9-7-2007

An Efficient Scheme for Removing Compromised Sensor Nodes from Wireless Sensor Networks

Yong Wang
University of Nebraska-Lincoln

Byrav Ramamurthy
University of Nebraska-Lincoln, bramamurthy2@unl.edu

Xukai Zou
Indiana University-Purdue University Indianapolis

Yuyan Xue
University of Nebraska-Lincoln

University of Nebraska–Lincoln, Computer Science and Engineering
Technical Report TR-UNL-CSE-2007-0017
Issued Sept. 7, 2007

An Efficient Scheme for Removing Compromised Sensor Nodes from Wireless Sensor Networks

Yong Wang and Byrav Ramamurthy
University of Nebraska-Lincoln
and
Xukai Zou
Indiana University-Purdue University Indianapolis
and
Yuyan Xue
University of Nebraska-Lincoln

Key management is a core mechanism to ensure the security of applications and network services in wireless sensor networks. It includes two aspects: key distribution and key revocation. Key distribution has been extensively studied in the context of sensor networks. However, key revocation has received relatively little attention. Existing key revocation schemes can be divided into two categories: centralized key revocation scheme and distributed key revocation scheme. In this paper, we first review and summarize the current key revocation schemes for sensor networks. Then, we present an efficient scheme of removing compromised sensor nodes from wireless sensor networks. Unlike most sensor node removal schemes focusing on removing the compromised keys, the proposed scheme, KeyRev, uses key update techniques to obsolesce the keys owned by the compromised sensor nodes and thus remove the nodes from the network. Our analyses show that the KeyRev scheme is secure inspite of not removing the pre-distributed key materials at compromised sensor nodes. Simulation results also indicate that the KeyRev scheme is scalable and performs very well compared with other key revocation schemes in wireless sensor networks.

Categories and Subject Descriptors: C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms: Security, Algorithms, Design, Performance

Additional Key Words and Phrases: Wireless sensor network, key management, centralized key revocation scheme, distributed key revocation scheme

This work is partially supported by NSF Grant No. CCR-0311577. A preliminary version of this article appears in the Proceedings of the IEEE International Conference on Communications (ICC), 2007.
Author's address: Department of Computer Science and Engineering, University of Nebraska-Lincoln,Lincoln, NE 68588 USA.

1. INTRODUCTION

Wireless sensor networks (WSNs) are promising solutions for many applications and security is an essential requirement of WSNs [Wang et al. 2006]. Among all security issues in WSNs, key management is a core mechanism to ensure the security of applications and network services in WSNs.

The goal of key management is to establish the required keys between sensor nodes which exchange data. A key management scheme includes two aspects: key distribution and key revocation. Key distribution refers to the task of distributing secret keys to sensor nodes to provide communication secrecy and authenticity. Key revocation refers to the task of securely removing keys which are known to be compromised. Key distribution has been exclusively studied under the constraints on computation and power consumption in sensor networks [Zhu et al. 2003; Eschenauer and Gligor 2002; Chan et al. 2003]. However, key revocation has received relatively little attention.

Because sensor nodes in WSNs may be deployed in hostile or insecure environments, the security of sensor nodes must be considered. In case a sensor node is captured or compromised, the sensor node must be removed securely from the network. The problem of sensor node removal is usually reduced to that of key revocation [Chan et al. 2003; Chan et al. 2005]. By revoking all of the keys belonging to a known compromised sensor node, the node can be removed from the network.

Most of the proposed key management schemes depend on some key materials being pre-distributed in the sensor nodes. These pre-distributed key materials might include an initial key shared by all sensor nodes [Zhu et al. 2003], a pairwise key shared between the base station and the sensor node [Eschenauer and Gligor 2002], or a key ring consisting of certain number of keys to be used in the future [Eschenauer and Gligor 2002; Chan et al. 2003]. The keys for secure communication, for example, *pairwise keys* [Eschenauer and Gligor 2002], *path keys* [Eschenauer and Gligor 2002], *cluster keys* [Zhu et al. 2003] used by sensor nodes are set up based on those pre-distributed materials in the bootstrap stage. When a sensor node is compromised, the keys set up on the fly and the pre-distributed materials must be revoked.

Revocation attacks must be considered in designing a revocation scheme. A revocation attack is a specific attack in which an adversary uses the node revocation protocol to selectively revoke uncompromised sensors from the network. Since compromised sensor nodes may act as an adversary's surrogates within a revocation protocol and subvert the execution of the revocation protocol [Chan et al. 2005], the resistance to compromised sensors must be evaluated in a revocation protocol. Further, after compromised sensors are removed from the network, new sensors might be re-deployed to replace those compromised sensors. The node addition problem must be considered. The node addition problem is usually reduced to the key distribution problem. In this paper, we focus on the key revocation issues.

A few schemes [Eschenauer and Gligor 2002; Chan et al. 2003; Chan et al. 2005] have been proposed to address the key revocation problem in WSNs. However, these schemes incur various difficulties when used in sensor networks. For example, the centralized key revocation scheme proposed in [Eschenauer and Gligor 2002] requires a signature key distributed in the non-revoked sensor nodes. However,

the signature key can only be distributed by unicasting which causes severe performance issues in large scale sensor networks. The distributed key revocation schemes proposed in [Chan et al. 2003; Chan et al. 2005] are based on some strong assumptions such as each node knowing its neighboring nodes before the sensor network is deployed. These assumptions are hard to satisfy. Thus, designing a new efficient scheme of removing compromised sensor nodes from wireless sensor networks is highly desirable.

In this paper, we present an efficient scheme, KeyRev, to remove compromised sensor nodes. The KeyRev scheme was first proposed in [Wang et al. 2007]. This paper is a significant improvement of the previous paper. First, we review and summarize the distributed key revocation scheme proposed in [Chan et al. 2003; Chan et al. 2005]. Second, we analyze and evaluate for the first time the distributed key revocation scheme in wireless sensor networks. Our analyses and simulation results reveal that the centralized key revocation scheme, which had been believed inefficient before, can also attain high efficiency in sensor networks.

Unlike most proposed key revocation schemes focusing on removing the compromised keys, we propose to use key update techniques to obsolesce the keys owned by the compromised sensor nodes and thus remove the nodes from the network. The proposed scheme depends on a unique key shared by all nodes in the network and the unique key is distributed to the network using an efficient group communication scheme [Liu et al. 2003]. In addition, our proposed scheme does not depend on any specific key distribution schemes and thus, the KeyRev scheme can be extended for implementation with other key distribution schemes, for example, the schemes proposed in [Chan et al. 2005; Cametepe and Yener 2004].

The KeyRev scheme is classified into the centralized key revocation schemes in this paper. However, unlike the key revocation schemes in [Eschenauer and Gligor 2002; Chan et al. 2003] which try to remove the keys shared with the compromised sensor nodes, there are really no keys to be removed from the sensor nodes in the KeyRev scheme. The sensor node removal problem is reduced to a key update problem in this paper. In the remainder of this paper, without specific explanation, the KeyRev scheme is also called a key revocation scheme.

Our contributions in this paper include the following:

(1) We present a novel scheme of removing compromised sensor nodes from wireless sensor networks utilizing key update techniques.
(2) We analyze and evaluate the performance of the distributed key revocation scheme in wireless sensor networks for the first time.
(3) Simulation results reveal that the centralized key revocation scheme can also attain high efficiency in wireless sensor networks.

Our analyses and simulation results show that the proposed scheme, KeyRev, is secure and efficient in computation, communication and storage usage. Simulation results also indicate that the KeyRev scheme is scalable and performs very well compared with other revocation schemes.

The remainder of this paper is organized as follows: Section 2 discusses the related work. Section 3 presents our proposed key revocation scheme. The security and performance analyses are presented in Section 4, and the simulation experiments and results in Section 5. Section 6 concludes the paper.

2. RELATED WORK

As discussed before, key management includes two aspects: key distribution and key revocation. Many key distribution schemes have been proposed in sensor networks. According to the network structure, the schemes can be divided into centralized key distribution schemes [Pietro et al. 2003] and distributed key distribution schemes [Eschenauer and Gligor 2002; Du et al. 2004]. According to the probability of key sharing between a pair of sensor nodes, the key distribution schemes can be classified into deterministic approaches [Zhu et al. 2003; Cametepe and Yener 2004] and probabilistic approaches [Eschenauer and Gligor 2002; Du et al. 2004]. An investigation of key distribution schemes for WSNs can be found in [Wang et al. 2006; Camtepe and Yener 2005]. In this paper, we focus on the key revocation problem.

Key revocation refers to the task of securely removing keys which are known to be compromised. To detect a compromised sensor, intrusion detection techniques are employed. Intrusion detection is out of the scope of this paper. We assume that there are some methods [Zhu et al. 2004; Ye et al. 2004; Wang et al. 2003] for a base station to detect a compromised sensor node. Another issue which must be considered is reconfiguration. The topology of the WSN needs to be rebuilt after the compromised sensors are removed. Sensors might be re-deployed to replace those compromised sensors. The rest of the section reviews several known key revocation schemes in WSNs.

Recent work conducted on key revocation for WSNs include [Eschenauer and Gligor 2002; Chan et al. 2003; Chan et al. 2005; Wang et al. 2007; Zhang et al. 2005] and no other schemes have been reported to date. These key revocation schemes can be divided into two categories: the centralized key revocation schemes [Eschenauer and Gligor 2002; Wang et al. 2007; Zhang et al. 2005] and the distributed key revocation schemes [Chan et al. 2003; Chan et al. 2005]. We discuss these in turn below.

2.1 Centralized key revocation scheme

In centralized key revocation scheme, a centralized authority (base station) is used to revoke compromised sensors [Eschenauer and Gligor 2002]. Eschenauer and Gligor presented a key management scheme for WSNs in [Eschenauer and Gligor 2002]. This scheme, which is called the basic random key scheme in this paper, is a centralized key revocation scheme. Before describing the key revocation scheme, we first introduce the key distribution scheme which will be used later to demonstrate how to revoke the compromised key materials in our scheme.

The key distribution scheme consists of three phases: key pre-distribution, shared-key discovery, and path key establishment.

In the key pre-distribution phase, each sensor is equipped with a *key ring* held in the memory. The key ring consists of k keys which are randomly drawn from a large pool of P keys. The association information of the key identifiers in the key ring and sensor identifier are also stored at the base station. Further, the authors assumed that each sensor i shares a pairwise key K^{ci} with the base station.

In the shared key discovery phase, each sensor discovers its neighbors within wireless communication range with which it shares keys. Two methods to accom-

plish this are suggested in [Eschenauer and Gligor 2002]. The simplest method is for each node to broadcast a list of identifiers of the keys in its key ring in plain text allowing neighboring nodes to check whether they share a key. However, an adversary may observe the key-sharing patterns among sensors in this way. The second method uses the challenge-response technique to hide key-sharing patterns among nodes from an adversary. For every key K_i on a key ring, each node could broadcast a list $\langle \alpha, E_{K_i}(\alpha) \rangle, i = 1, ..., k$ where α is a challenge. The decryption of $E_{K_i}(\alpha)$ with the proper key by a recipient would reveal the challenge and establish a shared key with the broadcasting node.

Finally, in the path-key establishment phase, a path-key is assigned between sensor nodes which are within wireless communication range but do not share a key at the end of the second phase.

If a node is compromised, the base station can send a message to all other sensors to revoke the compromised node's key ring. The revocation scheme in [Eschenauer and Gligor 2002] can be divided into three phases: signature key distribution, key revocation and link reconfiguration.

In the signature key distribution phase, the base station generates a signature key K_e and unicasts it to each node by encrypting it with a pairwise key K^{ci} shared by the base station with the i-th sensor node.

In the key revocation phase, the base station broadcasts a single message containing a list of key identifiers for the key ring to be revoked signed by the signature key. Each sensor verifiers the signature of the key revocation message, locates those identifiers in its key ring, and removes the corresponding keys.

Once the keys are removed from the key rings, some links may disappear, and the affected nodes need to reconfigure those links by restarting the shared-key discovery, and possibly the path-key establishment, phases.

The key revocation scheme, referred to as EsRev scheme, requires n unicast messages and one broadcast message. In a large scale sensor network, distributing the signature key might be a problem. Pre-distributing the signature key might be possible; however, once the signature key is compromised, the adversary could use the signature key to duplicate the revocation messages from the base station.

Zhang *et al.* proposed a location-based revocation scheme utilizing multiple revocation messages in [Zhang et al. 2005]. When the revocation area is large or complicated, the revocation area can be divided into sub-areas. For each sub-area, a revocation message is sent to a certain node within that area using GPSR protocol [Karp and Kung 2000], and then the revocation message is multicasted to the remaining sub-area. The revocation message includes the identifier of the sensor nodes to be revoked and the scope of the revocation area. On receiving the message, for each node, if it has received the message before or is outside of the revocation area, the message is dropped. If the sensor node is within the revocation area indicated by the revocation message, the sensor node records the identifier of the revoked sensor node, and rebroadcast the message to its neighboring nodes. The revocation scheme, referred to as the GPSRRev scheme, is also a centralized key revocation scheme.

2.2 Distributed key revocation scheme

In a distributed key revocation scheme, no centralized authority is used. Chan *et al.* proposed a distributed key revocation scheme for sensor networks in [Chan et al. 2003] and further investigated this scheme in [Chan et al. 2005]. In this distributed key revocation scheme, a vote is cast and collected among sensor nodes. If the vote tally against a sensor node exceeds a specified threshold, the sensor node will be revoked. Chan's scheme depends on the secret sharing scheme proposed in [Shamir 1979]. The distributed key revocation scheme is described below.

The revocation timeline is divided into sessions. Each sensor has at most s_{total} revocation sessions against any target nodes (compromised nodes). Before the sensor network is deployed, the setup server generates a t degree random polynomial $q_{A_s}(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}$ for each session s on sensor node A. For each node B of A's participants (a participant of A is a sensor which shares a key with node A), the setup server loads the revocation vote $(q_{B_s}(x_{AB_s}), x_{AB_s})$ from A against B on node A. This revocation vote is encrypted by a mask $Mask_{AB_s}$ that B gives to A. That is, the preloaded data on A against B is $E_{Mask_{AB_s}}(q_{B_s}(x_{AB_s}), x_{AB_s})$. For each vote, the setup server also loads the $\log m$ authentication hash values for the Merkle tree with leaves $(q_{B_s}(x_{iB_s}), x_{iB_s})$ for each node i in B's participants (a total of m leaves). The root R_B of the Merkle tree is also loaded on A. Finally, the setup server loads $H^2 q_{B_s}$, which is the hash of the hash of the revocation polynomial of B on A. This will allow nonlocal participants (a nonlocal participant of B is a sensor node which shares a key with B but multi-hops (> 1) away from B) to verify the authenticity of a revocation decision against B.

In the beginning of each session, the masks are exchanged among neighboring nodes. The purpose of the mask key is to ensure that each node is only able to revoke sensors within its immediate neighborhood.

Each revocation session is divided into three states: pending, active and completed. The pending state indicates that there is no voting occurs in the current session. When the first vote of the session is cast and received by A, A changes its state to active. The active state lasts for exactly Δ_s time for each node, after which the node transitions to the completed state for this session, and starts the pending state for the next session.

When A votes against B during the revocation session s, it decrypts the vote using B's mask key $Mask_{AB_s}$ and broadcasts $(q_{B_s}(x_{AB_s}), x_{AB_s})$ along with the $\log m$ Merkle authentication values. This is a local neighborhood broadcast. The broadcast only needs to go far enough to ensure complete dissemination in the neighborhood of B. The node receiving the vote verifies the authenticity of the vote using the Merkle authentication values. The node will disseminate the broadcast if the verification is successful. When A casts a vote on B, it will vote both in the current session and on the next session. Voting on the next session occurs immediately upon completion of the current session.

When A's state for the session has transited to completed state, it counts the number of votes it has received when it was in the active state. If A has at least t revocation votes, then A can use these t points to compute the random t-degree polynomial q_{B_s} using the secret sharing scheme in [Shamir 1979]. From this, A computes the hash of the polynomial, Hq_{B_s}. This value is then broadcasted through

the entire network. All participants of B receiving this value can verify it by computing the hash of the received value. If the verification is successful, the keys shared with B will be revoked.

Note that Chan's scheme, referred to as DistRev scheme, is built on some simplifying assumptions; for example, each node knows its neighboring nodes before deployment. It is hard to satisfy these requirements.

Compared with the centralized key revocation schemes, the distributed key revocation schemes are faster because they require local broadcast and avoid a single point of failure. However, the distributed key revocation schemes are also more complex than the centralized key revocation schemes and, hence, more prone to design error since compromised sensor nodes can participate in the revocation protocol and attempt to block or circumvent it. In addition, it is also possible to compromise enough nodes to sabotage the distributed key revocation scheme. For more detailed information about the distributed key revocation scheme, please refer to [Chan et al. 2003; Chan et al. 2005].

In the remainder of this paper, we present an efficient scheme, KeyRev, to remove compromised sensor nodes from wireless sensor networks. We use the following notation in the remainder of this paper:

—A, B are principals such as communicating nodes.
—$K_{A,B}$ denotes the secret pairwise key shared between A and B.
—M_K is the encryption of message M with key K.
—$MAC(K, M)$ denotes the computation of the message authentication code of message M with key K.
—$M_1|M_2$ denotes the concatenation of messages M_1 and M_2.
—$A \longrightarrow B$ denotes that A unicasts a message to B.

3. KEYREV: AN EFFICIENT SCHEME OF REMOVING COMPROMISED SENSORS FROM WSNS

Unlike most of the proposed key revocation schemes focusing on removing the compromised keys, our scheme, KeyRev, uses key update techniques to obsolesce the keys owned by the compromised sensor nodes and thus remove the nodes from the network. The KeyRev scheme does not depend on a specified key distribution scheme. Without loss of generality, we assume that the basic random key distribution scheme [Eschenauer and Gligor 2002] is used.

3.1 Assumptions of our protocol

We assume that the base station is secure and well protected. The sensor nodes are not tamper-resistant and thus can be compromised. If a sensor node is compromised, the attacker is capable of stealing all the key materials contained within that node. We also assume that all the sensor nodes are within reach of the base station. Next, we provide an overview of our scheme.

3.2 Overview

The basic random key distribution scheme establishes two kinds of keys among sensor nodes: the *pairwise keys* and the *path keys*. When a sensor node is compromised, the compromised keys must be revoked so that the compromised keys will

not be chosen again as the new secret keys. Instead of using the pairwise keys and the path keys directly for the communication secrecy and authenticity, we propose two kinds of keys for secure communication in the sensor network: the *encryption key* and the *message authentication code (MAC) key*. The encryption key and the MAC key are generated by a pseudo-random function which is bound to the pairwise key or the path key, and a *session key* distributed regularly by the base station to all the sensor nodes in the network. When the session key is updated, the encryption key and the MAC key are also changed. A sensor node always uses the latest encryption key and MAC key to encrypt and sign the outgoing messages or decrypt and verify the incoming messages. If there is a session key distribution scheme in which the compromised sensors cannot recover the new session key when they are revoked, these revoked sensors will be removed from the network because they cannot derive the new encryption keys and the MAC keys on the next session. Although an adversary may retain the pairwise keys and the path keys, the adversary cannot figure out the encryption keys and the MAC keys because of the pseudo-random function used. Thus, the key revocation problem is reduced to the session key update problem.

In the remainder of this section, we first present the KeyRev scheme assuming an effective session key distribution scheme is used, and then we introduce the session key distribution scheme.

3.3 KeyRev scheme

The lifetime of a WSN is partitioned into time intervals called *sessions*. The duration of sessions can be fixed or dynamic depending on the applications. The base station is responsible for distributing *session keys* to the sensor nodes. We use K_j to denote the j-th session key where $j \in \{1, 2, \cdots, m\}$ and m is the number of sessions.

We assume that each sensor is uniquely identified by an ID number i, where $i \in \{1, \ldots, n\}$ and n is the largest ID number. Each sensor maintains a list: *node revocation list (NRL)*. A *node revocation list* includes all the sensor identifiers which have been revoked in the network. The revocation list is empty initially and will be populated as the time goes by. The revocation list is checked for any incoming and outgoing messages to ensure that only valid sensors are members of the network. We also assume that the pairwise keys and the path keys have been set up by the basic random key distribution scheme.

We propose two kinds of keys for secure communication in the sensor network: the *encryption key* K_{encr} and the *message authentication code (MAC) key* K_{mac}. For any message transmitted in the network, authentication, confidentiality, and integration are required. Let A and B be two entities in a WSN, the complete message A sends to B is:

$$A \longrightarrow B : \{M|T_s\}_{K_{encr}}, MAC(K_{mac}, \{M|T_s\}_{K_{encr}})$$

where M is the message, T_s is the timestamp when sending the message, and $MAC(K, R)$ denotes the computation of the message authentication code of message R with key K.

Let K_j be the current session key and $K_{A,B}$ represent the pairwise key or path key shared between the sensor nodes A and B. The encryption key and the MAC

key used in session j can be generated as follows:

$$K_{encr} = F(MAC(K_{A,B}, K_j), 1) \quad (1)$$

$$K_{mac} = F(MAC(K_{A,B}, K_j), 2) \quad (2)$$

where $F(K, x)$ is a pseudo-random function and x is an integer 1 or 2 for generating K_{encr} or K_{mac} respectively.

The security of the communication between A and B is ensured by the encryption key K_{encr} and the MAC key K_{mac}. Both of them are bound to the session key and will be updated when the session key is updated. Any message that A sends to B is encrypted by the encryption key K_{encr} and signed by the MAC key K_{mac}. For any message that B receives from A, B always verifies the message first and then decrypts it. Further, a sensor node always uses the encryption key and the MAC key corresponding to the current session key to encrypt and sign the outgoing messages or decrypt and verify the incoming messages.

If there is a method to stop the compromised sensors from obtaining the new session keys and thus stop them from deriving K_{encr} and K_{mac}, then the compromised sensors can no longer decrypt new messages and authenticate themselves. For example, if A is compromised and A cannot recover the new session key, then A cannot derive the new encryption key and the MAC key while B can. Due to the lack of the proper keys to encrypt and sign the messages, A cannot send any valid messages to B from that time. Therefore, the sensor node A is removed from the network.

Next, we introduce the session key distribution scheme used in the KeyRev scheme.

3.4 Session key distribution scheme

To make the KeyRev scheme work, the session key distribution scheme must satisfy the following criteria:

(1) The compromised sensors should not be able to obtain the new session keys.

(2) The sensor network is time synchronized so that the current keys can be identified.

Criterion 2 is easily satisfied. For criterion 1, we derive a simple session key distribution scheme based on the personal key share distribution scheme in [Liu et al. 2003]. The session key distribution scheme can be divided into three phases, viz., setup, broadcast, and session key recovery.

(1) Setup: The setup server randomly picks m $2t$-degree masking polynomial, $h_j(x) = h_{j,0} + h_{j,1}x + \cdots + h_{j,2t}x^{2t}, j \in \{1, 2, \cdots, m\}$, over a finite field F_q where q is a sufficiently large prime number. For each sensor node A_i, the setup server loads the personal secrets, $\{h_1(i), h_2(i), \cdots, h_m(i)\}$, to the node A. The setup server also loads the polynomial, $h_j(x)$, to the base station. For each session key K_j, the setup server randomly picks a t-degree polynomial $p_j(x)$ and constructs $q_j(x) = K_j - p_j(x)$.

(2) Broadcast: Given a set of revoked group members, $R = \{r_1, r_2, \cdots, r_w\}$, $w \leq t$ in session j, the base station distributes the shares of t-degree polynomial $p_j(x)$

and $q_j(x)$ to non-revoked sensors via the following broadcast message:

$$\begin{aligned} B &= \{R\} \\ &\cup \{P_j(x) = g_j(x)p_j(x) + h_j(x)\} \\ &\cup \{Q_j(x) = g_j(x)q_j(x) + h_j(x)\} \end{aligned}$$

where the revocation polynomial $g_j(x)$ is constructed as $g_j(x) = (x - r_1)(x - r_2)\cdots(x - r_w)$.

(3) Session key recovery: If any non-revoked sensor node A_i receives such a broadcast message, it evaluates the polynomial $P_j(x)$ and $Q_j(x)$ at point i and gets $P_j(i) = g_j(i)p_j(i) + h_j(i)$ and $Q_j(i) = g_j(i)q_j(i) + h_j(i)$. Because A_i knows $h_j(i)$ and $g_j(i) \neq 0$, it can compute $p_j(i) = \frac{P_j(i)-h_j(i)}{g_j(i)}$ and $q_j(i) = \frac{Q_j(i)-h_j(i)}{g_j(i)}$. A_i finally can compute the new session key $K_j = p_j(i) + q_j(i)$.

The revoked sensors cannot recover $p_j(i)$ and $q_j(i)$ because $g_j(i) = 0$ and thus cannot recover the new session key. Without obtaining the new session key, the revoked sensors cannot derive the encryption key K_{encr} and the MAC key K_{mac} and thus cannot decrypt new messages and authenticate themselves to other sensor nodes in the network. The compromised sensor nodes can thus be removed.

To demonstrate the session key distribution process, an example is given below. We consider three sensors with ID numbers 1, 2, and 3 respectively. We assume sensor 2 is compromised in session 5 and will be revoked in session 6. In the setup phase, the setup server picks the masking polynomial $h_6(x) = 1 + x^8$ for session 6 and each sensor receives a secret $h_6(1) = 2$, $h_6(2) = 257$, and $h_6(3) = 6562$ respectively. Let $K_6 = 101$, $p_6(x) = 1 + x^4$ and thus we have $q_6(x) = 100 - x^4$ and $g_6(x) = x - 2$. In session 6, the base station broadcasts a message:

$$\begin{aligned} B &= \{2\} \\ &\cup \{P_6(x) = (x - 2)(1 + x^4) + 1 + x^8\} \\ &\cup \{Q_6(x) = (x - 2)(100 - x^4) + 1 + x^8\} \end{aligned}$$

When sensor 1 receives the message, sensor 1 calculates: $P_6(1) = 0$, $Q_6(1) = -97$ and thus $p_6(1) = 2$ and $q_6(1) = 99$. Sensor 1 computes the session key $K_6 = p_6(1) + q_6(1) = 101$; Similarly, sensor 3 calculates: $P_6(3) = 6644$, $Q_6(3) = 6581$ and thus $p_6(3) = 82$ and $q_6(3) = 19$. Sensor 3 can also compute the session key $K_6 = p_6(3) + q_6(3) = 101$. However, sensor 2 cannot calculate $p_6(2)$ and $q_6(2)$ because $g_6(2) = 0$ and thus sensor 2 cannot derive the new session key.

A missing link in the above scheme is how a base station broadcasts authenticated messages. In the absence of authentication of broadcast messages, an adversary can impersonate a base station and start a revocation attack. $\mu TESLA$ [Przydatek et al. 2003] and its extensions [Liu and Ning 2003; 2004] have been proposed to provide such services for sensor networks. We assume that a proper broadcast authentication scheme such as $\mu TESLA$ is used with the KeyRev scheme. Note that to use $\mu TESLA$ protocol, the sensor network should be loosely time synchronized to meet the requirements [Liu et al. 2005].

To add new nodes to the sensor network, pre-distributed key materials required by the basic random key distribution scheme and the broadcast authentication scheme must be loaded on the sensor nodes. In addition, the setup server must also

load the personal secrets, $\{h_j(i)\}_{j=1..m}$, required by the session key distribution scheme, to each added sensor node.

4. SECURITY AND PERFORMANCE ANALYSIS

In this section we first discuss the security of the protocol. Then, we analyze the computation, the communication costs, and the storage requirements of the KeyRev protocol.

4.1 Security analysis

Our proposed scheme, KeyRev, satisfies the following properties:

Property 1 The session key distribution process is secure.

The session key is distributed using the personal key distribution scheme [Liu et al. 2003]. To restore the session key, it requires some personal secret to be pre-distributed among the sensor nodes. Outsiders cannot recover the session key without the pre-distributed secret. Further, as we show in Section 3.4, the revoked sensors cannot recover the new session keys either. Thus, the session key distribution process is secure.

Property 2 The KeyRev scheme is secure inspite of the non-removal of the pre-distributed key materials at a compromised sensor node.

Although, due to the non-removal of the pre-distributed key materials, the compromised sensor may retain the pairwise keys, the adversaries cannot figure out the encryption key K_{encr} and the MAC key K_{mac} if the session key is updated. In the worst case, an adversary might use a chosen plaintext attack to crack the session key; however, the attack itself is also time consuming. As long as the duration of sessions is less than the session key cracking time, the proposed key revocation scheme is secure.

Property 3 The KeyRev scheme is immune to revocation attack assuming the base station is secure.

The KeyRev scheme depends on the base station to distribute and update the session key. Broadcast authentication schemes such as $\mu TESLA$ [Perrig et al. 2002] can be used to protect the authenticity of the broadcast messages. To start the revocation attack, an adversary must impersonate the base station. However, since the base station is the only one which can broadcast authenticated messages using $\mu TESLA$ protocol, the compromised sensor nodes cannot be used to start the revocation attack. Thus, the KeyRev scheme is immune to revocation attack if the base station is secure.

4.2 Performance analysis

4.2.1 *Computation cost.* To restore the session key, each sensor node must evaluate the polynomial $P_j(x)$ and $Q_j(x)$ at point i. The polynomial evaluation is fast and thus the session key recovery is efficient in computation.

4.2.2 *Communication cost.* The performance of the KeyRev scheme depends mainly on the session key updating process. The session key can be updated in one round using broadcasting. The maximum size of the broadcast message in bits is decided by S:

$$S = (5t + 2) \log q$$

Let B indicate the transmission rate of the base station, L be the maximum range between the base station and the sensor nodes. The session key distribution time can be calculated as:

$$t_s = \frac{S}{B} + \frac{L}{3 * 10^8}$$

Compared with the transmission time, the propagation delay is very small. Thus, we can approximately estimate the session key distribution time as:

$$t_s \approx \frac{(5t + 2) \log q}{B}$$

4.2.3 *Storage requirement.* To restore the session key, each sensor node needs to be loaded with m personal secrets. Since the encryption key and the message authentication code key can be set up on the fly, the extra storage units to implement the KeyRev scheme is $m \log q$.

Overall, the KeyRev scheme is efficient in consideration of the computation load, the communication cost, and the storage space.

4.3 Comparison

The KeyRev scheme is a centralized key revocation scheme. It depends on an efficient session key distribution scheme which can be done in one round using a broadcast message (Section 3.4). Compared with the EsRev scheme, in case a sensor node is compromised, the EsRev scheme requires two rounds of communications: distributing a signature key to the no-revoked sensors, followed by broadcasting a message containing a list of revoked key identifiers. Since the signature key is distributed to the network using unicasting, the EsRev scheme may cause heavy traffic in large scale sensor networks. Note that there is no need of the unicasting and the session key can be updated in one round using broadcasting, the KeyRev scheme is much better than the EsRev scheme.

By dividing the revocation field into sub-areas and using multiple revocation messages, the GPSRRev scheme performs better than the EsRev scheme. However, addition information, such as location of the sensor nodes, must be used. Further, the multicast of the revocation message in the sub-area is implemented using message flooding and it is still time and energy consuming. The KeyRev scheme is more efficient than the GPSRRev scheme since it uses broadcast instead of multicast.

The distributed key revocation scheme, DistRev, has been regarded to be faster than the centralized key revocation schemes due to the fact that it requires only broadcast messages of a few hops that reach the local destinations [Chan et al. 2005]. However, it is not true for the KeyRev scheme. In case a sensor node is compromised and revoked successfully from the network, the DistRev scheme requires four rounds of communications:

(1) Neighboring nodes exchange the masks to decrypt the votes for the current revocation sessions at the connection time.

(2) At least t sensor nodes cast their votes against the target node (compromised node) in the current session.

(3) The voting nodes also cast their votes against the target node on the next session.

(4) If a sensor node receives at least t revocation votes, a hash value containing the compromised sensor node information needs to be broadcasted through the entire network.

Although the first three rounds of the communications are local broadcast, the last one involves a broadcast through the entire network. The broadcast message can either be flooded from the sensor node which receives t revocation votes or be forwarded to the base station and broadcasted to the network by the base station. Either way, the KeyRev scheme is much better than the DistRev scheme since it requires only one broadcast and no local communication is required. Further, the DistRev scheme is also built on some simplifying assumptions, for example, each node knows its neighboring nodes before deployment, which are hard to satisfy in many sensor network applications.

Table I compares the four revocation schemes discussed in the paper, where n is the number of sensor nodes in the network, d is the number of sub-areas in the GPSRRev scheme, and t is the number of votes which a sensor node has to collect to revoke a compromised node in the DistRev scheme. We consider the situation when a single node is compromised and revoked successfully from the network.

Table I. Comparison of the key revocation schemes in wireless sensor networks.

	Scheme	Rnds	Unicast	Broadcast	Local Broadcast	Scalability
I	EsRev	2	n	1	0	Low
	GPSRRev	1	d	0	d	Medium
	KeyRev	1	0	1	0	Good
II	DistRev	4	0	1	$2 * t$	Good

Category I denotes centralized key revocation schemes and category II denotes distributed key revocation schemes.
Note that the GPSRRev scheme requires the location information of the compromised sensor nodes.

The comparison in Table I shows that the KeyRev scheme is better than other schemes in reducing the communication overhead caused by the revocation protocol. Notice that the KeyRev scheme requires a session key to be distributed to the network during each session. The duration of the session time could be set and adjusted dynamically according to the application to reduce the background traffic in the sensor network.

5. SIMULATION AND RESULTS

5.1 Experimental setting

The performance of the KeyRev scheme was evaluated in SENSIM [Wang and Ramamurthy 2006], a component-based discrete-event simulator for sensor networks. Each sensor node in SENSIM consists of six components, i.e., app, net, mac, phy, event generator, and battery. In the physical component, the free space propagation model is used. In the mac component, all the packets sent to MAC layer are

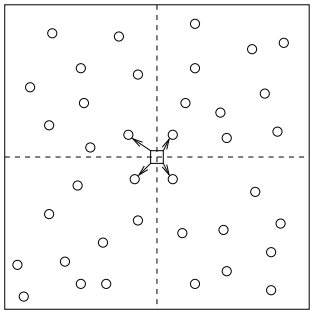

Fig. 1. Illustration of the GPSR-based revocation scheme. The revocation message is sent to a sensor node in each sub-area. Then, the revocation message is multicasted to the remaining sub-area.

guaranteed to be received at the receivers. Thus, no packet collisions are considered and the performance evaluated in the simulation are under ideal conditions.

We consider two sensor network experimental settings: a small-scale sensor network with 100 nodes uniformly dispersed in a field with dimension $100m \times 100m$ and a large-scale sensor network with 1000 nodes uniformly dispersed in a field with dimension $2000m \times 2000m$. In both the networks, we set the base station at the center of the field and we assume that all the sensor nodes are within reach of the base station.

We compare the KeyRev scheme with the centralized key revocation schemes, the EsRev scheme and the GPSRRev scheme. The sensor field in the GPSRRev scheme is divided into four areas as shown in Figure 1. The revocation message is sent to a sensor node in each sub-area. Then, the revocation message is multicasted to the remaining sub-area.

The evaluation metrics include the key revocation time t_v and the average energy consumption e_v per node to revoke a compromised sensor in the network. The key revocation time is the time duration from when the key revocation protocol starts until all the uncompromised sensor nodes receive the key revocation message.

We consider the KeyRev scheme operating on a finite field F_q, where q is a 56-bit integer. The polynomial degree t in the KeyRev scheme is set to $t = 4$. We use the simulator parameters that represent the Mica2 Mote radio characteristics. These parameters are shown in Table II. For each experimental sensor network, we randomly select one sensor to be revoked and run the simulation ten times. The average value is measured.

Table II. Characteristic data for the Mica2 sensor platform.

Field	Value
Effective data rate	19.2kbps
Transmit power	36mW
Receive power	14.4mW
Idle power	14.4mW
Sleep	0.015mW
Transition power	28.8mW
Transition time	800μs

5.2 KeyRev vs. EsRev vs. GPSRRev

Table III shows the key revocation time to revoke a compromised sensor node in the two networks. As the table shows, in the 100-node sensor network, the key revocation time by using the EsRev scheme and the GPSRRev scheme is about 83 times and 1.6 times that of the KeyRev scheme. In the 1000-node sensor network, the key revocation time by using the EsRev scheme and the GPSRRev scheme is 800 times and 6.5 times that of the KeyRev scheme. The KeyRev scheme is much better than the EsRev scheme and the GPSRRev scheme in the key revocation time.

Table IV shows the average energy consumption to revoke a compromised sensor in the 100-node and 1000-node sensor networks. As the table shows, in the 100-node sensor network, the average energy consumption to revoke a single node by using the EsRev scheme and the GPSRRev is about 71 times and 19 times that of the KeyRev scheme. In the 1000-node sensor network, the average energy consumption to revoke a single sensor by using the EsRev scheme and the GPSRRev is about 714 times and 29 times that of the KeyRev scheme. The KeyRev scheme is much better than the EsRev scheme and the GPSRRev scheme in the average energy consumption.

In both the experimental settings, the KeyRev scheme performs very well compared with the EsRev scheme and the GPSRRev scheme. Further, Tables III and IV also show that the key revocation time and the average energy consumption to revoke a single sensor node by using KeyRev scheme have only a slight difference between the 100-node sensor network and the 1000-node sensor network, which indicates that the KeyRev scheme is scalable to large-scale sensor networks. However, due to the long key revocation delay caused by the EsRev scheme, the EsRev scheme is not scalable to large-scale sensor networks. The performance of the GPSRRev scheme is better than the EsRev scheme but not as good as the KeyRev scheme.

Table III. Key revocation time.

Scheme	100-node WSN Time (seconds)	1000-node WSN Time (seconds)
EsRev	49.63	496.06
GPSRRev	1.02	4.04
KeyRev	0.59	0.62

Table IV. Average energy consumption per node to revoke a compromised sensor.

Scheme	100-node WSN Energy (joules)	1000-node WSN Energy (joules)
EsRev	0.71	7.14
GPSRRev	0.19	0.29
KeyRev	0.01	0.01

5.3 KeyRev vs. DistRev

To evaluate the performance of the KeyRev scheme, we also compare the KeyRev scheme with the DistRev scheme. The metrics we evaluate include the key revocation time and the average energy consumption. As we discussed in Section 2.2, each revocation session in the DistRev scheme consists of three states: pending, active, and completed. The critical part of the three states which decides the key revocation time is the active state. In the active state, a sensor node casts a vote and the vote is broadcasted locally among the neighboring nodes. Assume that the active state lasts for Δ_s time for each node and Δ_c is the maximum time that a message needs to completely propagate in a local neighborhood broadcast. We have $t_v > \Delta_s$ and $\Delta_s > 2\Delta_c$ since each sensor has to vote both in the current session and on the next session. Therefore, the key revocation time t_v of the DistRev scheme is at least twice that of Δ_c, $t_v > 2\Delta_c$. Similarly, let e_{Δ_s} be the energy consumption during the active state and e_{Δ_c} be the energy consumption consumed during the Δ_c period of time, We have $e_v > e_{\Delta_s}$, $e_{\Delta_s} > te_{\Delta_c}$ (to revoke a compromised sensor node, the sensor node must receive at least t revocation votes) and thus, $e_v > te_{\Delta_c}$.

The duration of Δ_c is decided by a maximum count L (max-hops) which the vote can be broadcasted to ensure complete dissemination in the neighborhood of a compromised sensor node (four-six hops can cover this area with high probability [Eschenauer and Gligor 2002]). We test the Δ_c in the 100-node and the 1000-node sensor networks. The sensor node casting the vote is set to the center of each testbed. Table V shows the number of sensor nodes in the coverage area when the max-hops changes.

Table V. The number of nodes in the covered area.

L (max-hops)	1	2	3	4	5	6
100-node WSN	100	n/a	n/a	n/a	n/a	n/a
1000-node WSN	15	44	85	142	219	299

Note: All the sensor nodes in the 100-node sensor network are in the cover area when the max-hops is set to 1.

In the 100-node sensor network, the simulation results show that $\Delta_c = 0.035$ seconds and $e_{\Delta_c} = 995$ nano-joules. Thus, we have $t_v > 0.070$ and $e_v > 995t$ nano-joules. Compared with the KeyRev scheme in the 100-node sensor network as shown in Tables III and IV, the DistRev scheme might be better than the KeyRev scheme but the performance of the KeyRev scheme is also very good in the 100-node sensor network.

Figure 2 shows the key revocation time of the DistRev scheme in the 1000-node sensor network when the max-hops changes. Note that the column value is not the real key revocation time t_v of the DistRev scheme but the value of the $2\Delta_c$. The actual key revocation time is $t_v > 2\Delta_c$. The dotted horizontal line shows the key revocation time of the KeyRev scheme in the 1000-node sensor network. From the figure, we can draw the conclusion that the KeyRev scheme is better than the DistRev scheme in terms of the key revocation time since the max-hops is definitely greater than one in the DistRev scheme to ensure full coverage of the neighboring nodes of the target node (compromised node).

Figure 3 shows the average energy consumption per node in the DistRev scheme in the 1000-node sensor network when the max-hops changes. The column value is also not the real average energy consumption e_v of the DistRev scheme but the value of $2e_{\Delta_c}$ (we set t to the minimum value 2, $t = 2$). The actual average energy consumption is $e_v > te_{\Delta_c}$. The dotted horizontal line shows the average energy consumption of the KeyRev scheme in the 1000-node sensor network. The figure indicates that the KeyRev scheme is better than the DistRev scheme even if we set the number of votes to revoke a sensor node to the minimum value of two.

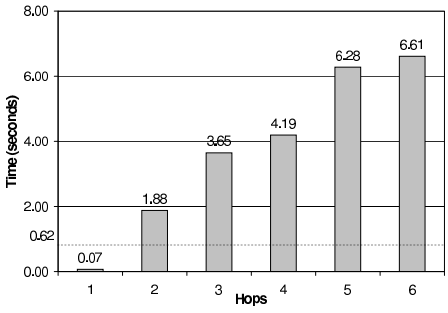

Fig. 2. Key revocation time in the 1000-node sensor network. The column value is not the real key revocation time t_v of the DistRev scheme but the value of the $2\Delta_c$.

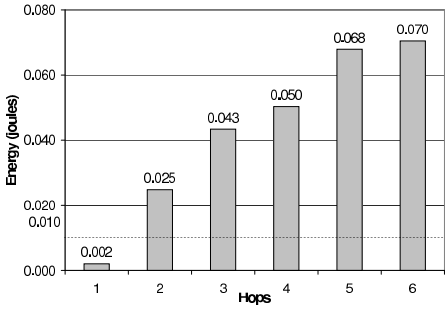

Fig. 3. Average energy consumption per node to revoke a compromised sensor in the 1000-node sensor network. The column value is also not the real average energy consumption e_v of the DistRev scheme but the value of $2e_{\Delta_c}$.

To ensure the neighborhood of the target node (compromised node) is fully covered, the max-hops cannot be set too small. Thus, our proposed scheme, KeyRev,

is better than the DistRev scheme. From Figures 2 and 3, we can estimate the performance of the KeyRev scheme and the DistRev scheme. For example, if the max-hops is set to five, the key revocation time of the DistRev scheme is at least 10.1 times that of the KeyRev scheme and the average energy consumption of the DistRev scheme is at least 6.8 times that of the KeyRev scheme.

Overall, the KeyRev scheme is much better than the previously proposed centralized key revocation schemes, such as the EsRev scheme and the GPSRRev scheme. It is also superior to the distributed key revocation scheme, the DistRev scheme. The superior performance of the KeyRev protocol is due to the efficient session key distribution scheme presented in Section 3.4.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a key revocation scheme, KeyRev, for wireless sensor networks. Unlike most of the key revocation schemes proposed in the literature (such as [Eschenauer and Gligor 2002; Chan et al. 2003; Chan et al. 2005]) focusing on removing the compromised keys, our proposed scheme focuses on updating the session key and thus removing the compromised sensor nodes from the network.

Previous research on key revocation have concluded that the distributed key revocation schemes are faster than the centralized key revocation schemes. For example, Chan *et al.* in [Chan et al. 2005] proposed and analyzed the security of the DistRev scheme. However, they did not evaluate its performance. In this paper, we evaluated and estimated the minimum value of the key revocation time and the average energy consumption of the DistRev scheme. To the best of our knowledge, this is also the first paper which evaluates the performance of a distributed key revocation protocol in a wireless sensor network. We found that our proposed centralized scheme, KeyRev, is much better than the distributed key revocation scheme proposed in [Chan et al. 2005]. It goes counter to the conclusion in the paper [Chan et al. 2005] which claims that the distributed key revocation scheme has better performance than any centralized key revocation scheme.

As the simulation results show, the performance of the KeyRev scheme is much better than that of other revocation schemes and the KeyRev scheme is also scalable to large-scale sensor networks. The KeyRev scheme depends on an effective session key distribution scheme in the network, which is currently based on the personal key share distribution scheme proposed in [Liu et al. 2003]. Further investigation on different session key distribution schemes will be conducted in the future. Further, the KeyRev scheme is a centralized revocation scheme and the base station is the single point of failure. A distributed key revocation scheme might be still attractive due to the avoidance of single points of failure. The integration of both centralized and distributed key revocation scheme merits under further investigation. Finally, the KeyRev scheme depends on a globally distributed session key in the network, which requires the sensor network be synchronized. Since most broadcast authentication schemes, such as $\mu Tesla$, require the synchronization of all sensor nodes in the network, it is not a problem if such broadcast authentication schemes are used. Our future work will extend the framework to scenarios where the sensor network is not synchronized.

REFERENCES

Cametepe, S. A. and Yener, B. 2004. Combinatorial design of key distribution mechanisms for wireless sensor networks. In *Proceedings of 9th European Symposium on Research Computer Security*. Sophia Antipolis, France.

Camtepe, S. A. and Yener, B. 2005. Key distribution mechanisms for wireless sensor networks: A survey. Tech. Rep. TR-05-07, Computer Science Department at RPI.

Chan, H., Gligor, V., Perrig, A., and Muralidharan, G. 2005. On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Transactions on Dependable and Secure Computing 2,* 3 (July-Sept.), 233–247.

Chan, H., Perrig, A., and Song, D. 2003. Random key predistribution schemes for sensor networks. In *Proceedings of IEEE Symposium on Security and Privacy*. Oakland, CA, USA.

Du, W., Deng, J., Han, Y. S., Chen, S., and Varshney, P. K. 2004. A key management scheme for wireless sensor networks using deployment knowledge. In *Proceedings of IEEE INFOCOM*. 586–597.

Eschenauer, L. and Gligor, V. D. 2002. A key-management scheme for distributed sensor networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. ACM Press, New York, NY, USA, 41–47.

Karp, B. and Kung, H. T. 2000. GPSR: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM Press, New York, NY, USA, 243–254.

Liu, D. and Ning, P. 2003. Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks. In *Proceedings of the 10th Annual Network and Distributed System Security Symposium*. San Diego, CA, USA, 263–276.

Liu, D. and Ning, P. 2004. Multi-level μTESLA: Broadcast authentication for distributed sensor networks. *Trans. on Embedded Computing Sys. 3,* 4, 800–836.

Liu, D., Ning, P., and Sun, K. 2003. Efficient self-healing group key distribution with revocation capability. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. ACM Press, New York, NY, USA, 231–240.

Liu, D., Ning, P., Zhu, S., and Jajodia, S. 2005. Practical broadcast authentication in sensor networks. In *MobiQuitous '05: Proceedings of The 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*. San Diego, CA, USA, 118–129.

Perrig, A., Szewczyk, R., Wen, V., Culler, D., and Tygar, J. D. 2002. SPINS: Security protocols for sensor networks. *Wireless Networks 8,* 5 (September), 521–534.

Pietro, R. D., Mancini, L. V., Law, Y. W., Etalle, S., and Havinga, P. J. M. 2003. LKHW: A directed Diffusion-Based secure multicast scheme for wireless sensor networks. In *ICPPW '03: Proceedings of the 32nd International Conference on Parallel Processing Workshops*. IEEE Computer Society Press, 397–406.

Przydatek, B., Song, D., and Perrig, A. 2003. SIA: secure information aggregation in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM Press, New York, NY, USA, 255–265.

Shamir, A. 1979. How to share a secret. *Commun. ACM 22,* 11, 612–613.

Wang, G., Zhang, W., Cao, C., and Porta, T. L. 2003. On supporting distributed collaboration in sensor networks. In *Proceedings of MILCOM*.

Wang, Y., Attebury, G., and Ramamurthy, B. 2006. A survey of security issues in wireless sensor networks. *IEEE Communications Surveys and Tutorials 8,* 2.

Wang, Y. and Ramamurthy, B. 2006. SENSIM: SEnsor Network SIMulator (Version 0.1).

Wang, Y., Ramamurthy, B., and Zou, X. 2007. KeyRev: An efficient key revocation scheme for wireless sensor networks. In *ICC '07: Proceedings of IEEE International Conference on Communications*. Glasgow, Scotland, U.K.

Ye, F., Luo, H., Lu, S., and Zhang, L. 2004. Statistical en-route filtering of injected false data in sensor networks. In *Proceedings of IEEE INFOCOM*.

Zhang, W., Song, H., Zhu, S., and Cao, G. 2005. Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks. In *MobiHoc '05: Pro-*

ceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing. ACM Press, New York, NY, USA, 378–389.

Zhu, S., Setia, S., and Jajodia, S. 2003. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. ACM Press, New York, NY, USA, 62–72.

Zhu, S., Setia, S., Jajodia, S., and Ning, P. 2004. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In *Proceedings of IEEE Symposium on Security and Privacy*. Oakland, CA, USA, 259–271.

Received August 2007;