Securing Sensor Networks with Location-Based Keys

Yanchao Zhang*, Wei Liu*, Wenjing Lou† and Yuguang Fang*
*Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611
Email: {yczhang@, liuw@, fang@ece}ufl.edu
†Department of Electrical and Computer Engineering
Worcester Polytechnic Institute, Worcester, MA 01609
Email: wjlou@ece.wpi.edu

***Abstract*— Wireless sensor networks are often deployed in unattended and hostile environments, leaving individual sensors vulnerable to security compromise. This paper proposes the novel notion of *location-based keys* for designing compromise-tolerant security mechanisms for sensor networks. Based on location-based keys, we develop a node-to-node authentication scheme, which is not only able to localize the impact of compromised nodes within their vicinity, but also to facilitate the establishment of pairwise keys between neighboring nodes. Compared with previous proposals, our scheme has perfect resilience against node compromise, low storage overhead, and good network scalability. We also demonstrate the use of location-based keys in combating a few notorious attacks against sensor network routing protocols.**

I. Introduction

Many sensor network applications demand security, especially when the sensor network is deployed to protect or monitor critical infrastructures. In large-scale untethered sensor networks, however, it is quite difficult to prevent each individual node from either physical or logical attack. This situation poses the demand for *compromise-tolerant* security mechanisms. That is, the rest of the network should remain secure even when a few nodes are compromised and their cryptographic materials are exposed to adversaries.

One of the fundamental problems in sensor network security is how to bootstrap secure communications, i.e., how to establish pairwise keys between neighboring nodes. In their seminal paper [1], Eschenauer and Gligor proposed a probabilistic key pre-distribution scheme, in which each node is pre-loaded with a random subset of keys from a global key pool in such a way that any two nodes can share at least one common key with a certain probability. Subsequently, several other proposals [2]–[6] were proposed to improve [1] in many aspects such as network connectivity, memory usage, and network resilience against node compromise among others. Though beautiful in theory, the probabilistic key pre-distribution schemes have several disadvantages that may impede their practical use. First, a small number of compromised nodes may expose a large fraction of pairwise keys between non-compromised nodes. Even worse, adversaries who compromised sufficiently many nodes could reconstruct the complete key pool and thus break the scheme, as pointed out in [7]. Second, most of these schemes fail to provide node-to-node authentication between neighboring nodes, which is indispensable for guaranteeing link-layer security. Third, but not the last, these schemes often assume a pre-planned fixed network size and require each node to store sometimes up to hundreds of keys to attain a desired network connectivity, thus leading to the poor network scalability.

In contrast to mobile ad hoc networks, most sensor networks have an intrinsic property that nodes are stationary, that is, fixed at where they are deployed. As a result, nodes[1] can be addressed with their geographic locations instead of traditional IP-type addresses or meaningless node IDs. Nodal location information has played an important role in many sensor network applications, including target tracking, geographic routing, location directory service, etc. However, its potential in securing sensor networks has so far received little attention.

Our contributions in this paper are mainly threefold. First, we propose the novel notion of *location-based* keys (LBKs), each of which corresponds to a node's unique geographic location. Second, we design a node-to-node neighborhood authentication scheme using LBKs, which is not only able to localize the impact of compromised nodes within their vicinity, but also to establish pairwise shared keys between neighboring nodes at the same time. Last, we demonstrate how LBKs can act as efficient countermeasures against a few notorious attacks on sensor network routing protocols.

II. Preliminaries

A. Pairing Concept

Pairing has recently found a number of interesting applications in cryptography and it is the cryptographic foundation of this paper. Let $\mathbb{G}_1, \mathbb{G}_2$ be two groups of the same primer order q. We view $\mathbb{G}_1$ as an additive group and $\mathbb{G}_2$ as a multiplicative group throughout the paper. Pairing is a computable *bilinear*

This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464 and under grant N000140210554.

[1] In this paper, we use the terms sensors, sensor nodes, and nodes interchangeably.

map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ if $\forall\ P, Q, R, S \in \mathbb{G}_1$, we have

$$\hat{e}(P+Q, R+S) = \hat{e}(P,R)\hat{e}(P,S)\hat{e}(Q,R)\hat{e}(Q,S).^2 \quad (1)$$

Modified Weil pairing [8] and Tate pairing [9] on supersingular elliptic curves are examples of such bilinear maps, for which the *Bilinear Diffie-Hellman Problem* (BDHP) is believed to be hard, i.e., it is believed that, given $< P, xP, yP, zP >$ for random $x, y, z \in \mathbb{Z}_q^{*3}$ 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 non-negligible probability. We refer readers to [8], [9] for further details on pairing.

B. Adversarial Model

Adversaries in sensor networks can be classified into two categories, namely, *external* adversaries and *internal* adversaries. The former are outside the sensor network and may just perform passive eavesdropping on data transmissions, or inject bogus data or routing messages into the network to consume network resources. In contrast, internal adversaries might be either compromised nodes running malicious code or adversaries who stole the cryptographical materials from legitimate nodes. Internal adversaries can mount more subtle attacks and are more difficult to defend against than external adversaries. This paper aims to offer countermeasures against both external and internal adversaries.

III. Location-based Security Schemes

A. Pre-deployment Phase

We examine a sensor network consisting of hundreds of or thousands of low-end untethered stationary sensors. We assume that sensors have the same transmission range *R* and communicate with each other via bi-directional wireless links. Sensors perform a collaborative monitoring of the designated sensor field and report the sensed events to the distant sink, which is a well-protected data collection center with sufficient processing capabilities and resources. We further assume that each node has a unique, integer-valued, non-zero ID. In view of the cost constraints, sensors are assumed to be not tamper-resistant in the sense that adversaries can extract all the cryptographic materials and data stored on a compromised node, thus having full access to the contents of messages forwarded by a compromised node.

Before deployment, we assume that a trusted authority (TA), e.g., the system administrator or network planner, first determines[4] two *q*-order cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$, one bilinear map $\hat{e}$, a system master-key $\kappa \in \mathbb{Z}_q^*$, a generator W of $\mathbb{G}_1$, and two collision-resistant cryptographic hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ mapping arbitrary strings to elements in $\mathbb{G}_1$, and $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_\beta^*$ mapping arbitrary strings to integers in $\mathbb{Z}_\beta^*$ ($0 < \beta \leq q$), e.g., SHA-1 [10]. After that, the TA pre-loads each node and the sink with the public system parameters $<\mathbb{G}_1, \mathbb{G}_2, \hat{e}, W, H_1, H_2>$, but only the sink with κ. The security of our schemes depends on the secrecy of κ, which is only known to the well-protected sink and the TA who does not appear in the sensor field.

Moreover, each node, say A, obtains from the TA an ID-based key, or IBK for short, $IK_A = \kappa H_1(ID_A) \in \mathbb{G}_1$. Since the discrete log problem (DLP)[5] is believed to be hard in $\mathbb{G}_1$ [8], given a $< ID_A, IK_A >$ pair, adversaries cannot deduce the system master-key κ with non-negligible probability. As a result, even if compromising an arbitrary number of nodes and their IBKs, adversaries are unable to exploit the acquired knowledge to calculate the IBKs of the remaining non-compromised nodes.

B. Sensor Deployment and Localization

Corke *et al*. [11] presented an interesting, practical approach by using mobile robots to deploy and localize individual sensors, i.e., providing per-node location information. The efficacy of their scheme has been justified through field study. For simplicity, we assume a similar approach in this paper.[6]

During the deployment phase, regular sensors are dropped from a helicopter or a Unmanned Aerial Vehicle (UAV) into the field of interest. Sensors cover the designated field uniformly but not necessarily regularly. Subsequently, a few mobile robots, either ground or flying, are dispatched to sweep across the whole sensor field along pre-planned paths. Mobile robots have GPS capabilities as well as more powerful computation and communication capacities than regular sensors. Each robot is also equipped with a sensor that allows the robot to communicate with the rest of the network. Mobile robots are capable of collectively or independently determining the geographic location of each individual sensor.

In the end, each sensor, say *A*, obtains its unique geographic location pos_A from mobile robots. For convenience only, we postulate that there is no location errors and a location is encoded as two two-byte quantities, for *x* and *y* coordinate values, e.g., $pos_A =< x_A, y_A >$. Provided that there is no overlapping sensors having the same coordinates, each sensor can be uniquely identified and addressed by its location rather than its ID. In other words, each node uses its own location to identify itself when performing such network tasks as routing and collaborative monitoring. In the rest of this paper, we refer to node pos_A as the node who has the location pos_A.

C. Generation and Distribution of Location-Based Keys

In addition to its IBK, each node, say A, should possess a location-based key, or LBK for short, $LK_A = \kappa H_1(pos_A) = \kappa H_1(x_A \parallel y_A) \in \mathbb{G}_1$, where "$\parallel$" denotes the concatenation of messages. To do this, mobile robots are equipped with both the above system parameters $<\mathbb{G}_1, \mathbb{G}_2, \hat{e}, W, H_1, H_2>$ and

[2] In particular, $\forall\ P, Q \in \mathbb{G}_1,\ \forall\ a, b \in \mathbb{Z}_q^*,\ \hat{e}(aP, bQ) = \hat{e}(aP, Q)^b = \hat{e}(P, bQ)^a = \hat{e}(P, Q)^{ab}$ etc.

[3] $\mathbb{Z}_q^*$ is the *multiplicative group* of integers modulo q. In particular, if *q* is a prime, $\mathbb{Z}_q^* = \{a \mid 1 \leq a \leq q-1\}$.

[4] Boneh and Franklin gave some guidelines on how to choose the pairing parameters in their seminal paper [8].

[5] The DLP in the additive group $\mathbb{G}_1$ is to find an integer $n \in \mathbb{Z}_q^*$ for two given group elements *P* and *Q* such that $Q = nP$ whenever such an integer exists.

[6] An alternative approach is to physical install and hence localize sensors one by one, though it may be unrealistic in many cases.

the system master-key κ so that they can generate LBKs for individual nodes[7]. Mobile robots should be well-protected and programmed in such a way that they should be able to erase the system-master key κ completely after the deployment phase.

This paper is targeted for sensor networks that are used to protect or monitor critical infrastructures. In such structural monitoring applications, it would be a reasonable assumption that the sensor field is under super surveillance only during the deployment phase which usually does not last too long. That is, we assume that adversaries do not actively catch or attack either mobile robots or individual sensor nodes in this phase because otherwise they would run a high risk of exposing themselves. However, adversaries might be able to perform passive eavesdropping on data transmissions. They might also send bogus LBKs to sensor nodes. For this reason, mobile robots cannot simply send LBKs in plaintext to individual nodes and they are required to execute the following protocol:

$$\begin{aligned} T_1 \rightarrow A: &\quad \text{“}helloLBK\text{”} \text{ (broadcast)};\\ A \rightarrow T_1: &\quad ID_A \text{ (unicast)};\\ T_1 \rightarrow A: &\quad \{ID_A, pos_A, LK_A\}_{IK_A} \text{ (unicast)}, \end{aligned}$$

where T_1 denotes a robot and A is is one of the nodes to be initialized (we just use one node for the ease of explanation).

In the above protocol, T_1 first broadcasts a special “helloLBK” message to announce its existence. If seeing such a message and still uninitialized, node A responds with its identifier ID_A by unicast. Then T_1 proceeds to determine pos_A and generate $IK_A = \kappa H_1(ID_A)$ and $LK_A = \kappa H_1(pos_A)$. After that, T_1 encrypts ID_A, pos_A, and LK_A with the encryption key IK_A using any efficient secret-key function such as RC5 [13], and unicasts the ciphertext to node A. Node A can then decrypt the ciphertext with the pre-loaded IK_A. Note that, in the third step above, T_1 can also pack together the responses for several nodes and broadcast them in one message to the target sensors so that the communication overhead can be reduced.

The purpose of embedding ID_A in the ciphertext is to thwart the attack that an adversary may send a forged location/LBK pair to node A. Since adversaries do not have the knowledge of IK_A, they cannot form the appropriate ciphertext which would be decrypted to produce the correct first field ID_A. As a result, if its own ID_A does not match the first field of the decrypted result, A should discard the packet because it might come from an adversary. Otherwise, A accepts the packet and saves the embedded pos_A and LK_A for later use. Notice that an adversary might as well send to T_1 a forged node identifier, but it would be unable to interpret the following ciphertext from T_1 for the lack of the IBK corresponding to the claimed identifier. Furthermore, once initialized, node A should not respond to subsequent “helloLBK” messages which might come from either mobile robots or adversaries. It is obvious that our protocol guarantees the secure distribution of both node locations and the corresponding LBKs. This guarantee is extremely important for many sensor network applications, e.g., geographic routing [14], where secure locations of individual nodes are a must.

[7]A more reliable way would be to distribute κ among mobile robots using Shamir’s secret sharing technique [12]. For the lack space, we leave this extension in another separate paper.

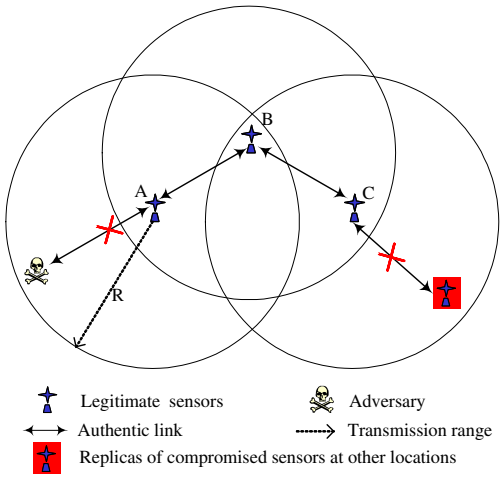

Fig. 1. Node-to-node mutual authentication between neighbors.

D. Location-based Node-to-Node Authentication

Mutual authentication between neighboring nodes is prerequisite in supporting many security services in wireless sensor networks. For example, a node should only accept and/or forward messages from authenticated neighbors. Otherwise, external adversaries can easily inject arbitrary broadcast messages into the network to deplete the scarce network resources.

During the post-deployment phase, each node is required to discover and perform mutual authentication with neighboring nodes. Fig. 1 shows an example of neighborhood authentication, where node B is the neighboring node of both A and C, while A and C are non-neighbors of each other.

To achieve mutual authentication with neighboring nodes, node A locally broadcasts an authentication request including its location $pos_A =< x_A, y_A >$ and a random nonce n_A. Upon seeing A’s request, node B with location $pos_B =< x_B, y_B >$ first needs to ascertain that the claimed location pos_A is in its transmission range by checking

$$(x_A - x_B)^2 + (y_A - y_B)^2 \leq R^2. \qquad (2)$$

This check is necessary because otherwise an adversary might send to B an authentication request including the location of one compromised node, say D, who is out of the transmission range of B, by boosting the transmission power. In this case, B might be tricked into belief that D is its authentic neighbor in that D has the correct LBK corresponding to the claimed location so that it can pass the following authentication process.

If the check fails, node B simply discards the request because node A is by no means a neighboring node. Otherwise, B returns a reply including its own location $pos_B =< x_B, y_B >$,

a random nonce n_B, and an authenticator V_B calculated as

$$V_B = H_2(\hat{e}(LK_B, H_1(pos_A)) \parallel n_A \parallel n_B \parallel 0). \quad (3)$$

Once receiving *B*'s reply, node *A* can determine whether *B* is in its transmission range by checking if Eq. (2) holds as well. If so, *A* proceeds to compute a verifier V'_B as

$$V'_B = H_2(\hat{e}(H_1(pos_B), LK_A) \parallel n_A \parallel n_B \parallel 0). \quad (4)$$

According to Eq. (1), if and only if both *A* and *B* have the authentic LBKs corresponding to their claimed locations, they can have

$$\begin{aligned}\hat{e}(LK_B, H_1(pos_A)) &= \hat{e}(H_1(pos_B), LK_A) \\ &= \hat{e}(H_1(pos_B), H_1(pos_A))^{\kappa} \in \mathbb{G}_2.\end{aligned} \quad (5)$$

After verifying the equality of V'_B and V_B, *A* can ascertain that *B* is an authentic neighbor with the claimed location pos_B. Node *A*, in return, should send to *B* its own authenticator V_A computed as

$$V_A = H_2(\hat{e}(H_1(pos_B), LK_A) \parallel n_A \parallel n_B \parallel 1). \quad (6)$$

By a simple calculation, node *B* can as well determine whether *A* is an authentic neighbor with the claimed location pos_A. Based on this three-way handshaking, nodes *A* and *B* can achieve mutual authentication and establish an authentic link between them as shown in Fig. 1. Following the same procedure, node *A* can achieve mutual authentication with all its neighboring nodes. Notice that if all the neighboring nodes simultaneously send replies to the same broadcast request from node *A*, a possible collision may occur. In this paper, we assume the reliable transmission of such authentication requests/replies. It can be achieved for instance through MAC-layer retransmission or by using a random jitter delay for which each node has to wait before responding to 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. In these cases, a new node is required to execute the authentication protocol once initialized properly as in Section III-C.

Discussion

The above location-based authentication scheme is secure against various malicious attacks. For example, in a *location forgery attack*, an adversary might send a forged location location within *A*'s transmission range as shown in Fig. 1, but would not have been in possession of the LBK corresponding to the forged location. Therefore, he/she cannot successfully finish the authentication procedure and thus cannot deceive *A* into belief that he/she is an authentic neighbor. Moreover, two powerful colluding adversaries might attack the conventional ID-based authentication schemes by tunnelling authentication messages received at one location of the network over an invisible, out-of-band, low-latency channel to another network location which is typically multi-hop away. By doing that, they attempt to make two nodes far away from each other believe that they are authentic neighbors. This tunnelling of authentication messages attack is infeasible either with our scheme because each legitimate node will deny authentication requests from nodes that are not physically within its transmission range. In addition, an adversary might put a replica of one compromised node at other locations into the vicinity of a legitimate node, say *C*. Most ID-based authentication schemes are vulnerable to this attack because one node like *C*, without dependence on any central authority, has great difficulty in differentiating between legitimate authentication requests and malicious ones from replicas of a compromised node. With our scheme in place, node *C* will simply discard the replica's authentication request because the replica should not appear in its transmission range.

It is worth pointing out that our scheme itself cannot prevent a compromised node or its replicas in its transmission range from achieving mutual authentication with its legitimate neighbors. But our scheme can guarantee that the compromised node receives nothing more than some random numbers and the public locations from the legitimate nodes. This ensures that the compromised node cannot impersonate its legitimate neighbors to other nodes. Therefore, our location-based authentication scheme can localize the impact of a compromised node to its vicinity, more specifically, within a small transmission range centered at its true location. In other words, once compromising a node, adversaries can no longer utilize the acquired knowledge to launch network-wide attacks as in a traditional ID-based authentication scheme. What they can only do is to misbehave around the current location of the compromised node. If so, they might run a high risk of being detected by legitimate nodes if effective misbehavior detection mechanisms are available.

We notice that an adversary may mount the denial-of-service attack on a target node by continuously sending authentication requests to the victim. To cope with this situation, if a legitimate node detects too many bogus authentication requests in a short time window, we assume that there are available efficient mechanisms for it to report such an abnormality to the sink. For lack of space, the further investigation on this issue is left to the extended version of this paper.

E. Pairwise Key Establishment

Most of previous proposals in sensor network security, such as [1]–[6], focus on the establishment of pairwise keys between neighboring nodes. Such pairwise keys are indispensable for guaranteeing link-layer security, i.e., authenticating, integrity-protecting, and encrypting messages exchanged between neighboring nodes.

Notice that after a successful three-way handshaking described in Section , two authenticated neighboring nodes, say *A* and *B*, have implicitly established a shared master secret as $PK_{AB} = \hat{e}(H_1(pos_B), H_1(pos_A))^{\kappa}$. An eavesdropper may overhear the authentication messages exchanged between *A* and *B*, but cannot calculate PK_{AB} for the lack of the LBKs of *A* and *B*. Therefore, PK_{AB} is only known to *A* and *B* and is referred to as the pairwise master secret between them hereafter. From PK_{AB}, *A* and *B* can derive various shared

keys for different security purposes by feeding PK_{AB} into the collision-resistant hash function H_2. For example, they can use $K_0 = H_2(PK_{AB} \parallel 0)$ for message encryption while $K_1 = H_2(PK_{AB} \parallel 1)$ for message authentication. In the similar way, each node can establish pairwise shared keys with all its authenticated neighbors after the neighbor discovery and authentication phase. Since pairwise keys are the by-products of the node-to-node mutual authentication process, there is no extra communication and computation overhead in establishing them.

In contrast to the probabilistic key pre-distribution schemes, our scheme has strong resistance to node compromise because pairwise keys are built upon the private LBKs of individual nodes. No matter how many nodes and their respective LBKs are compromised, the LBKs of non-compromised nodes always remain secure and so do the pairwise keys shared between them. In addition, our scheme only requires each node to memorize its own IBK and LBK, leading to favorable memory savings. Furthermore, our scheme puts no limitation on the network size and thus is highly scalable.

IV. Secure Sensor Network Routing Protocols

Karlof and Wagner [15] have identified a variety of attacks against existing sensor network routing protocols. In this section, we demonstrate how our location-based keys (LBKs) can act as efficient countermeasures against some most notorious attacks reported there.

A. The 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 identities. Karlof and Wagner [15] and Newsome *et al.* [16] pointed out that the Sybil attack is extremely detrimental to many important functions of the sensor networks, including routing, fair resource allocation, misbehavior detection, data aggregation, distributed storage, etc. As mentioned before, we require each node to perform location-based mutual authentication with its neighbors. When a malicious or compromised node intends to impersonate a legitimate node, he/she does not have the authentic LBK and thus cannot pass the screening process by other legitimate neighboring nodes. For the same reason, a malicious node cannot claim multiple locations as well. Therefore, the Sybil attack is efficiently defeated by our location-based key management and authentication schemes.

B. The identity replication attack

The *identity replication attack* [16] takes place when adversaries put multiple replicas of a compromised node in different geographic locations. It may cause the inconsistence of the routing information, as well as jeopardizing other sensor network functions. Conventional defenses often involve a central authority, e.g., the sink, that either keeps a record of each node's location [16] or centrally counts the number of connections a node has and revokes those with too many connections [2]. In these solutions, node-to-node authentication and/or pairwise key establishment have to be performed through the central authority to withstand the identity replication attack, which causes intensive communication overhead and the lack of scalability.

The identity replication attack is infeasible when location-based node-to-node mutual authentication 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.

C. Wormhole and sinkhole attacks

Wormhole [15], [17] and sinkhole [15] attacks are two notorious attacks against routing protocols that are difficult to defend against, especially when the two are used in combination.

In the wormhole attack, adversaries tunnel messages received at one location of the network over an invisible, out-of-band, low-latency channel to another network location which is typically multi-hop away. Hu *et al.* [17] presented a technique called *packet leashes* to withstand the wormhole attack, but it requires extremely tight time synchronization and is thus infeasible for most sensor networks [15]. In contrast, each node in our scheme only accepts messages from authenticated neighbors and will discard those messages tunnelled from multi-hop-away locations. Therefore, the wormhole attack is efficiently defeated.

In the sinkhole attack, a compromised node attempts to attract all the traffic from its surrounding nodes by announcing a high-quality route to the sink. Our scheme can withstand sinkhole attacks in minimum-hop routing protocols. For example, if a compromised node announces a rather small hop-count to the sink, its surrounding nodes can roughly verify the authenticity of its advertisement based on its location, transmission range, and the sink's location. In addition, geographic routing protocols such as [14] have been identified in [15] as promising solutions resistant to sinkhole and wormhole attacks because they construct the routing topology on demand using only localized interactions and geographical information. However, the location information advertised from neighboring nodes must be authenticated. Our scheme provides such a guarantee by binding nodes' locations to their respective private keys and requiring location-based node-to-node mutual authentication between neighbors. We notice that our scheme itself cannot prevent sinkhole attacks in routing protocols that use advertised information such as remaining energy or end-to-end reliability as routing metrics because this information is hard to verify. To the best of our knowledge, there is no workable solutions proposed in the literature, so it is an interesting topic worthy of further study.

V. More Discussion

Our location-based security schemes are built upon identity-based public-key cryptography (ID-PKC). First introduced by

Shamir in 1984 [18], ID-PKC allows public keys to be directly derived from publicly available information that uniquely and undeniably identifies entities, e.g., sensor locations in this paper. It thus eliminates the need for public-key certificates of conventional PKC such as RSA. This inbred feature makes ID-PKC more suitable for wireless sensor networks than RSA-type PKC, because sensor nodes no longer need to expend scarce resources in exchanging and verifying certificates.

It was a common belief that PKC is too complex, slow and power hungry, and thus ill-suited for use in resource-constrained environments like wireless sensor networks. For this reason, PKC has often been ruled out for establishing pairwise keys in sensor networks and most previous proposals [1]–[6] are purely based on secret-key cryptography. As we mentioned before, such proposals suffer from the lack of authentication, scalability, and resilience to node compromise due to the inherent limitations of secret-key cryptography. In addition, the probabilistic schemes often involve tens or even hundreds of secret-key encryption/decryption operations if the secure "puzzle-solving" method to discover a common key between two nodes is applied [1]–[3]. By contrast, pairwise keys in our scheme are the by-product of the node-to-node neighboring authentication process and each pair of neighbors only need to evaluate the pairing once during the network bootstrapping phase. In this sense, the actual computation or energy savings from the use of secret-key cryptography may be not that significant. Considering many other nice properties of our scheme, we argue that it is worthwhile introducing the use of pairing-based LBKs in securing sensor networks.

Most recently, many researchers [19]–[22] have challenged the traditional belief by showing that traditional PKC is, in fact, rather viable on low-power, low-cost sensor nodes such as the 8-bit, 7.3828-MHz UC Berkeley MICA2 mote. As an emerging technique, pairing-based ID-PKC is under rapid development. For example, according to the recent result reported in [23], the Tate pairing can be evaluated up to 10 times better than previously reported implementations. As far as we know, there is no published result on the implementation of pairing on low-end embedded devices such as sensor nodes and our exploration along this line is under way. We postulate that, just as traditional PKC, pairing-based ID-PKC will soon be proven with practical implementations to be tractable and beneficial in securing sensor networks.

VI. Concluding remarks

In this paper, we first proposed the notion of location-based keys for design compromise-tolerant security mechanisms for wireless sensor networks. We then presented a novel location-based node-to-node authentication scheme being able to localize the impact of compromised nodes within their vicinity. Another nice feature of our scheme is that, once finishing mutual authentication, two involved neighboring nodes have established a pairwise key indispensable for guaranteeing link-layer security. We also demonstrated the use of location-based keys in combating a few notorious attacks against sensor network routing protocols.

As the future research, we plan to evaluate the performance of our scheme in practical sensor networks. We also intend to further investigate the potential applications of location-based keys in securing sensor networks.

References

[1] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in *ACM CCS*, Washington, DC, Nov. 2002.

[2] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symposium on Security & Privacy*, Oakland, CA, May 2003.

[3] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *ACM CCS*, Washington, DC, Oct. 2003.

[4] W. Du, J. Deng, Y. Han, and P. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in *ACM CCS*, Washington, DC, Oct. 2003.

[5] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *ACM SASN*, Fairfax, VA, Oct. 2003.

[6] W. Du, J. Deng, Y. Han, S. Chen, and P.K.Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *IEEE INFOCOM*, HongKong, China, March 2004.

[7] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, June 2004.

[8] D. Boneh and M. Franklin, "Identify-based encryption from the weil pairing," *SIAM Journal of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[9] P. Barreto, H. Kim, B. Bynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Proc. CRYPTO'02*, ser. LNCS, vol. 2442. Springer-Verlag, 2002, pp. 354–368.

[10] NIST, "Digital hash standard," Federal Information Processing Standards PUBlication 180-1, April 1995.

[11] P. Corke, R. Peterson, and D. Rus, "Networked robots: Flying robot navigation using a sensor net," in *Proc. of 11th International Symp. of Robotics Research (ISRR)*, Oct. 2003.

[12] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[13] R. Rivest, "The rc5 encryption algorithm," in *Fast Software Encryption*, ser. LNCS, vol. 1008. Springer-Verlag, 1995, pp. 86–96.

[14] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *ACM MobiCom*, Rome,Italy, July 2001.

[15] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2, 2003.

[16] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis & defenses," in *Proc. of the Third International Symposium on Information Processing in Sensor Networks (IPSN 2004)*, Berkeley, CA, April 2004.

[17] Y. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hoc networks," in *IEEE INFOCOM*, San Francisco, CA, April 2003.

[18] A. Shamir, "Identity based cryptosystems and signature schemes," in *Proc. CRYPTO'84*, ser. LNCS, vol. 196. Springer-Verlag, 1984, pp. 47–53.

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

[20] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and rsa on 8-bit cpus," in *CHES'04*, Boston, MA, Aug. 2004.

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

[22] G. Gaubatz and B. S. J. Kaps, "Public keys cryptography in sensor networks – revisited," in *ESAS'04*, EURESCOM, Heidelberg, Germany, Aug. 2004.

[23] P. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups," in *Selected Areas in Cryptography – SAC'2003*, ser. LNCS, vol. 3006. Springer-Verlag, 2004, pp. 17–25.