Uneven Key Pre-Distribution Scheme for Multi-Phase Wireless Sensor Networks

Onur Catakoglu and Albert Levi

Sabanci University, 34956, Orhanli, Tuzla, Istanbul, Turkey
catakoglu@sabanciuniv.edu, levi@sabanciuniv.edu

Abstract. In multi-phase Wireless Sensor Networks (WSNs), sensor nodes are redeployed periodically to replace nodes whose batteries are depleted. In order to keep the network resilient against node capture attacks across different deployment epochs, called *generations,* it is necessary to refresh the key pools from which cryptographic keys are distributed. In this paper, we propose Uneven Key Pre-Distribution (UKP) scheme that uses multiple different key pools at each generation. Our UKP scheme provides self healing that improves the resiliency of the network at a higher level as compared to an existing scheme in the literature. Moreover, our scheme provides perfect local and global connectivity. We conduct our simulations in mobile environment to see how our scheme performs under more realistic scenarios.

Keywords: wireless sensor network, key pre-distribution, multi-phase sensor network, resiliency, security

1 Introduction

Wireless Sensor Networks (WSNs) are used to carry wide range of data for various kinds of applications such as military, security, smart homes, telehealth, environmental observation and industry automation. Information that is transferred via those networks may contain not only temperature readings for habitat monitoring but also classified military data for battlefield surveillance which should not be seen by an unauthorized person. Therefore, security should be prioritized for these applications. WSNs have very limited resources in terms of memory and computational power. Hence, symmetric key cryptography is mostly used for existing key management schemes. However, pre-distribution of the symmetric keys effectively and efficiently in terms of resource usage are always been a challenge in WSNs.

An attacker can learn key rings that are inside of any node by corrupting the node and use these keys to compromise links between other sensor nodes. In Random Key Pre-distribution (RKP) scheme by Eschenauer and Gligor [1], an adversary corrupts sensor nodes of the network persistently (i.e. constant attacker) will eventually learn the whole key pool of the corresponding sensor network.

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under grant 110E180.

Most of the recent studies do not consider mobile environment i.e. they assume that sensor nodes are static. However, it is not a realistic assumption to make, because there are many types of applications in commercial, environmental and military studies such as housekeeping robots, service industry, wildlife tracking, patient tracking, autonomous deployment, shooter detection [3] which require a new network topology that takes mobility of nodes into consideration.

In this paper, we propose Uneven Key Pre-distribution (UKP) scheme for multiphase wireless sensor networks in mobile environment. The main idea of our method is the uneven pre-distribution of keys, which are taken from distinct key pools. At every deployment, newly deployed nodes will have their keys not only from the previous key pools, but also from a new distinct key pool. Therefore, keys in the network will be renewed at each redeployment phase and this will provide *self healing* to the network. Our results showed that we have better resiliency than RoK scheme without decreasing the local connectivity of network and without adding any additional memory overhead. Differently from the most of schemes, UKP uses multiple distinct key pools to refresh keys instead of using forward and backward hash operations for the sake of resiliency. In our scheme, hash operation is used only for creating a session key between two nodes from common keys.

The rest of the paper is organized as follows: Section 2 explains the related work on WSN security. Section 3 provides a comparative overview of our scheme and explains it in more detail. Section 4 presents the performance evaluations and Section 5 concludes the paper.

2 Related Work

Because public key cryptography is a very costly option for WSNs, most of the studies use symmetric cryptography. *RKP* is the most popular scheme that is proposed by Eschenauer and Gligor [1]. In this work, each wireless node picks keys from the same key pool before the deployment and if two nodes have at least one common key, they can establish a secure communication. However, a constant attacker eventually learns all the keys in the key pool and he can compromise the whole network [5].

Chan et al. [2] improved *RKP* scheme by using a threshold value, $q > 1$, for the number of common keys that are used for establishing a connection. Yet it requires more keys to be stored before the deployment which causes memory overhead, or it requires fewer keys in the key pool that leads to increase chance of same key being used more than once.

In RoK scheme [5], Castelluccia and Spognardi improved *RKP* scheme by using forward and backward hash chains to form a resilient network against node capture attacks. RoK scheme will be explained in detail at the third section.

There are some other works inspired by RoK that focus on multiphase networks. RPoK [6] is a polynomial-based *RKP* scheme proposed by Ito et al. for multiphase WSNs. Using private sub-key that is indirectly stored into every each node, they are able to establish a resilient network. Yi et al. [4] separates work time of the nodes into phases. They proposed a hash chain based scheme (HM scheme) for multiphase WSNs by using different key matrices for every phase.

3 Uneven Key Pre-Distribution Scheme

In this section, we firstly explain preliminaries and definitions that are used in explaining RoK [5] and our UKP schemes. Then, we overview the RoK [5] scheme and finally we explain the proposed UKP scheme in detail.

3.1 Preliminaries and Definitions

Notation used for RoK and UKP is explained in Table 1.

Table 1. Symbols used for RoK and UKP

Symbol	Explanation
A	Sensor A
n	Last generation of the network
FKP^j	Forward key pool at gen.
BKP^j	Backward key pool at gen.
P^j	Key pool of gen. j
m^j	Number of keys that are taken from key pool j
p_A^j	Number of keys that are taken from key pool j for node A
P	Key pool size
FKR_A^j	Forward key ring of A at gen.
BKR_A^j	Backward key ring of A at gen.
KR_A^j	Key ring of A that deployed at gen. j
G_w	Generation window
fk_t^j	t-th forward key at gen. j
bk_t^j	t-th backward key at gen. j
$k_{t_u}^j$	t_u -th key of P^j
k_{AB}	Common secret key between sensor A and B
$H(.)$	Secure hash function
m	Key ring size

Because sensor nodes are battery operated systems, they have to be redeployed periodically for the sake of connectivity of the network. These new nodes are assumed to be deployed at regular epochs which are called *generations*. Also, lifetime of a node is assumed to have an upper bound and it is determined by *generation window*, G_w. A newly deployed sensor node's battery at generation j will deplete before generation $j + G_w$.

In RoK [5], keys are updated and refreshed at the end of each phase. Therefore, two nodes which are from different generations can establish a secure channel with this update mechanism. UKP follows a different mechanism for that purpose. It is based on average age of nodes in the network i.e. keys are pre-distributed to a node according to its life time. Every sensor node from generation j can communicate with another sensor node from different generation in the range of $[j - G_w, j + G_w]$ as in the RoK. However in UKP, instead of taking m number of keys from a key pool, a node takes its keys from G_w number of key pools. In other words, our scheme pre-distributes the keys not just from the key pool of the current generation, but also from key pools of future generations.

3.2 Overview of RoK Scheme

In the RoK scheme [5], key pools evolve for each new generation and sensors update their key rings by hashing their keys. In other words, keys have lifetimes and they are refreshed when a new generation is deployed. This mechanism is achieved by using forward and backward hash chains. Each sensor node takes its keys from both forward and backward key pools, FKP and BKP, that are associated to its generation. Each key pool has $P/2$ random keys.

Forward key pool at generation j defined as $FKP^j = \{fk_1^j, fk_2^j, \dots fk_{P/2}^j\}$ where $fk_t^{j+1} = H(fk_t^j)$. Similiarly backward key pool at generation j will be $BKP^j = \{bk_1^j, bk_2^j, \dots bk_{P/2}^j\}$ where $bk_t^j = H(bk_t^{j+1})$. While a node at generation j takes its forward keys from FKP^j, it takes its backward keys from BKP^{j+G_w-1}. Therefore, key rings of the node will be formally represented as;

$FKR_A^j = \{fk_u^j | u = h(id_A \| i \| j), i = 1,2, \dots, {}^m/_2\}$ and

$BKR_A^j = \{bk_u^{j+G_w-1} | u = h(id_A \| i \| j), i = 1,2, \dots, {}^m/_2\}$

for forward and backward key ring respectively.

A sensor B deployed at generation i in the range of $[j - G_w, j + G_w]$ communicates with sensor A while their common keys' indices are $t_1, t_2, \dots, t_z$ respectively as follows.

while $i \leq j$,

$$k_{AB} = H(fk_{t_1}^j \| bk_{t_1}^{i+G_w-1} \| fk_{t_2}^j \| bk_{t_2}^{i+G_w-1} \| \dots fk_{t_z}^j \| bk_{t_z}^{i+G_w-1})$$

If two neighboring nodes have multiple shared common keys, all of them are used for the session key, k_{AB}. An adversary cannot compute keys from past generations by using forward keys, and cannot compute keys from future generations by using backward keys. Therefore, this mechanism provides forward and backward secrecy.

3.3 Proposed Uneven Key Pre-distribution Scheme

In this paper we propose Uneven Key Pre-Distribution (UKP) scheme for multiphase wireless sensors in mobile environment. The main idea of UKP is to distribute keys considering nodes' average life time statistic that is also represented in RoK [5] as shown in the Fig. 1. According to the statistic, when G_w = 10 and average life time of a node is $G_w/2 = 5$ with Gaussian distribution, most of the nodes in the network are newly deployed or young. After the age of four, the number of old nodes decreases dramatically.

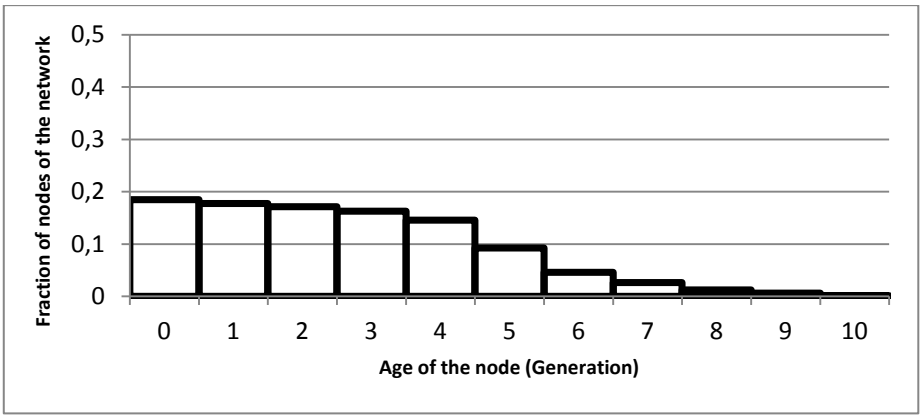

Fig. 1. Distribution of average age of the nodes.

Pools and Key Assignments. In UKP, there are n distinct pools that cannot be associated with each other. A sensor node takes its keys from G_w number of consecutive key pools in terms of generations. In order to decide how much to take from a key pool, we use the statistic shown in Fig. 1. In that case, a sensor will have the most keys from its descendant key pool and takes fewer and fewer keys from key pools that belong to further generations. For instance, a node at generation j takes its keys from $P^j, P^{j+1} \dots P^{j+G_w-1}$. Relationship between key counts is $m^j > m^{j+1} > \cdots > m^{j+G_w-1}$. Based on the statistic on Fig. 1, while the difference between m^j, m^{j+1} and $m^{j+2}$is smaller, difference between m^{j+4} and m^{j+5} is much greater and it will continue to grow in further generations until the generation window is reached. Hence, keys that are captured are only valid between $[j - G_w]^{\text{th}}$ and $[j + G_w]^{\text{th}}$ generations and this provides forward and backward secrecy. The key ring of node A, which is denoted as KR_A^j is composed of all the key sets coming from different generations of key pools as stated above. Similarly, if node B is at generation $j+1$, key ring of node B which is KR_B^{j+1} will take its keys from key pools $P^{j+1}, P^{j+2} \dots P^{j+G_w}$. Node B does not have any keys from P^j pool. In other words, if there are G_w number of differences in terms of generations between two nodes, these two nodes will not share any common keys. In this way, we provide self healing because compromised keys become outdated in time. We can represent key ring of a node A at generation j as follows.

$$KR_A^j = \begin{cases} k_{t_u}^j \,|\, u = 1,2, \dots m^j \\ k_{t_u}^{j+1} \,|\, u = 1,2, \dots m^{j+1} \\ \dots \\ k_{t_u}^{j+G_w-1} \,|\, u = 1,2, \dots m^{j+G_w-1} \end{cases}$$

where, $\mathrm{k}_{\mathrm{t_u}}^{\mathrm{i}}, i = j \dots j + G_w - 1$, are the keys selected from corresponding P^i using uniform random distribution with replacement.

The size of the key ring produced in this way, m, is calculated as follows.

$$m = m^j + m^{j+1} \dots + m^{j+G_w-1}$$

The purpose of having an uneven key distribution, i.e., using more keys from closer key pools in terms of generation is to achieve higher local connectivity in network. Moreover, this will strengthen the *self healing* property, since a compromised key has less chance to exist in further generations. In other words, most of the keys will be outdated sooner than the remaining ones and resiliency will be enhanced by the arrival of the new nodes with fresh keys.

Session Key Establishment. Any two nodes, say node $\boldsymbol{A}$ and node $\boldsymbol{B}$, can establish a session key only if they share at least one common key in their key rings. The session key is computed as the hash of all common keys that nodes $\boldsymbol{A}$ and $\boldsymbol{B}$ share. This key is denoted as $\boldsymbol{k_{AB}}$. The common keys used in session key establishment are chosen irrespective of the generations of keys. In other words, even if the two nodes come from different generations, they use all of the keys in their key rings to find common keys for session key establishment. Let us say that node $\boldsymbol{A}$ comes from generation $\boldsymbol{j}$, node $\boldsymbol{B}$ comes from generation $\boldsymbol{i}$ and the condition $\boldsymbol{i \leq j}$ holds for node generations. Then, if the common keys $\boldsymbol{A}$ and $\boldsymbol{B}$ share are denoted as

$$CK_{AB}^{x} = \{\forall\, k_v^x \,|\, k_v^x \in KR_A^j \,, k_v^x \in KR_B^i \; and \; k_v^x \in P^x \; for \; i \leq j \leq x\}$$

then, the session key is computed as follows.

$$\boldsymbol{k_{AB} = H\{CK_{AB}^{x} \,|\, x = j \ldots i + G_w - 1\}}$$

Example. As an example, if node A comes from generation j, node B comes from generation $j-2$ as shown in Fig. 2, and the set of common keys they share are $k_{t_1}^{j}$, $k_{t_2}^{j}$, $k_{t_1}^{j+1}$, $k_{t_2}^{j+1}$, $k_{t_9}^{j+4}$, $k_{t_3}^{j+5}$ and $k_{t_{11}}^{j+8}$, the session key is computed as follows.

$$k_{AB} = H(k_{t_1}^{j+2} \| k_{t_2}^{j} \| k_{t_1}^{j+1} \| k_{t_2}^{j+1} \| k_{t_9}^{j+4} \| k_{t_3}^{j+5} \| k_{t_{11}}^{j+8})$$

Again, the common keys coming from different key pools $(P^j, P^{j+1}, P^{j+4}, P^{j+5}, P^{j+8})$ and consequently different generations are used together to form the session key.

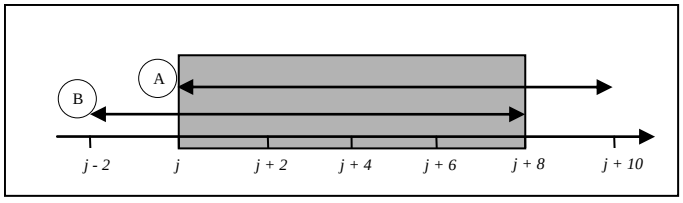

Fig. 2. Generation windows and overlapping generations of nodes A and B.

4 Performance Evaluation

We evaluated performance of our scheme with various simulations. In this section, we first explain performance metrics and then give simulation results together with the configuration and parameters.

4.1 Performance Metrics

Local connectivity is an important metric that shows the performance of the key distribution mechanism. It is defined as the probability of sharing a common key between two neighboring sensor nodes.

High local connectivity shows that a node can establish a secure communication with most of its neighbors. However, high local connectivity does not guarantee high global connectivity. Global connectivity is used to check if there are any nodes that are not reachable from the rest of the network. It is calculated as the ratio of the number of nodes in the largest isolated component to the number of nodes in the whole network.

In order to evaluate resiliency of the network we look at the ratio of additionally compromised links in the event of node capture. In other words, resiliency is computed as the number of indirectly corrupted channels divided by the number of all establishes links. We have better resiliency when the number is smaller. In our proposed scheme, we evaluated resiliency for active channels.

4.2 Evaluation by Simulation

For the sake of a fair comparison, we used similar setup as in the RoK [5] scheme for our simulation. The number of nodes in the network is taken as 1.000. We set the number of keys in each pool, P, as 10.000 and key ring size, m, as 500 for each node. Note that m value will be $m/2$ for forward and backward key rings of the RoK scheme. Generation window, G_w, is taken as 10 and we assume that a node's lifetime is determined according to a Gaussian distribution with mean $G_w/2$. and with standard deviation $G_w/6$. Deployment area is 500 x 500 meters and sensor node's wireless communication range is 40 meters. Nodes are distributed in that area with uniform random distribution. Speed of a node is decided randomly between 5-15 meters per minute. Note that, we assumed network topology does not change over time for the sake of simplicity. In other words, nodes whose lifetimes expired will be replaced with new ones. We take average of 25 runs to get more realistic results.

We developed our simulation code in C# using MS Visual Studio 2010. Simulations are conducted on a computer with 64-bit Windows 7 running on Intel Core i7-2600 CPU, 8.00 GB RAM.

Instead of using a static environment, we run our simulation with mobile nodes to expand the usage of UKP. The mobility models that we use are explained next. After that we explain the attacker and the simulation results.

Mobility Models. In order to simulate node mobility, we used two models: (a) random walk mobility model, (b) reference point group mobility model. These models are summarized below and detailed information can be found in [7].

In *Random Walk Mobility Model*, a sensor node chooses a direction and speed randomly using uniform distribution. Then it moves in that direction for a fixed amount of time, which is taken as 1 minute in our simulations. When it finishes its movement, this process repeats itself with new direction and speed. Past location and speed information are not stored, so no memory usage is needed.

Reference Point Group Mobility Model covers both groups' random movement and random movement of individual nodes inside a group. Each group moves based on a node that is chosen as central node. Individual nodes pick a reference point randomly around the central node and move as in the Random Walk Mobility Model .Reference points are updated with the movement of central node.

Attacker Model. We assume that attacker can learn keys of a node by capturing it. As stated in RoK scheme, if a forward key captured in generation j, it is possible to compute key with same index for generations after j and it is also possible for backward keys for generations before $j + G_w$. Because key pools in the UKP scheme are distinct, there is no such association between keys of different generations. However, the attacker learns all the keys in a captured node including keys that belong to further generations. In our model, an *eager* attacker captures nodes at constant rate at each round and attack will not stop until the end of the simulation.

Simulation Results. We computed local connectivity, global connectivity and resiliency performance of our UKP scheme in comparison with RoK scheme [5]. Both RoK and our UKP schemes under both mobility models have perfect local and global connectivity. In other words, every single node is reachable in the network and every node share at least one key with its neighbors.

For the evaluation of the network resiliency, we consider an attacker who captures sensor nodes with rates 1, 3 and 5 nodes per round (1 *generation* = 10 *rounds*). In our simulations, attacker starts compromising nodes at *generation* 5 in order to allow some time for network stabilization. Since keys in the network renewed by arrival of new nodes, rate of compromised keys sharply decreases at every new generation.

Figure 3 shows that in random walk model UKP scheme has almost same ratio of compromised keys under light attack (capture rate = 1). As attacker captures more nodes per round, our scheme outperforms RoK [5] model in terms of resiliency. Figure 4 also gives us similar results with reference point group mobility model. Compared to RoK, we have better results with capture rates 3 and 5.

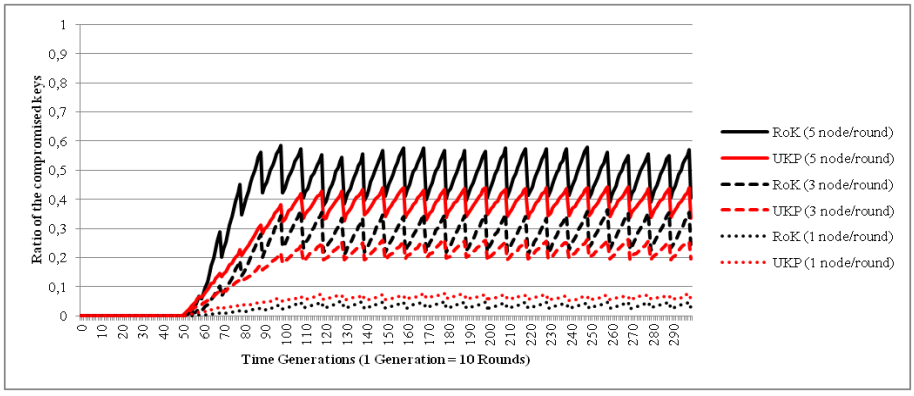

Fig. 3. Resiliency of RoK and UKP in case of an eager attacker with capture rates of 1, 3, and 5 nodes per round with Random Walk model

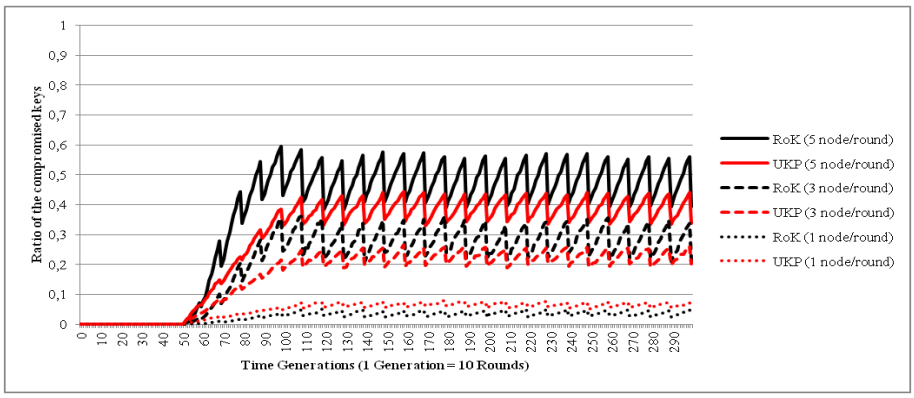

Fig. 4. Resiliency of RoK and UKP in case of an eager attacker with capture rates of 1, 3, and 5 nodes per round with Reference Point Group Mobility model.

5 Conclusions

In this paper, we proposed uneven key pre-distribution (UKP) scheme for multiphase wireless sensor networks in mobile environment. Our scheme is based on using different and distinct key pools at each generation. In this way, we improve the resiliency against heavy node capture attacks as compared to RoK scheme [5], while still maintaining perfect local and global connectivity.

References

1. Eschenauer, L. and Gligor, V. D.: A key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 41--47. Washington, DC, USA (2002)
2. Chan, H., Perrig, A., Song, D.: Random Key Predistribution Schemes for Sensor Networks. In: IEEE SP'03, pp. 197--213 (2003)
3. Amundson, I., Koutsoukos, X.D.: A Survey on Localization for Mobile Wireless Sensor Networks. In: Proceedings of the 2Nd International Conference on Mobile Entity Localization and Tracking in GPS-less Environments, pp. 235--254, Springer (2009)
4. Yi, S., Youngfeng, C., Liangrui, T.: A multi-phase key pre-distribution scheme based on hash chain. In: Fuzzy Systems and Knowledge Discovery, pp. 2061--2064, (2012)
5. Castelluccia, C. and Spognardi, A.: RoK: A robust key pre-distribution protocol for multi-phase wireless sensor networks. In: SecureComm2007, Third International Conference on Security and Privacy in Communication Networks. Baltimore, MD, USA (2007)
6. Ito, H., Miyaji, A., Omote, K.: RPoK: A Strongly Resilient Polynomial-Based Random Key Pre-Distribution Scheme for Multiphase Wireless Sensor Networks. In: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, pp.1--5. (2010)
7. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. In: Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483--502. (2002)