18

A Compromise-resilient Pair-wise Rekeying Protocol in Hierarchical Wireless Sensor Networks

Song Guo
School of Computer Science and Engineering The University of Aizu, Japan
sguo@u-aizu.ac.jp

Zhuzhong Qian
State Key Laboratory for Novel Software Technology Nanjing University, China
qzz@nju.edu.cn

1. Introduction

Wireless sensor networks (WSNs) have been envisioned to be very useful for a broad spectrum of emerging civil and military applications (Akyildiz et al., 2002). However, sensor networks are also confronted with many security threats such as node compromise, routing disruption and false data injection, because they are normally operated in an unattended, harsh or hostile environment. Among all these threats, the WSNs are particularly vulnerable to the node compromise because sensor nodes are not tamper-proof devices. When a sensor network is deployed in unattended and hostile environments such as battlefield, the adversaries may capture and reprogram some sensor nodes, or inject some malicious sensor nodes into the network and make the network accept them as legitimate nodes. After getting control of a few nodes, the adversary can mount various attacks from inside the network (Zhang et al., 2008). Therefore, it is desirable to design key distribution protocols to support secure and robust pair-wise communication among any pair of sensors.

This is a challenging task in sensor networks because they have scarce resources in energy, computation and communication. As a result, the conventional asymmetric key cryptosystem, such as RSA (Rivest et al., 1978) and Diffie-Hellman (Diffie & Hellman, 1976), can not be implemented in sensor nodes due to their very limited capacities and only lightweight energy efficient key distribution mechanisms are affordable. Furthermore, sensor nodes are low-cost and they cannot afford tamper-resistance hardware. Recent advances in physical attack show that even memory chips with built-in tamper-resistance mechanisms are subject to various memory read-out attacks. Thus, an adversary might easily capture the sensor devices to acquire their sensitive data and keys and then abuse them to further compromise the communication between other non-captured nodes. In order to conquer such *node capture attack* (NCA) problem, it is desirable to design protocols to support secure and robust pair-wise communication among any pair of sensors.

To defend against such attack, the security mechanisms in WSNs are required. Most of existing key management schemes focus on the efficiency of bootstrapping session keys which has been intensively studied in the literature of WSNs (Cheng & Agrawal, 2005; Du et al., 2003; Eschenauer & Gligor, 2002). Traditionally, once such key system is adopted, the whole security system is established and fixed. However, when the WSN runs for a long time using a fixed key, it enhances the probability for the adversaries to decrypt the key by analyzing the adequate messages eavesdropped or capturing some nodes. Under this circumstance, the entire network security might be threatened. Thus, it is necessary to update this key with a new key periodically to maintain backward secrecy (Mishra, 2002). The idea is to prevent a node with the new key from going backwards in time to decipher previous content encrypted with prior keys. Likewise, when a node leaves, it is necessary to update the key to maintain forward secrecy (Mishra, 2002). The idea is to prevent a node from using an old key to continue to decrypt new content.

WSNs can be broadly classified into flat WSNs and hierarchical WSNs. It has been shown in (Cheng & Agrawal, 2007) that a hierarchical architecture can provide better performance, in terms of communication overhead, than a flat architecture in such networks. This is the major reason why most recent lightweight energy efficient rekeying mechanisms are proposed for hierarchical WSNs. In a flat WSN, all senor nodes have the same computational and communication capacities. In a hierarchical WSN, however, some special sensor devices, called Cluster Head (CH), have much higher capacities than other sensor nodes. By applying some clustering algorithms like (Heinzelman et al., 2002), the whole set of sensor devices could be partitioned into several distinct clusters such that each cluster has at least one CH. Under this arrangement, each sensor node forwards the generated packets to its local CH by short-range transmissions, and the CH then performs a pre-processing for the raw data received from all other senor nodes in the cluster and finally forwards the aggregated data to the sink node, or Base Station (BS), by long-range transmissions.

Most existing polynomial-based rekeying schemes suffer the node capture attack. Let us examine Chadha's rekeying protocol proposed in (Chadha et al., 2005) as an example to show its vulnerability to NCA. The basic idea is that the rekeying message from a CH can disallow the compromised nodes to renew their pair-wise keys. In the pre-loading phase, each sensor node S_i is pre-loaded the secret values $h(S_i)$ obtained from a $2t$-degree masking polynomial $h(x)$. This scheme assumes that each CH has the intrusion detection capacity. In the rekeying phase, the CH generates a t-degree secrecy polynomial $f(x)$ and constructs $w(x)$ as $w(x) = g(x)f(x) + h(x)$, where $g(x)$ is constructed using the Ids of all compromised nodes. Once $g(x)$ is evaluated at the Id of any malicious node, the result will be equal to 0. The CH then broadcasts $w(x)$ and the Id list of all detected compromised nodes throughout the whole group members. Upon receiving the message, any non-revoked node S_i can compute the new pair-wise key $f(S_i)$ between sensor node S_i as follows: $f(S_i) = (w(S_i) - h(S_i))/g(S_i)$. We observe that if there are $(2t + 1)$ nodes are compromised in an arbitrary rekeying phase, the $2t$-degree polynomial $h(x)$ can be derived. Recalling that the polynomials $w(x)$ and $g(x)$ are public, we conclude that $f(x)$ can be derived as well and used to calculate the pair-wise key in any given rekeying phase. In addition, their vulnerability to the node capture attack disables them from supporting both forward and backward secrecy. This motivates us to design a new compromise-resilient pair-wise rekeying scheme with strong resistance to such attack.

The rest of this paper is organized as follows. Section 2 presents our system model and gives an overview of background knowledge. Section 3 describes a perturbation based pair-wise rekeying protocol. Sections 4 and 5 evaluate the security and the performance of our proposal, respectively. Section 6 summarizes our findings.

2. Preliminaries

2.1 Network Model

As in other hierarchical models of sensor network (Cheng & Agrawal, 2007; Zhang et al., 2005), our system also assumes that a sensor network is divided into clusters, which are the minimum unit for detecting events. A cluster head coordinates all the actions inside a cluster and each pair of cluster heads in their transmission range can communicate directly with each other. Each low-cost sensor node (SN) has low data processing capability, limited memory storage and battery power supplies, and short radio transmission range. The CHs are equipped with richer resources (*e.g.*, higher power batteries, large memory storages, powerful antenna, *etc.*) and higher data processing capacities, and thus can execute relatively complicated numerical operations. Moreover, we assume a single base station (BS) or an access point (AP) in the network and works as the network controller to collect event data. The information collected by cluster heads from all its sensor nodes is retrieved by a BS or a AP periodically. During the information retrieval operation, the BS/AP broadcasts a beacon to activate cluster heads in its coverage area. Activated cluster heads then transmit their data to the BS/AP through a common wireless channel. As the most powerful node in a WSN, the BS/AP has virtually unlimited memory storage capacity and sufficiently large radio transmission range to reach all other devices in a network.

Under such model, we say the link (v, u), corresponding to the wireless communication channel between nodes v and u, is secure if they share a secret pairwise key $K_{v,u}$. Due to the constrained resources, computationally expensive and energy-intensive operations for pairwise key establishment are not favorable for such systems. In addition, each sensor node is not tamper-resistant. Once a sensor node is captured, the adversary can read its memory to get all information stored there. Schemes for key predistribution enable nodes in a large network to agree on pairwise secret keys. The sensor network is administrated by an offline authority, which is responsible for node initialization and deployment. Before deploying a node, the authority assigns the node a unique identity (ID) from a set of legitimate IDs and some secret information that will be used to allow any two nodes v and u to agree on a shared key $K_{v,u}$.

2.2 Symmetric Polynomial Function

As the basis of our pair-wise rekeying protocol for any wireless link between a CH and a SN, the polynomial-based key predistribution scheme originally proposed in (Blundo et al., 1993) works as follows.

Let F_q be a finite field, in which q is the maximum prime number satisfying $q < 2^{\ell}$ that can accommodate a cryptographic key with ℓ bits. The elements of F_q can be used as pairwise keys. To achieve t-resilience using the Blundo's scheme (Blundo et al., 1993), the authority chooses a random symmetric bivariate polynomial $f \in F_q[x, y]$ of degree t in each variable as the master secret polynomial:

$$f(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{ij} x^i y^j. \tag{1}$$

The coefficients a_{ij} $(a_{ij} = a_{ji})$ are randomly chosen from F_q. A node with Id $u \in F_q$ is preloaded the univariate polynomial:

$$g_u(y) = f(u, y). \tag{2}$$

The shared key $K_{v,u}$ between nodes v and u is

$$g_v(u) = f(v, u) = g_u(v), \tag{3}$$

which both parties can compute using the fact that $f(x, y)$ is symmetric. The security proof in (Blundo et al., 1993) ensures that this scheme is unconditionally secure and t-collusion resistant; *i.e.*, a coalition of no more than t compromised nodes cannot know anything about the key shared by any two non-compromised nodes. However, an attacker who compromises $t+1$ nodes can use interpolation to recover the master polynomial $f(x, y)$.

By applying the symmetric property, a secure link can be easily built up by just exchanging the IDs of transmission nodes. On the other hand, a t-degree bivariate polynomial key scheme can only keep secure against coalitions of up to t compromised sensors. Although increasing the value of t can improve the security property of bivariate polynomial key scheme, it is not suitable for wireless sensor networks due to the limited memory size of sensors.

2.3 Perturbation Polynomial Function

Our proposed pair-wise rekeying protocol exploits the characteristic of the perturbation polynomial, which was originally introduced in (Zhang et al., 2007). Given a finite field F_q, a positive integer r ($r < \ell$), and a set of node Ids S ($S \subset \{0, \cdots, q-1\}$), a polynomial set Φ is a set of perturbation polynomials regarding r and S if any polynomial $\phi(\cdot) \in \Phi$ has the following *limited infection* property:

$$\forall x \in S, \phi(x) \in \{0, \cdots, 2^r - 1\}. \tag{4}$$

According to the above definition, the value of a perturbation polynomial will not be larger than $(2^r - 1)$, *i.e.*, it has at most r bits. This property is used to design perturbation-based scheme. If let an r-bit number add to a ℓ-bit number, only the least significant r-bit of the ℓ-bit numer will be directly affected. Wheather the most significant $(\ell - r)$ bits are changed or not will hinge on if a carry is generated from the least significant r bits in the addition process. For example, we assume $\ell = 6$ and $r = 4$. The addition $(101001)_2 + (0101)_2 = (101110)_2$ changes the least significant 4-bits but not the most $\ell - r = 2$ significant bits of the first operand, but $(101001)_2 + (1100)_2 = (110101)_2$ not only changes least significant 4-bits but also the most significant 2 bits, because a carry is generated from the least significant 4-bits.

3. A Pair-wise Rekeying Protocol

In general, the design of a light-weight compromise-recilient rekeying scheme in WSNs is difficult because of the vulnerability of sensor nodes and the constrained system resources. Due to these challenges, a practical pair-wise rekeying scheme for WSNs should be resilient to large number of node compromises, be efficient in computation, communication, and storage, and allow both full and direct key establishment. In this section, we present a perturbation-based pair-wise rekeying protocol that can achieve all these goals.

In the basic polynomial-based scheme (Blundo et al., 1993), where any two nodes (with IDs u and v) are given shares ($f(u, y)$ and $f(v, y)$) of a symmetric polynomial $f(x, y)$, they can always find a match $f(u, v)$ to be used as the shared key of size ℓ bits. Different from this, our rekeying scheme does not use shares generated from symmetric polynomial but perturbation polynomials such that (1) a match can still be achived and (2) the shared key is difficult to crack by large-scale NCAs. To further explain the above basic idea, we now introduce the three major steps of the rekeying scheme: system initialization, pre-distribution of perturbed polynomials, and key establishment and rekeying. In order to present it in a formal way, we list the notations used in our protocol descriptions in Table 1 for convenience to the readers.

3.1 System Initialization

We assume that there are n sensor nodes to be deployed in the network. The node deployment can be done by only once, or several times in order to extend the lifetime of the network with

Notation	**Description**
CH_a	The Id of cluster head a
CS_k	The Id of compromised sensor node k
$E(\texttt{data, } K)$	An encryption function using K as a key
$f(x,y)$	a symmetric polynomial
F_q	a finite field with any element that can be represented by ℓ bits
$g_u(y)$	the univariate polynomial for node u obtained by $g_u(y) = f(u,y)$
$\bar{g}_u(y)$	the perturbed polynomial preloaded to node u
$H^k(x)$	the hashed value based on the most significant k bits of x
$K_{a,b}$	the shared pairwise key between nodes a and b
ℓ	the minimal integer satisfying $2^\ell > q$
n	the total number of sensor nodes to be deployed, $n < q$
n_a	the number of sensor nodes in a cluster
n_c	the number of compromised sensor nodes in a cluster
m	the total number of perturbation polynamials, $m = \lvert\Phi\rvert$
$p_u(y)$	a randomly generated univariate rekeying polynomial at node u
q	a large prime number
r	a positive integer such that $2^r < q$
S	a set of legitimate IDs for sensor nodes, $S \subset \{0, \cdots, q-1\}$
SN_i	The Id of sensor node i
t	the degree of both variables x and y in the symmetric polynomial $f(x,y)$
$\phi_u(y)$	a perturbation polynamial assigned for node u
Φ	a set of perturbation polynamials satisfying the limited infection property regarding r and S

Table 1. Notations

the renewed nodes. Based on the number n, a large prime number q is chosen such that $n < q$ and let ℓ be the minimal integer satisfying $2^\ell > q$.

The offline authority arbituary constructs a bivariate symmetric polynomial $f(x,y) \in F_q[x,y]$, where the degrees of x and y are both t, and for any $x, y \in F_q$, $f(x,y) = f(y,x)$. It then applies the method in (Zhang et al., 2007) to construct the legitimate ID set S for sensor nodes and the perturbation polynamial set Φ, which satisfies the limited infection property regarding r and S with m ($m \geq 2$) number of bivariate symmetric polynomials. Finally, we note that the desired number of bits for any pairwise key is $\ell - r$.

3.2 Pre-distribution of Perturbed Polynomials

Before sensor devices are deployed into usage, some secret information should be pre-assigned as follows. Each cluster head a needs to be preloaded with a unique Id $CH_a \in S$ and a perturbed polynomial $\overline{g}_{CH_a}(y)$:

$$\overline{g}_{CH_a}(y) = f(CH_a, y) + \phi_{CH_a}(y) = g_{CH_a}(y) + \phi_{CH_a}(y). \tag{5}$$

Similarly, for each sensor node i, the security server preloads it with a unique Id $SN_i \in S$ and a perturbed polynomial $\overline{g}_{SN_i}(y)$:

$$\overline{g}_{SN_i}(y) = f(SN_i, y) + \phi_{SN_i}(y) = g_{SN_i}(y) + \phi_{SN_i}(y). \tag{6}$$

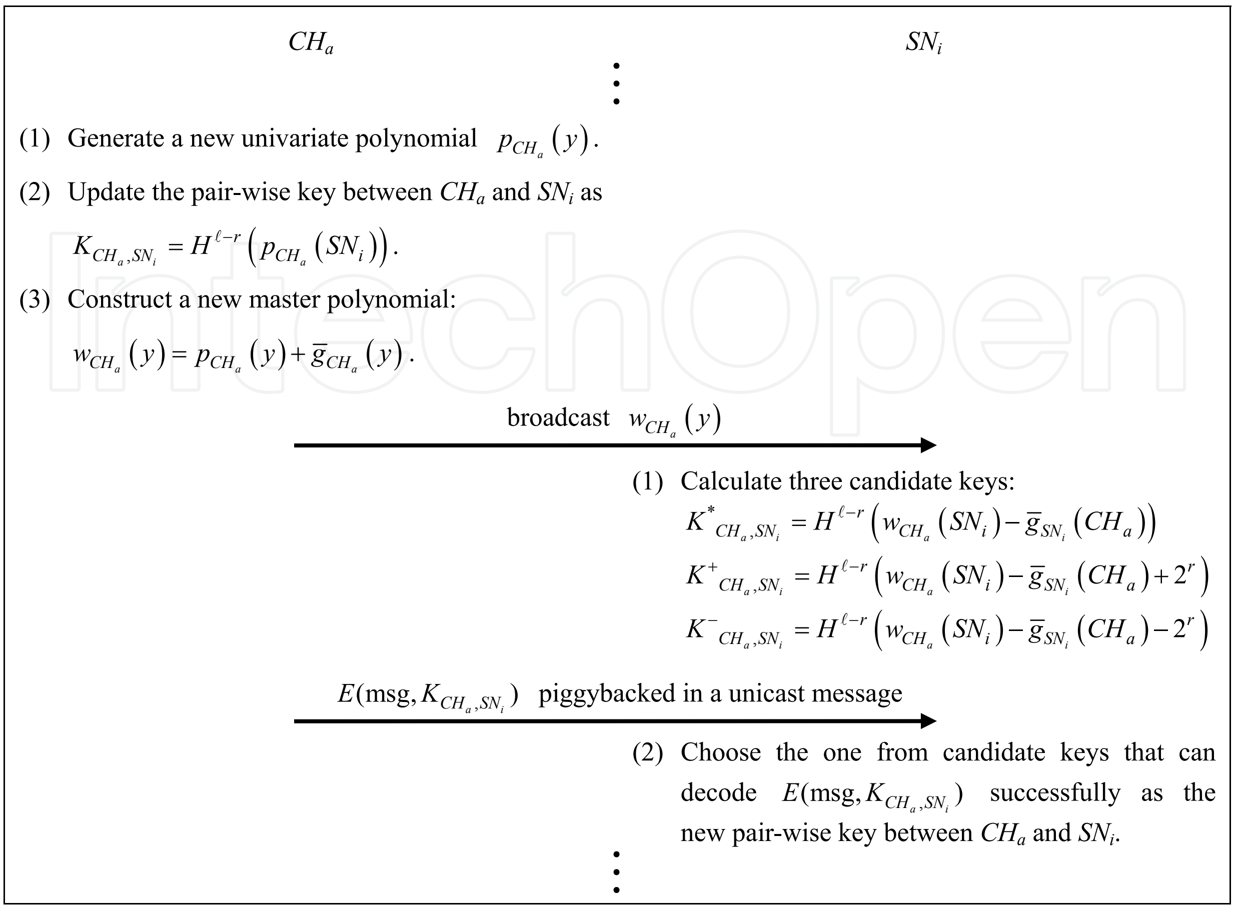

CH_a
SN_i
(1) Generate a new univariate polynomial $p_{CH_a}(y)$.
(2) Update the pair-wise key between CH_a and SN_i as
$K_{CH_a,SN_i} = H^{\ell-r}\left(p_{CH_a}(SN_i)\right)$.
(3) Construct a new master polynomial:
$w_{CH_a}(y) = p_{CH_a}(y) + \overline{g}_{CH_a}(y)$.
broadcast $w_{CH_a}(y)$
(1) Calculate three candidate keys:
$K^{*}_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a)\right)$
$K^{+}_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a) + 2^r\right)$
$K^{-}_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a) - 2^r\right)$
$E(\text{msg}, K_{CH_a,SN_i})$ piggybacked in a unicast message
(2) Choose the one from candidate keys that can decode $E(\text{msg}, K_{CH_a,SN_i})$ successfully as the new pair-wise key between CH_a and SN_i.

Fig. 1. The protocol for pair-wise key establishment and rekeying

Note that the security authority only preloads each sensor device u (a CH or SN) the coefficients of $\overline{g}_u(y)$. Hence, each sensor device cannot extract from $\overline{g}_u(y)$ the coefficients of the original polynomial shares of either $f(x,y)$, $f_u(y)$, or $\phi_u(y)$ ($\phi_u(\cdot) \in \Phi$). Furthermore, each sensor device is equipped with the same one-way hash function $H^k(x)$, which returns the hashed value based on the most significant k bits of x.

3.3 Pair-wise Key Establishment and Rekeying

After the key pre-assignment phase, wireless sensors are randomly distributed in a given area, and later on, some clustering algorithm, *e.g.*, (Heinzelman et al., 2002), shall organize the network into a hierarchical structure. The following intra-cluster protocol, as illustrated in Figure 1, is to establish the new pair-wise key between a cluster head a and one of its member sensor nodes i in a new round of rekeying phase, in which the orignal pair-wise key establishment is treated the same as the subsequent rekeyings. The inter-cluster rekeying protocol for CH-CH links works in a similar manner and thus is omitted here.

- Step 1: At the beginning of each rekeying phase, CH_a randomly generates a new t-degree univariate rekeying polynomial function $p_{CH_a}(y)$. For each of its sensor node SN_i, CH_a updates the corresponding pair-wise key K_{CH_a,SN_i} as

$$K_{CH_a,SN_i} = H^{\ell-r}(p_{CH_a}(SN_i)). \tag{7}$$

- Step 2: CH_a uses $p_{CH_a}(y)$ and the preloaded polynomial $\overline{g}_{CH_a}(y)$ to construct a master polynomial $w_{CH_a}(y)$:
$$w_{CH_a}(y) = p_{CH_a}(y) + \overline{g}_{CH_a}(y) \tag{8}$$
and broadcasts its ID CH_a and this polynomial $w_{CH_a}(y)$ to all its sensor nodes by a single transmission.
- Step 3: Upon receiving the broadcast message, each SN_i evaluates the preloaded polynomial $\overline{g}_{SN_i}(y)$ at $y = CH_a$ and evaluates the receieved master polynomial $w_{CH_a}(y)$ at $y = SN_i$. After that, three candidate keys $K^*_{CH_a,SN_i}$, $K^+_{CH_a,SN_i}$ and $K^-_{CH_a,SN_i}$ will be calculated as follows, respectively.
$$K^*_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a)\right) \tag{9}$$
$$K^+_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a) + 2^r\right) \tag{10}$$
$$K^-_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a) - 2^r\right) \tag{11}$$
- Step 4: At a later time, a encoded information $E(\text{msg}, K_{CH_a,SN_i})$ will be piggybacked in a normal unicast message sent from CH_a to SN_i. The exact new pair-wise key is determined by SN_i once such message can be decoded successfully using one of the candidate keys.

Note that due to the characteristic of the perturbation polynomial (Zhang et al., 2007), only one of the candidate keys (9) - (11) will be validated as the new pair-wise key between SN_i and CH_a, *i.e.*,
$$K_{CH_a,SN_i} \in \left\{K^*_{CH_a,SN_i}, K^+_{CH_a,SN_i}, K^-_{CH_a,SN_i}\right\}. \tag{12}$$
The unicast message can be also sent from SN_i to CH_a. Under this circumstance, the new pair-wise key will be calculated at SN_i as $K_{CH_a,SN_i} = H^{\ell-r}\left(w_{CH_a}(SN_i) - \overline{g}_{SN_i}(CH_a)\right)$, while three candidate keys will be evaluated at CH_a as $K^*_{CH_a,SN_i} = H^{\ell-r}\left(p_{CH_a}(SN_i)\right)$, $K^+_{CH_a,SN_i} = H^{\ell-r}\left(p_{CH_a}(SN_i) + 2^r\right)$, and $K^-_{CH_a,SN_i} = H^{\ell-r}\left(p_{CH_a}(SN_i) - 2^r\right)$. All remaining rekeying processes are the same and conclusion in (12) will be also made.

3.4 Examples

To help understand the details of our rekeying protocol, we provide the following simplified example with $CH_a = 3$ and $SN_i = 2$. In system initialization, we set $q = 127$, $t = 2$, $\ell = 7$, and $r = 3$. All arithmetic operations are over finite field F_{127}. The bivariate symmetric polynomial is $f(x, y) = xy^2 + x^2y + 2xy + 5$ and the corresponding univariate polynomials for CH_a and SN_i are $g_3(y) = f(3, y) = 3y^2 + 15y + 5$ and $g_2(y) = f(2, y) = 2y^2 + 8y + 5$, respectively. Now, we consider the following cases in a rekeying phase, in which CH_a generates a new univariate polynomial function $p_3(y) = 3y^2 + 15y + 9$ under different preloaded perturbed polynomials.

Case 1: Suppose the perturbation polynomials for CH_a and SN_i are $\phi_3(y) = y^2 - 3y + 5$ and $\phi_2(y) = y^2 - 4y + 5$, respectively. Note that both polynomials satisfy the limited infection property: $\phi_3(2) = 3 \in \{0, 1, \cdots, 7\}$ and $\phi_2(3) = 2 \in \{0, 1, \cdots, 7\}$. Their preloaded polynomials are therefore $\overline{g}_3(y) = g_3(y) + \phi_3(y) = 4y^2 + 12y + 10$ and $\overline{g}_2(y) = g_2(y) + \phi_2(y) = 3y^2 + 4y + 10$, respectively, as illustrated in Figure 2. In rekeying, CH_a calculates the new pair-wise key as $K_{3,2} = H^4(p_3(2)) = H^4(51) = H^4(0110011)$ and

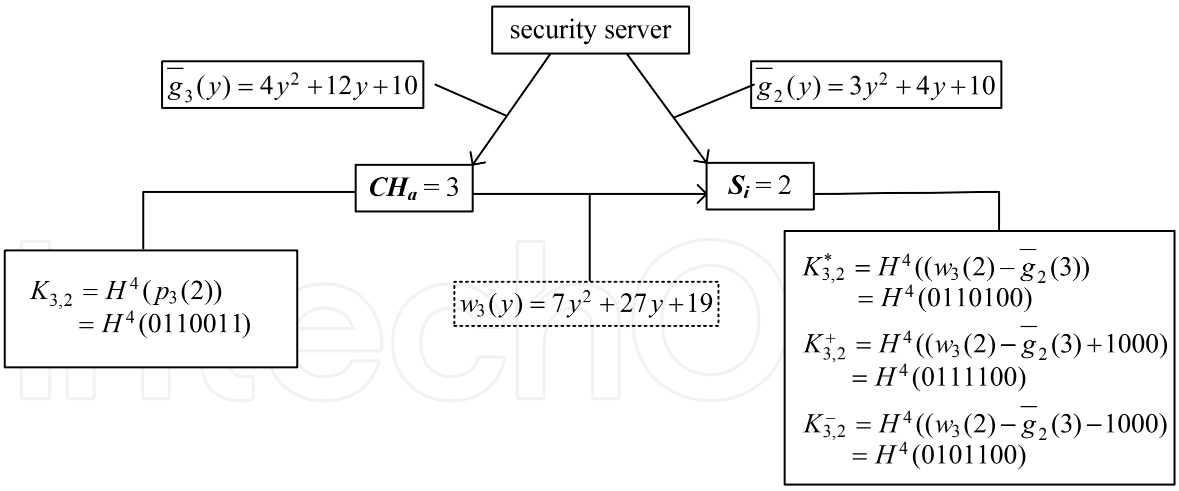

security server
$\overline{g}_3(y) = 4y^2 + 12y + 10$
$\overline{g}_2(y) = 3y^2 + 4y + 10$
$CH_a = 3$
$S_i = 2$
$K_{3,2} = H^4(p_3(2))$
$= H^4(0110011)$
$w_3(y) = 7y^2 + 27y + 19$
$K^*_{3,2} = H^4((w_3(2) - \overline{g}_2(3))$
$= H^4(0110100)$
$K^+_{3,2} = H^4((w_3(2) - \overline{g}_2(3) + 1000)$
$= H^4(0111100)$
$K^-_{3,2} = H^4((w_3(2) - \overline{g}_2(3) - 1000)$
$= H^4(0101100)$

Fig. 2. Example of $K_{CH_a,SN_i} = K^*_{CH_a,SN_i}$

sends the master polynomials $w_3(y) = p_3(y) + \overline{g}_3(y) = 7y^2 + 27y + 19$ to SN_i. At SN_i side, it then calculates three candidate keys: $K^*_{3,2} = H^4(w_3(2) - \overline{g}_2(3)) = H^4(52) = H^4(0110100)$, $K^+_{3,2} = H^4(60) = H^4(0111100)$, and $K^-_{3,2} = H^4(44) = H^4(0101100)$. We observe that $K_{CH_a,SN_i} = K^*_{CH_a,SN_i}$ ($H^4(0110011) = H^4(0110100)$) is achieved.

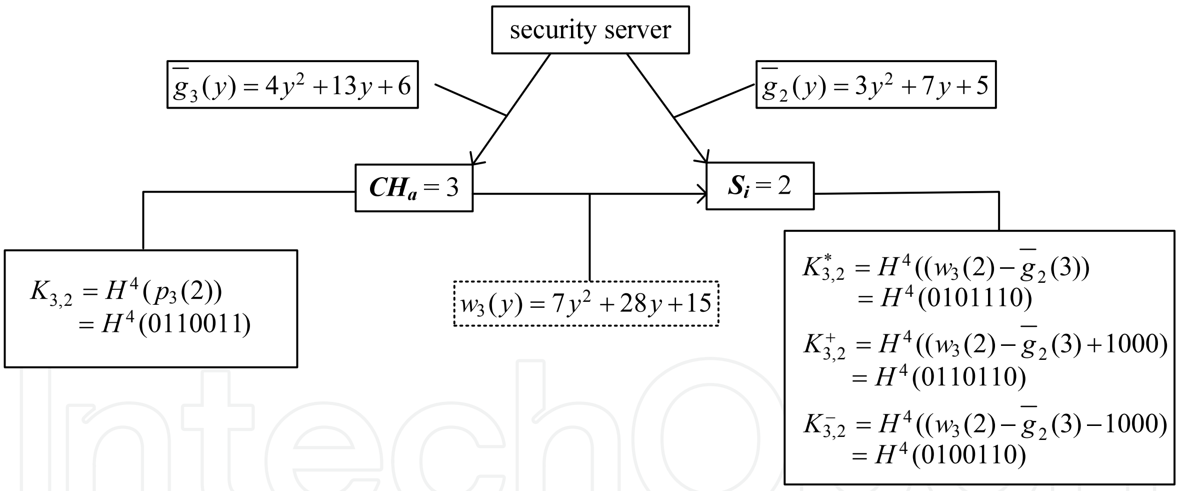

security server
$\overline{g}_3(y) = 4y^2 + 13y + 6$
$\overline{g}_2(y) = 3y^2 + 7y + 5$
$CH_a = 3$
$S_i = 2$
$K_{3,2} = H^4(p_3(2))$
$= H^4(0110011)$
$w_3(y) = 7y^2 + 28y + 15$
$K^*_{3,2} = H^4((w_3(2) - \overline{g}_2(3))$
$= H^4(0101110)$
$K^+_{3,2} = H^4((w_3(2) - \overline{g}_2(3) + 1000)$
$= H^4(0110110)$
$K^-_{3,2} = H^4((w_3(2) - \overline{g}_2(3) - 1000)$
$= H^4(0100110)$

Fig. 3. Example of $K_{CH_a,SN_i} = K^+_{CH_a,SN_i}$

Case 2: Under different perturbation polynomials $\phi_3(y) = y^2 - 2y + 1$ ($\phi_3(2) = 1$) for CH_a and $\phi_2(y) = y^2 - y$ ($\phi_2(3) = 6$) for SN_i, we can obtain $\overline{g}_3(y) = g_3(y) + \phi_3(y) = 4y^2 + 13y + 6$, $\overline{g}_2(y) = g_2(y) + \phi_2(y) = 3y^2 + 7y + 5$, and $w_3(y) = p_3(y) + \overline{g}_3(y) = 7y^2 + 28y + 15$. Eventually, we observe $K_{CH_a,SN_i} = K^+_{CH_a,SN_i}$ ($H^4(0110011) = H^4(0110110)$) as shown in Figure 3.

Case 3: Similarly, the perturbation polynomials $\phi_3(y) = y^2 - 6y + 14$ ($\phi_3(2) = 6$) and $\phi_2(y) = y^2 - 7y + 13$ ($\phi_2(3) = 1$) are for CH_a and SN_i, respectively. We then obtain $\overline{g}_3(y) = g_3(y) + \phi_3(y) = 4y^2 + 9y + 19$, $\overline{g}_2(y) = g_2(y) + \phi_2(y) = 3y^2 + y + 18$, and $w_3(y) = p_3(y) + \overline{g}_3(y) =$

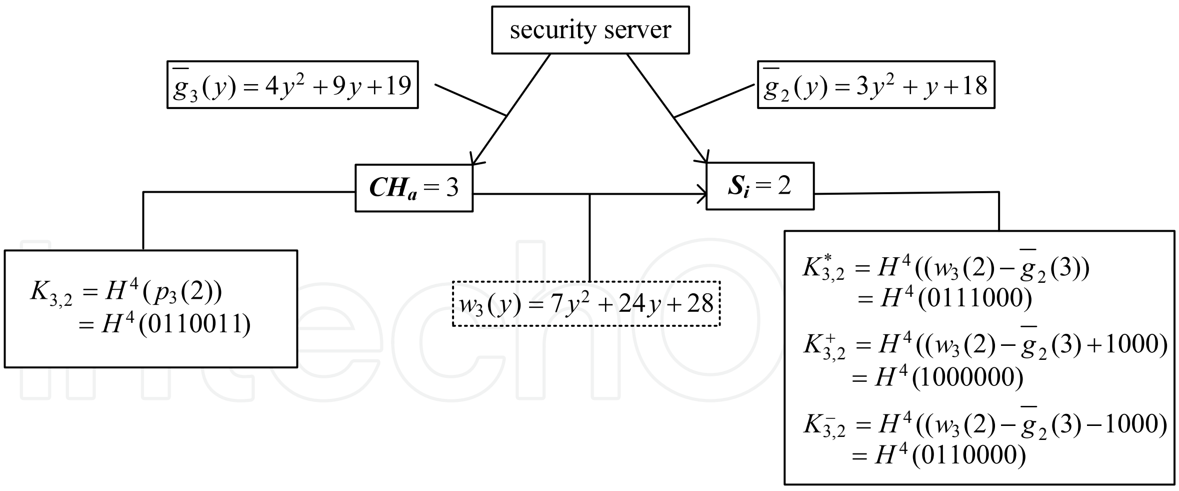

security server
$\overline{g}_3(y) = 4y^2 + 9y + 19$
$\overline{g}_2(y) = 3y^2 + y + 18$
$\boldsymbol{CH_a} = 3$
$\boldsymbol{S_i} = 2$
$K_{3,2} = H^4(p_3(2)) = H^4(0110011)$
$w_3(y) = 7y^2 + 24y + 28$
$K^*_{3,2} = H^4((w_3(2) - \overline{g}_2(3)) = H^4(0111000)$
$K^+_{3,2} = H^4((w_3(2) - \overline{g}_2(3) + 1000) = H^4(1000000)$
$K^-_{3,2} = H^4((w_3(2) - \overline{g}_2(3) - 1000) = H^4(0110000)$

Fig. 4. Example of $K_{CH_a,SN_i} = K^-_{CH_a,SN_i}$

$7y^2 + 24y + 28$. The final case $K_{CH_a,SN_i} = K^-_{CH_a,SN_i}$ $(H^4(0110011) = H^4(0110000))$ is shown in Figure 4.

4. Security Analysis

In this section, we give a security analysis for our proposed rekeying scheme and compare it to other proposals in terms of robustness to the node capture attack.

4.1 Breaking Rekeying Polynomial $p_{CH_a}(y)$

We assume that an adversary has compromised n_c sensor nodes in cluster a, denoted as CS_k $(k = 1, \cdots, n_c > t)$, and has obtained all their preloaded information.

To derive the polynomial $p_{CH_a}(y)$ that is used to generate the new pair-wise key as shown in (7), the adversary needs to break $\overline{g}_{CH_a}(y)$ because $p_{CH_a}(y) = w_{CH_a}(y) - \overline{g}_{CH_a}(y)$, in which $w_{CH_a}(y)$ is the public information broadcasted by CH_a. Furthermore, for any sensor node y of CH_a, the corresponding pair-wise key $K_{CH_a,y}$ satisfies:

$$\begin{aligned} K_{CH_a,y} &= H^{\ell-r}\left(w_{CH_a}(y) - \overline{g}_{CH_a}(y)\right) \\ &= H^{\ell-r}\left(w_{CH_a}(y) - g_{CH_a}(y) - \phi_{CH_a}(y)\right) \\ &= \begin{cases} H^{\ell-r}\left(w_{CH_a}(y) - g_{CH_a}(y)\right), & \text{or} \\ H^{\ell-r}\left(w_{CH_a}(y) - g_{CH_a}(y) - 2^r\right). \end{cases} \end{aligned}$$

The above equation shows that to break $\overline{g}_{CH_a}(y)$ is equivalent to break $g_{CH_a}(y)$ or $f(CH_a, y)$. This can be done by collecting a number of polynomials $\overline{g}_{CS_k}(y)$ stored in the compromised sensor nodes, which satisfy

$$\overline{g}_{CS_k}(y) = f(CS_k, y) + \phi_{CS_k}(y). \tag{13}$$

It can be formulated as a linear equation system as follows.

$$\sum_{i=0}^{t} a_{ij} \cdot (CS_k)^i + b_{kj} = d_{kj}, 0 \le j \le t, 1 \le k \le n_c \tag{14}$$

Note that a_{ij} and b_{kj} are the variables of this linear equation system, which are defined by (1) and the following equation

$$\phi_{CS_k}(y) = \sum_{j=0}^{t} b_{kj} \cdot y^j, 1 \le k \le n_c, \tag{15}$$

respectively. On the other hand, the values of d_{kj} are known to the adversary:

$$\overline{g}_{CS_k}(y) = \sum_{j=0}^{t} d_{kj} \cdot y^j, 1 \le k \le n_c. \tag{16}$$

By applying a similar reasoning technique in (Zhang et al., 2007), we can derive that the probabilities to find the solution of the linear equation system (14) in one attempt is $m^{-(t+1)}$, in which m is the total number of perturbation polynamials, *i.e.*, $m = |\Phi| \ge 2$. In other words, to break $f(x,y)$, or $g_{CH_a}(y) = f(CH_a, y)$, in one attempt is $m^{-(t+1)}$. Finally, we can conclude that the computational complexity for breaking $p_{CH_a}(y)$ under the condition of $t+1$ compromised nodes is $\Omega\left(m^{t+1}\right)$.

4.2 Node Capture Attack

After deployment, each cluster head and each sensor node can be captured and compromised by attackers due to the unattended deployment environments and their lack of tamper-resistance. The adversary can read out all information stored in the node to get all secret information. In addition, the attackers may collect the secrets owned by compromised nodes, and attempt to derive the secrets held by innocent nodes (and therefore can cheat these innocent nodes or impersonate as them). This is the well-known node capture attack.

In the Chadha's scheme (Chadha et al., 2005), each sensor node SN_i is pre-loaded a $2t$-degree masking polynomial $h(x)$ in its storage. After $2t$ sensor nodes are compromised, the whole network will crash. In our proposed pair-wise rekeying protocol, in order to derive the rekeying polynomial $p_{CH_a}(y)$ of cluster head a, the adversary needs to break the original symmetric polynomial $f(x,y)$ with extremely low probability.

Assume that the degree of polynomial function is $t = 80$, the NCA-robustness comparison of these two protocols are illustrated in Figure 5. As we observe that after a number of sensor nodes are compromised, Chadha's schemes will disclose the polynomials that can generate any group key in the past or future. On the contrary, our proposed scheme can achieve both forward and backward secrecy because such polynomials are extremely hard to be broken in our approach.

5. Performance Analysis

In this section, we evaluate the performance of our proposal by comparing with Chadha's scheme (Chadha et al., 2005). The performance metrics include the computational complexity, communication overhead, and storage overhead. Table 2 summarizes the performance results. In the Chadha's scheme, each cluster head first constructs $w(x) = g(x)f(x) + h(x)$ and calculates $n_a - n_c$ pair-wise keys for all innocent nodes, in which n_a and n_c are number of all sensor nodes and compromised sensor nodes, respectively, in a cluster. It needs $O(n_c^2 + n_c t + (n_a - n_c)t) = O(n_c^2 + n_a t)$ multiplications. Upon receiving $w(x)$, each sensor node needs to derive its personal key using $O(t)$ multiplications. In our proposed pair-wise

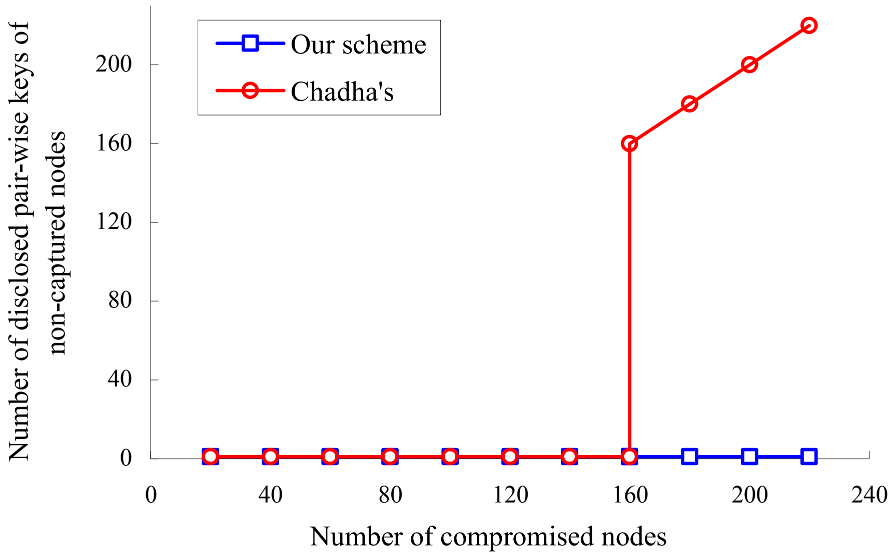


Fig. 5. NCA robustness comparison ($t = 80$)

		Chadha's	Our Scheme
Computation	Cluster head	$O(n_c^2 + n_a t)$ mul.	$O((n_a - n_c)t)$ mul. $n_a - n_c$ hash fun.
	Sensor node	$O(t)$ mul.	$O(t)$ mul. 3 hash fun.
Communication	Cluster head	$(2t + n_c + 1) \cdot \ell$	$(t + 1) \cdot \ell$
	Sensor node	0	0
Storage	Cluster head	$(2t + 1) \cdot \ell$	$(t + 1) \cdot \ell$
	Sensor node	ℓ	$(t + 1) \cdot \ell$

Table 2. Performance analysis

rekeying scheme, each cluster head needs to recalculate $n_a - n_c$ pair-wise keys using the rekeying polynomial with $O((n_a - n_c)t)$ multiplications. Each key generation involves a hash function operation as well. For each sensor node, it needs to calculate three candidate keys, which takes $O(t)$ multiplications and 3 hash function operations.

In the Chadha's scheme, each cluster head broadcasts a new $2t$-degree polynomial $w(x)$ and n_c Ids of detected compromised nodes to all the sensor nodes in the cluster. Such broadcast message has $(2t + n_c + 1) \cdot \ell$ bits. No message transmission at sensoe node side. The only communication overhead in our proposed scheme is the broadcast message for sending the t-degree master polynomial with $(t + 1) \cdot \ell$ bits. Note that, the overhead of the piggybacked short message for key agreement are considered as normal traffic and not included in Table 2.

In the evaluation of storage overhead, we consider the space requirement of the preloaded information in each sensor node and cluster head for the rekeying schemes. In Chadha's scheme, each cluster head is pro-loaded a $2t$-degree masking polynomial function $h(x)$. All coefficients for the polynomial require $(2t + 1) \cdot \ell$ bits. Each sensor node S_i needs to store one secret values $h(S_i)$ with ℓ bits. In our scheme, each sensor device (both cluster head and sensor node) is preloaded one t-degree perturbed polynomial taking $(t + 1) \cdot \ell$ bits.

6. Conclusion

The traditional polynomial based pair-wise rekeying protocol suffers the large-scale node capture attack. Once $t + 1$ nodes are compromised, all previous and future keys for any pair of nodes will be disclosed. We present a compromise-resilient pair-wise rekeying scheme based on a three-tier WSN. It can significantly improve the security level by reducing this probability from 1 down to $m^{-(t+1)}$ ($m \geq 2$). Our proposed scheme also achieves both forward and backward secrecy.

7. References

Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor Networks: A Survey, *Journal of Computer Networks*, Vol. 38, No. 4, 393–422.

Blundo, C.; De Santis, A.; Herzberg, A.; Kutten, S.; Vaccaro, U. & Yung, M. (1993). Perfectly-secure key sistribution for dynamic conferences, *LNCS*, Vol. 740, 471–486.

Chadha, A.; Liu, Y. & Das, S. (2005). Group key distribution via local collaboration in wireless sensor, *IEEE SECON*, pp. 46–54, July 2005.

Cheng, Y. & Agrawal, D. P. (2005). Efficient pairwise key establishment and management in static wireless sensor networks, *IEEE MASS*, November 2005.

Cheng, Y. & Agrawal, D. P. (2007). A improved key distribution mechanism for large-scale hierarchical wireless sensor networks, *Journal of Ad Hoc Networks*, Vol. 5, No. 1, 35–48.

Diffie, W. & Hellman, M. E. (1976). New direction in cryptography, *IEEE Transactions on Information Theory*, Vol. 22, No. 6, 644–654.

Du, W. L.; Deng, J.; Han, Y.& Varshney, P. K. (2003). A pairwise key pre-distribution scheme for wireless sensor network, *ACM Conference on Computer and Communications Security*, pp. 42–51, October 2003.

Eschenauer, L. & Gligor, V. (2002). A key-management scheme for distributed sensor networks, *ACM CCS*, pp. 41–47, November 2002.

Heinzelman, W. R.; Chandrakasan, A. P. & Balakrishnan, H. (2002). An application specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, 660–670.

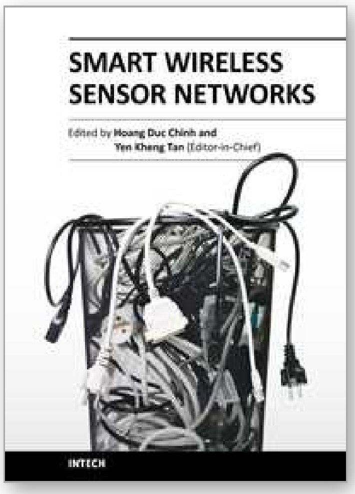
Mishra, S. (2002). Key management in large group multicast, *Technical Report CU-CS-970-02*, University of Colorado.

Rivest, R.; Shamir, A. & Adleman, L. (1978). A method for obtaining digital signatures and public key cryptosystems, *Communications of ACM*, Vol. 21, No. 2, 120–126.

Zhang, W.; Song, H.; Zhu, S. & Cao, G. (2005). Least privilege and privilege deprivation: Towards to tolerating mobile sink compromises in wireless sensor networks, *ACM MobiHoc*, pp. 378–389, May 2005.

Zhang, W.; Tran, M.; Zhu, S. & Cao, G. (2005). A random perturbation-based scheme for pairwise key establishment in sensor networks, *ACM MobiHoc*, pp. 90–99, September 2007.

Zhang, W.; Subramanian, N.; Zhu, S. & Wang, G. (2005). Lightweight and compromise-resilient message authentication in sensor networks, *IEEE INFOCOM*, pp. 1418–1426, April 2008.


SMART WIRELESS
SENSOR NETWORKS
Edited by Hoang Duc Chinh and
Yen Kheng Tan (Editor-in-Chief)
INTECH

Smart Wireless Sensor Networks
Edited by Yen Kheng Tan

ISBN 978-953-307-261-6
Hard cover, 418 pages
Publisher InTech
Published online 14, December, 2010
Published in print edition December, 2010

The recent development of communication and sensor technology results in the growth of a new attractive and challenging area â€“ wireless sensor networks (WSNs). A wireless sensor network which consists of a large number of sensor nodes is deployed in environmental fields to serve various applications. Facilitated with the ability of wireless communication and intelligent computation, these nodes become smart sensors which do not only perceive ambient physical parameters but also be able to process information, cooperate with each other and self-organize into the network. These new features assist the sensor nodes as well as the network to operate more efficiently in terms of both data acquisition and energy consumption. Special purposes of the applications require design and operation of WSNs different from conventional networks such as the internet. The network design must take into account of the objectives of specific applications. The nature of deployed environment must be considered. The limited of sensor nodesâ€™ resources such as memory, computational ability, communication bandwidth and energy source are the challenges in network design. A smart wireless sensor network must be able to deal with these constraints as well as to guarantee the connectivity, coverage, reliability and security of networkâ€™s operation for a maximized lifetime. This book discusses various aspects of designing such smart wireless sensor networks. Main topics includes: design methodologies, network protocols and algorithms, quality of service management, coverage optimization, time synchronization and security techniques for sensor networks.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Song Guo (2010). A Compromise-Resilient Pair-wise Rekeying Protocol in Large-Scale Wireless Sensor Networks, Smart Wireless Sensor Networks, Yen Kheng Tan (Ed.), ISBN: 978-953-307-261-6, InTech, Available from: http://www.intechopen.com/books/smart-wireless-sensor-networks/a-compromise-resilient-pair-wise-rekeying-protocol-in-large-scale-wireless-sensor-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820

www.intechopen.com

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821