A Key Distribution Scheme for WSN Based on Deployment Knowledge and Dual-Key Pools

Lin You¹ and Younan Yuan²

¹College of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China

²System Department, Baidu R & D center in Shanghai, Shanghai 201210, China
mryoulin@gmail.com

Abstract

By using deployment Knowledge and cellular model, a novel key distribution scheme based on deployment and dual-key pools is proposed for WSN. Each sub-key pool in the network contains two types of key pools: one doubled by the offline server is shared by three adjacent cells before the deployment and the other one distributed to each cell is non-overlapping. The server distributes four random numbers for each cell and ensures each adjacent cell sharing one random number. In order to improve the network security, nodes among the neighboring cells adopt a binary hash function and a corresponding random number to generate pair-wise keys. Compared with Du scheme and q-composite scheme, the simulation results show that our proposed scheme not only has efficiently improved the probability of pair-wise key establishment and invulnerability, but also has certain advantage on the nodes' storage space.

Keywords: Wireless sensor network (WSN), Security, Deployment knowledge, Cell, Dual-key pool, Hash function

1. Introduction

Wireless sensor network (WSN) is a self-organizing network with a large number of ultra-small autonomous sensor nodes which are designed to monitor information, collect data, *etc.*, [1]. There exist lots of constraints in a WSN that make its security different from the conventional networks, such as the limited energy, resource and computation ability in the nodes, *etc.*, Among these issues, pair-wise key distribution schemes are much more difficultly established in WSN than the traditional networks.

In [2], Eschenauer and Gligor proposed a random key pre-distribution scheme (E-G scheme) for the large-scale network based on the probability theory. According to the scheme, each node selects a random subset of keys as its key ring from a domain pool. The neighbor nodes generate pair-wise keys only when they have at least a common key in their key rings. In the basic E-G scheme, Chan *et al.*, proposed a *q*-composite random key pre-distribution schemes for sensor networks [3] (*called q-composite scheme for short*), and it is a well-known key distribution scheme. The neighbor nodes demanded at least *q* common keys in their key rings to improve the security of their links. Based on the multiple key spaces constructed by the use of Blom's scheme which have the threshold characteristic, Du, *et al.*, [4] proposed another key pre-distribution scheme which substantially improved the security of the networks. This scheme will be called Du scheme for short in the following discussions. In [5], Liu, *et al.*, employed some polynomials in a prime field instead of Blom matrix to construct pair-wise keys, but it will increase the nodes' computation cost. Using the

deployment knowledge of the nodes, Du, *et al.*, [6] put forward a novel scheme to establish the pair-wise keys among the neighbor nodes, and the scheme could enhance the local connectivity probability and the security of the networks. Afterwards, many improved schemes were presented [7-9].

Based on dual-key pools (a pair of key pools) and deployment knowledge, we propose a pre-distribution scheme for WSN, which is called KPDK for short. Before the network deployment, the offline server generates two types of key pools using the deployment knowledge. At the same time, the server distributes four random numbers and extends the size of the sub-key pools (using hash function) for each cell. Using these random numbers and the distributive keys, our scheme can effectively improve the local connectivity and the invulnerability.

2. Notations and Network Model

2.1. Notations

In Table 1, we list the common notations and their descriptions which will be used in our following discussion.

Table 1. Notations

Notations	Descriptions	Notations	Descriptions
$cell_{i,j}$	i -th row and j -th column of the cell	$h(x)$	One-way hash function
S	Domain key pool of the network, the size equals to $ S $	m	The size of the nodes' key ring
$sa_{i,j}$	i -th row and j -th column of the key pools (its type is sa) and its size equals to $ sa = (sc - sb)/3$	$sa'_{i,j}$	Extended set of $sa_{i,j}$ using a hash function, it includes 3 key pools with type sa'
$SA_{i,j}$	The set of 3 adjacent key pools with type sa in $sc_{i,j}$	$g(r, x)$	One-way hash function for variables r and x
$sb_{i,j}$	Sub-key pool with type sb in $sc_{i,j}$ and its size equals to $ sb $	$r_{i,j}$	Random number in $cell_{i,j}$
$sc_{i,j}$	$sc_{i,j} = \{sa_{i,j}, sb_{i,j}\}$, and its size equals to $ sc = 3 sa + sb $	$sc'_{i,j}$	$sc'_{i,j} = \{sa'_{i,j}, sb_{i,j}\}$, its size equals to $ sc' $
$sa'_{i,j}$	The extended set of $sa_{i,j}$ using hash function and its size equals to $2 sa $, and it is a set with type sa'	k_{uv}	Pair-wise key between the nodes u and v

2.2. Network Model

In certain applications, the positions of nodes may be predicated, and we can employ the deployment feature to improve the local connectivity and the security of WSN. In Du scheme [4], it was thought that the probability that the nodes will become some neighbor nodes airdropped in the same deployment batch is higher than that in the different batches. According to the theory, we divide the destination area into the different groups in our scheme.

Triangle, square and hexagon are three common network models, and we utilize the hexagon model for its large covering area advantage [10] in our proposed scheme. The node

position probability can be approximately estimated as two-dimensional Gaussian distribution. Each hexagon is called a *cell* as shown in Figure 1. Assuming that the number of the cells is $n \times t$ and each cell is identified as $cell_{i,j}$, where $1 \leq i \leq n, 1 \leq j \leq t$.

3. KPDK Scheme

KPDK scheme adopts two types of sub-key pools: sa and sb . For the neighbor nodes among the adjacent cells to $cell_{i,j}$, they can generate the pair-wise keys only using $sa_{i,j}$. As shown in the following Eq. (1), Eq. (2) and Eq. (3), $sc_{i,j}$ is consisted of $sb_{i,j}$ and $sa_{i,j}$, where $sa_{i,j}$ is composed of three key pools in type sa . $sa'_{i,j}$ is composed of three key pools in type sa' , where type sa' is extended from type sa by $h(x)$. $sa'_{i,j}$ and $sb_{i,j}$ are mixed together to become $sc'_{i,j}$. The server distributes four random numbers to each cell and enlarges the size of the key pools using the hash function for each cell. Using the random numbers and the distributive keys, any neighbor node can compute its pair-wise keys.

Figure 1. Network Structure Model

3.1. Construction of Sub-Key Pools

Take $cell_{i,j}$ for example. The final sub-key pool $sc'_{i,j}$ in $cell_{i,j}$ includes $sb_{i,j}$ and the three different key pools in type sa' as shown in Eq. (1) and Eq. (2). We mark the three key pools in type sa' as $sa'_{i,j,k}$ generated by $sa_{i,j,k}$ through the hash function $h(x)$, where $k = 1, 2, 3$. According to Eq. (1), we get $sa_{i,j} = \{sa_{i,j,1}, sa_{i,j,2}, sa_{i,j,3}\}$ and $sa'_{i,j} = \{sa'_{i,j,1}, sa'_{i,j,2}, sa'_{i,j,3}\}$. And $sa_{i,j}$ are non-overlapped each other and shared by the three adjacent cells as shown in Figure 1, where $1 \leq i \leq n, 1 \leq j \leq t$. For example, $sa_{2,3}$ is shared by $cell_{2,2}$, $cell_{2,3}$ and $cell_{3,3}$, and $sa_{2,4}$ is shared by $cell_{2,3}$, $cell_{1,4}$ and $cell_{2,4}$. For $sa'_{i,j,k}$, it is mixed with $sa_{i,j,k}$ and its corresponding hash values, so we have $|sa'_{i,j,k}| = 2|sa|$ with $k = 1, 2, 3$. Eq. (1) and Eq. (2) show the construction of the final sub-key pool $sc'_{i,j}$, where $1 \leq i \leq n, 1 \leq j \leq t$.

$$\begin{cases} sa_{i,j} = \{sa_{i+1,j}, sa_{i,j}, sa_{i,j+1}\}, & \text{if } j \bmod 2 = 0 \\ sa_{i,j} = \{sa_{i,j}, sa_{i-1,j}, sa_{i,j+1}\}, & \text{if } j \bmod 2 = 1 \end{cases} \quad (1)$$

$$\begin{cases} SA'_{i,j} = \begin{cases} \{Sa'_{i+1,j}, Sa'_{i,j}, Sa'_{i,j+1}\}, & \text{if } j \bmod 2 = 0 \\ \{Sa'_{i,j}, Sa'_{i-1,j}, Sa'_{i,j+1}\}, & \text{if } j \bmod 2 = 1 \end{cases} \\ Sc'_{i,j} = \{SA'_{i,j}, Sb_{i,j}\}. \end{cases} \quad (2)$$

3.2. Description of KPDK Scheme

In KPDK scheme, the pair-wise keys are established by the following four steps:

- (1) Pre-distribution of keys and random numbers.
- (2) Establishment of pair-wise keys.
- (3) Final key processing.
- (4) Key path establishment.

3.2.1. Pre-Distribution of Keys and Random Numbers: Besides the pre-distributive keys, the offline server also needs to distribute four random numbers to each node in KPDK scheme.

Figure 2. Distributions of Random Numbers among Different Cells

- a) **Key Pre-Distribution:** Take the nodes in $cell_{i,j}$ for example. After the construction of the sub-key pools $Sc'_{i,j}$ in Section 3, each node chooses m keys and the corresponding indexes randomly from $Sc'_{i,j}$. Therefore, each node may contain two sorts of keys: one calculated by $SA'_{i,j}$ and the other calculated by $Sb_{i,j}$.
- b) **Random Number Pre-Distribution:** Before the deployment of the nodes, a random number r_0 is assigned to $cell_{i,j}$ by the server, where $r_0 = r_{i,j}$. In addition to r_0 , three other random numbers (labeled as r_1 , r_2 and r_3 respectively which are chosen from the adjacent cells whose directions are the north, the southwest and the southeast to $cell_{i,j}$) are distributed to $cell_{i,j}$. For example, $cell_{2,4}$ contains $r_0 = r_{2,4}$, $r_1 = r_{3,4}$, $r_2 = r_{2,3}$ and $r_3 = r_{2,5}$, and their directions are shown in Figure 2. It indicates that our scheme can ensure any two adjacent cells to share one random number.

3.2.2. Establishment of pair-wise keys: There are two cases in the phase of pair-wise key establishment: one is the key establishment in the same cell, and the other is the key establishment among the neighboring cells.

According to the previous discussion, we can see that the nodes in $cell_{i,j}$ have contained m keys and some of them may be chosen from $Sb_{i,j}$ and $SA'_{i,j}$. Assuming that U and V are

two neighboring nodes and the index of the common key between them is kid_c . The procedure of key establishment is as follows:

a) **Pair-wise key establishment in the same cell:** For this case, suppose that U and V are the neighbor nodes in the same cell, then there are two possible situations:

i) The key with the index of kid_c is included in $Sb_{i,j}$. Under this case, kid_c corresponds to a key (labeled as v_c) uniquely. We can get the pair-wise key between U and V using $k_{uv} = k_{vu} = h(v_c)$.

ii) The key with the index kid_c is included in $SA'_{i,j}$. $SA'_{i,j}$ is generated by the Eq. (1) and Eq.(2), and $SA'_{i,j}$ is extended through $SA_{i,j}$ by $h(x)$. It means that an index of the key may correspond to two keys. We mark the two keys as va_0 and va_1 , where $va_1 = h(va_0)$. In other words, in this case, there exist two keys with the same index kid_c . There are four possible situations to generate the pair-wise keys. Obviously, we have $k_{uv} = k_{vu}$.

(1) If U stores va_0 and so does V, then $k_{uv} = h(va_0)$, $k_{vu} = h(va_0)$.

(2) If U stores va_0 and V stores va_1 , then $k_{uv} = h(h(va_0))$, $k_{vu} = h(va_1)$.

(3) If U stores va_1 and V stores va_0 , then $k_{uv} = h(va_1)$, $k_{vu} = h(h(va_0))$.

(4) If U stores va_1 and so does V, then $k_{uv} = h(va_1)$, $k_{vu} = h(va_1)$.

b) **Key establishment among neighboring cells:** For this case, U and V are the neighbors among adjacent cells. According to the above distribution principle of the random numbers, any two adjacent cells share a random number. Assuming that $cell_a$ and $cell_b$ are two arbitrary neighboring cells and r is their sharing random number. Based on the above analysis, any two neighboring cells can find a sharing key pool with type sa . U and V can generate the pair-wise key only when they find a sharing key from the common type sa key pool. So, it must be va_0 or va_1 ($va_1 = h(va_0)$) with the index kid_c . There are four possible situations for the generation of the pair-wise keys. Obviously, we have $k_{uv} = k_{vu}$.

(1) If U stores va_0 and so does V, then $k_{uv} = g(r, va_0)$, $k_{vu} = g(r, va_0)$.

(2) If U stores va_0 and V stores va_1 , then $k_{uv} = g(r, h(va_0))$, $k_{vu} = g(r, va_1)$.

(3) If U stores va_1 and V stores va_0 , then $k_{uv} = g(r, va_1)$, $k_{vu} = g(r, h(va_0))$.

(4) If U stores va_1 and so does V, then $k_{uv} = g(r, va_1)$, $k_{vu} = g(r, va_1)$.

If U and V find more than one sharing keys when the nodes are in the same cell or in the neighboring cells, then they will generate more than one temp pair-wise key according to the above discussions by using each sharing key respectively. The final pair-wise key can be computed by XOR operation on the temp pair-wise keys.

3.2.3. Final Key Processing: Assuming that U is in $cell_{i,j}$ and r_1, r_2 and r_3 are the three random numbers selected from the three adjacent cells to $cell_{i,j}$, respectively.

Firstly, for keys in $Sb_{i,j}$, the nodes in $cell_{i,j}$ store the hash values of the keys instead of the original keys.

Secondly, for keys in $SA'_{i,j}$, the nodes in $cell_{i,j}$ need to take the following procedures.

Figure 2 shows that any key pool $sa_{i,j}$ (a key pool with type sa) can be shared by three neighboring cells. Suppose that $sa'_{i,j}$ is the extended key pool of $sa_{i,j}$ and K_0 is the set of keys generated by $sa'_{i,j}$ in U. U can compute some new keys which are used to replace the keys in K_0 by $g(r, x)$. The three random numbers r_i are stored in U. For the key $key_i \in K_0$, U stores the following keys:

$$\begin{cases} g(r_1, key_i) \\ g(r_2, key_i) \\ g(r_3, key_i) \end{cases} \quad 0 \leq i \leq |K_0| \quad (3)$$

If U has already possessed a key established in the process of the above key establishment, then it just need to store one of them.

Thirdly, the nodes delete the all stored random numbers and some keys so that the number of the stored keys in each node equal to m .

3.2.4. Path Pair-Wise Key Establishment: After the processing of the key establishment, it still may exist some neighbor nodes that can't generate their pair-wise keys directly, so our scheme will start the path pair-wise key establishment.

KPDK scheme sets up the path pair-wise keys using multi-hops. Considering the energy consumption in the nodes, it is important to set a threshold to limit the hops: if the number of hops is less than the threshold, then the pair-wise keys can be produced as the way described in Section 3.2.2. Otherwise, the nodes refuse to generate the keys immediately.

4. Performance Analyses and Comparison

In this section, we evaluate the performances of KPDK scheme, Du scheme [6] and q -composite scheme [3]. In order to compare their performances, our analysis and simulations are installed as the following:

- 1) The size of the domain key pool is $|S| = 100000$ and the wireless communication range for each node is $R = 40$ meters.
- 2) The sharing factor of sub-key pools in Du scheme is $a = b = \lambda = 0.125$.
- 3) The ratio of the sharing key pool is $a = |sa|/|sc| = 0.125$ in our scheme, and so we get $|sb|/|sc| = (1 - 3a) = 0.625$;
- 4) The deployment area is 1000×1000 square meters and the number of the cells is $n \times t$ in our scheme and Du scheme, where $n = 10$, $t = 10$.

4.1. Connectivity Analyses

In KPDK scheme, the local connectivity probability is considered in two cases: the probability that the neighbor nodes are in the same cell (denoted as p_{com}) and the probability that the neighbor nodes are among different cells (denoted as p_{dif}). According to Eq. (1) and Eq.(2), the sub-key pool $sc'_{i,j}$ contains $sa'_{i,j}$ and $sb_{i,j}$, where $sa'_{i,j}$ is consisted of three key pools with type sa' (denoted as $sa'_{i,j,1}$, $sa'_{i,j,2}$ and $sa'_{i,j,3}$, respectively). $sa'_{i,j,k}$ (Its initial key pool is $sa_{i,j,k}$) is a key pool extended by the hash function $h(x)$ and $sa_{i,j,k}$ ($k = 1, 2, 3$). Hence, we have $|sa'_{i,j,k}| = 2|sa|$ and $|sc'_{i,j}| = 3 \times 2|sa| + |sb| = |sc| + 3|sa|$.

4.1.1. Probability p_{com} : p_{com} represents the local connectivity probability when the neighbor nodes are in the same cell. Assuming that U is in $cell_{i,j}$, some keys stored in U may be included in $SA'_{i,j}$ and $Sb_{i,j}$ simultaneously. Let $p(k)$ be the probability that there exist k keys out of the m keys in the node that are chosen from $SA'_{i,j}$. Let $p(k_1, k_2)$ be the probability that the neighbor nodes U and V can generate the pair-wise key if in the nodes U and V there are respectively k_1 and k_2 keys which are selected from $SA'_{i,j}$. Then, the three probabilities $p(k)$, $p(k_1, k_2)$ and p_{com} can be computed as the following:

$$p(k) = \frac{\binom{6|SA|}{k} \binom{|Sb|}{m-k}}{\binom{Sc}{m}} \quad (4)$$

$$\begin{aligned} p(k_1, k_2) &= 1 - \frac{\binom{6|SA|}{k_1} \binom{6|SA|-2k_1}{k_2} \binom{|Sb|}{m-k_1} \binom{|Sb|-(m-k_1)}{m-k_2}}{\binom{6|SA|}{k_1} \binom{|Sb|}{m-k_1} \binom{6|SA|}{k_2} \binom{|Sb|}{m-k_2}} \\ &= 1 - \frac{\binom{6|SA|-2k_1}{k_2} \binom{|Sb|-(m-k_1)}{m-k_2}}{\binom{6|SA|}{k_2} \binom{|Sb|}{m-k_2}} \end{aligned} \quad (5)$$

$$p_{com} = \sum_{k_1=0}^{\min(m, 6|SA|)} \sum_{k_2=0}^{\min(m, 6|SA|)} p(k_1) p(k_2) p(k_1, k_2) \quad (6)$$

Figure 3. Probabilities that the Key Establishment of KPDK, Du and q -composite with the Neighboring Nodes in the Same Cells

p_{com} can be computed in three cases: a) $Sc_{i,j}$ is equal to $Sb_{i,j}$ (then $|Sc| = |Sb|$ and $SA'_{i,j}$ is null); b) $Sc_{i,j}$ is equal to $SA'_{i,j}$ and $|Sc| = 6|SA|$ (then $Sb_{i,j}$ is null); c) $Sc_{i,j}$ is consisted of both $Sb_{i,j}$ and $SA'_{i,j}$.

Figure 3 shows the three probability curves p_{com} of KPDK scheme, Du scheme and q -composite scheme. The probability performance of the key establishment with neighbor nodes in the same cell indicates that the probability of KPDK scheme is larger than that of Du scheme when the nodes have the same number of the stored keys. Under the same condition, the local connectivity of q -composite scheme is the worst because it is a probabilistic scheme without using other effective measures. So the probability p_{com} in KPDK scheme is the largest, while the probability in q -composite scheme is the smallest. As Figure 3 shown, p_{com} in KPDK scheme is higher than Du scheme and q -composite when $0 < m \leq 90$, but p_{com} in Du scheme is approaching to KPDK scheme (approximately equal to 1) when $m > 90$.

The reason that the performance of p_{com} in our scheme is better than the two others lies in the following two aspects:

a) The size of the domain key pool $|S_c'|$ extended by $h(x)$ in our scheme is smaller than Du scheme. Based on the work in [6], we know that the size of the domain key pool in Du scheme is $|S_c|$, where

$$|S_c| = |S| / (nt - (2nt - t - n)a - 2(nt - t - n)b).$$

Take the given parameters for example. The size of the domain key pool in Du scheme is 1817 while the extended domain key pool in our KPDK scheme is 1375.

b) Each set $SA'_{i,j}$ which is contained in $Sc'_{i,j}$ is consisted of $3|SA|$ pairs of keys (each pair-wise key includes a key and its hash value). Let a pair of the keys be marked as $(v_0, h(v_0))$. The neighbor nodes use any v_0 or $h(v_0)$ instead of both of them to establish the pair-wise keys. And it will improve the probability p_{com} .

4.1.2. Probability p_{dif} : p_{dif} represents the probability of key establishment with the neighbor nodes among the different cells. We know that each cell shares one random number and $2|SA|$ keys (keys in SA' are extended from $|SA|$ keys by $h(x)$). p_{dif} Can be computed as $1 - Pr(2 \text{ nodes have no sharing keys})$. We first compute the probability $Pr(2 \text{ nodes have no sharing keys})$ as follows: A node chooses t keys from the shared $2|SA|$ keys in some fixed $Sc'_{i,j}$, and then selects $m - t$ keys from the non-shared $|Sc'| - 2|SA|$ keys in $Sc'_{i,j}$. In the meantime, from the remaining keys in $Sc'_{i,j}$, another node selects m keys that any two of which cannot consist of a pair-wise key. It follows that

$$Pr(2 \text{ nodes have no sharing keys}) = \frac{\sum_{t=0}^{\min(m, 2|SA|)} \binom{2|SA|}{t} \binom{|Sc'| - 2|SA|}{m-t} \binom{|Sc'| - 2t}{m}}{\binom{|Sc'|}{m} \binom{|Sc'|}{m}}.$$

Hence, we have

$$p_{dif} = 1 - Pr(2 \text{ nodes have no sharing keys})$$

$$= 1 - \frac{\sum_{i=0}^{\min(m, 2|S_a|)} \binom{2|S_a|}{t} \binom{|S_c'| - 2|S_a|}{m-t} \binom{|S_c'| - 2t}{m}}{\binom{|S_c'|}{m} \binom{|S_c'|}{m}} \quad (7)$$

Figure 4. Probabilities that the Key Establishment of the Schemes KDPK, Du and q-composite with the Neighboring Nodes among Different Cells

Figure 4 shows the probabilities of key establishment with the neighbor nodes among different cells for KDPK scheme, Du scheme and q -composite scheme ($q=1, 2$). It indicates that the probability of p_{dif} in KDPK scheme is higher than the other two schemes. This is mainly due to the following aspects:

a) The size of the final extended sub-key pool $s_{c,i,j}$ in KDPK scheme is smaller than that in Du scheme, and the size of the adjacent shared key pools are expanded by the hash function to be twice as that of Du scheme. It leads to that the probability will increase when the same number of the keys is selected for the pair-wise key establishment in the neighbor nodes using the hash functions.

b) Du scheme created the pair-wise keys based only on whether the two neighbor nodes have shared keys in a sharing key pool, but did not employed any other measurement.

c) q -composite scheme regarded the domain key pool as a whole and would create a pair-wise key when the number of the common keys in the neighbor nodes was larger than or equal to q . It did not use any deployment knowledge or other node information as our scheme does. It leads to that, under the same conditions, q -composite has the lowest local connection probability.

Hence, the probability p_{dif} in KDPK scheme is the largest, while p_{dif} in q -composite scheme is the smallest.

In a word, for both p_{com} and p_{dif} , our scheme shows the best, while q -composite shows the worst. Hence, as Figure 5 shown, the average network connectivity performance of our KDPK

scheme is superior to that of Du scheme, while the average network connectivity performance of Du scheme is superior to that of q-composite.

Figure 5. Average Network Connectivity among Du scheme, q-composite and our Scheme

4.2. Security Analyses

After the key processing phase described in Section 3.2, each node still stores m keys. As Figure 1 shown, the size of the original domain key pool is $|S| = nt(|S_b| + |S_a|)$.

Because of the adjacent random numbers and the using of a hash function, the sizes of the key pools with type S_a are extended three times as the their original sizes, that is to say, the sizes of the key pools with type S_a' are extended six times as the sizes of the type S_a key pools. But due to the deletion of certain relevant keys, the enlargement of the key pools with type S_a can be considered as the extension of the domain key pool. For example, for the random keys r_1, r_2, r_3 and a given key x_0 , the node will store keys $\{g(r_1, x_0), g(r_2, x_0), g(r_3, x_0)\}$ generated by Eq. (3), that is, the number of each key in the networks is three times as its original number. The domain key pool S is enlarged by the sharing random keys and $g(r, x)$. The enlarged domain key pool is denoted as S' . From the formula $|S_a| = a|S_c| = a(|S_c|/(nt))$, we have

$$\begin{aligned}
 |S'| &= nt \times (|S_b| + 3|S_a|) \\
 &= nt \times (|S_b| + |S_a|) + 2nt|S_a| \\
 &= |S| + 2nt|S_a| = (2a + 1)|S|
 \end{aligned} \tag{8}$$

According to the above analysis, when x nodes have been captured, the probability (denoted as p_c) of that the corresponding links will be compromised is

$$\begin{aligned}
 p_c &= 1 - \left(1 - \frac{m}{|S'|}\right)^x \\
 &= 1 - \left(1 - \frac{m}{(2a + 1)|S|}\right)^x, \text{ where } 0 \leq a \leq \frac{1}{3}.
 \end{aligned} \tag{9}$$

When both the key number m and the number of the captured nodes are determinate, Eq. (9) means that the larger the sharing key pool ratio a ($= |S_a|/|S_c|$), the lower the probability p_c . Fig.6 shows that the link security at $m = 50, 100, 150$ when 1500 nodes are captured, respectively. If a and x are determinate, then it is clear that the more the keys stored in the nodes, the higher the probability p_c , and so the lower the nodes' security. On the contrary, the less the keys stored in the nodes, the higher the nodes' security. Similarly, when a and m are determinate, the more the number of the captured nodes, the higher the probability p_c , and so the lower the nodes' security.

Figure 6. Relationship of the Fraction of the Compromised Link among the Non-Compromised Nodes vs. the Ratio a when an Adversary has Compromised 1500 Nodes

The following Figure 7 and Figure 8 show the relationship of the fraction of the compromised link between the non-compromised nodes vs. the number of the compromised nodes among the different schemes when the network connectivity probabilities are 0.33 and 0.5, respectively. We can see that the performance of KPDK scheme is the best. There are two key reasons for that our scheme is superior to the two other scheme in security. One is that the numbers of the keys in the nodes in our KPDK scheme is the smallest. For example, in Figure 5 and Table 2, when the network connectivity probability equals to 0.33 and 0.5, respectively, the number of the keys in the nodes is respectively 30 and 44 in KPDK scheme, while the numbers of the keys in the nodes are respectively 46 and 68 in Du scheme, and it is respectively 200 and 262 in q -composite scheme ($q=1$), or is respectively 341 and 408 in q -composite scheme ($q=2$). The other is that Eq. (9) shows that the size of the domain key pool in KPDK scheme will increase to $(2a + 1)|S|$ instead of $|S|$. If the number of the captured nodes remains the same, the number of the keys stored in the nodes, the higher the probability that the corresponding link is broken. It follows that our scheme has the highest security compared with the two other schemes since the stored keys in the final domain key pool in our scheme is the least.

Figure 7. Relationship of the Fraction of Compromised Link among the Non-Compromised Nodes when the Network Connectivity Probability Equals to 0.33 for the Different Schemes

Figure 8. Relationship of the Fraction of Compromised Link among the Non-Compromised Nodes when the Network Connectivity Probability Equals to 0.5 for the Different Schemes

4.3. Storage Space Analyses

According to the phase of the key distribution and the establishment, the storage space of the nodes will increase during the process. Suppose that the average numbers of the keys including the random numbers in the nodes is $\overline{M}$ in this process, and let $\overline{M}_1$ be the average number of the stored keys in the nodes without including the stored random numbers.

Eq. (3) shows that each key belonging to the type s_a will be extended by $g(r, x)$, and each cell contains three type s_a key pools. The number $\overline{M}_1$ is only affected by $g(r, x)$. Hence, we have

$$\begin{aligned}\overline{M}_1 &= m + |m_a| = m + \frac{6|S_a|}{|S_c'|}m = m \left(1 + \frac{6|S_a|}{|S_c| + 3|S_a|} \right) \\ &= m \left(1 + \frac{6a \frac{S}{nt}}{\frac{S}{nt} + 3a \frac{S}{nt}} \right) = m \left(1 + \frac{6a}{1 + 3a} \right).\end{aligned}\quad (10)$$

In addition to the $\overline{M}_1$ keys, the nodes still have 4 random numbers. Thus,

$$\overline{M} = \overline{M}_1 + 4 = m(1 + 6a/(1 + 3a)) + 4, \text{ where } 0 \leq a \leq 1/3.$$

Clearly, we have

$$m + 4 \leq \overline{M} \leq 2m + 4.$$

But after the phase of the key processing, each node ensures to have m keys instead of $\overline{M}$. Hence, from the pre-phase of the key establishment to the post-phase of the key establishment, the number of the keys stored in the nodes changes, but the final number remains m . From Figure 2 and Table 2, we can know that the number of the keys stored in the nodes in KPDK scheme is the smallest at the same average network connectivity probabilities. Hence, our scheme will save the most amount of the storage space than the two other schemes at the same conditions.

Table 2. Average Numbers of the Keys Stored in the Nodes at the Different Average Network Connectivity Probabilities

PL		0.2	0.33	0.5	0.7
KDPK		22	30	44	62
Du		32	46	68	106
<i>q-composite</i>	<i>q=1</i>	148	200	262	346
	<i>q=2</i>	289	341	>400	>400

4.4. Expansibility Analyses

When the network has run for a period of time, the node energy may run out and need to add some new nodes constantly. There are two ways to add new nodes: 1) Replacement of the dead nodes; 2) Extension of the network.

4.4.1. Replacement of the Dead Nodes: If some new nodes are added into $cell_{i,j}$, then the offline server will take the following actions:

- Distribute four random numbers (allocated to $cell_{i,j}$ as in Section 3.2.1) to the new nodes.
- Keys in the three key pools of type s_a , which are included in $S_{c_{i,j}}$, should be replaced with their hash values (let the three substitutions of the key pools of type s_a be denoted as S_{a_1} , S_{a_2} and S_{a_3}).

- c) Use sa_1 , sa_2 and sa_3 to replace κ_0 successively, and the final sub-key pool will be formed.
- d) Each new node selects m keys from the final sub-key pool. Then, the pair-wise keys can be generated similarly as E-G scheme [2] did.

4.4.2. Extension of Network: In Figure 2, the server only needs to add some cells at the edge of the deployment area when the network needs to expand, it will construct the sub-key pools and distribute four random numbers according to Section 3.1. Finally, the new additive sensor nodes would be treated as those nodes that are used to replace the dead nodes.

5. Conclusions

Based on the deployment and the dual-key pools, we propose a novel key distribution for WSN in a cell model. Each cell has a key pool consisted of two types of key pool: type sa and type sb . Both the key establishment probability and the security are improved by the extended key pool of type sa , the sharing random numbers and the hash function. Compared with Du scheme and q -composite scheme, the simulation results show that our KDPK scheme performs the best in the pair-wise key establishment probability, the invulnerability and the node storage space.

Now, based on different technologies, some other key distribution schemes with different characteristics for WSN [11-15] have been proposed, and each of them has its own advantages and some disadvantages to some extent. In our future work, we plan to give a survey on the most representative key distribution systems for WSN and evaluate their performance in security, connectivity, the number of the stored keys and some others in their practical application scenarios. And we try to develop a more efficient key distribution scheme for WSN based on our survey and our evaluation on these most representative key distribution systems.

Acknowledgements

This work has been supported by Zhejiang Natural Science Foundation of Outstanding Youth Team Project (No.R1090138) and the National Cipher Development Fund of China during the 12th Five-Year Plan Period (No.MMJJ2011010019).

References

- [1] M. Tubaishat and S. Madria, "Sensor networks: an overview", IEEE Potentials, vol. 22, no. 2, (2003), pp. 20-23.
- [2] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington D.C., USA, (2002) November 18-22, pp. 41-47.
- [3] H. Chan, A. Perrig and D. Song, "Random key Pre-distribution schemes for sensor networks", Proceedings of the 2003 IEEE Symposium on Security and Privacy, (2003) May11-14, , pp. 197-213, Berkeley, CA, USA.
- [4] W. Du, J. Deng, Y. S. Han and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks", Proceedings of the 10th ACM Conference on Computer and Communications Security, (2003) October 27-30, pp. 42-51, Washington D.C., USA.
- [5] D. Liu, P. Ning and R. Li, "Establishing pairwise keys in distributed sensor networks", ACM Trans. Inf. Syst. Secur, vol. 8, no. 1, (2005), pp. 41-77.
- [6] W. Du, J. Deng, Y. Han, S. Chen and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, (2004) March 7-11, pp. 597-608, Hong Kong, China.
- [7] F. Anjum, "Location dependent key management in sensor networks without using deployment knowledge", Wireless Networks , vol. 16, no. 6, (2010), pp. 1587-1600.

- [8] P. De, Y. Liu and S. K. Das, "Deployment-aware modeling of node compromise spread in wireless sensor networks using epidemic theory", *ACM Trans. on Sensor Networks*, vol. 5, no. 3, **(2009)**, pp. 1-33.
- [9] Z. Yu and Y. Guan, "A key pre-distribution scheme using deployment knowledge for wireless sensor networks", *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, **(2005)** April 25-27, pp. 261-268, Los Angeles, California, USA.
- [10] J. Wang, L. Xia and J. Jing, "Analysis for location-based Key Pre-distribution in wireless sensor networks", *Proceedings of the 2th International Conference on Information and Computing Science*, **(2009)** May 21-22, pp. 297-300, Manchester, England, UK.
- [11] B. Tian, S. Han, J. K. Hu and T. Dillon, "A mutual-healing key distribution scheme in wireless sensor networks", *Journal of Network and Computer Applications*, vol. 34, no. 1, **(2011)**, pp. 80-88.
- [12] C. Y. Chen and H. C. Chao, "A survey of key distribution in wireless sensor networks", *Security Comm. Networks*, Published online, **(2011)**.
- [13] K. Shaila, S. H. Manjula, K. R. Venugopal and L. M. Patnaik, "Mobile node authentication using key distribution scheme in wireless sensor networks", *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 12, no. 1, **(2013)**, pp. 34-45.
- [14] H. Yu, J. S. He, T. Zhang and P. Xiao, "A group key distribution scheme for wireless sensor networks in the Internet of Things scenario", *International Journal of Distributed Sensor Networks*, vol. **(2012)**, Article ID 813594, **(2012)**, pp. 1-12.
- [15] S. M. Ruj, A. Nayak and I. Stojmenovic, "Pairwise and triple key distribution in wireless sensor networks with applications", *IEEE Trans. On Computers*, vol. 62, no. 11, **(2013)**, pp. 2224-2237.

Authors

Lin You, he received the B.S. and M.S. degree in mathematics from Jiangxi Normal University in 1985 and from Wuhan University in 1988, respectively. He received the Ph.D. degree from Dalian University of Technology in 2003. He is currently a professor in College of Communication Engineering of Hangzhou Dianzi University, China. His current research interests include algebra, information security and cryptography, biometrics recognition and their applications.

Younan Yuan, he received his B.S. degree in electronics and information engineering from Jiangxi University of Science and Technology, Ganzhou, China, in 2009, and received M.S. degree in communication engineering from Hangzhou Dianzi University, Hangzhou, China, 2012. He currently works as a network security engineer in the System Department, Baidu R & D center in Shanghai, China. His research interests include key management for WSN, network security.