2007

# KeyRev: An Efficient Key Revocation Scheme for Wireless Sensor Networks

Yong Wang
*University of Nebraska - Lincoln*, ywang@cse.unl.edu

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

Xukai Zou
*Indiana University-Purdue University Indianapolis*, xkzou@cs.iupui.edu

*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.*
ICC '07. IEEE International Conference on Communications, 2007.
Digital Object Identifier: 10.1109/ICC.2007.213 Publication Year: 2007 , Page(s): 1260 - 1265

# KeyRev: An Efficient Key Revocation Scheme for Wireless Sensor Networks

Yong Wang, Byrav Ramamurthy
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588 USA
Email: {ywang, byrav}@cse.unl.edu

Xukai Zou
Department of Computer and Information Science
Indiana University-Purdue University Indianapolis
Indianapolis, IN 46202 USA
Email: xkzou@cs.iupui.edu

***Abstract*— 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 summarize the current key revocation schemes for sensor networks. Then, we propose an efficient centralized key revocation scheme, KeyRev, for wireless sensor networks. Unlike most proposed key revocation schemes focusing on removing the compromised keys, we propose to use key updating 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 in wireless sensor networks.**

## I. Introduction

Wireless sensor networks (WSNs) are promising solutions for many applications and security is an essential requirement of WSNs [1]. 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 [2], [3], [4]. 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 [4], [5]. 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 [2], a pairwise key shared between the base station and the sensor node [3], or a key ring consisting of certain number of keys to be used in the future [4], [3]. The keys for secure communication, for example, *pairwise keys* [3], *path keys* [3], *cluster keys* [2] 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 attack is a specific attack in which an adversary uses the node revocation protocol to selectively revoke uncompromised sensors from the network. Revocation attack must be considered in designing a revocation scheme. Moreover, compromised nodes may act as an adversary's surrogates within a revocation protocol and subvert the execution of the revocation protocol [5]. Thus, the resistance to compromised sensors should also be evaluated in a revocation protocol. Finally, after compromised sensors are removed from the network, new sensors might be re-deployed to replace those compromised sensors. The node addition problem must also be considered.

A few schemes [3], [4], [5] 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 [3] requires a signature key distributed in the 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 [4], [5] are based on some strong assumptions such as each node knowing its neighboring nodes and each node knowing its neighboring node's neighbors before deployment. These assumptions are hard to satisfy. Thus, designing a new efficient key revocation scheme is highly desirable.

In this paper, we propose a key revocation scheme, KeyRev, for wireless sensor networks. Unlike most proposed key revocation schemes focusing on removing the compromised keys, we propose to use key updating techniques to obsolesce the keys owned by the compromised sensor nodes and thus remove the nodes from the network. Our analysis and simulation results show that our proposed key revocation scheme is secure and efficient in computation, communication and storage usage.

The remainder of this paper is organized as follows: Section II discusses the related work. Section III presents our proposed key revocation scheme. The security and performance analyses are presented in Section IV, and the simulation experiments and results in Section V. Section VI concludes the paper.

## II. Related work

As discussed before, 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 [6], [7], [8] 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.

Existing key revocation schemes can be divided into two categories: centralized key revocation scheme [3] and distributed key revocation scheme [4], [5]. We discuss these in turn below.

### A. Centralized key revocation scheme

In centralized key revocation scheme, a centralized authority (base station) is used to revoke compromised sensors [3]. Eschenauer and Gligor presented a key management scheme for WSNs in [3]. 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 accomplish this are suggested in [3]. 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 $\alpha$ 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 [3] 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 in [3] 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.

### B. 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 [4] and further investigated this scheme in [5]. 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 [9].

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

Compared with the centralized key revocation scheme, the distributed key revocation scheme is faster because it requires local broadcast and avoids a single point of failure. However, the distributed key revocation scheme is more complex than the centralized key revocation scheme. Further, 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 [4], [5]. This paper focuses on a centralized key revocation scheme.

In the remainder of this paper, we present an efficient key revocation scheme, KeyRev, for wireless sensor networks. We use the following notation in the remainder of this paper:

- $A, B$ are principals such as communicating nodes.
- $K_{AB}$ 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 $B$.

## III. KeyRev: an efficient key revocation scheme for WSNs

Unlike most of the proposed key revocation schemes focusing on removing the compromised keys, our scheme, KeyRev, uses key updating 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 [3] is used. Here, we provide an overview of our scheme.

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 revoked 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 in 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.

### A. 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, encryption and authentication 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 $K_{AB}$ 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:

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

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

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 KeyRev.

### *B. 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 [10]. 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. $\mu TESLA$ [11] and its extensions [12], [13] have been proposed to provide such services for sensor networks. We assume that a proper broadcast authentication scheme such as $\mu TESLA$ is used in the paper. Note that to use $\mu TESLA$, the sensor network should be loosely time synchronized to meet the requirements [14].

## IV. Security and performance analysis

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

### *A. Security analysis*

Our proposed key revocation 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 [10]. 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 III-B, 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 $\mu TESLA$ [15] can be used to protect the broadcast messages. To start the revocation attack, an adversary must impersonate the base station. The KeyRev scheme is immune to revocation attack if the base station is secure.

### *B. Performance analysis*

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. The performance of the KeyRev scheme depends mainly on the session key updating process. The session key can be updated in one round by broadcasting a message to all sensor nodes. Nevertheless, the centralized revocation scheme proposed in [3] depends on a signature key distributed in the sensor network by unicasting and thus incurs additional inter-sensor communication cost. Our proposed KeyRev scheme is scalable to large scale sensor networks.

To add new nodes to the sensor network, pre-distributed key materials required by the basic random key distribution scheme must be loaded on to 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 updating process, to each added sensor node.

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

## V. Simulation and results

The performance of the KeyRev scheme was evaluated in SENSIM [16], 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 guaranteed to be received at the receivers. Thus, no packet collisions are considered and the performance evaluated in the simulation is 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 networks, we set the base station at the center of 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 scheme in [3] (which is referred as EsRev scheme in the remainder of this section). The evaluation metrics include the key revocation time and the average energy consumption 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 both the KeyRev scheme and the EsRev 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 I. For each testbed, we randomly select one sensor to be revoked and run the simulation ten times. The average value is measured.

TABLE I
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 | 800$\mu$s |

Figures 1 and 2 show the key revocation time and the average energy consumption to revoke a compromised sensor in the 100-node sensor network. As the figures show, the key revocation time and the average energy consumption to revoke a single sensor by using the EsRev scheme is about 83 and 73 times that of the KeyRev scheme.

Figures 3 and 4 show the key revocation time and the average energy consumption to revoke a compromised sensor in the 1000-node sensor network. As the figures show, the key revocation time and the average energy consumption to revoke a single sensor by using the EsRev scheme are about 805 times that of the KeyRev scheme.

In both the experimental settings, the KeyRev scheme performs far better than the EsRev scheme. Further, unlike the EsRev scheme, the figures also show that the key revocation time and the average energy consumption to revoke a single sensor node by using the 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, as figure 3 shows, it takes a long time ($> 8mins$) for the EsRev scheme to remove a compromised sensor node in the 1000-node sensor network. Thus, the EsRev scheme is not scalable to large-scale sensor networks.

## VI. 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 [3], [4], [5] focusing on removing the compromised keys, our proposed scheme focuses on updating the session key and thus avoids the communication overhead to distribute the signature key to each sensor in the network.

As the simulation results show, the performance of the KeyRev scheme is far better than that of the EsRev scheme and the KeyRev scheme is also scalable to large-scale sensor networks.

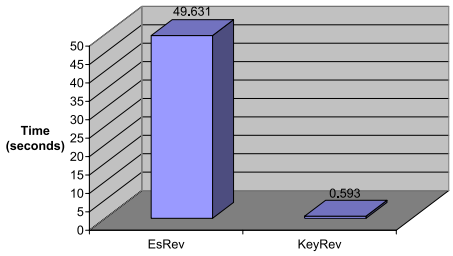


Fig. 1. Key revocation time in the 100-node sensor network. The EsRev key revocation time is 83 times that of the KeyRev scheme.

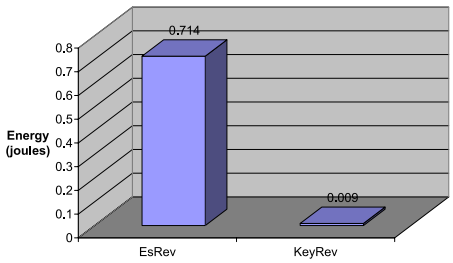


Fig. 2. Average energy consumption per node to revoke a compromised sensor in the 100-node sensor network. The average energy consumption to execute the EsRev protocol per node is 73 times that of the KeyRev scheme.

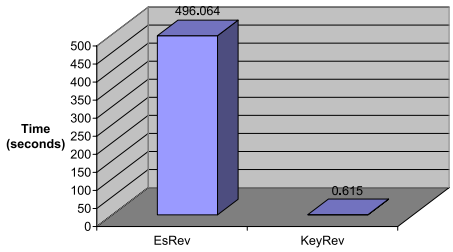


Fig. 3. Key revocation time in the 1000-node sensor network. The EsRev key revocation time is 806 times that of the KeyRev scheme.

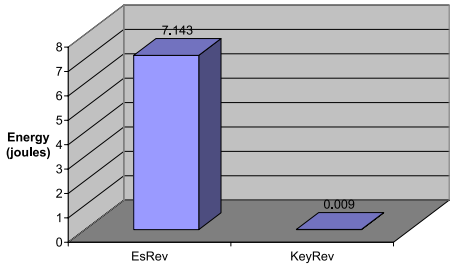


Fig. 4. Average energy consumption to revoke a compromised sensor in the 1000-node sensor network. The average energy consumption per node to revoke a compromised sensor is 805 times that of the KeyRev scheme.

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 [10]. Further investigation on different session key distribution schemes will be conducted in the future.

## ACKNOWLEDGEMENTS

This work is partially supported by NSF Grant No. CCR-0311577. The authors thank the anonymous reviewers for their valuable comments on this manuscript.

## REFERENCES

[1] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 2, 2006.

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

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

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

[5] H. Chan, V. Gligor, A. Perrig, and G. Muralidharan, "On the distribution and revocation of cryptographic keys in sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, pp. 233–247, July-Sept. 2005.

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

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

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

[9] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

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

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

[12] D. Liu and P. Ning, "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, February 2003, pp. 263–276.

[13] D. Liu and P. Ning, "Multi-level $\mu$TESLA: Broadcast authentication for distributed sensor networks," *Trans. on Embedded Computing Sys.*, vol. 3, no. 4, pp. 800–836, 2004.

[14] D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical broadcast authentication in sensor networks," in *MobiQuitous '05: Proceedings of The 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, July 2005, pp. 118–129.

[15] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, September 2002.

[16] Y. Wang and B. Ramamurthy, "SENSIM: SEnsor Network SIMulator (Version 0.1)," August 2006. [Online]. Available: http://cse.unl.edu/∼ywang/sensim.htm