ON EFFICIENT KEY PRE-DISTRIBUTION IN LARGE SCALE WIRELESS SENSOR NETWORKS

Kui Ren, Kai Zeng and Wenjing Lou

Department of ECE, Worcester Polytechnic Institute, Worcester, MA 01609
{kren, kzeng, wjlou}@wpi.edu

ABSTRACT

In a wireless sensor network, 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, such as limited storage capability, low computation capability, and limited battery life, key pre-distribution schemes should be highly efficient, namely requiring as little storage space as possible, and at the same time, maintain a strong security strength, i.e., 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 random 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 computes the shared secret key during the bootstrapping phase on-the-fly using efficient hash operations. The proposed scheme outperforms the previous random key pre-distribution schemes by significantly reducing the storage requirement while holding the same security strength as shown by the detailed analysis.

I. 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.

In this paper, we focus on the random key pre-distribution scheme without network pre-deployment knowledge. The drawback of the previous random key pre-distribution schemes [2], [3] is that they are not suitable for large scale sensor networks as they require each node to be loaded with 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 4KB of data memory [1]). The problem becomes even worse when the network size is larger. This fact makes the previous proposed random key distribution schemes less practical for large-scale WSNs.

Identifying these limitations, 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 is that, instead of requiring the sensor nodes store all the chosen keys, the majority of the keys are represented and stored in term of key-generation key sets with a very small size by carefully designing the key pool, and therefore, significantly reduces storage overhead while holding the same security strength. The contribution of the proposed scheme is twofold: 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 works closely related to ours in Section II. Then we defines the terms and notation and describe our own scheme in Section III. Next we discuss the performance and security strength of the proposed scheme in Sections IV and V. Finally, the conclusion is drawn in Section VI.

II. BACKGROUND AND RELATED WORK

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 Erdós 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 "nonexistent" 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 any real constant then

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

where P_c denotes the desired possibility that the key graph is connected. In addition, n denotes 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 in $G(V, E)$:

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

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:

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

This implies that any two nodes in the WSN should share at least one secret key with probability no less than $p_{required}$. 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 Eq. (3) with Eq. (5) or (6).

III. THE PROPOSED SCHEME

A. 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 of 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 of 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 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.

The proposed key pre-distribution scheme takes (n, n', R_{max}) as its input parameters, where n is the network size, n' is neighborhood size and R_{max} is the upper bound of storage capability in terms of number of keys that can be stored by each sensor node. The proposed scheme then outputs an optimized 6-element tuple (K, L, r_0, r_1, q) as the scheme parameters, according to which the key pool is organized and each sensor node is assigned its own key ring. The computation of the values of these parameters will be discussed later.

B. Random Key Pre-distribution Scheme

The proposed key pre-distribution scheme consists of two phases: key assignment phase and shared-key discovery & 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.

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: $\mathcal{K} = \cup_i \mathcal{C}_i$ $(i = 0, .., L-1)$ and $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ $(i \neq j)$. Each key chain

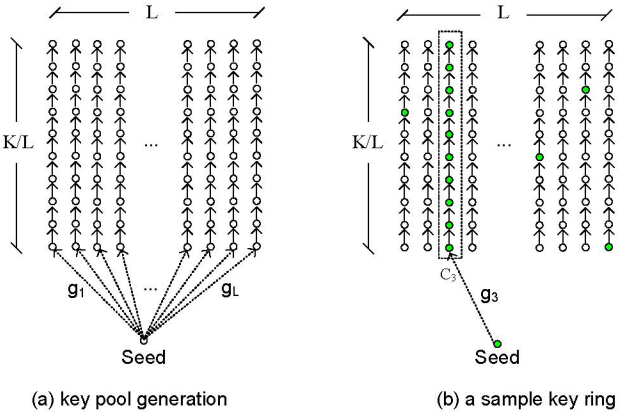

Fig. 1. A sample key pool and key ring

$\mathcal{C}_i$ 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

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

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,

$$\mathcal{C}_i = \cup_{l=1}^{K/L} k_{c_i,l} \tag{9}$$

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 random keys. To be more specific, for node i, $\mathcal{R}_i = \mathcal{R}_{i,1} \cup \mathcal{R}_{i,2}$. The assigning 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, i.e., $|\mathcal{R}_{i,1}| = r_0$. 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 Fig. 1(b), where the green key chain and keys 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.

Shared-key discovery & path-key establishment phase: During the network bootstrapping phase, each sensor node is required to broadcast the key index information of its key ring, i.e., $\mathcal{R}_i$, 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 Eq. (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, we use the same path-key establishment technique as described in [2] to establish a pairwise key between them. Note that in our setting, we do not count in the situations that 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, $\mathcal{R}_{i,2} \cup \mathcal{R}_{j,2} = \emptyset$ and $\mathcal{R}_{i,1} \cup \mathcal{R}_{j,2} = \emptyset$ and $\mathcal{R}_{i,2} \cup \mathcal{R}_{j,1} = \emptyset$ and $\mathcal{R}_{i,1} \cup \mathcal{R}_{j,1} \neq \emptyset$. 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 all its neighbor nodes, respectively. We use the same method as in [3] to generate the link key $k_{link} = hash(k_1|k_2|...|k_i)$ to secure the communication link 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. In the proposed scheme, shared key discovery involves keyed hash operations. We use universal hash function WH which is specifically designed for sensor nodes [9] in the proposed scheme instead of general purposed hash algorithms. As mentioned above, WH has a message block size of w-bit with a 2^{-w} collision probability. WH is highly power efficient. The implementation of WH shows that it consumes only $11.6\mu W$ at 500 kHz.

IV. PERFORMANCE ANALYSIS

We evaluate the proposed two schemes in terms of required storage space (i.e., key ring size) at the sensor node, when the required key sharing probability $p_{required}$ is given. Once network size n and neighborhood size n' of a WSN are fixed, $p_{required}$ is calculated using Eq. (3). Then the key pool size K and key ring size R can be properly chosen according to Eq. (5) [2] and Eq. (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:

$$R = |\mathcal{R}_1| + |\mathcal{R}_2| + 1 = r_0 + r_1 + 1 \tag{10}$$

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:

$$(I) = \binom{L}{r_0}\binom{L-r_0}{r_1}\binom{K/L}{1}^{r_1}$$

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.

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

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:

$$(III) = \sum_{t=0}^{r_o}\sum_{s=0}^{r_0-t}\binom{L-r_0-r_1}{r_0-s-t}\binom{r_0}{s}\binom{r_1}{t}\sum_{i=0}^{r_0-s}$$
$$\sum_{j=0}^{min(r_1-t,r_0-s)}\binom{L-2r_0-r_1+s+t}{r_1-i}\binom{r_1-t}{j}$$
$$\binom{r_0-s}{i}\sum_{m=0}^{j}\binom{j}{m}\binom{\frac{K}{L}}{1}^{r_1-j}\binom{\frac{K}{L}-1}{1}^{j-m},$$

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 $P\{|\mathcal{R}_i \cap \mathcal{R}_j| = 0\} = \frac{(II)}{(I)}$, and the probability that

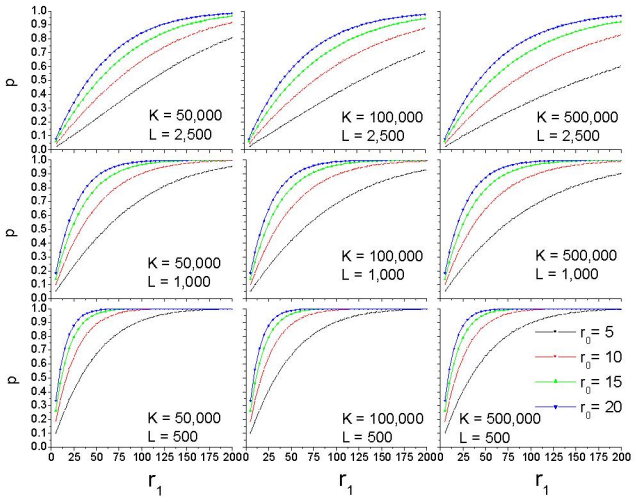

Fig. 2. The proposed basic scheme: p vs. r_0 and r_1 under different values of K and L, when network size n is $10,000$.

any two nodes share exactly x keys is $P\{|\mathcal{R}_i \cap \mathcal{R}_j| = x\} = \frac{(III)}{(I)}$. Hence, for the basic scheme, we have

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

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

$$p_{required} = 1 - \frac{(II)}{(I)} - \frac{(III)(x=1)}{(I)} \tag{12}$$

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, respectively. 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 $n = 10,000$. The key pool sizes K will be 5, 10, and 50 times of the corresponding network size. The number of key chains L is set to be 0.05, 0.1 and 0.25 times of the corresponding network size. Fig. 2 shows the performance of the proposed basic scheme at $n = 10,000$. Fig. 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, When $n = 10,000$ and $p_{required} = 0.5$, 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. However, when similar security strength is assumed, the

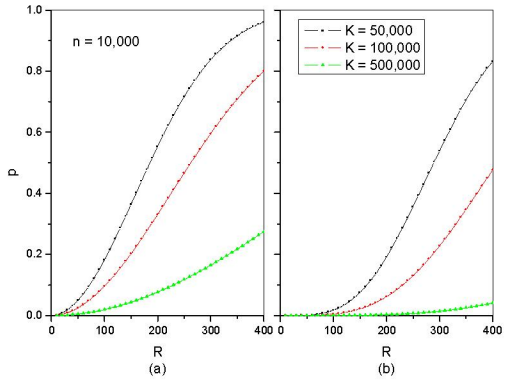

Fig. 3. (a) Performance of Gligor's scheme and (b) performance of Chan's Scheme ($q = 2$) when network size $n = 10,000$.

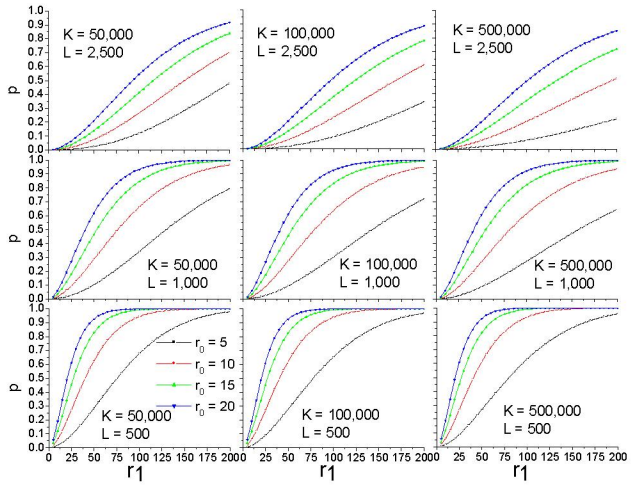

Fig. 4. The proposed q-composite scheme: p vs. r_0 and r_1 under different values of K and L, when network size $n = 10,000$ and $q = 2$.

required key ring size in the proposed scheme is around 50% less than that of Eschenauer et. al.'s scheme as will be shown in the next section. The evaluation of the proposed q-composite scheme is shown in Fig. 4 and as comparison, the performance of Chan et. al.'s q-composite scheme under the same settings is illustrated in Fig. 3(b). The performance improvement is again very significant. For instance, when $n = 10,000$ and $p_{required} = 0.5$, 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 ($q = 2$).

The improvement of the proposed two schemes goes higher as the network size n grows. For example, when $n = 50,000$ and $p_{required} = 0.5$, the proposed basic scheme requires as low as 100 keys with $K = 250,000$ as shown in Fig. 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

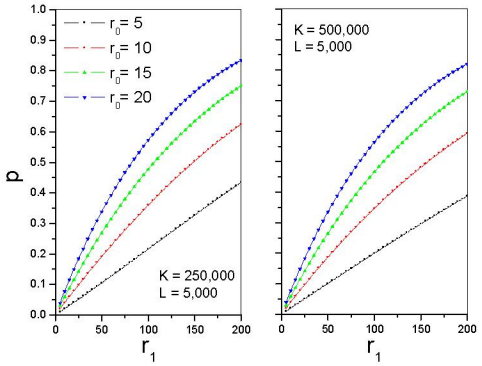

Fig. 5. The proposed basic scheme: p vs. r_0 and r_1, when network size $n = 50,000$.

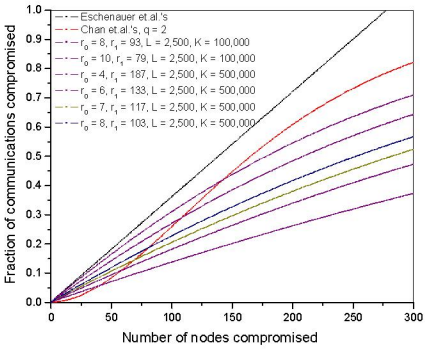

Fig. 6. Security strength of the proposed basic scheme with $n = 10,000$, $p_{required} = 0.5$ and $R_{max} = 192$

network size due to the extremely limited storage space of the sensor nodes.

Fig. 2) and Fig. 4) also illustrate how the performance of the proposed two schemes vary under different system settings, i.e., different values of K, L and (r_0, r_1) pairs. We could find that under a given network size n, the performance of the proposed schemes decreases as either K or L increases. From Eq. (14) 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 given. 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 Eq. (10) and the values of (r_0, r_1) are subject to Eq. (11) or Eq. (12).

V. SECURITY ANALYSIS

In this section, we study the security resilience of the proposed scheme against node capture by calculating the

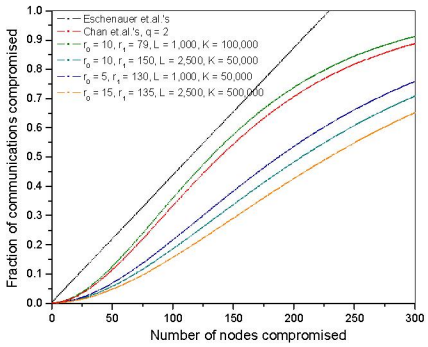

Fig. 7. Security strength of the proposed q-composite scheme with $n = 10,000$, $p_{required} = 0.5$, $q = 2$ and $R_{max} = 161$

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_{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

$$(1 - (1 - (\frac{r_0}{L} + \frac{r_1}{K}))^m)^q \tag{13}$$

Therefore, averagely the compromising probability is

$$\sum_{s=q}^{m} (1 - (1 - (\frac{r_0}{L} + \frac{r_1}{K}))^m)^s \frac{P(|\mathcal{R}_i \cap \mathcal{R}_j| = s)}{\sum_{t=q}^{m} P(|\mathcal{R}_i \cap \mathcal{R}_j| = t)} \tag{14}$$

Eq. (14) also represents the fraction of additional communications that an adversary can compromise based on the key information retrieved from m captured nodes in the worst case. Fig. 6 shows the security strength of the proposed basic scheme, where $n = 10,000$, $p_{required} = 0.5$ and $R_{max} = 192$. Obviously, the proposed scheme could offer a much better resilience property while requiring a much smaller key ring size when compared with Eschauer and Gligor's. Fig. 7 illustrates the security strength of the proposed q-composite scheme, where $n = 10,000$, $p_{required} = 0.5$, $q = 2$ and $R_{max} = 161$. Again the proposed q-composite scheme offers a much better 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. Fig. 8 shows the security strength of the proposed basic scheme, when $n = 10,000$, $p_{required} = 0.5$ and key

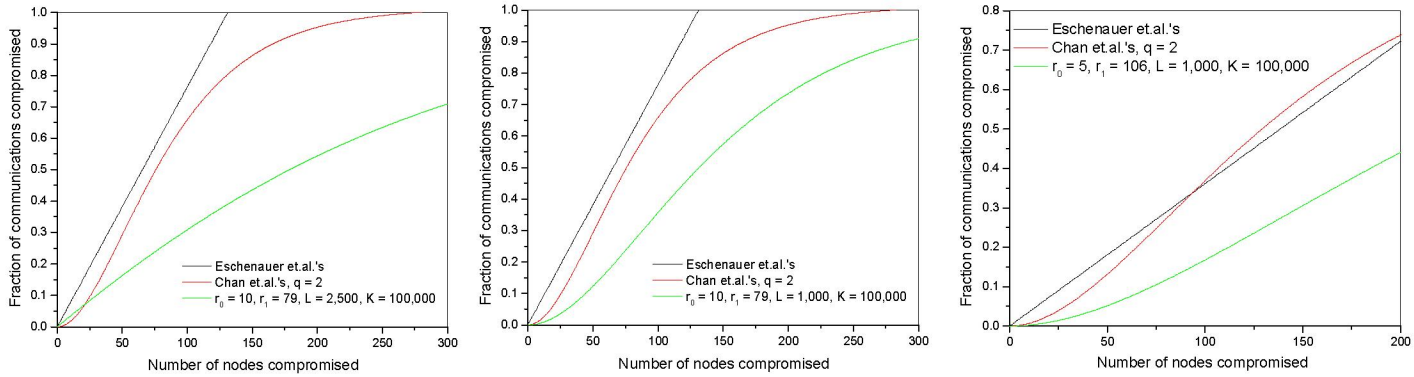

Fig. 8. Security strength of left) basic scheme with $n = 10,000$, $p_{required} = 0.5$ and $R = 90$, middle) q-composite scheme with $n = 10,000$, $p_{required} = 0.5$, $q = 2$ and $R = 90$ and right) q-composite scheme with $n = 10,000$, $p_{required} = 0.33$, $q = 2$ and $R = 112$.

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. Fig. 9 shows the significant resilience improvement of the proposed q-composite scheme when $n = 10,000$, $p_{required} = 0.5$, $q = 2$ 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 Fig. 10.

VI. CONCLUSION

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 term of key generation 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 easily outperformed 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] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, "SPINS: Security Protocols for Sensor Networks". In Proc. of Mobicom 2001, Rome Italy, July 2001.

[2] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks", In Proc. of ACM CCS 2002, Washington, DC, 2002.

[3] H. Chan, A. Perrig and D. Song, "Random Key Predistribution Schemes for Sensor Networks", In Proc. of IEEE Symposium on Research in Security and Privacy 03, 2003

[4] D. Liu and P. Ning, "Location-Based Pairwise Key Establishments for Relatively Static Sensor Networks", SASN03, 2003.

[5] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks", In Proc. of ACM CCS03, 2003.

[6] W. Du, J. Deng, Y. Han, and P. Varshney, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks", In Proc. of ACM CCS 03, 2003.

[7] W. Du, J. Deng, Y. Han, S. Chen and P. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge", IEEE INFOCOM04, Hongkong, 2004.

[8] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach", In Proc. of ICNP03, 2003.

[9] K. Yuksel, J. Kaps, and B. Sunar, "Universal Hash Functions for Emerging Ultra-Low-Power Networks", In Proc. of CNDS 04, 2004.

[10] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware Key Management Scheme for Wireless Sensor Networks", SASN 04, 2004.

[11] Erdos and Renyi, "On random graphs", I. Publ. Math. Debrecen, 6:290-297, 1959.

[12] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", IETF INTERNET RFC 2104, 1997.