reconstructed from any t shares of it, but is irretrievable from any $(t-1)$ or fewer shares. In particular, let $T_{m,n}$ denote the number of nodes in cell $\langle m, n\rangle$ and Ω be a t-order subset of $\{1, \ldots, T_{m,n}\}$. We can compute

$$\mathcal{K}_{m,n} = \sum_{i \in \Omega} \lambda_i \mathcal{K}^i_{m,n} \tag{3}$$

where $\lambda_i = \prod_{j \in \Omega \setminus \{i\}} (\mathrm{ID}^j_{m,n} \parallel l^j_{m,n}) / (\mathrm{ID}^j_{m,n} \parallel l^j_{m,n} - \mathrm{ID}^i_{m,n} \parallel l^i_{m,n})$. Regarding the choice of t, there is a tradeoff between resilience to node compromise and node density. Basically, the larger t, the more resilient the network is to node compromise, the higher the required node density is, and *vice versa*. This issue is closely related to the well-studied t-coverage problem [30]. We refer interested readers to [30] about how to strike a good balance between these two competing metrics.

To ensure high-level t-coverage of cell boundaries with regard to security, it is also important to let some nodes possess cell-key shares of adjacent cells. In particular, we require that the nodes out of a cell but within c of the cell boundary also hold cell-key shares of that cell. For example, if $mr - X^i_{m,n} \le c$, node $\mathrm{ID}^i_{m,n}$ also has the authenticator vector $\vec{V}_{m+1,n}$ and a share of cell key $\mathcal{K}_{m+1,n}$. Likewise, if $nr - Y^i_{m,n} \le c$, it owns $\vec{V}_{m+1,n}$ and a share of $\mathcal{K}_{m,n+1}$ as well. In addition, for the boundaries of the sensor field, it is often necessary to purposely deploy some sensors beyond the field boundaries. The choice of c represents a tradeoff between cell-boundary t-coverage and tolerance to node compromise. The greater c, the higher level t-coverage of cell boundaries, the more vulnerable a cell key is to node compromise because more nodes have a cell-key share, and *vice versa*. Its concrete value is also germane to that of t and node density.

2) Range-Free Cell-Key Distribution: In this method, each node is preloaded with the polynomial $\mathcal{F}(x)$ in addition to the network master secret κ. Consider again node $\mathrm{ID}^i_{m,n}$ as an example. Once determining its own location $l^i_{m,n}$, it also knows that it resides in cell $\langle m, n\rangle$. Therefore, besides generating its LBK (cf. Section III-B2), node $\mathrm{ID}^i_{m,n}$ employs κ to first derive $\mathcal{K}_{m,n}$ and then its share $\mathcal{K}^i_{m,n}$. Moreover, it computes the authenticator vector $\vec{V}_{m,n}$.[4] If within c of adjacent cells' boundaries, node $\mathrm{ID}^i_{m,n}$ should as well compute a cell-key share and the authenticator vector for each of those cells. Upon finishing all these operations, it should securely erase κ, $\mathcal{F}(x)$ and all the complete cell keys from its memory.

[3] $\mathbb{G}_1^*$ denotes the set $\mathbb{G}_1 \setminus \{\mathcal{O}\}$, where $\{\mathcal{O}\}$ is the identity element of $\mathbb{G}_1$.

[4] The authenticators $v^{(j)}_{m,n}$ $(1 \le j \le t-1)$ may be precalculated and preloaded to each node to reduce the computational overhead.

C. Performing Threshold-Endorsements of Data Reports

Now, we explain how to perform threshold-endorsements on data reports. Without loss of generality, we take cell $\langle m,n\rangle$ as an example in the following description.

In general, sensor nodes generate a report when triggered by a special event such as the appearance of adversaries, or in response to a query made by the sink. Assume that such a stimulus occurs in cell $\rangle m,n\langle$ and is detected by $s \geq t$ nodes. If the event occurs closely to the cell boundary, then the s nodes may include nodes in different adjacent cells. To simplify our presentation, however, we assume that all of them are in cell $\langle m,n\rangle$. By local interactions, the detecting nodes can reach a consensus on a final report, denoted by Λ and containing application-dependent information such as the type, occurrence time and location of the event.

The detecting nodes are required to elect among themselves an AP. To obtain a threshold-endorsement of Λ, the AP chooses a random $\alpha \in \mathbb{Z}_q^*$ and computes $\theta = \hat{e}(W,W)^\alpha$ broadcasted to the other detecting nodes. Upon receipt of θ, each detecting node $\mathrm{ID}_{m,n}^i$ endorses the report Λ by computing $U_{m,n}^i = \mathcal{K}_{m,n}^i h(\Lambda||\theta)$. It then sends to the AP $U_{m,n}^i$ encrypted and authenticated with the pairwise key shared with the AP [cf. Section III-D]. Once receiving over t such endorsements, the AP randomly selects t of the endorsers, denoted by a set notation Ω which may include itself. It then calculates $U_{m,n} = \sum_{i\in\Omega} \lambda_i U_{m,n}^i = \mathcal{K}_{m,n} h(\Lambda||\theta)$ (cf. (3)) and $\Upsilon_{m,n} = U_{m,n} + \alpha W$. The threshold-endorsement of Λ is $(\Upsilon_{m,n}, h(\Lambda \parallel \theta))$ and the final report is of format $\langle \Lambda, \Upsilon_{m,n}, h(\Lambda||\theta)\rangle$.

It is possible that some of the endorsers have been compromised and, thus, may provide the AP with falsely computed endorsements. Fortunately, our LTE scheme can well handle this situation. In particular, once deriving $U_{m,n}$, the AP is required to verify its authenticity by checking if the equation $\hat{e}(U_{m,n}, W) = (v_{m,n}^{(0)})^{h(\Lambda||\theta)}$ holds. The check should succeed for a valid $U_{m,n}$ because $\hat{e}(U_{m,n}, W) = \hat{e}(\mathcal{K}_{m,n}, W)^{h(\Lambda||\theta)}$ by the bilinearity of $\hat{e}$ and $v_{m,n}^{(0)} = \hat{e}(\mathcal{K}_{m,n}, W)$. Otherwise, the AP proceeds to verify each received $U_{m,n}^i$ by checking if

$$\hat{e}\left(U_{m,n}^i, W\right) = \prod_{j=0}^{t-1} \left(v_{m,n}^{(j)}\right)^{\left(\mathrm{ID}_{m,n}^i || l_{m,n}^i\right)^j \cdot h(\Lambda||\theta)}.$$

The verification works because of the following equations:

$$\begin{aligned}
\hat{e}\left(U_{m,n}^i, W\right) &= \hat{e}\left(\mathcal{K}_{m,n}^i, W\right)^{h(\Lambda||\theta)} \\
&= \hat{e}\left(\sum_{j=0}^{t-1} F_j\left(\mathrm{ID}_{m,n}^i \parallel l_{m,n}^i\right)^j, W\right)^{h(\Lambda||\theta)} \\
&= \prod_{j=0}^{t-1} \hat{e}(F_j, W)^{\left(\mathrm{ID}_{m,n}^i || l_{m,n}^i\right)^j \cdot h(\Lambda||\theta)} \\
&= \prod_{j=0}^{t-1} \left(v_{m,n}^{(j)}\right)^{\left(\mathrm{ID}_{m,n}^i || l_{m,n}^i\right)^j \cdot h(\Lambda||\theta)}. \qquad (4)
\end{aligned}$$

The third-line equation holds because $\hat{e}$ is bilinear. If the check succeeds, the AP considers node $\mathrm{ID}_{m,n}^i$ legitimate and compromised, otherwise. In this way, the AP is able to pinpoint all the endorsers offering false endorsements and delete them from Ω. Subsequently, it replenishes Ω with the corresponding number of endorsers randomly selected from the unused ones, and recalculates $(\Upsilon_{m,n}, h(\Lambda \parallel \theta))$. As long as there are at least t legitimate endorsers, a correct threshold-endorsement can always be generated.

It is worth noting that the pinpoint-identification capability of the AP may deter the compromised endorsers (if any) from providing false endorsements. As a result, it is highly possible that the AP can derive an authentic threshold-endorsement in the first round. In the light of this, we let the AP verify the individual endorsements only when the threshold-endorsement is incorrect rather than at the beginning, thereby reducing its computational load.

In some cases, the AP itself may be a compromised node. It may either not at all send a final report to the sink or transmit a bogus report with an incorrect Λ or a wrong $(\Upsilon_{m,n}, h(\Lambda||\theta))$ or both. Both attacks can be easily detected by the legitimate detecting nodes which, in turn, elect a new AP among themselves to generate a new threshold-endorsement and send the final report to the sink. Also, note that dealing with the latter attack requires the legitimate detecting nodes to verify the threshold-endorsement in the final report. The verifications are similar to the filtering operations by intermediate nodes on the way to the sink, which are explained in what follows.

D. Probabilistic Enroute Filtering of Data Reports

The AP sends to the sink the final report along a multihop path discovered via the underlying routing protocol. Depending on different applications, end-to-end and/or link-layer security measures can be enforced on the report transmission (cf. Sections III-D and III-E). We denote by p_s the *sampling probability* which is a system-wide parameter.

Upon receipt of a report $\langle \Lambda, \Upsilon_{m,n}, h(\Lambda \parallel \theta)\rangle$ to be forwarded, with probability p_s, each intermediate node, say A, deduces the originating cell information $\rangle m,n\langle$ from the event location embedded in Λ. It then computes

$$\theta' = \hat{e}(\Upsilon_{m,n}, W)\hat{e}\left(H(m \parallel n), -W_{\mathrm{pub}}\right)^{h(\Lambda||\theta)} \qquad (5)$$

where $W_{\mathrm{pub}} = \kappa W$ is the public system parameter defined in Section III-A. If the report is authentic, we will have

$$\begin{aligned}
\theta' &= \hat{e}(\Upsilon_{m,n}, W)\hat{e}\left(H(m \parallel n), W_{\mathrm{pub}}\right)^{-h(\Lambda||\theta)} \\
&= \hat{e}\left(\mathcal{K}_{m,n} h(\Lambda \parallel \theta) + \alpha W, W\right)\hat{e}\left(H(m \parallel n), \kappa W\right)^{-h(\Lambda||\theta)} \\
&= \hat{e}\left(\mathcal{K}_{m,n} h(\Lambda \parallel \theta) + \alpha W, W\right)\hat{e}\left(\kappa H(m \parallel n), W\right)^{-h(\Lambda||\theta)} \\
&= \hat{e}(\mathcal{K}_{m,n}, W)^{h(\Lambda||\theta)}\hat{e}(W,W)^{\alpha}\hat{e}(\mathcal{K}_{m,n}, W)^{-h(\Lambda||\theta)} \\
&= \theta. \qquad (6)
\end{aligned}$$

Therefore, if $h(\Lambda||\theta') = h(\Lambda||\theta)$, node A considers the report authentic and then forward it to the next hop. Otherwise, it thinks of the report a fabricated one and simply dumps it. Our LTE scheme is a simplified adaptation of the provably secure threshold version [32] of Hess's ID-based signature scheme [33].

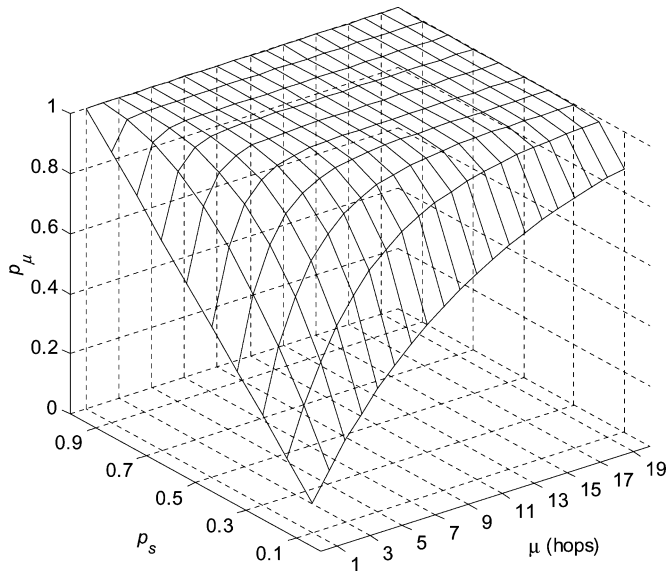


Fig. 2. The probability p_μ of filtering one bogus report as a function of the sampling probability p_s and the number μ of hops a bogus report travels.

E. Efficacy and Security Analysis

We first quantify the efficacy of probabilistic enroute filtering of fabricated data reports. There might be compromised nodes lying on the forwarding path to the sink which just relay bogus reports to the next hop without verifying them. Since we are only interested in the energy consumption of legitimate intermediate nodes, we merely consider a "valid" forwarding path from which compromised nodes are extracted. Given the sampling probability p_s, the probability that a bogus report can be detected and dropped within μ hops is $p_\mu = 1-(1-p_s)^\mu$, and the average number of hops a bogus report traverses is

$$\bar{\mu} = \sum_{j=1}^{\infty} j p_s (1-p_s)^{j-1} = \frac{1}{p_s}. \tag{7}$$

Fig. 2 shows how p_μ changes with p_s and μ. We can see that, even when p_s assumes a small value, say 0.3, over 83% of bogus reports can be filtered within five hops, and less than 3% of them can travel beyond ten hops. Therefore, for large-scale WSNs often involving very long forwarding paths, our LTE is highly effective in filtering bogus reports during their early transmission stages, thereby saving the precious energy of legitimate nodes.

Due to the probabilistic verifications at intermediate nodes, a bogus report might escape the filtering and reach the sink with a small probability $(1-p_s)^{\text{len}-1}$, where len indicates the forwarding path length. As the last line of defense, the sink is required to verify the threshold-endorsement of each received report and discard those failing the test.

The choice of p_s represents a tradeoff between the early filterability of bogus reports and the computational overhead involved in verifying authentic reports. On the one hand, if p_s is too small, a bogus report will statistically traverse more hops before being filtered. On the other hand, if p_s is too large, it may incur unnecessary computational overhead on intermediate nodes in verifying authentic reports. p_s can be either fixed or dynamically adjusted as time goes on. For example, if the sink receives many alarms of bogus reports from sensor nodes or detects many unfiltered bogus reports by itself during a predetermined time period, it can increase p_s by a certain amount or else decrease it. The new p_s can be securely conveyed to sensor nodes using a μTESLA-like [34] broadcast authentication protocol.

Our LTE scheme has strong resilience against node compromise. It guarantees that, as long as there are less than t compromised nodes holding cell-key shares of a same cell, adversaries are unable to forge data reports that seem to originate from that cell and can escape the filtering by enroute intermediate nodes and the sink. In the worst-case scenario, adversaries may manage to compromise at least t nodes with cell-key shares of a same cell. We refer to this event as *cell compromise*. Fortunately, adversaries can only utilize the reconstructed cell key to fabricate reports in that cell but not in other cells, due to the location-dependent nature of the cell key. Therefore, if the sink initially accepts a report with a correct endorsement but finally finds that it is a bogus one by further field investigations or other means, the sink can immediately detect the cell-compromise event and take corresponding remedy actions that are outside the paper scope.

Adversaries might launch denial-of-service attacks by trapping legitimate nodes into endless verifications of data reports. Consequently, if a legitimate node detects too many bogus reports in a short time window, we assume that there are efficient ways for it to report such an abnormality to the sink. Another possible attack is that a compromised intermediate node may stall the reporting of real events to the sink by either directly dropping any received report or tampering with the report content before forwarding it to the next hop. This attack is orthogonal to the bogus data injection attack we focus on, but we would like to suggest several possible ways to withstand it. One way is to utilize a SPREAD-like [35] secure multipath routing protocol to transmit copies of a report along multiple disjoint paths to the sink. Another possible approach is through local monitoring enabled by the broadcast nature of radio transmissions. In particular, if an intermediate node receives a report from the prehop node, multiple neighbors of it can hear that packet as well. Likewise, these neighbors can overhear the packet it transmits to the next hop and, thus, be able to tell whether it behaves good or not. We leave the further investigation on this issue and its combination with the bogus data injection attack to a separate paper.

F. Performance Evaluation

In this section, we evaluate the performance of our LTE in achieving energy savings.

1) Pairing parameters: In our evaluation, the bilinear map $\hat{e}$ used is the Tate pairing [20]. The elliptic curve $\mathbb{E}$ is defined over $\mathbb{F}_p$, where p is a 512-bit prime. The order q of $\mathbb{G}_1$ and $\mathbb{G}_2$ is a 160-bit prime. According to [19], our chosen parameters deliver an equivalent level of security to that of 1024-bit RSA.

We use the following method to quantify the computation time and energy consumption of the Tate pairing. We assume that the sensor CPU is a low-power high-performance 32-bit Intel PXA255 processor at 400 MHz. The PXA255 has been widely used in many sensor products such as Sensoria WINS

3.0 and Crossbow Stargate. According to [36], the typical power consumption of PXA255 in active and idle modes are 411 and 121 mW, respectively. It was reported in [37] that it takes 752 ms to compute the Tate pairing with the similar parameters as ours on a 32-bit ST22 smartcard microprocessor at 33 MHz. Therefore, the computation of the Tate pairing on PXA255 roughly needs $33/400 \times 752 \approx 62.04$ ms, and the energy consumption E_p is approximately 25.5 mJ.

2) Overhead Analysis: For an authentic report forwarded along a ξ-hop path, LTE statistically involves ξp_s filtering operations, while it takes only one filtering operation to detect and dump a bogus report. A filtering operation requires one exponentiation in $\mathbb{G}_2$, one hash function evaluation and two evaluations of the Tate pairing. Due to the stationarity of sensor nodes, each sensor is more likely to forward reports from the same set of cells. As a result, each node can evaluate a limited set of values $\{\hat{e}(H(m \parallel n), W_{\text{pub}})\}$ beforehand, each corresponding to a potential cell from which a report may come from. By doing so, one of the pairing evaluations can be eliminated. As noted in [33], the pairing evaluation by far takes the most running time of a filtering operation. Thus, for the sake of simplicity, we use E_p to approximate the energy consumption of an enroute filtering operation.

Our LTE requires each report to carry a threshold-endorsement of format $(\Upsilon_{m,n}, h(\Lambda \parallel \theta))$ in addition to the normal fields. Since $\Upsilon_{m,n}$ is a point of $\mathbb{E}/\mathbb{F}_p$, only one of its X and Y coordinates needs to be transmitted because the other can be easily derived using the curve equation, resulting in an overhead of 512 bits. Also, assume that the hash function h is implemented using SHA-1 [21] with a 20-byte output. Then, the total packet overhead introduced by LTE is $L_o = 84$ bytes to achieve a high level of security as that of 1024-bit RSA.

3) Energy Savings: Our LTE aims to save the energy of intermediate nodes along the forwarding path to the sink through its early detection and dropping of bogus data reports. On the other hand, the introduced packet overhead and the probabilistic enroute filtering operations incur both communication and computation energy consumption. In the following, we employ a similar model to that of [4] to analyze the energy savings caused by LTE. For the sake of simplicity, we ignore the energy consumption of the report generation process, which is considered to be negligible as compared with that of transmitting it to the distant sink.

We denote by E_{tr} the hop-wise energy consumption for transmitting and receiving one byte. As reported in [38], a Chipcon CC1000 radio used in Xrossbow MICA2DOT motes consumes 28.6 and 59.2 μJ to receive and transmit one byte, respectively, at an effective data rate of 12.4 kb/s. Thus, we have $E_{\text{tr}} = 87.8$ μJ, which is used as an exemplary value throughout our evaluation.

We also denote by L_n the byte length of an original data report without using LTE, and by ξ the average number of hops an original report travels toward the sink. To simplify our evaluation, we assume that L_n is fixed to be 256 bytes. We further assume that the ratio of legitimate data traffic to bogus data traffic is $1 : \rho$ and ρ is called the *bogus traffic ratio* hereafter. As mentioned before, our LTE spends ξp_s filtering operations in verifying an authentic report, while merely one filtering operation to sift a bogus report. Let E_{sum} and E'_{sum} be the normalized energy consumed to deliver all the traffic without and with LTE in place, respectively. Then, we have

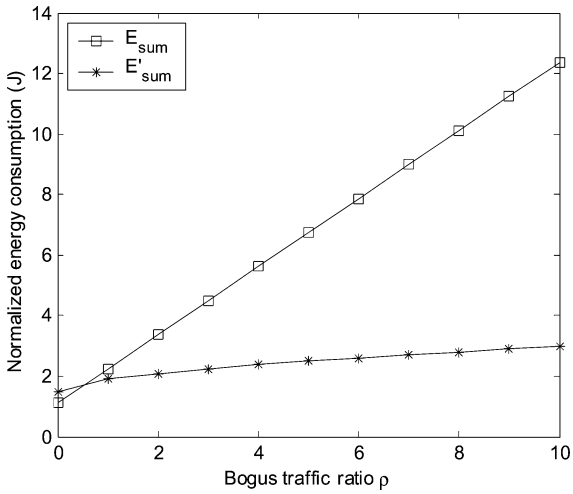


Fig. 3. Comparison of E_{sum} and E'_{sum} as a function of the bogus traffic ratio ρ, where $\xi = 50$ and the optimal p_s's are used.

$$E_{\text{sum}} = L_n E_{\text{tr}} \xi (1 + \rho) \tag{8}$$

and

$$\begin{aligned} E'_{\text{sum}} &= (L_n + L_o) E_{\text{tr}} (\xi + \rho \bar{\mu}) + (\xi p_s + \rho) E_p \\ &= (L_n + L_o) E_{\text{tr}} \left(\xi + \frac{\rho}{p_s} \right) + (\xi p_s + \rho) E_p \\ &\geq (L_n + L_o) E_{\text{tr}} \xi + \rho E_p + 2\sqrt{(L_n + L_o) E_{\text{tr}} \rho \xi E_p} \end{aligned} \tag{9}$$

with equality if and only if $p_s = \sqrt{(L_n + L_o) E_{\text{tr}} \rho / \xi E_p}$.

Fig. 3 compares E_{sum} with E'_{sum}, where the optimal p_s's are used and $\xi = 50$. We can see that E_{sum} increases dramatically along with the increase of bogus data reports, while E'_{sum} always maintains a rather stable level. The reason is that most bogus reports can be detected and dropped during their early transmission stages with LTE in place. In addition, when there is no bogus traffic, our LTE increases the energy consumption by about 32% due to the introduced packet overhead. However, when the bogus traffic starts to exceed the legitimate traffic, LTE demonstrates growingly remarkable energy savings. For example, when $\rho = 2$ and 5, our LTE saves more than 37% and 63% of energy, respectively.

In most WSN applications, data delivery is event-driven and legitimate traffic occurs only when some events of interest appear in the sensor field. In contrast, to increase the impact of their attacks, adversaries often inject into the network a large amount of bogus traffic, which is often several orders of magnitude greater than that of legitimate traffic [4]. Our LTE is particularly useful for these scenarios in saving a great deal of energy by early filtering bogus data reports.

In reality, it is often difficult to obtain an accurate estimate of the bogus traffic ratio ρ. Therefore, to some extent, Fig. 3 reflects the upper-bound performance of our LTE. There are

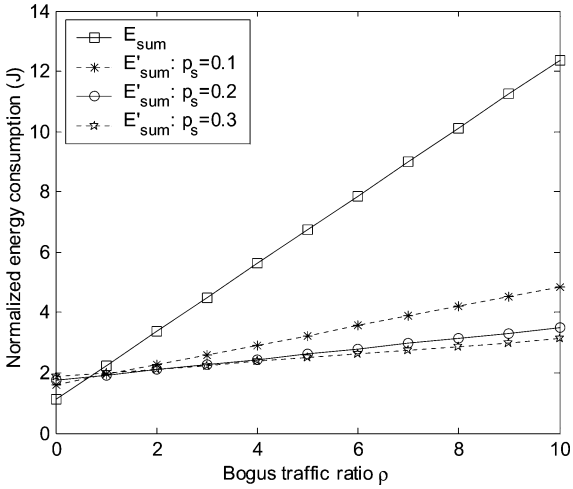


Fig. 4. Comparison of $E_{\rm sum}$ and $E'_{\rm sum}$ as a function of the bogus traffic ratio ρ, where $\xi = 50$ and nonoptimal p_s's are used.

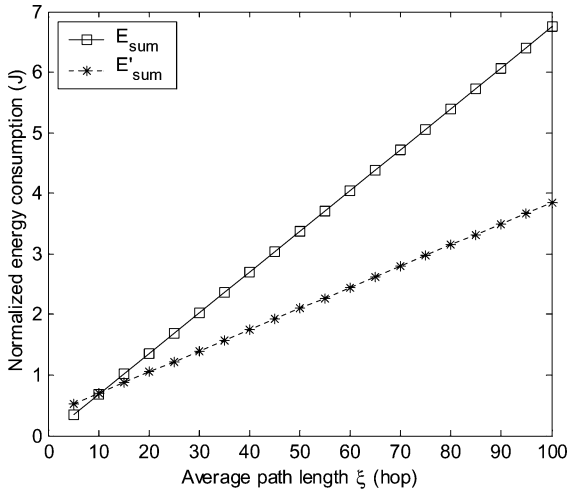


Fig. 5. Comparison of $E_{\rm sum}$ and $E'_{\rm sum}$ as a function of the average path length ξ, where $\rho = 2$ and $p_s = 0.2$.

two possible ways to approach this upper bound. In the first approach, the sink estimates the current ρ based on the received reports and possible alarms from sensor nodes. It then derives the optimal sampling probability p_s, which is conveyed to sensor nodes using a μTESLA-like [34] broadcast authentication protocol. The other approach is for each node itself to estimate the ρ as the ratio of bogus traffic to legitimate traffic in the total traffic sampled during a certain period. Then, it can compute the new p_s locally optimal to itself.

Even if without using an optimal p_s, the energy savings resulting from our LTE are still remarkable. Fig. 4 depicts the case that nonoptimal values of p_s are used. The advantages of using our LTE are quite obvious under all the three sampling probabilities. Another observation is that, when ρ becomes larger, p_s should be increased as well in order to filter bogus data reports as early as possible. Likewise, the new p_s can either be determined by the sink as a network-wide common value, or be decided individually by each node based on its local observations.

Next, we investigate the impact of the average path length ξ on the energy-saving performance of LTE. As can be seen from Fig. 5, the further the originating cells of bogus data reports are away from the sink, the more energy savings our LTE can achieve. We note that adversaries may inject bogus data reports to consume the energy resources of the nodes that are only several hops away from the sink. For this case, our LTE might not achieve the desirable objective because the energy savings from early filtering bogus reports may be offset by the energy consumption incurred by our scheme. However, bogus reports injected in the distant cells away from the sink are much more detrimental than those injected in the sink's vicinity because their transmissions involve many more intermediate nodes. In addition, we believe that it is much easier for the sink to detect the bogus data injection attack mounted in its vicinity than in the distant cells.

VI. Related Work

Recent years have witnessed growing interest in sensor network security. Due to space limitations, here we merely discuss prior art that is more germane to this paper.

How to set up a pairwise shared key between two sensors is a topic which by far has attracted extensive attention. As a pioneering solution, Eschenauer and Gligor propose a probabilistic key predistribution scheme [7]. The main idea is to preload each sensor with a random subset of keys from a global key pool in a way that any two nodes can share at least one common key with a certain probability. This scheme has been improved later by several other proposals such as [8]–[10] in terms of network connectivity, memory usage, and resilience against node compromise, among others. Unfortunately, these probabilistic schemes suffer from a few drawbacks that may limit their potential in large-scale WSNs demanding a high level of security.

First of all, as noted in [39], these schemes are vulnerable to node compromise attacks in that adversaries who compromised sufficiently many nodes could also obtain a large fraction of pairwise keys shared between noncompromised nodes. Second, they are subject to all the attacks discussed in Section IV. Third, they are designed to establish pairwise shared keys among neighboring nodes. As a result, they are both inefficient and insecure in setting up a pairwise key shared between two nonneighboring nodes or two neighboring nodes without a priori shared knowledge. Fourth, most of them fail to provide secure neighborhood authentication, which is prerequisite for guaranteeing link-level security. Although the random pairwise keys scheme in [8] offers mutual authentication between two neighbors having a preloaded pairwise key, the resulting cost is the much restricted supportable network size [3]. Fifth, these schemes all have an upper limit on the network size and often require each node to store tens or even hundreds of keys, leading to the poor network scalability. Finally, all of them do not offer support for nonrepudiation of digital signatures, which is one of the fundamental security requirements.

As compared with the above schemes, our schemes enable deterministic, secure and efficient establishment of a shared key between any two network nodes, be they immediate neighbors or multiple hops apart. Our IPK and MPK establishment methods both have perfect resilience against node compromise because of their reliance on the private LBK's of individual nodes. In addition, our schemes can not only limit the impact of compromised nodes to their vicinity, but also withstand other notorious attacks like those mentioned in Section IV.

Moreover, our schemes provide secure location-based neighborhood authentication and support nonrepudiation of digital signatures. Furthermore, our schemes merely require each node to memorize its own IBK and LBK, and allow the addition of an arbitrary number of new nodes.

Some other proposals [11]–[14] propose to use the known deployment information to facilitate more secure and efficient pairwise key establishment. These solutions still belong to the category of the probabilistic schemes, thereby suffering from either some or even all of the aforementioned drawbacks. In addition, concrete geographic locations of individual nodes are not used in all of them. More recently, Lazos *et al.* [40] present a location-based solution to deal with the wormhole attack. This solution addresses neither the establishment of multihop pairwise keys, nor the issue of node addition (or the network scalability issue).

Aside from the probabilistic schemes, another notable work called LEAP is proposed by Zhu *et al.* in [27]. In LEAP, each node is preloaded with a global shared secret, through which it can authenticate neighboring nodes and establish pairwise shared keys with them once deployed. However, the MPK establishment method of LEAP suffers from both the significant communication overhead and the vulnerability to the compromise of intermediate nodes. In addition, LEAP does not support nonrepudiation of digital signatures.

We are aware of two existing solutions to the bogus data injection attack, namely, SEF [4] and IHA [5]. Both schemes can achieve the same objective of energy savings as our LTE by detecting and dropping bogus reports as early as possible. However, adversaries who compromised nodes carrying keys from t different key partitions can render SEF completely useless, as noted in [4]. Likewise, IHA breaks down once adversaries compromise over t nodes and, thus, are able to forge data reports seeming to originate from arbitrary network locations. In a large-scale WSN with many more than t nodes, however, it seems unlikely to prevent adversaries from compromising over t nodes. In addition, IHA suffers from the considerable communication overhead in maintaining the per-route interleaved structure of nodes as compared with both SEF and our LTE. By comparison, our LTE is able to localize the impact of compromised nodes to their vicinity due to its location-dependent nature. It can tolerate the compromise of up to $(t-1)$ nodes holding cell-key shares of the same cell and, thus, many more nodes regarding the whole network. Therefore, our LTE exhibits much better compromise-tolerant performance than both SEF and IHA.

There are many other related work in sensor network security. Carman *et al.* [41] investigate the performance of a number of key management schemes over different hardware platforms. Basagni *et al.* [6] utilize tamper-resistant hardware in periodically updating the key shared by all the nodes. Perrig *et al.* [34] propose SNEP, a protocol for data confidentiality and two-party data authentication, and μTESLA, a protocol for broadcast data authentication. μTESLA is further improved by Liu and Ning in [42]. Przydatek *et al.* [43] construct efficient random sampling mechanisms and interactive proofs to ensure secure information aggregation in WSNs. Karlof and Wagner [1] discuss various attacks against existing sensor network routing protocols and point out some possible solutions. Newsome *et al.* [3] analyze in detail the impact of the Sybil attack on sensor networks and propose several defenses.

VII. Discussion

In this section, we discuss the use of symmetric-key versus public-key cryptography (PKC) in WSNs.

It was a common belief that PKC is too complex, slow and power hungry and, thus, ill-suited for use in resource-constrained WSNs. For this reason, PKC has often been ruled out for securing WSNs and most previous proposals such as [7]–[14] are purely based on symmetric-key cryptography. However, many researchers [38], [44]–[47] have recently challenged this belief by showing that traditional PKC such as RSA or elliptic-curve cryptography is rather viable in WSNs.

Moreover, we have mentioned previously that the pure symmetric-key solutions have a number of drawbacks due to the inherent limitations of symmetric-key cryptography. In addition, they may not be so energy efficient as they are claimed to be. For example, most of the probabilistic key predistribution schemes such as [7]–[10] require a secure "puzzle-solving" method to set up a shared key between two neighboring nodes. In particular, one node broadcasts a key-discovery message containing a challenge α and m ciphertexts $\{\alpha\}_{k_i}$ for $i = 1, \ldots, m$, where k_i is a potential pairwise key the other node may have. If the other node can correctly decrypt any of the m ciphertexts, it can establish a pairwise key with the broadcasting node. Since there are often several tens or even hundreds of potential pairwise keys, the total energy consumption caused by communication and symmetric-key encryption and decryption operations may have been already higher than that of a public-key solution. Therefore, we believe that it is both necessary and feasible to design public-key solutions for security-sensitive WSNs to establish shared keys for subsequent use with efficient symmetric-key algorithms.

Our proposed schemes are public-key solutions built upon the pairing-based IBC, which is more appropriate than traditional PKC for WSNs (cf. Section II-A). Therefore, our schemes eliminate the need for transmitting and verifying conventional public-key certificates. As an emerging technique, IBC is under rapid development. For example, according to the recent result in [48], the Tate pairing can be evaluated up to ten times faster than previously reported implementations. We have also been aware of the efficient hardware implementations of the Tate pairing on smartcards [37], PDAs [49], and FPGAs [50]. The real implementation of the pairing on sensor node hardware is part of our ongoing work.

VIII. Conclusion

To counteract the impact of compromised nodes, this paper presents a comprehensive set of location-based compromise-tolerant security mechanisms for WSNs. We first propose the notion of LBKs by binding private keys of individual nodes to both their IDs and concrete geographic locations. We then develop an LBK-based neighborhood authentication protocol which is able to constrain the impact of compromised nodes to their vicinity. We also present efficient methods to set up pairwise shared keys between any two network nodes, be they di-

rect neighbors or multihop away. In addition, we demonstrate the capability of LBKs in withstanding some notorious attacks against WSNs. Moreover, we design a LTE scheme to filter bogus traffic injected by adversaries during their early transmission stages. The remarkable energy savings resulting from LTE have been confirmed by detailed performance evaluation. As the future research, we plan to evaluate the performance of the proposed schemes in real sensor platforms. We also intend to further investigate the potential applications of LBKs in WSNs, such as misbehavior detection, secure distributed storage, secure routing, and target tracking.

REFERENCES

[1] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Netw.*, vol. 1, no. 2, pp. 293–315, 2003.

[2] J. R. Douceur, "The Sybil attack," in *Proc. 1st Int. Workshop on Peer-to-Peer Syst.*, Cambridge, MA, Mar. 2002, pp. 251–260.

[3] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis & defenses," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, Berkeley, CA, Apr. 2004, pp. 259–268.

[4] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004, pp. 2446–2457.

[5] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proc. IEEE Symp. Security Privacy*, Oakland, CA, May 2004, pp. 259–271.

[6] E. R. S. Basagni, K. Herrin, and D. Bruschi, "Secure pebblenets," in *Proc. ACM MobiHoc*, Long Beach, Oct. 2001, pp. 256–263.

[7] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. ACM CCS*, Washington, DC, Nov. 2002, pp. 41–47.

[8] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. IEEE Symp. Security, Privacy*, Oakland, CA, May 2003, pp. 197–213.

[9] W. Du, J. Deng, Y. Han, and P. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *Proc. ACM CCS*, Washington, DC, Oct. 2003, pp. 42–51.

[10] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proc. ACM CCS*, Washington, DC, Oct. 2003, pp. 52–61.

[11] ——, "Location-based pairwise key establishments for static sensor networks," in *Proc. ACM SASN*, Fairfax, VA, Oct. 2003, pp. 72–82.

[12] W. Du, J. Deng, Y. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004, pp. 586–597.

[13] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks," in *Proc. ACM SASN*, Washington, DC, Oct. 2004, pp. 29–42.

[14] Y. Zhou, Y. Zhang, and Y. Fang, "LLK: A link-layer key establishment scheme in wireless sensor networks," in *Proc. IEEE WCNC*, New Orleans, LA, Mar. 2005, pp. 1921–1926.

[15] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," in *Proc. ACM SIGCOMM Workshop Data Comm. Latin America and the Caribbean*, Costa Rica, Apr. 2001, pp. 20–41.

[16] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM MobiCom*, Boston, MA, Aug. 2000, pp. 243–254.

[17] K. Barr and K. Asanovic, "Energy aware lossless data compression," in *Proc. 1st Int. Conf. Mobile Syst., Applicat., Services*, San Francisco, CA, May 2003, pp. 231–244.

[18] A. Shamir, "Identity based cryptosystems and signature schemes," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 1984, vol. 196, Proc. CRYPTO, pp. 47–53.

[19] D. Boneh and M. Franklin, "Identify-based encryption from the Weil pairing," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 2001, vol. 2139, Proc. CRYPTO, pp. 213–229.

[20] P. Barreto, H. Kim, B. Bynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 2002, vol. 2442, Proc. CRYPTO, pp. 354–368.

[21] *Digital Hash Standard*, Federal information processing standards publication 180-1, Apr. 1995.

[22] S. Čapkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. IEEE INFOCOM*, Miami, FL, March 2005, pp. 1917–1928.

[23] Y. Zhang, W. Liu, and Y. Fang, "Secure localization in wireless sensor networks," in *Proc. IEEE MILCOM*, 2005.

[24] L. Lazos and R. Poovendran, "Serloc: Secure range-independent localization for wireless sensor networks," in *Proc. ACM WiSe*, Philadelphia, PA, Oct. 2004, pp. 21–30.

[25] D. Liu, P. Ning, and W. Du, "Attack-resistant location estimation in sensor networks," in *Proc. IPSN*, Apr. 2005, pp. 99–103.

[26] W. Du, L. Fang, and P. Ning, "LAD: Localization anomaly detection for wireless sensor networks," in *Proc. IPDPS*, Denver, CO, Apr. 2005.

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

[28] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings," Cryptology ePrint Archive, 2002.

[29] Y. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2003, pp. 1976–1986.

[30] S. Kumar, T. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proc. ACM MobiCom*, Philadelphia, PA, Sep./Oct. 2004, pp. 144–158.

[31] A. Shamir, "How to share a secret," in *Commun. ACM*, vol. 22, 1979, pp. 612–613.

[32] J. Baek and Y. Zheng, "Identity-based threshold signature from the bilinear pairings," in *Proc. Int. Conf. Inf. Tech.: Coding Comput.*, Las Vegas, NV, Apr. 2004, pp. 124–128.

[33] F. Hess, "Efficient identity based signature schemes based on pairings," in *Proc. SAC*, St. John's, NF, Canada, Aug. 2002, pp. 310–324.

[34] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. Culler, "Spins: Security protocols for sensor networks," *ACM Wireless Netw.*, pp. 521–534, Sep. 2002.

[35] W. Lou, W. Liu, and Y. Fang, "SPREAD: Enhancing data confidentiality in mobile ad hoc networks," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004, pp. 2404–2413.

[36] Intel PXA255 Processor Electrical, Mechanical, and Thermal Specification, http://www.intel.com/design/pca/applicationsprocessors/manuals/278780.h% tm.

[37] G. Bertoni, L. Chen, P. Fragneto, K. Harrison, and G. Pelosi1, "Computing Tate pairing on smartcards," *White Paper STMicroelectronics*, 2005. [Online]. Available: http://www.st.com/stonline/products/families/smartcard/ast_ibe.htm.

[38] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Chang, "Energy analysis for public-key cryptography for wireless sensor networks," in *Proc. IEEE PerCom*, Pisa, Italy, Mar. 2005.

[39] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," in *Commun. ACM*, vol. 47, Jun. 2004, pp. 53–57.

[40] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. Chang, "Preventing wormhole attacks on wireless ad hoc networks: A graph theoretic approach," in *Proc. IEEE WCNC*, New Orleans, LA, Mar. 2005, pp. 1193–1199.

[41] D. Carman, P. Kruus, and B. Matt, "Constraints and approaches for distributed sensor network security," NAI Labs, Tech. Rep. 00-010, Sep. 2000.

[42] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," in *Proc. NDSS*, San Diego, CA, Feb. 2003, pp. 263–276.

[43] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure information aggregation in sensor networks," in *Proc. ACM SenSys'03*, Los Angeles, CA, Nov. 2003, pp. 255–265.

[44] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *Proc. IEEE SECON*, Santa Clara, CA, Oct. 2004, pp. 71–80.

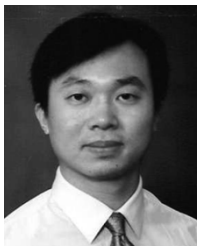
[45] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUS," in *Proc. CHES*, Boston, MA, Aug. 2004, pp. 119–132.

[46] R. Watro, D. Kong, S. fen Cuti, C. Gardiner, C. Lynn, and P. Kruus, "Tinypk: Securing sensor networks with public key technology," in *Proc. ACM SASN*, Washington, DC, Oct. 2004, pp. 59–64.

[47] G. Gaubatz, J. Kaps, and B. Sunar, "Public keys cryptography in sensor networks—revisited," in *Proc. ESAS*, Heidelberg, Germany, Aug. 2004, pp. 2–18.

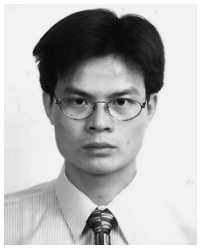
[48] P. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 2004, vol. 3006, Proc. Sel. Areas Cryptography, pp. 17–25.

[49] M. Scott, "Computing the Tate pairing," in *Proc. Cryptographers' Track at the RSA Conf.*, San Francisco, CA, Feb. 2005, pp. 293–304.
[50] T. Kerins, W. Marnane, E. Popovici, and P. Barreto, "Efficient hardware for the Tate pairing calculation in characteristic three," in *Proc. Workshop on Cryptographic Hardware and Embedded Syst.*, Edinburgh, Scotland, Aug./Sep. 2005, pp. 412–426.



Yanchao Zhang (S'03) received the B.E. degree in computer communications from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1999, and the M.E. degree in computer applications from Beijing University of Posts and Telecommunications, Beijing, China, in 2002. He is currently working towards the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Florida, Gainesville.

His research focuses on security, trust, and privacy in network and distributed systems, with current emphases on mobile ad hoc networks, wireless sensor networks, wireless mesh networks, and hybrid wired/wireless networks.

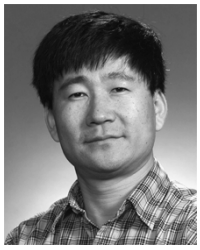


Wei Liu received the B.E. and M.E. degrees in electrical and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 2005.

Currently, he is a Senior Technical Member with Scalable Network Technologies. His research interest includes cross-layer design, and communication protocols for mobile ad hoc networks, wireless sensor networks, and cellular networks.

Wenjing Lou (S'01–M'03) received the M.A.Sc. degree from Nanyang Technological University, Singapore, in 1998, the B.E. degree in computer science and engineering, the M.E. degree from Xi'an Jiaotong University, China, in 1993 and 1996, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 2003.

She is an Assistant Professor in the Electrical and Computer Engineering Department, Worcester Polytechnic Institute, Worcester, MA. From 1997 to 1999, she worked as a Research Engineer in the Network Technology Research Center, Nanyang Technological University. Her current research interests are in the areas of ad hoc and sensor networks, with emphases on network security and routing issues.



Yuguang Fang (S'95–M'99–SM'99) received the B.S. and M.S. degrees in mathematics from Qufu Normal University, Qufu, Shandong, China, in 1984 and 1987, respectively, the Ph.D. degree in systems and control engineering from the Department of Systems, Control and Industrial Engineering, Case Western Reserve University, Cleveland, OH, in 1994, and the Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, Boston University, Boston, MA, in 1997.

From 1987 to 1988, he held Research and a Teaching position in both the Department of Mathematics and the Institute of Automation, Qufu Normal University. From 1989 to 1993, he was a Teaching/Research Assistant in the Department of Systems, Control and Industrial Engineering, Case Western Reserve University, where he held a Research Associate position from January 1994 to May 1994. He held a Postdoctoral position in the Department of Electrical and Computer Engineering, Boston University from 1994 to 1995. From 1995 to 1997, he was a Research Assistant in the Department of Electrical and Computer Engineering, Boston University. From 1997 to 1998, he was a Visiting Assistant Professor in the Department of Electrical Engineering, University of Texas at Dallas. From 1998 to 2000, he was an Assistant Professor in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark. In May 2000, he joined the Department of Electrical and Computer Engineering, University of Florida, Gainesville, where he received an early promotion to Associate Professor with tenure in August 2003, and to Full Professor in August 2005. He has published over 150 papers in refereed professional journals and conferences. His research interests span many areas including wireless networks, mobile computing, mobile communications, wireless security, automatic control, and neural networks.

Dr. Fang is a member of the Association for Computing Machinery (ACM). He received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He also received the 2001 CAST Academic Award. He is listed in *Marquis Who's Who in Science and Engineering*, *Who's Who in America*, and *Who's Who in the World*. He has actively engaged in many professional activities. He is an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON MOBILE COMPUTING, *ACM Wireless Networks*, and the *IEEE Wireless Communications*. He was an Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS: Wireless Communications Series, an Area Editor for *ACM Mobile Computing and Communications Review*, an Editor for *Wiley International Journal on Wireless Communications and Mobile Computing*, and Feature Editor for Scanning the Literature in *IEEE Personal Communications*. He has also been actively involved with many professional conferences such as ACM MobiCom'02 (Committee Co-Chair for Student Travel Award), MobiCom'01, IEEE INFOCOM'06, INFOCOM'05 (Vice-Chair for Technical Program Committee), INFOCOM'04, INFOCOM'03, INFOCOM'00, INFOCOM'98, IEEE WCNC'04, WCNC'02, WCNC'00 (Technical Program Vice-Chair), WCNC'99, IEEE Globecom'04 (Symposium Co-Chair), Globecom'02, and the International Conference on Computer Communications and Networking (IC3N) (Technical Program Vice-Chair).