State-Based Random Key Pre-Distribution Scheme for Wireless Sensor Networks

Jaemin Park, Zeen Kim, and Kwangjo Kim

International Research center for Information Security (IRIS)

Information and Communications University (ICU), Korea

Abstract

In wireless sensor networks (WSNs), for secure communications, the random key pre-distribution arises as the practical solution for sharing common keys between sensor nodes. Since sensor networks suffer from the resource constraints, we should consider the small computation, small number of keys, etc. while supporting the same security level, i.e., high resilience against node capture. In this paper, we propose a new random key pre-distribution scheme that utilizes new pre-deployment knowledge, *state of sensors*, to avoid unnecessary key assignments while supporting higher connectivity and resilience against node capture. The analysis of this paper shows that the outstanding performance of our proposed scheme with respect to the connectivity, memory usage, and resilience against node capture.

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. Since WSNs are vulnerable to malicious attacks, it is important to protect communications among sensor nodes to maintain confidentiality and integrity.

Recent research suggests that symmetric secret key pre-distribution is the only practical approach for establishing secure channels among sensors due to the low-power and limited computational capacity. Many random key pre-distribution schemes have been proposed [2-7]. The basic idea behind Eschenauer et al. [2] is to have a large pool of keys, from which a set of keys is randomly chosen and stored in each sensor node. Any of two nodes able to find common keys within their key subsets can use those shared keys for secure communication. Our scheme is based on the Eschenauer et al., and we refer to this scheme as the *basic scheme* throughout this paper. In case that certain pre-deployment knowledge is available, the performance of the key pre-distribution can be improved by exploiting such knowledge. Several schemes that utilize the pre-deployment knowledge such as location of sensors are proposed [4,7]. Based on the location information of each sensor, the probability that two sensors share a common key is improved.

However, above schemes still require each node to carry a large number of keys for large scale sensor networks. For example, to implement the random key pre-distribution schemes proposed in [2, 3] for a WSN of size 10,000, at least 200 keys are required for each node, which is almost half of the available memory (assume 64-bit keys and less than 4KB data memory [1]). Also, because useful communications are occurred only among active sensors, if two sensors sharing some keys are hardly in active-state at the same time, these keys are unnecessary and inefficient.

To address above problems, we propose a new approach for random key pre-distribution that exploits new pre-deployment knowledge, state of sensors. By facilitating new pre-deployment knowledge, we can improve the connectivity and reduce the number of required keys that each sensor should carry.

This paper is organized as follows: In section II, we model our pre-deployment knowledge. We propose our scheme in section III and we analyze our proposed scheme and compare it with the

본 연구는 산업자원부 지역협력연구사업
(R12-2003-004-01004-0) 지원으로 수행되었음

previous schemes in section IV. Finally, we conclude our paper in section V.

II. Modeling State of Sensors

1. Classification of State

We consider two major operational states: active and sleep. We define that sensors in sleep state consume the lowest amount of the node power; while being asleep, a node cannot interact with the external world. On the other hand, the sensors in active-state can interact with the external world with higher node power consumption.

2. Active-State Group Modeling

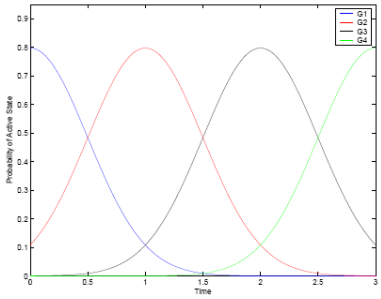
The state of sensor depends on the scheduler implemented in sensors, events that sensors may receive, MAC protocol and other variable factors. The probability of active-state is determined by the sleep scheduling algorithm, job scheduler, and randomness of other variable factors.

Since all possibilities related to the sensors' state are probabilistic and random, the probability of active-state for all sensors may have the different pdfs(probability density function). However, in our proposed scheme, keys are pre-distributed to each group classified by the probability of active-state at each time-interval. Therefore, all sensors in each group can be assumed to have same pdfs. In this paper, we assume such a group-based key pre-distribution, and we model each group follows Gaussian distribution. We also assume that each group has different time point when the probability is maximized. Based on these assumptions, we define that *Active-State Group* is the group of sensors which are highly probable in the active-state at the same time-interval. If a sensor k in G_i is the most probable in active-state at time t_m, the pdf of node k in group G_i is as follow:

$$f_k^i(t|k \in G_i) = \frac{1}{\sqrt{2\Pi}\,\rho} e^{-(t-t_m)^2/2\rho^2} = f(t-t_m) \quad (1)$$

where $f(t) = \frac{1}{\sqrt{2\Pi}\,\rho} e^{-t^2/2\rho^2}$. Without loss of generality, we assume that the pdf for each group is identical except the value of t_m, so we use $f_k(t|k \in G_i)$ instead of $f_k^i(t|k \in G_i)$ throughout this paper.

<Fig. 1> depicts the pdf of each group. We can find out that if one group has the highest probability of active-state at one time-interval, it also has the moderately high probability at nearby time-interval. Therefore, two neighbor groups are probable to be in active-state at the same time-interval with the moderate probability.



<Fig. 1> Probability Distribution of Active-State Group

III. The Proposed Scheme

1. Assumptions and Security Threats

To use the state as the pre-deployment knowledge, we define the following assumptions:

- Whole lifetime of WSN can be divided into many small time-intervals and each of them repeats periodically.
- There is no time-interval when all sensors are in sleep state.

WSN is vulnerable to several security threats. We consider two major security threats; node capture and eavesdropping. First, adversary can monitor communications between sensors due to the characteristic of the radio broadcast signal. Second, adversary can capture nodes and analyze all information embedded in each sensor.

2. Design Requirements

To address the security threats and the problems of existing key pre-distribution schemes, we propose our scheme which satisfies following requirements:

- *Small number of keys*: To address the limited memory constraint, small number of keys should be promised while supporting the same or higher level of security.
- *Higher connectivity*: With smaller number of keys, the probability that two sensors share at least one common session key at given time-interval should be higher.
- *Resilience against node capture*: Sensors are easily captured by adversaries. Once captured, they are analyzed and may reveal secret information to the adversaries. The proposed scheme should be resilient against node capture.

3. Notations and Terminologies

We utilize following notations and terminologies for convenience of description.

- *Global Key Pool*: A global key pool S is a pool of random symmetric keys, from which a group key pool is generated. (Cardinality=$|S|$)
- *Group Key Pool*: A group key pool S_i(i=1,2,3,…) is a subset of global key pool, S. (Cardinality=$|S_G|$)
- *Time-Interval*: A time-interval, T, is a part of lifetime of WSN. T is divided into the small time-intervals, T_i(i=1,2,3,…).
- *Group*: A group, G_i(i=1,2,3,…) is a set of sensors estimated to be in active-state at specific time-interval, T_i with high probability.
- *Key Ring*: A key ring $R_{i,j}$,(i,j=1,2,3,…) is a subset of group key pool, which is independently assigned to each sensor i classified as the G_j. (Cardinality=$|R|$)
- *Key-Sharing Graph*: Let V represent all sensors. A Key-Sharing Graph $G(V,E)$ is constructed in the following manner: For any two sensor nodes i and j in V, there exists an edge between them if and only if (1) nodes i and j have at least one common key, and (2) nodes i and j can reach each other within the wireless transmission range, i.e., in a single hop.

4. Key Pre-Distribution Scheme

Using state of sensors modeled in the previous section, terms and notations, and assumptions, we propose a new random key pre-distribution scheme that satisfies all requirements listed in the previous section. Our proposed scheme consists of three phases: key pre-distribution phase, shared-key discovery, and path-key establishment.

1) Key Pre-Distribution Phase

This phase is performed off-line and before deployment. Key setup server estimates the probability of active-state at all time-intervals for each sensor using the information about all sensors like MAC protocol, sleep scheduler, job scheduler, and so on. Based on the estimation, it classifies all sensors into groups so that sensors more probable in active-state at the same time can share common session keys. We assume that L different groups are found while estimation.

After grouping of all sensors, key setup server generates a large global key pool S, and divides it into L group key pools S_i (for i =1,2,3,…,L), for group G_i. Two group key pools are neighbors if their corresponding time-intervals are previous or next. The purpose of setting up the group key pool S_i is to allow the neighbor groups to share more keys. We will describe the detail group key pool setup step later.

After completion of group key pool setup, for each sensor in the active-state group G_i, randomly selected R keys from its corresponding group key pool S_i and their indices are loaded into the memory of each sensor.

Because key assignments for sensors are determined by the probability of active-state, in some cases sensors may be in active-state even though they are not assumed to be. Therefore, all sensor should share keys with the other groups to communicate with others. Since we assume that the probability of active-state follows the Gaussian distribution, sensors are moderately probable to be in active-state at the previous and next time-interval. Therefore, each sensor should carry some portion of the group key pools from the previous and next time-interval.

2) Shared-Key Discovery Phase

After deployment, the state of each sensor is switching depending on the scheduler, events, and other variable factors at each time-interval. For secure communication with active-state node at given time-interval, each active node broadcasts a message containing the indices of the keys it carries. Each active node can use these broadcast messages to find out if there exists a common key it shares with the broadcasting node. If such a key exists, the active node uses this key to secure its communication channel with the broadcasting node. For disclosing the indices of keys each sensor carries, the challenge-response technique can be utilized to avoid sending the indices [2], that is for every key K_i on a key ring, each sensor can broadcast a list $(\alpha, E_{K_i}(\alpha))$ (i =1,…,k), where α is a challenge. By decrypting the $E_{K_i}(\alpha)$ with the proper key, a recipient can reveal the challenge α and establish a shared key with the broadcasting sensor. After above step, entire sensor network forms a key-sharing graph G.

3) Path-Key Establishment Phase

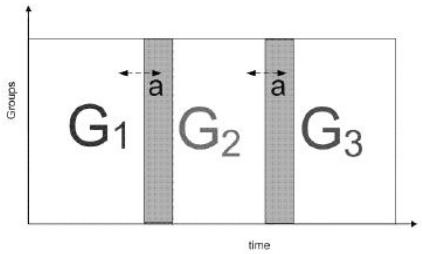
It is possible that two active sensors do not share a pre-distributed key. They should perform path-key establishment phase.

Suppose that i wants to share a session key with j, who do not share a common key each other. The idea is to use the secure channels that have already been established in the key-sharing graph G: as long as the graph is connected, two active nodes i and j can always find a path in G from i to j. Two sensors need to find an intermediate active-state sensor node that shares common keys with both of them to help establish a session key. Either of these two sensors may broadcast a request message with their own IDs. We assume that i sends this request. Suppose sensor v receives this request, and v shares a common key $K_{i,v}$ with i, and a common key $K_{j,v}$ with j. Sensor v then generates a random session key k and sends a message back to i, which contains $E_{K_{i,v}}(k)$ and $E_{K_{j,v}}(k)$. These are the session key k encrypted with $K_{i,v}$ and $K_{j,v}$, respectively. Upon receiving this reply message, node i can have the session key by decrypting $E_{K_{i,v}}(k)$, and inform sensor j by forwarding $E_{K_{j,v}}(k)$ to j.

4) Setting up Group Key Pools

We will show how to assign keys to each group key pool S_i, for i=1,2,3,$\cdots$,L, such that group key pools corresponding to nearby time-intervals have a certain number of common keys. We assume that a determines the certain number of common keys between two nearby time-interval groups. In our scheme, one group key pool shares exactly $a|S_G|$ with nearby time-interval group key pool($0 \leq a<1$). We call a an overlapping factor.

To achieve this property, we divide the keys in each group key pool into two partitions like illustrated in <Fig.2>. Keys in each partition are those keys that are shared between corresponding nearby time-interval group key pools. In <Fig.2>, the left partition of G_2 consists of $a|S_G|$ keys shared between G_1 and G_2.



<Fig. 2> Shared keys between nearby group key pools

Given the global key pool S and overlapping factor a, we now describe how to select keys for each group key pool S_i. First, keys for the first group key pool S_1 are selected from S; then remove $|S_G|$ keys from S. For each group key pool S_i (i=2,...,L), select $a|S_G|$ keys from group key pool S_{i-1}; then select $(1-a)|S_G|$ keys from the global key pool S, and remove the selected w keys from S. After group G_1 selects $a|S_G|$ keys from its nearby time-interval group G_2, no other group can select any one of the $a|S_G|$ keys. That is, these $a|S_G|$ keys are only shared between G_1 and G_2.

With above strategies, we can generate group key pool for each group. Then, now we calculate the number of keys in each group key pool. Since keys selected from the other groups are all distinct, the sum of all the number of keys should be equal to the $|S|$. Therefore, we have the following equation: $|S_G| = \frac{|S|}{L-aL+a}$ where L is the number of groups and a is the overlapping factor.

IV. Performance Analysis

In this section, we analyze our proposed scheme in detail. We present the probability that two sensors share a common key, and analyze our proposed scheme.

1. Connectivity

We calculate $p_{connect}$, the probability that two active-state sensors share at least one common key after deployment at given time-interval. Let A be the event that two sensors are in active-state at given time-interval, and B be the event that two sensors share at least one common key. Hence, $p_{connect} = P[B|A] = \frac{P[B \cap A]}{P[A]}$.

The probability that two sensors share at least one common key can be expressed as 1 - P[two sensors do not share any key]. Note that when the size of the key pool is $|S_G|$, the number of keys shared between two key pools is $\lambda|S_G|$, where the possible values of λ are 1, a, and 0.

According to the value of λ, we should consider three cases for finding the required probability. First, two sensors come from same group (λ=1). Second, two sensors come from

the nearby two groups (λ=a). Third, two sensors come from the different groups which are not neighbor each other. (λ=0).

Since we adopts the same overlapping key pool method used in [7], here we just briefly introduce the procedures and equations for calculation. The first node selects i keys from the $\lambda|S_G|$ shared keys, it then selects the remaining $R-i$ keys from the non-shared keys. To avoid sharing any key with the first sensor, the second sensor cannot select any of the i keys from those $\lambda|S_G|$ shared keys already selected by the first sensor, so it has to select R keys from the remaining $(|S_G|-i)$ keys from its key pool. Therefore, $p(\lambda)$, the probability that two sensors share at least one key when their key pools have $\lambda|S_G|$ keys in common, can be calculated as follow:

$$p(\lambda)=1-\frac{\sum_{i=0}^{\min(R,\lambda|S_G|)}\binom{\lambda|S_G|}{i}\binom{(1-\lambda)|S_G|}{R-i}\binom{|S_G|-i}{R}}{\binom{|S_G|}{R}} \quad (2)$$

Here, if λ=1, the above equation can be reduced as follow:

$$p(\lambda)=1-\frac{\binom{|S_G|-R}{R}}{\binom{|S_G|}{R}}$$

If λ=0, required probability is zero, $p(\lambda)=0$.

Then, we need to find out the probability that two sensors are in active-state at given time-interval. For this we need to consider two cases as follows:

- Case 1: two sensors are in same group during key pre-distribution phase.
- Case 2: two are in different group during key pre-distribution phase, and two groups are neighbors each other.

For each case, we can calculate the probability that two sensors are in active-state at given time-interval using Eq.(1).

Suppose that time-interval T_1 is given as $t_1 \le t \le t_2$. Then, the probability that G_1 is in active-state at given time-interval can be found as follow:

$$\begin{aligned} h(T_i) &= F(t_2)-F(t_1) \\ &= \Phi\left(\frac{t_2-t_m}{\rho}\right)-\Phi\left(\frac{t_1-t_m}{\rho}\right) \\ &= Q\left(\frac{t_2-t_m}{\rho}\right)-Q\left(\frac{t_1-t_m}{\rho}\right) \end{aligned}$$

where i(=1,2,3,$\cdots$) is the index of the time-interval, $F(x)$ is the cdf(cumulative distribution function) of Gaussian function, $\Phi(x)$ is the cdf of Gaussian function with m=0 and ρ=1, and $Q(x)$ is the Q-function.

For case 1, since two sensors are estimated to be in active-state at same time-interval, the probability of case 1 can be calculated simply as:

$p_{Case1}=h(T_x)^2$ where T_x is the given time-interval.

For case 2, we only need to consider the cases that two sensors are in active-state at the nearby time-interval. That is, if one sensor is active at T_x, the other may be in active at T_{x+1} or T_{x-1}. Then, the probability of case 2 can be simplified as follow:

$p_{Case2}=h(T_x)\times h(T_{x\pm1})$ where $T_{x\pm1}$ means the previous or next time-interval of T_x.

Then, we can define the probability that two sensors are in active-state as follow:

$$H(i,j)=\begin{cases} h(T_i)^2, & \text{if } i=j \\ h(T_i)\times h(T_{i+1}), & \text{if } i-j=1 \\ h(T_i)\times h(T_{i-1}), & \text{if } i-j=-1 \\ 0, & otherwise \end{cases} \quad (3)$$

Finally, we can now calculate $p_{connect}$ using Eq.(2) and (3).

We define Ψ as the set of all groups in our scheme. Suppose that two sensors, s_i and s_j, are selected from G_i and G_j (i,j=1,2,3,$\cdots$) of Ψ. Since the event that sensor $s_i(\in G_i)$ and sensor $s_j(\in G_j)$ share at least one common key is independent of the event that node s_i and node s_j are in active-state at given time-interval, we can calculate the probability that s_i and s_j are in active-state at given time-interval, and two sensors share at least one common key using Eq. (2) and (3) as:

$p(\lambda(i,j))\cdot H(i,j)$ (4)

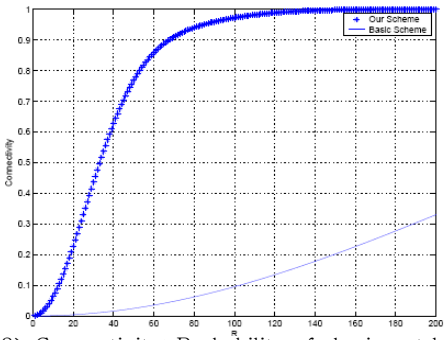
where $\lambda(i,j)$ is defined as follow:

$$\lambda(i,j)=\begin{cases} 1, & \text{if } i=j \\ a, & \text{if } |i-j|=1 \\ 0, & otherwise \end{cases}$$

Then, $p_{connect}$ is the average of the value in Eq.(4) for all groups, and can be calculated as follow:

$$p_{connect} = \frac{\sum_{i\in\Psi}\sum_{j\in\Psi} F(i,j) \cdot p(\lambda(i,j))}{\sum_{i\in\Psi}\sum_{j\in\Psi} F(i,j)}$$

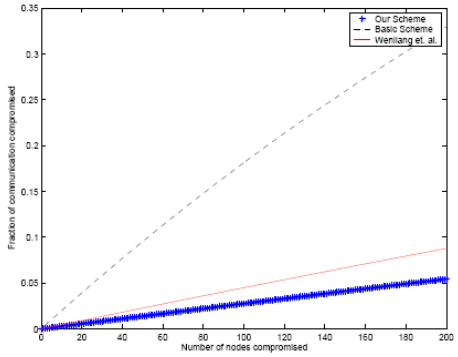
<Fig.3> illustrates the probability that two sensors share at least one common key when they are in active-state versus the number of keys each sensor carries under $|S|$ = 100,000, L = 100, and a = 0.25. <Fig.3> also illustrates the comparison between our proposed scheme and basic scheme [2]. To achieve same probability, our proposed scheme requires much smaller number of keys than basic scheme in [2].



<Fig.3> Connectivity: Probability of sharing at least one key between two sensor nodes

2. Resilience against Node Capture

To evaluate our key pre-distribution scheme against node capture, we apply the same method used in [7]. According to [7], the estimation of the expected fraction of total keys being compromised is calculated by $1-(1-\frac{R}{S})^x$ where x is the number of compromised nodes.



<Fig. 4> Resilience Against Node Capture when p=0.33

<Fig.4> illustrates the simulation results. We compare our scheme with existing random key pre-distribution schemes such as *basic scheme* in [2] and Wenliang et. al.'s scheme in [7]. The figure shows that our proposed scheme lowers the fraction of compromised communication after x nodes are compromised. The most important reason for this improvement is that, to achieve the same probability that two sensors share at least one common key while using the same key pool size S, our proposed scheme only requires much smaller R keys. For instance, to achieve p = 0.33 under S = 100,000, the *basic scheme* and Weliang et. al.'s scheme require R = 200 and 46, respectively. However, our scheme only needs R = 25. Carefully looking at the equation for find the fraction of communications compromised, we can find out the smaller value of R strengthens the networks against node capture. By adopting new deployment knowledge, we enable to reduce the number of unnecessary keys.

V. Conclusion

In this paper, we propose a new random key pre-distribution scheme that utilizes new pre-deployment knowledge, *state of sensors*. Using this, we can make keys be shared with sensors which are more probable in active-state at the same time. Therefore, we can achieve the higher connectivity with smaller number of keys compared to the previous schemes. Through this accomplishment, we can expect that sensor can save lots of memory usage and also improvement of resilience against node captures. We show the outstanding performance and security strength through the simulation.

[References]

[1] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, "SPINS: Security Protocols for Sensor Networks", The 7th Annual ACM International Conf. on Mobicom2001, Rome Italy, July 2001.

[2] Laurent Eschenauer, and Virgil D. Gligor, "A key-management scheme for distributed sensor networks", The 9th ACM Conf. on CCS02, Washington D.C., USA.

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

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

George W. Johnson Center at George Mason University, Fairfax, VA, USA. October, 2003
[5] D. Liu,and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks", The 10th ACM Conf. on CCS03, Washington D.C., October, 2003.
[6] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod Varshney, "A Pairwise Key Predistribution Scheme for Wireless Sensor Network", The 10th ACM Conf. on CCS 03, Washington D.C., October, 2003.
[7] Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen, and Pramod Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge", IEEE INFOCOM 04, Hong Kong. March, 2004