

# A new approach for random key pre-distribution in large-scale wireless sensor networks

Kui Ren, Kai Zeng and Wenjing Lou<sup>\*,†</sup>

*Department of ECE, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609, U.S.A.*

## Summary

In a wireless sensor network (WSN), pre-distribution of secret keys is possibly the most practical approach to protect network communications. To meet the stringent resource constraints of the sensor nodes, key pre-distribution schemes should be highly efficient, require as little storage space as possible, and at the same time, maintain a strong security strength, that is, high resilience against node capture. In this paper, a new approach for random key pre-distribution is proposed to achieve both efficiency and security goals. The novelty of this approach lies in that, instead of using a key pool consisting of random keys, a key generation technique is carefully designed such that a large number of random keys can be represented by a small number of key-generation keys. Then, instead of storing a big number of random keys, each sensor node stores a small number of key-generation keys while computing the shared secret keys during the bootstrapping phase on the fly using the computationally efficient hash function. The proposed scheme outperforms the previous random key pre-distribution schemes in that it reduces the storage requirement significantly while holding the comparable security strength, as shown by our thorough analysis and simulation. Copyright © 2006 John Wiley & Sons, Ltd.

---

**KEY WORDS:** sensor network; security; key management; efficiency

---

## 1. Introduction

A wireless sensor network (WSN) consists of a large number of tiny sensor nodes with limited computation capacity, storage space, and power resource. Typically, WSNs are deployed at high density in regions requiring surveillance and monitoring. In military applications, sensor nodes may be deployed in unattended or hostile environments such as battlefields. WSNs are, therefore, vulnerable to various kinds of malicious attacks like eavesdropping, masquerading, traffic-analysis, etc. Hence, it is important to protect communications

among sensor nodes to maintain message confidentiality and integrity. Recent research suggests that symmetric secret key pre-distribution is possibly the only practical approach for establishing secure channels among sensor nodes since the low-power sensor nodes have very limited computational capacity which excludes the applicability of computation-intensive public key cryptographic algorithms.

Recently, many random key pre-distribution schemes have been proposed [2–8, 10]. Random key pre-distribution was first proposed by Eschenauer *et al.* [2]. The basic idea behind this scheme is to have a large

\*Correspondence to: Wenjing Lou, Department of ECE, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609, U.S.A.

†E-mail: wjlou@wpi.edu

pool of keys, from which a set of keys is randomly chosen and stored in each sensor node. Any two nodes which are able to find common keys within their key subsets can use those shared keys for secure communication. Chan *et al.* [3] extended the above scheme to enhance the security and resilience of the network using  $q$ -compositeness. In the  $q$ -composite scheme, at least  $q$  common keys are required to establish the secure channel between two nodes instead of using only one key. This method achieves higher security strength when a network is prone to small-scale attacks (less than 100 captured nodes) but not large-scale attacks. However, a higher value of  $q$  makes the network less scalable—it requires a larger number of keys stored at each node in order to maintain the necessary probability of finding  $q$  keys. Du *et al.* [6] and Liu *et al.* [5] further extended random key pre-distribution approach to pairwise key pre-distribution in which the shared key between any two sensor nodes is uniquely computed so that the resilience against node capture is significantly improved. All above mentioned schemes assume no network pre-deployment knowledge. In case that certain pre-deployment knowledge is available, the performance of the key pre-distribution can be improved by exploiting such knowledge [7,4].

In this paper, we focus on the random key pre-distribution scheme without network pre-deployment knowledge. The drawback of the above mentioned random key pre-distribution schemes [2,3] is that they are not suitable for large-scale sensor networks as they require each node to load a large number of keys. For instance, implementation of random key distribution schemes in [2,3] results in a storage overhead of at least 200 keys at each sensor node for a WSN of size 10 000, which is almost half of the available memory (assume 64-bit keys and less than 4 KB of data memory [1]). The problem becomes even worse when the network size is larger. This fact makes the previously proposed random key distribution schemes less practical for large-scale WSNs.

We propose a highly efficient random key pre-distribution scheme in this paper, which combines the random key pre-distribution technique and the hash chain technique. The novelty of our scheme lies in that, instead of requiring each sensor node to store all the chosen keys, the majority of the keys a node possesses are represented and stored in the form of a small number of key-generation keys by carefully designing the key pool, and therefore, the storage overhead is significantly reduced while the same security strength holds. Compared with the existing schemes, the proposed scheme is more scalable and more secure in the

sense that (1) Under the given resilience requirement against node capture, the proposed scheme requires a much smaller key ring size than the previous schemes; (2) Under the given maximum allowed key ring size, the proposed scheme has a much better resilience property against node capture than the previous schemes. The performance of the proposed scheme is justified by our thorough analysis and simulation.

The rest of the paper is organized as follows. We describe the background and related work in Section 2. Then we define the terms and notation and describe our new scheme in Section 3. Next we discuss the performance and security strength of the proposed scheme in Sections 4 and 5. Finally, the conclusion is drawn in Section 6.

## 2. Background on Key Management in WSN

In a WSN without pre-deployment knowledge, sensor nodes can be viewed as random points which are uniformly distributed (i.e., with equal probability). Thus, the sufficiency problem of the secure links resided in a WSN can be reduced to the connectivity problem of the generalized random graph, which, hence, can be mathematically treated using the well-known connectivity theory for random graph by Erdős and Rényi [11]. The connectivity of a *key graph*  $G(V, E)$  is then given as: for monotone properties, there exists a value of  $p$  such that the property moves from ‘non-existent’ to ‘certainly true’ in a very large random graph. The function defining  $p$  is called the threshold function of a property. If  $p = \frac{\ln(n)}{n} + \frac{c}{n}$ , with  $c$  being any real constant then

$$P_c = \lim_{n \rightarrow \infty} Pr([G(n, p) \text{ connected}]) = e^{-e^{-c}} \quad (1)$$

where  $P_c$  denotes the desired probability that the key graph is connected. In addition,  $n$  denotes the network size and  $d$  denotes the node degree (i.e., the average number of edges connected to each node) necessary to assure that the key graph is connected with probability  $P_c$ ;  $p$  is the probability that an edge between any two nodes exists on  $G(V, E)$ :

$$p = \frac{d}{n} \quad (2)$$

Due to the inherent communication constraints in WSNs, a sensor node can only communicate directly with its  $n'$  neighboring nodes. Since the expected node

degree must be at least  $d$  as calculated, the required probability of successfully performing key-setup with some neighboring node is now:

$$p_{\text{required}} = \frac{d}{n' - 1} \quad (3)$$

This implies that any two nodes in the WSN should share at least one secret key with probability no less than  $p_{\text{required}}$ . Further, the probability of two nodes  $i$  and  $j$  sharing at least one secret key can be computed as follows:

$$p = P(\mathcal{R}_i \cap \mathcal{R}_j \neq \emptyset) = 1 - P(|\mathcal{R}_i \cap \mathcal{R}_j| = 0) \quad (4)$$

For the key pre-distribution scheme in [2],  $p$  is computed as

$$p = 1 - \frac{\binom{K-R}{R}}{\binom{K}{R}} \quad (5)$$

where  $K$  is the size of the *key pool*, and  $R$  is the size of the *key ring*. In  $q$ -composite scheme proposed in [3], the above calculation is now

$$p = P(|\mathcal{R}_i \cap \mathcal{R}_j| \geq q) = 1 - \sum_{s=0}^{q-1} P(|\mathcal{R}_i \cap \mathcal{R}_j| = s) \quad (6)$$

Note that in [2,3]

$$P(|\mathcal{R}_i \cap \mathcal{R}_j| = s) = \frac{\binom{K}{s} \binom{K-s}{2(R-s)} \binom{2(R-s)}{m-s}}{\binom{K}{R}^2} \quad (7)$$

Therefore, key pool size  $K$  and key ring size  $R$  can be calculated by relating Equation (3) with Equations (5) or (6).

### 3. The Proposed Random Key Pre-Distribution Scheme

#### 3.1. Terms and Notation

In this paper, we use the following notation and terms for the convenience of description.

- *Key Pool*: A *key pool*  $\mathcal{K}$  with  $|\mathcal{K}| = K$  is a pool of random symmetric keys, from which each sensor node is independently assigned a subset, namely, a *key ring* in the key pre-distribution scheme for a WSN. The cardinality of  $\mathcal{K}$  equals to  $K$ .

- *Key Chain*: A *key chain*  $\mathcal{C}$  with  $|\mathcal{C}| = C$  is a subset of  $\mathcal{K}$ , and  $L$  equal-sized *key chains* in total form a complete *key pool*. Therefore, we have  $C = K/L$ . Each *key chain* is independently generated via a unique generation key, namely,  $g_i$  and a publicly known seed, namely, *seed*, by applying a keyed hash algorithm repeatedly. The value of the publicly known seed is the same for every key chain. Each key chain is uniquely indexed by its ID, namely,  $\mathcal{C}_i$  and  $\mathcal{C}_i \in [0, L - 1]$ .
- *Key Ring*: A *key ring*  $\mathcal{R}_i$  with  $|\mathcal{R}_i| = R$  is a subset of *Key Pool* with the cardinality of  $R$  ( $R \leq K$ ), which is independently assigned to a sensor node  $i$  following the assignment rules defined by the key pre-distribution scheme. Note that  $R$  is the same for every sensor node.
- *Key Graph*: Let  $V$  represent all sensor nodes in a WSN. A *key graph*  $\mathcal{G}(V, E)$  is constructed in the following manner: for any two nodes  $i$  and  $j$  in  $V$ , there exists an edge  $e_{ij} \in E$  between them if and only if  $\mathcal{R}_i \cap \mathcal{R}_j \neq \emptyset$ . Note that  $|V| = n$  for a WSN of size  $n$ . We say that a *key graph*  $\mathcal{G}(V, E)$  is *connected* if and only of any two nodes  $i$  and  $j$  belonging to  $V$  can reach each other via edge set  $E$  only.
- In a WSN of size  $n$ , each network node is uniquely identified through its ID, which ranges from 0 to  $n - 1$ . The length of a node ID is therefore up to  $\log_2 n$  bits.

In this paper, we say that a *key graph*  $\mathcal{G}(V, E)$  is *connected* if and only if any two nodes  $i$  and  $j$  belonging to  $V$  can reach each other via edge set  $E$  only. In  $q$ -composite scheme [3], a *key graph*  $\mathcal{G}(V, E)$  is *connected* if and only if any two nodes  $i$  and  $j$  belonging to  $V$  can reach each other through no less than two independent paths via edge set  $E$  only.

A cryptographically secure one-way hash function  $\mathcal{H}$  has the following property: for  $y = \mathcal{H}(x, k)$ , (1) given  $x$ , it is computationally infeasible to find  $y$  without knowing the value of  $k$ ; (2) given  $y$  and  $k$ , it is computationally infeasible to find  $x$ . A keyed hash algorithm like HMAC is provably secure and can be easily constructed on top of any secure one-way hash algorithms like SHA-1 [12]. However, a general purpose hash algorithm like SHA-1 is not suitable for sensor nodes, because (1) it is too complicated for an 8-bit micro-processor; (2) its message block length is at least 512-bit, which might be too large for sensor nodes and thus is not energy efficient. In [9], a class of universal hash functions WH is proposed for sensor nodes, whose message block is  $w$ -bit with a  $2^{-w}$  collision probability. This hash function is highly

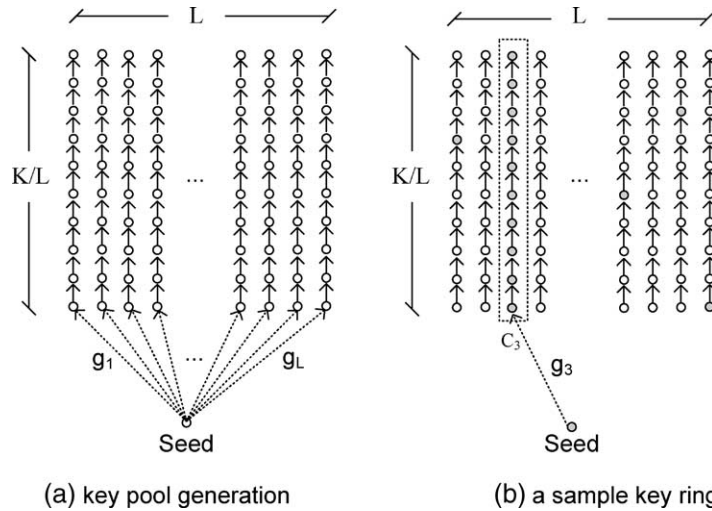


Fig. 1. A sample key pool and key ring; (a) key pool generation; (b) a sample key ring.

power efficient. The implementation of WH shows that it consumes only  $11.6 \mu W$  at 500 kHz. In the proposed scheme, we use WH in our key chain generation. The input and output length will be both 64-bit and no padding operation is needed at all. By applying the keyed hash function  $\mathcal{H}$  repeatedly on an initial value  $m$ , one can obtain a chain of outputs. Based on the properties described above, we know that these outputs are independent with each other and without knowing the secret key used by  $\mathcal{H}$ , one can not deduce any value on the chain even from other values of the same hash chain.

### 3.2. Random Key Pre-Distribution Scheme

The proposed key pre-distribution scheme consists of two phases: key assignment phase and shared-key discovery and path-key establishment phase. Although the way to find shared keys is different, the shared-key discovery and path-key establishment phase is more or less the same as in the previous schemes. In our scheme, the most significant difference lies in the key assignment phase. We propose two different schemes: the basic scheme and the  $q$ -composite scheme for key assignment phase. The details of the proposed schemes are described below.

#### 3.2.1. Key assignment phase

- *Key pool generation:* Key pool  $\mathcal{K}$  is determined by the following two parameters: key pool size  $K$  and the number of key chains  $L$ . Therefore, a key pool  $\mathcal{K}$  consists of  $L$  different key chains:  $\mathcal{K} = \bigcup_i \mathcal{C}_i$  ( $i = 0, \dots, L - 1$ ) and  $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$  ( $i \neq j$ ). Each

key chain  $\mathcal{C}_i$  is generated via a unique generation key  $g_i$  and the publicly known seed  $seed$  by applying a keyed hash algorithm repeatedly. Thereby, the  $l$ -th key of key chain  $\mathcal{C}_i$  is conceptually computed as

$$k_{c_i,l} = \mathcal{H}^l(seed, g_i) \quad (8)$$

where  $\mathcal{H}^l(seed, g_i) = \mathcal{H}(\mathcal{H}^{l-1}(seed, g_i), g_i)$  and so on. Note that  $g_i$  is only known to its assigned sensor nodes and should be strictly kept secret from other nodes in the WSN. At the same time, we use the pair  $(\mathcal{C}_i, l)$  to index the corresponding key. Hence,

$$\mathcal{C}_i = \bigcup_{l=1}^{K/L} k_{c_i,l} \quad (9)$$

A graphical illustration of the concepts of key pool and key chains is shown in Fig. 1(a).

- *Key ring loading:* In this step, each node is loaded with its assigned key ring  $\mathcal{R}$ , which contains two parts,  $\mathcal{R}_1$  and  $\mathcal{R}_2$ , where  $\mathcal{R}_1$  is the generation knowledge of a number of key chains and  $\mathcal{R}_2$  is a set of individual random keys from different key chains. To be more specific, for node  $i$ ,  $\mathcal{R}_i = \mathcal{R}_{i,1} \cup \mathcal{R}_{i,2}$ . The assignment rules are as follows. First, node  $i$  is assigned with  $r_0$  randomly selected key chains. However, instead of storing all the  $K/L$  keys in each key chain, node  $i$  only stores the corresponding key chain generation keys (one key per key chain). Therefore, it stores  $r_0$  keys for this part, that is,  $|\mathcal{R}_{i,1}| = r_0$ . From these  $r_0$  key-generation keys,  $r_0 \times (K/L)$  random keys can be calculated effectively. Second, node  $i$  is additionally assigned with  $r_1$  randomly selected

keys each from a different key chain. Hence, we have  $|\mathcal{R}_{i,2}| = r_1$ . An example is shown in Figure 1(b), where the key chain and keys in green (filled with color) can be a sample key ring, where  $r_0 = 1$ . For the proposed  $q$ -composite scheme, the assigning rules are the same but with larger  $r_0, r_1$  values in general.

### 3.2.2. Shared-key discovery and path-key establishment phase

During the network bootstrapping phase, each sensor node is required to broadcast the key index information of its key ring, that is,  $\mathcal{R}_i$ , to expose its key information to the neighbor nodes. Hence, each node will know which keys its neighbors have. Each node then examines the key index information of its own key ring to find or calculate the keys it shares with the neighbor nodes. For node  $i$  to find the shared key(s) with node  $j$ , it matches the key indexes of  $\mathcal{R}_i$  and  $\mathcal{R}_{j,2}$ . If  $\mathcal{R}_{i,2} \cap \mathcal{R}_{j,2} \neq \emptyset$ , those are the keys node  $i$  shared with node  $j$ . If  $\mathcal{R}_{i,1} \cap \mathcal{R}_{j,2} \neq \emptyset$ , node  $i$  needs to calculate the key(s) in common. For example, if node  $x$  contains a key indexed as  $k_{c_i,l}$  and node  $y$  contains key chain  $\mathcal{C}_i$ , node  $y$  immediately knows that it shares key  $k_{c_i,l}$  with node  $x$  upon receiving node  $x$ 's broadcast message. Node  $y$  then simply calculates  $k_{c_i,l}$  following Equation (8). If node  $y$  also contains key  $k_{c_i,l}$ , then there is no need for calculation. If there are more than one shared key, the final pairwise key is simply computed as the hash value of the shared keys. The concatenation sequence of the shared keys can be easily enforced to ensure the same output hash value. For example, if  $ID_x < ID_y$ , then the keys sent by node  $x$  becomes the first in the concatenation. In case that two neighbor nodes share no common key directly, we use the same path-key establishment technique as described in [2] to establish a pairwise key between them. Note that in our setting, no shared key is established when two nodes only share one or more key chains, that is, we do not count in the situations that for any two nodes  $i$  and  $j$ ,  $\mathcal{R}_{i,2} \cap \mathcal{R}_{j,2} = \emptyset$  and  $\mathcal{R}_{i,1} \cap \mathcal{R}_{j,2} = \emptyset$  and  $\mathcal{R}_{i,2} \cap \mathcal{R}_{j,1} = \emptyset$  and  $\mathcal{R}_{i,1} \cap \mathcal{R}_{j,1} \neq \emptyset$ . We treat this case the same as that the two nodes do not share any key and use the path-key establishment technique to establish a shared key between them. At this point, each node now shares at least a key with each of its neighbor nodes, respectively. We use the same method as in [3] to generate the link key  $k_{\text{link}} = \text{hash}(k_1|k_2|\dots|k_i)$  to secure the communication between two sensor nodes, where  $i$  ( $q \leq i \leq r_0 + r_1$ ) is the number of keys it actually shares with a particular neighbor node.

## 4. Performance Analysis and Simulation

In this section, we evaluate the proposed two schemes in terms of required storage space (i.e., key ring size) at the sensor node, given the required key sharing probability  $p_{\text{required}}$ . For a WSN of network size  $n$  and neighborhood size  $n'$ ,  $p_{\text{required}}$  can be calculated using Equation (3). Then the key pool size  $K$  and key ring size  $R$  can be properly chosen according to Equation (5) [2] and (6) [3], respectively. We first develop the equations to calculate the probability that two nodes sharing at least one or  $q$  keys for the proposed two schemes. We next compare the performance of the proposed schemes with that of [2] and [3], respectively. From the description of the scheme, we know that key ring  $\mathcal{R}$  contains two parts:  $\mathcal{R}_1$  and  $\mathcal{R}_2$  in addition to a public seed. Hence,  $R$  is calculated as follows:

$$R = |\mathcal{R}_1| + |\mathcal{R}_2| + 1 = r_0 + r_1 + 1 \quad (10)$$

### 4.1. Connectivity Calculation

We consider the probabilities that any two nodes, say  $n_i$  and  $n_j$ , share at least one key (for the basic scheme) and at least  $q$  keys (for the  $q$ -composite scheme).

For any node, say  $n_i$ , the number of possible key ring assignments can be calculated as follows:

$$(I) = \binom{L}{r_0} \binom{L-r_0}{r_1} \left(\frac{K}{L}\right)^{r_1}$$

For the other node, say  $n_j$ , the number of possible key ring assignments that do not share any key with node  $n_i$  can be calculated as follows. Note that the two nodes may share common key chains.

$$(II) = \sum_{s=0}^{r_0} \binom{L-r_0-r_1}{r_0-s} \binom{r_0}{s} \\ \times \sum_{i=0}^{r_1} \binom{L-2r_0-r_1+s}{r_1-i} \\ \times \binom{r_1}{i} \left(\frac{K}{L}\right)^{r_1-i} \left(\frac{K}{L}-1\right)^i$$

Similarly, the number of possible key ring assignments at the other node  $n_j$  that share exactly  $x$  ( $1 \leq x \leq r_0 + r_1$ ) keys with node  $n_i$  (excluding key chain to key chain overlapping) can be computed as

follows:

$$\begin{aligned}
 (III) &= \sum_{t=0}^{r_0} \sum_{s=0}^{r_0-t} \binom{L-r_0-r_1}{r_0-s-t} \binom{r_0}{s} \binom{r_1}{t} \sum_{i=0}^{r_0-s} \\
 &\quad \times \sum_{j=0}^{\min(r_1-t, r_0-s)} \binom{L-2r_0-r_1+s+t}{r_1-i} \\
 &\quad \times \binom{r_1-t}{j} \binom{r_0-s}{i} \sum_{m=0}^j \\
 &\quad \times \binom{j}{m} \binom{K}{L}^{r_1-j} \binom{K}{L-1}^{j-m}
 \end{aligned}$$

where  $t + i + m = x$  and  $t + i + m \leq r_0 + r_1 - t$ .  
 Therefore, the probability that any two nodes share no key is

$$Pr\{\mathcal{R}_i \cap \mathcal{R}_j = 0\} = \frac{(II)}{(I)}$$

and the probability that any two nodes share exactly  $x$  keys is

$$Pr\{|\mathcal{R}_i \cap \mathcal{R}_j| = x\} = \frac{(III)}{(I)}$$

Hence, for the basic scheme, we have

$$p_{\text{required}} = 1 - \frac{(II)}{(I)} \tag{11}$$

For the proposed  $q$ -composite scheme ( $q = 2$ ), we have

$$\begin{aligned}
 p_{\text{required}} &= 1 - Pr\{|\mathcal{R}_i \cap \mathcal{R}_j| \\
 &= 0\} - Pr\{|\mathcal{R}_i \cap \mathcal{R}_j| = 1\} \tag{12}
 \end{aligned}$$

$$= 1 - \frac{(II)}{(I)} - \frac{(III)(x=1)}{(I)} \tag{13}$$

### 4.2. Performance Evaluation

In order to thoroughly examine the performance of the proposed two schemes, we vary the values of  $r_0$  and  $r_1$  under different network size  $n$ , key pool size  $K$ , and the number of key chains  $L$  to see how the connectivity varies accordingly. The key ring size  $R$  is calculated as  $r_0 + r_1 + 1$ . Also note that in the proposed schemes, the value of  $L$  is a function of that of network size  $n$ . The value of  $L$  determines the security strength against node capture as will be discussed in detail in the next section. The network size is first set as  $n = 10000$ . The key pool size  $K$  is set to 5, 10, and 50 times of the network size. The number of key chains

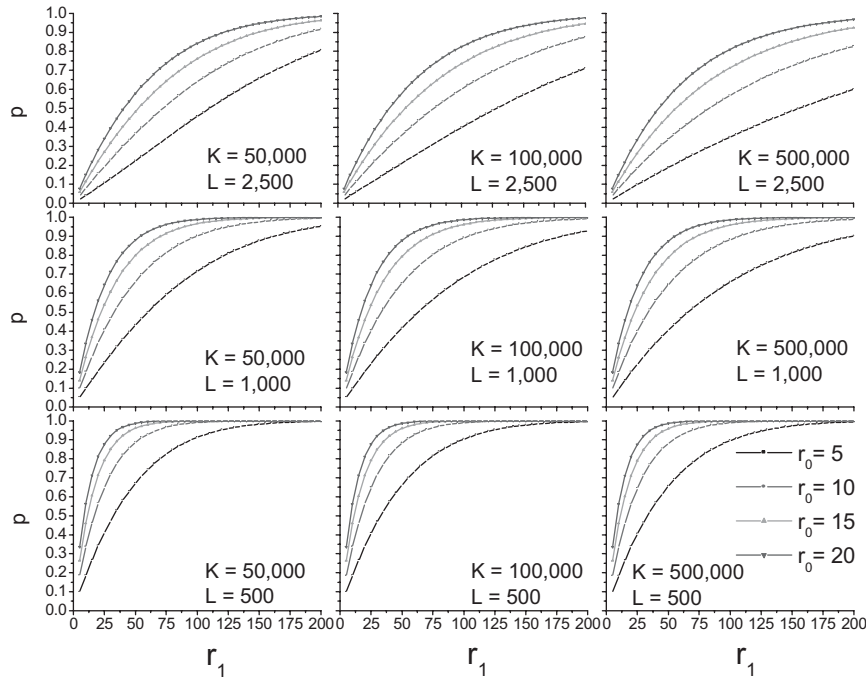


Fig. 2. The proposed basic scheme:  $p$  versus  $r_0$  and  $r_1$  under different values of  $K$  and  $L$ , when network size  $n$  is 10000.

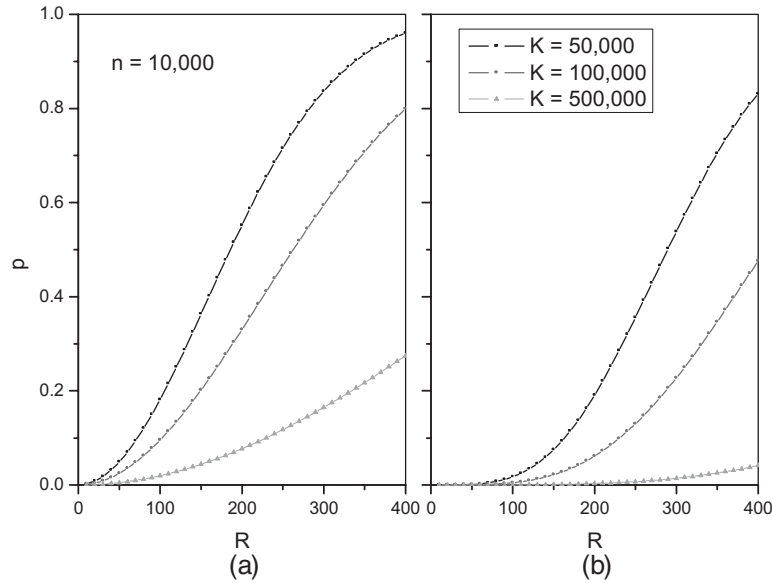


Fig. 3. (a) Performance of Gligor's Scheme and (b) performance of Chan's Scheme ( $q = 2$ ) when network size  $n = 10\,000$ .

$L$  is set to 0.05, 0.1 and 0.25 times of the network size. Figure 2 shows the performance of the proposed basic scheme at  $n = 10\,000$ . Figure 3(a) illustrates the performance of Eschenauer *et al.*'s scheme at the same network size. The proposed basic scheme offers a great performance improvement as compared to Eschenauer *et al.*'s scheme. For example, Where  $n = 10\,000$  and

$p_{\text{required}} = 0.5$ ,  $R$  is required to be around 260 given  $K = 100\,000$  in [2]; on the other hand, under the same settings  $R$  can be as low as 30 in the proposed scheme, although this choice is not good as it has a low security strength against node capture, as we will show in Section 5. However, when similar security strength is assumed, the required key ring size in the proposed

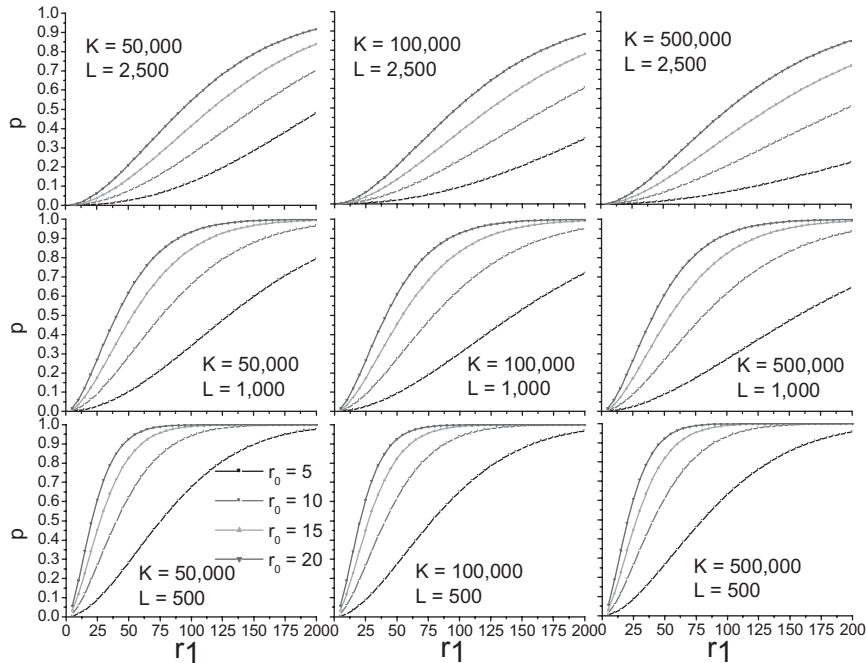


Fig. 4. The proposed  $q$ -composite scheme:  $p$  versus  $r_0$  and  $r_1$  under different values of  $K$  and  $L$ , when network size  $n = 10\,000$  and  $q = 2$ .

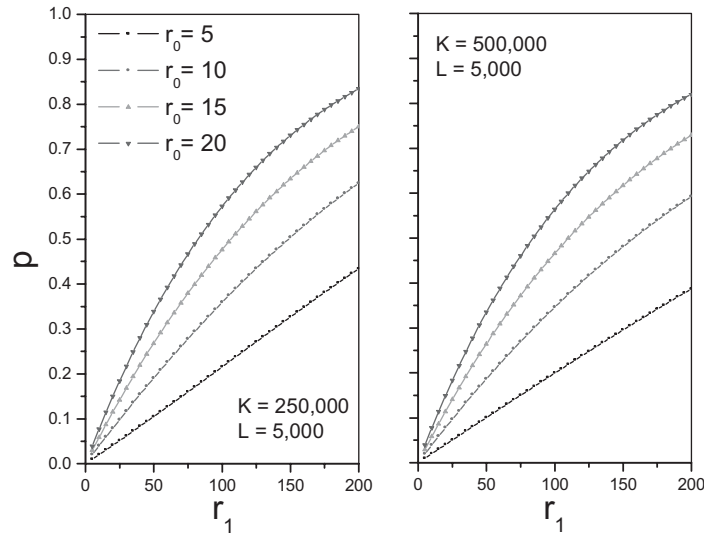


Fig. 5. The proposed basic scheme:  $p$  versus  $r_0$  and  $r_1$ , when network size  $n = 50\,000$ .

scheme is around 50% less than that of Eschenauer *et al.*'s scheme as will be shown in Section 5. The evaluation of the proposed  $q$ -composite scheme is shown in Figure 4 and as comparison, the performance of Chan *et al.*'s  $q$ -composite scheme under the same settings is illustrated in Figure 3(b). The performance improvement is again very significant. For instance, when  $n = 10\,000$  and  $p_{\text{required}} = 0.5$ ,  $R$  is required to be around 275 ( $q = 2$ ) given  $K = 50\,000$  in [3]; on the other hand, in the proposed scheme  $R$  can be as low as 50 ( $q = 2$ ) in the proposed scheme.

The improvement of the proposed two schemes goes higher as the network size  $n$  grows. For example, when  $n = 50\,000$  and  $p_{\text{required}} = 0.5$ , the proposed basic scheme requires as low as 100 keys with  $K = 250\,000$  as shown in Figure 5, while 410 keys are required in Eschenauer *et al.*'s scheme for comparable security strength. This fact shows that our scheme is highly scalable to the larger network sizes. At the same time, a requirement of  $R = 410$  implies that the scheme is no longer practical under the given network size due to the extremely limited storage space of the sensor nodes.

The above figures (Figures 2 and 4) also illustrate how the performance of the proposed two schemes vary under different system settings, that is, different values of  $K$ ,  $L$ , and  $(r_0, r_1)$  pairs. We find that under a given network size  $n$ , the performance of the proposed schemes decreases as either  $K$  or  $L$  increases. From Equation (15) developed below, we know that the values of  $K$  and  $L$  also determine how resilient the proposed schemes is against node capture. On one hand, we desire smaller values of  $K$  and  $L$  to achieve better key sharing probability with  $R$  fixed; on the other

hand, the proposed schemes present better resilience property against node capture when larger values of  $K$  and  $L$  are used. Therefore, this can be formulated as a constrained optimization problem:

Under the given system parameters of networks size  $n$  and neighborhood size  $n'$ , minimize  $R$ , where  $R = r_0 + r_1 + 1$  as defined in Equation (10) and the values of  $(r_0, r_1)$  are subject to Equation (11) or Equation (13).

## 5. Security Strength Analysis

To study the security strength of the proposed scheme, we first prove that without the knowledge of the corresponding key chain generation key  $g_i$ , whatever number of keys of a key chain that are compromised will not affect the security of the remaining keys in that key chain.

**Lemma.** For a given key chain  $C_i$  of size  $K/L$ , the knowledge of any combination of  $\frac{K}{L} - 1$  keys except for the key in question can not result any advantage on the knowledge of the remaining key without knowing the corresponding key chain generation key  $g_i$ .

**Proof.** In the proposed scheme, a key chain is generated using the keyed hash function following Equation (8). Hence, any key  $k_{c_i,l}$  inside a key chain holds the following relationship with other keys of the same key chain:

$$\dots, k_{c_i,l} = \mathcal{H}(k_{c_i,l-1}, g_i), k_{c_i,l+1} = \mathcal{H}(k_{c_i,l}, g_i), \dots$$

Therefore, it is computationally infeasible to compute  $k_{c_i,l}$  from either  $k_{c_i,l+1}$  or  $k_{c_i,l-1}$  without the secret key



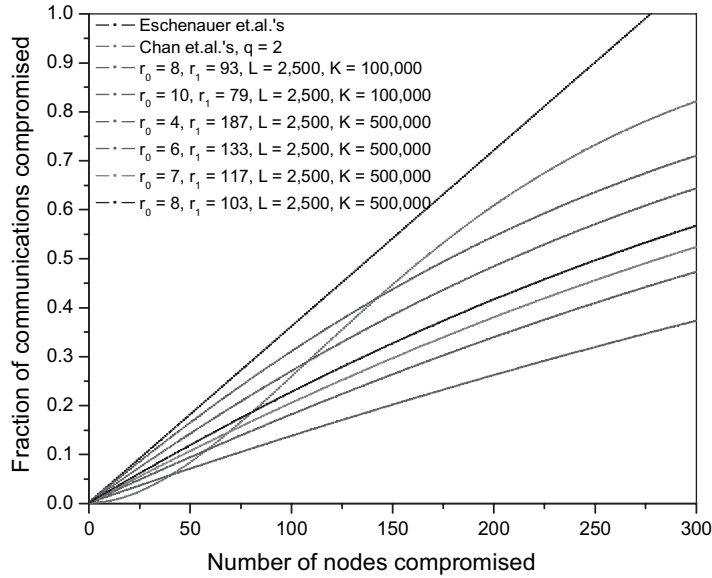


Fig. 6. Security strength of the proposed basic scheme with  $n = 10\,000$ ,  $p_{\text{required}} = 0.5$  and  $R_{\text{max}} = 192$ .

$g_i$  because the keyed hash function is used. On the other hand, it is also computationally infeasible to recover the key chain generation key  $g_i$  from any combination of its generated keys because of the same reason. ■

Next we study the resilience property of the proposed scheme against node capture by calculating the fraction of links in the network that are compromised due to key revealing resulted from node capture. In the proposed scheme, since each node actually has the knowledge of  $\frac{r_0 K}{L} + r_1$  keys, the probability that a given key does not belong to a node is  $1 - (\frac{r_0}{L} + \frac{r_1}{K})$ . Therefore, if there are

$m$  compromised nodes, the probability that a given key is not compromised should be  $(1 - (\frac{r_0}{L} + \frac{r_1}{K}))^m$ . The expected fraction of total keys compromised is thus  $1 - (1 - (\frac{r_0}{L} + \frac{r_1}{K}))^m$ . If the communication link between two nodes has its link key  $k_{\text{link}}$  computed from  $s$  ( $s \geq q$ ) shared keys, the probability of that link being compromised is then  $(1 - (1 - (\frac{r_0}{L} + \frac{r_1}{K}))^m)^s$  and hence, in the worst case the compromising probability is

$$\left(1 - \left(1 - \left(\frac{r_0}{L} + \frac{r_1}{K}\right)\right)^m\right)^q \quad (14)$$

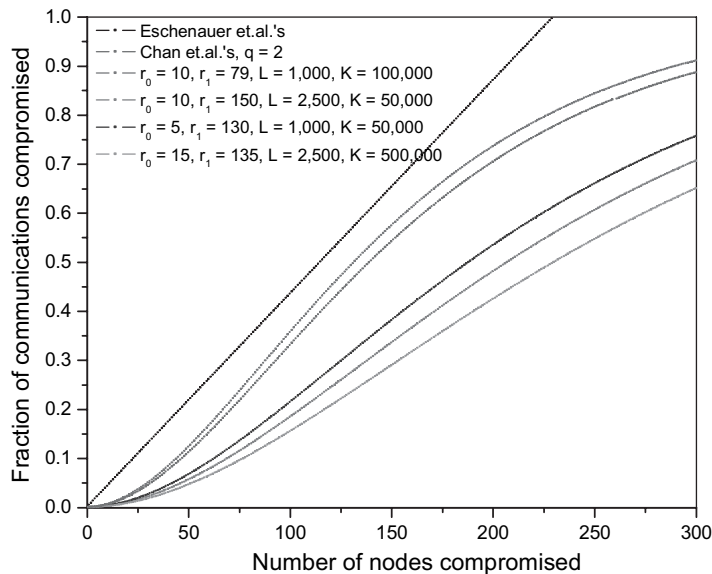


Fig. 7. Security strength of the proposed  $q$ -composite scheme with  $n = 10\,000$ ,  $p_{\text{required}} = 0.5$ ,  $q = 2$ , and  $R_{\text{max}} = 161$ .

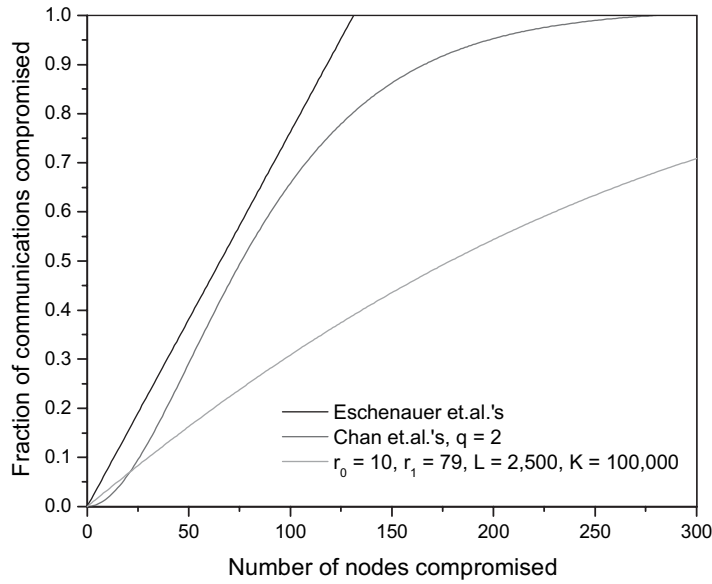


Fig. 8. Security strength of the proposed basic scheme with  $n = 10\,000$ ,  $p_{\text{required}} = 0.5$  and  $R = 90$ .

Therefore, averagely the compromising probability is

$$\sum_{s=q}^m \left(1 - \left(1 - \left(\frac{r_0}{L} + \frac{r_1}{K}\right)\right)^m\right)^s \frac{P(|\mathcal{R}_i \cap \mathcal{R}_j| = s)}{\sum_{t=q}^m P(|\mathcal{R}_i \cap \mathcal{R}_j| = t)} \quad (15)$$

Equation (15) also represents the fraction of additional communications that an adversary can com-

promise based on the key information retrieved from  $m$  captured nodes in the worst case. Figure 6 shows the security strength of the proposed basic scheme, where  $n = 10\,000$ ,  $p_{\text{required}} = 0.5$  and  $R_{\text{max}} = 192$ . Obviously, the proposed scheme offers a much better resilience property while requiring a much smaller key ring size when compared with Eschenauer and Gligor's. Figure 7 illustrates the security strength of the proposed  $q$ -composite scheme, where  $n = 10\,000$ ,  $p_{\text{required}} = 0.5$ ,  $q = 2$  and  $R_{\text{max}} = 161$ . Again the proposed  $q$ -composite scheme offers a much better

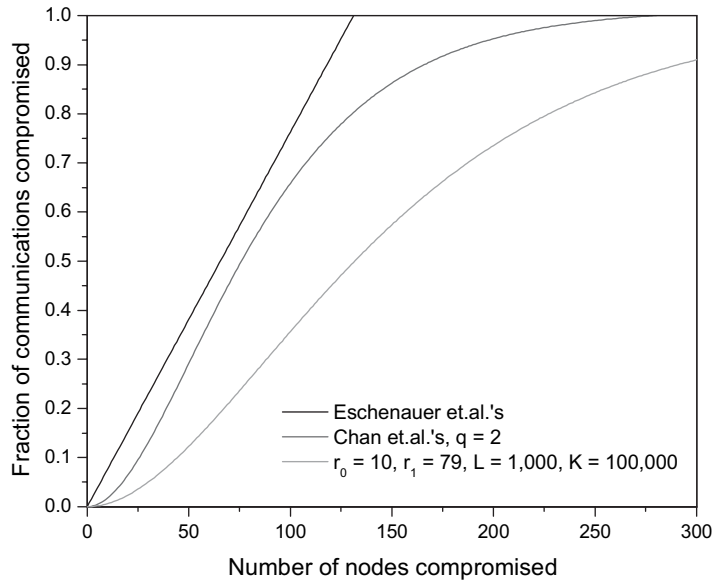


Fig. 9. Security strength of the proposed  $q$ -composite scheme with  $n = 10\,000$ ,  $p_{\text{required}} = 0.5$ ,  $q = 2$ , and  $R = 90$ .

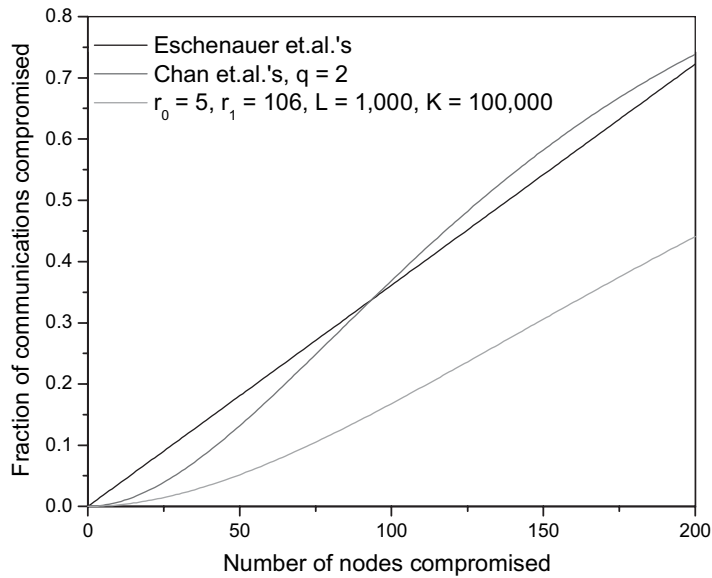


Fig. 10. Security strength of the proposed  $q$ -composite scheme with  $n = 10\,000$ ,  $p_{\text{required}} = 0.33$ ,  $q = 2$ , and  $R = 112$ .

resilience property as compared to that of Chan *et al.*'s. To exactly illustrate how much is the improvements gained by the proposed scheme, we now fix the key ring size  $R$  for each scheme and other system settings remain the same. Figure 8 shows the security strength of the proposed basic scheme, when  $n = 10\,000$ ,  $p_{\text{required}} = 0.5$  and key ring size  $R$  is fixed as 90. We can see that when the fraction of the compromised communication has reached to 100% in Eschenauer, the proposed basic scheme only has a value of 38% under the same settings. Figure 9 shows the significant resilience improvement of the proposed  $q$ -composite scheme when  $n = 10\,000$ ,  $p_{\text{required}} = 0.5$ ,  $q = 2$  and key ring size  $R$  is fixed as 90. To compromise 10% communications among the remaining network nodes, only 25 compromised nodes are required; however, 50 nodes are required in the proposed scheme. The improvement is around 100%. More importantly, the proposed  $q$ -composite scheme holds a much better security strength under both small-scale attack and large-scale attack, which overcomes the shortcomings presented in Chan *et al.*'s scheme, that is, achieving better security strength under small scale attack while trading off increased vulnerability in the face of a large scale attack on network nodes. This situation is illustrated in Figure 10.

## 6. Concluding Remarks

In this paper, we have proposed a new approach for random key pre-distribution in WSNs. The novelty of this

approach is that, instead of requiring the sensor nodes store all the assigned keys, the majority of the keys are represented and stored in terms of key generation key sets with a very small size by carefully designing the key pool, which significantly reduces storage space while holding the same security strength. The proposed scheme is hence, highly scalable to the larger network sizes. The proposed scheme outperforms the previous random key pre-distribution schemes under both small scale and large scale attacks, especially when the network size is large ( $\geq 10\,000$ ) as shown by our thorough analysis.

As the future work, we would like to extend the proposed scheme to the case of pair-wise key pre-distribution in order to further improve the security resilience against node capture. Further, we will take different types of active attacks into consideration besides random node capture attack and optimize the scheme accordingly.

## References

1. Perrig A, Szewczyk R, Wen V, Culler D, Tygar J. SPINS: Security protocols for sensor networks. In *Proceedings of Seventh Annual ACM International Conference on Mobile Computing and Networks (Mobicom 2001)*, Rome Italy, July 2001.
2. Eschenauer L, Gligor VD. A key-management scheme for distributed sensor networks. Conference on Computer and Communications Security. *Proceedings of the 9th ACM Conference on Computer and Communications Security 2002*, Washington, DC, USA.
3. Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, 2003.

4. Liu D, Ning P. Location-based pairwise key establishments for relatively static Sensor networks. *2003 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, George W. Johnson Center at George Mason University, Fairfax, VA, USA October 31, 2003.
5. Liu D, Ning P. Establishing pairwise keys in distributed sensor networks. *The 10th ACM Conference on Computer and Communications Security (CCS'03)*, Washington D.C., USA October, 2003
6. Du W, Deng J, Han YS, Varshney P. A pairwise key predistribution scheme for wireless sensor networks. *In Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS)*, Washington DC, USA October 27–31, 2003.
7. Du W, Deng J, Han YS, Chen S, Varshney P. A key management scheme for wireless sensor networks using deployment knowledge. *IEEE INFOCOM'04*, Hongkong, March 7–11, 2004.
8. Zhu S, Xu S, Setia S, Jajodia S. Establishing pair-wise keys for secure communication in ad hoc networks: A probabilistic approach. *In Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)*, Atlanta, Georgia, November 4–7, 2003.
9. Yuksel K, Kaps JP, Sunar B. Universal hash functions for emerging ultra-low-power networks. *The Communications Networks and Distributed Systems Modeling and Simulation Conference (CNDS) 2004*, San Diego, CA, January, 2004.
10. Huang D, Mehta M, Medhi D, Harn L. Location-aware key management scheme for wireless sensor networks. *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*, October 25, 2004.
11. Erdos, Renyi. On random graphs I *Publ Math Debrecen*, 6: 290–297, 1959.
12. Krawczyk H, Bellare M, Canetti R. HMAC: Keyed-hashing for message authentication. *Internet RFC 2104*, Feb. 1997.
13. Ren K, Zeng K, Lou W. On efficient key pre-distribution in large scale wireless sensor networks, *IEEE Milcom 2005*, Atlantic City, NJ, October 2005.

## Authors' Biographies



**Kui Ren** received his B.E. and M. E. both from Zhejiang University, China, in 1998 and 2001, respectively. He worked as a research assistant at Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences from March 2001 to January 2003, at Institute for Infocomm Research, Singapore from January 2003 to August

2003, and at Information and Communications University, South Korea from September 2003 to June 2004. Currently he is a PhD student in the ECE department at Worcester Polytechnic Institute. His research interests include ad hoc/sensor network security, wireless mesh network security, Internet security, and security and privacy in ubiquitous computing environments.



**Kai Zeng** received his BS degree in Communication Engineering and MS degree in Communication and Information System from Huazhong University of Science and Technology, China, respectively in June 2001 and June 2004. He is currently a Ph.D. student in the Electrical and Computer Engineering department at Worcester

Polytechnic Institute. His research interests are in wireless networking, especially on mobile wireless ad hoc networks and sensor networks with an emphasis on routing and network security.



**Wenjing Lou** is an assistant professor in the Electrical and Computer Engineering department at Worcester Polytechnic Institute. She obtained her Ph.D. degree in Electrical and Computer Engineering from University of Florida in 2003. She received the M.A.Sc degree from Nanyang Technological University, Singapore, in 1998, the M.E. degree and the B.E. degree in Computer Science and Engineering from Xi'an Jiaotong University, China, in 1996 and 1993, respectively. From December 1997 to July 1999, she worked as a Research Engineer in Network Technology Research Center, Nanyang Technological University. Her current research interests are in the areas of ad hoc and sensor networks, with emphases on network and system security and routing.