A Resilient Key Predistribution Scheme for Multi-Phase Wireless Sensor Networks

Murat Ergun, Albert Levi and Erkay Savaş
Sabancı University, Istanbul, Turkey
mergun@su.sabanciuniv.edu, levi@sabanciuniv.edu, erkays@sabanciuniv.edu

***Abstract*—In wireless sensor networks, sensor nodes eventually die due to battery depletion. Wireless Sensor Networks (WSNs) in which new nodes are periodically redeployed with certain intervals, called generations, to replace the dead nodes are called *multi-phase wireless sensor networks.* In the literature, there are several key predistribution schemes proposed for secure operation of WSNs. However, these schemes are designed for single phase networks which are not resilient against continuous node capture attacks; even under temporary attacks on the network, the harm caused by the attacker does not heal in time. However, the periodic deployments in multi-phase sensor networks could be utilized to improve the resiliency of the WSNs by deploying nodes with fresh keys. In the literature, there is limited work done in this area. In this paper, we propose a key predistribution scheme for multi-phase wireless sensor networks which is highly resilient under node capture attacks. In our scheme, called RGM (Random Generation Material) key predistribution scheme, each generation of deployment has its own random keying material and pairwise keys are established between node pairs of particular generations. These keys are specific to these generations. Therefore, a captured node cannot be abused to obtain keys of other generations. We compare the performance of our RGM scheme with a well-known multi-phase key predistribution scheme and showed that RGM achieves up to three-fold more resiliency. Even under heavy attacks, our scheme's resiliency performance is 50% better in steady state.**

Keywords—Multi-phase wireless sensor networks, security, key predistribution, generation keys.

I. Introduction and Related Work

Wireless Sensor Networks (WSNs) are composed of small devices called sensor nodes. Wireless sensor networks are well-suited for wide spectrum of purposes such as environment security, military tracking, medical and scientific experiments. Sensor nodes are powered via irreplaceable batteries; therefore, a particular sensor node can function as long as its battery is alive. However, WSNs should function for longer period of time compared to the lifetime of a sensor node. Therefore, as the nodes die, new nodes should be deployed in certain intervals, called *generations*, during operation of the network. This kind of WSNs is called *multi-phase wireless sensor networks*.

When sensor networks are deployed in critical environments, security becomes an important concern. In case of an existence of an attacker, sensor nodes can be captured and their keys can be obtained for message eavesdropping and injection purposes.

In wireless sensor network, security solutions based on CPU-efficient symmetric key cryptography are preferred. Moreover, in-network processing necessities require the symmetric keys to be distributed in node-to-node (i.e. link) basis.

In the literature, the problem of key distribution in WSNs is addressed by several probabilistic key predistribution schemes such as [3,5,6,7]. One of the first key management schemes using this approach is Eschenauer and Gligor's basic scheme [3]. Basic scheme is composed of three phases: key predistribution, shared key discovery, and path-key establishment phases. In key predistribution, for each sensor node τ keys are randomly drawn from a key pool of size P where $\tau << P$. Those τ keys form keyring in a node. Since all keys are drawn from the same pool, any two sensor nodes may keep a shared key with a probability less than 1. Shared key discovery phase starts after all sensor nodes are deployed and they discover neighbor nodes in their communication range. In this phase, all the nodes try to find a key shared between their neighbors. If there is such a key, it is used to secure communication between those two; otherwise, they run path-key establishment phase in which common secure neighbors help in key establishment. In the basic scheme, it is likely that a particular key exists in several nodes' keyrings. This is actually a must because otherwise the probability that a common key is found in shared key discovery phase, called *local connectivity*, reduces. However, having multiple copies of a key is also a potential security problem. An attacker can capture some nodes and acquire their keyrings. Established links secured by using the same keys in acquired keyrings are automatically compromised by the attacker. This weakens the security of the network.

Chan et al. [7] proposed another scheme to increase the resistance of basic scheme. Instead of using only one shared key, Chan et al. offered using as two or more shared keys. In this way, they achieve more durable system against attacks. This additional feature can lower connectivity value.

In [4], Blom proposed a multipurpose deterministic key pre-distribution scheme which uses single key space. Each node is able to calculate a pairwise key by storing only $\lambda+1$ keys in a network of size $N(\lambda << N)$. In this scheme, there is a property that an attacker cannot compromise any link unless no more than λ nodes have been captured. Besides, if λ nodes have been captured, whole system gets compromised. Du et al. [6] further improved Blom's scheme and transformed it to a general probabilistic case that is

This work is supported by Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant 104E071.

Murat Ergun is supported by TÜBİTAK BİDEB scholarship.

978-1-4244-5023-7/09/$25.00 ©2009 IEEE

directly applicable to WSNs. Du et al.'s scheme uses a multi-space approach. This scheme has similar phases with basic scheme.

In all random key predistribution schemes, there is a trade-off between local connectivity and resiliency against node capture attacks. Having a large keyring size increases the probability of direct key sharing (local connectivity), but this also gives more keys to the attacker when a node is captured.

Previously discussed systems are all designed for single-phase WSNs. Even if they allow dynamic node additions to the network, their key pools contain static keys that do not change in redeployments. As a result, if the network encounters a long term attack and new nodes are added to the system dynamically after this attack, they will be integrated to the network with some already compromised keys in their keyring. If the attacker continues his/her attack by capturing nodes and acquiring the keyring of captured nodes, he/she will eventually discover all of the key pool and the network would totally collapse. However, periodic redeployments in multi-phase sensor networks present an important opportunity to reduce the effect of an attacker. In each redeployment, a fresh set of keys may be deployed. So after a temporary attack, key pool can recover itself and remove the effects of the compromised keys. In addition to that, in case of a continuous attack, key pool can keep the rate of damage within a certain level. Here, the tradeoff is again at resiliency and local connectivity. Moreover, the connectivity among the nodes in different deployment generation should also be sustained. There is limited work done in the literature about key distribution is multi-phase sensor networks. One of them, RoK scheme [1], is explained in the next section in detail.

In this paper, we propose a novel random key predistribution scheme for multi-phase wireless sensor networks which is called RGM (Random Generation Material) scheme. In our RGM scheme, each generation of deployment has its own random keying material. During shared key discovery, unique pairwise keys are established between node pairs of particular generations. Here by uniqueness, we mean that nodes of other generation cannot know these keys. Therefore, a captured node cannot be used to obtain keys of other generations. This significantly improves the resiliency of RGM. We conducted simulative performance analyses and compared RGM scheme with RoK [1]. Our analyses show that RGM scheme is up to three-fold more resilient to node capture attacks as compared to RoK scheme. We also show that under heavy attacks, RoK scheme reveals 50% more secure link keys as compared to our RGM scheme. Moreover, our scheme provides 90% local connectivity, which is more than sufficient for a WSN.

The rest of this paper is organized as follows. Section II gives background information on key distribution in multi-phase WSNs. The RGM scheme, our contribution, is explained in Section III. Section IV discusses the comparative performance evaluation. Section V concludes the paper.

II. Key Distribution in Multi-phase Wireless Sensor Networks

Sensor nodes operate using battery power that eventually depletes. Wireless sensor networks are set up to function for longer period of time as compared to the lifetime of sensor nodes. So, new nodes need to be deployed in some intervals to provide continuity of network in *multi-phase sensor networks*. These intervals are called *generations*. In the beginning of each generation, dead nodes are replaced by new nodes.

Castelluccia and Spognardi [1] proposed a key management scheme called Robust Key Distribution (RoK) for multi-phase wireless sensor networks, in which predistributed keys have limited lifetimes. This is achieved by refreshing key pools for each generation of deployment. Refreshed key pools allow a network that is temporarily attacked to be self-healed in time.

If key pool is refreshed with random keys in each deployment, attacker cannot guess the upcoming pool by knowing previous keys or cannot learn former pools by knowing current one. But in the same way, sensor nodes deployed at different generations cannot establish secure links. In order to achieve connectivity between nodes belonging to different generations, there should be some kind of relation between key pools at different generations.

TABLE 1. Symbols used in RoK and RGM

FKP^j	forward key pool at generation j
BKP^j	backward key pool at generation j
FKR_A^j	forward keyring of node A deployed at generation j
BKR_A^j	backward keyring of node A deployed at generation j
GKR_A^g	generation keyring of node A deployed at generation g
GKR_A^{fg}	generation sub-keyring of node A deployed at generation g containing keys used to establish link with nodes deployed at generation f
fk_t^j	forward key with index t at generation j
bk_t^j	backward with index t at generation j
gk_t^{fg}	generation key with index t between generations f and g
k_{AB}	link key between nodes A and B
G_w	Generation window
$h(\cdot)$	secure hash function
$f(\cdot)$	hash function
$H(\cdot)$	secure hash function padded with key
m	number of current generation keys in a generation keyring
n	number of future generation keys in a generation keyring for each next generation

RoK uses two key pools: forward and backward key pools, *FKP* and *BKP*. In order to provide connectivity between different generations, *FKP* is updated by hashing keys of previous generation and *BKP* is updated using Lamport hash chain [2].

Table 1 gives the symbols used in the explanations of RoK scheme. The same symbol table will be referred for the

explanation of our proposed RGM scheme, which will be given in the next section.

$FKP^j = \{fk_1^j, fk_2^j, ..., fk_P^j\}$ is the forward key pool at generation j where P is the pool size.

$FKP^{j+1} = \{fk_1^{j+1}, fk_2^{j+1}, ..., fk_P^{j+1}\}$ is the forward key pool at generation $j+1$ where $fk_t^{j+1} = h(fk_t^j)$.

$BKP^j = \{bk_1^j, bk_2^j, ..., bk_P^j\}$ is the backward key pool at generation j where P is the pool size.

$BKP^{j+1} = \{bk_1^{j+1}, bk_2^{j+1}, ..., bk_P^{j+1}\}$ is the backward key pool at generation $j+1$ where $bk_t^j = h(bk_t^{j+1})$.

It is assumed that each node has an upper bound of lifetime and this upper-bound defines generation window, G_w, which is a system parameter. A node may live at most as long as generation window. A node A deployed at generation j is given two keyrings, forward keyring and backward keyring. Forward keyring, FKR_A^j, consists of forward keys of generation j drawn randomly from forward key pool at generation j, FKP^j. Similarly, backward keyring, BKR_A^j, consists of keys of $j + G_w - 1$ drawn randomly from backward key pool at generation $j + G_w - 1$, BKP^{j+G_w-1}. These keyrings are formally shown below.

$$FKR_A^j = \{fk_u^j \mid u = f(id_A \| i \| j), i = 1,2,..., m/2\}$$

$$BKR_A^j = \{bk_u^{j+G_w-1} \mid u = f(id_A \| i \| j), i = 1,2,..., m/2\}$$

Node A can produce a forward key fk_u^f where $f > j$ and backward key bk_u^b where $b < j + G_w - 1$. Each node, deployed at generation j, have certain probability to share a common key with another node B which is deployed at generation i, where i is in interval $]j - G_w, j + G_w[$. The generations between which two nodes can produce the same forward and backward keys are called *overlapping generations*. Let's suppose $i \le j$, then their overlapping generations would be between j and $i + G_w - 1$.If nodes A and B have common keys of indices $t_1, t_2, ..., t_z$, they compute their link key as the following:

$k_{AB} = h(fk_{t_1}^j \| bk_{t_1}^{i+G_w-1} \| fk_{t_2}^j \| bk_{t_2}^{i+G_w-1} \| ... \| fk_{t_z}^j \| bk_{t_z}^{i+G_w-1})$, where $i \le j$

Forward keys provide forward secrecy since the attacker cannot learn previous keys even if it learns a forward key at a generation. Similarly, backward keys provide backward secrecy since the attacker cannot learn future keys even if it learns a backward key at a generation. When an attacker learns some forward and backward keys by capturing a sensor node, previous forward key are not revealed since a forward key is calculated from previous forward key by a one-way function. Similarly, future backward keys are also protected. Similarly regular sensor nodes cannot find out these previous forward keys and future backward keys even if they keep keys of same index in their keyrings. This property provides a lifetime to the keyring. The lifetime of a keyring also limits the capability of an attacker. He/she can use a compromised keyring for a short period of time. Since the keyrings have limited lifetime and key pools are refreshed periodically, compromised keys expire like all the other keys as time passes. In this way, network gradually removes the traces of an attack and heals itself. If this attack is a temporary type, in a certain time network comes to the state before the attack has started. if it is a permanent type of attack, network can keep the ratio of corrupted links within a certain limit.

III. Our Contribution

In this section, the proposed RGM (Random Generation Material) key predistribution scheme for multi-phase WSNs is designed.

A. Overview

In our RGM method, the concept of overlapping generations is not used. Instead of forward and backward keyrings, one keyring, called the *generation keyring*, is used. In contrast to *RoK*, generation keys are generated randomly by a distribution center at each generation. Generation keys evolve in a different way, independent of evolution of generation key pool. This will be discussed in detail below.

Nodes will be loaded with randomly selected *m* generation keys prior to deployment in order to establish secure communication with other nodes in same generation. Keys in generation keyring are loaded to be used generation-wise.

In our scheme, generation key pool randomly refreshes in time. Therefore, there is no relation between past and future states of the pool in our scheme. This property is very important in order to limit attacker's capability. On the other hand, we need to provide connectivity between nodes deployed at different generations. Connectivity will be obtained by evolving not the generation key pool itself, but evolving generation keys individually.

Another improvement of our method is that generation keys are distributed to be used generation-wise. This has an advantage over RoK [1] in restricting the information an attacker acquires if he/she captures a node. In such a case, he/she can compromise the generation keys used between generation at which captured node has been deployed and other generations within the generation window. However, if the attacker captures a node at generation *j*, the nodes of other generations do not get affected even if *j* is between their deployment generations. Hence if the attacker wants to compromise a predefined link indirectly, he/she must compromise nodes which have generation keys in their keyring with the same shared indices and have been deployed at the same generations with one of the ends of that predefined link. What our method tries to do is to increase the number of constraints that an attacker should obey. In this way, we improve the resiliency of the WSN.

B. Predistribution of Generation Material

In multi-phase WSNs, each node should be able to establish secure links with its neighbor nodes in all

generations inside its generation window. In RGM, nodes use generation-wise keys to produce link keys. That is why they should be able to access generation-wise keying material of those generations. If a node stores this information for all generations in its generation window, it brings extra burden to the memory capacity of sensor nodes. Fortunately in RGM, a particular node *A* only needs to store generation keys to be used in establishing links with nodes deployed at future generations and current generation key which is used to establish links between node *A*, and other nodes deployed at the same generation. Node *A* can produce past generation keys (generation keys to be used in establishing links between node *A* and other nodes deployed at previous generations) by itself, when needed.

A node can simply generate past generation keys by concatenating current generation key to a well known plain text such as all zeros and apply a secure hashing algorithm like *SHA-1* or *SHA-256* depending on the key size. This is the case just for producing the closest past generation key. The past generation keys after the first past generation key are calculated by concatenating current generation key to previous past generation key and applying secure hashing algorithm to this input. The reason behind always concatenating current generation key before applying secure hashing algorithm is to make it very hard to calculate the next past generation key if current generation key is not known. While current generation keys of various indices belonging to generation, let's say, *g,* is known by only sensor nodes deployed at generation *g* again, past generation keys of generation *g* can be calculated by none of the sensor nodes, but nodes deployed at generation *g*. Past generation keys of generation *g* which are used to establish secure links between nodes deployed at generations *g* and $g-1$, and *g* and $g-2$ are calculated as follows.

$$gk_v^{g(g-1)} = H\left(00...0 \,||\, gk_v^{gg}\right)$$

$$gk_v^{g(g-2)} = H\left(gk_v^{g(g-1)} \,||\, gk_v^{gg}\right)$$

In general, past generation key of generation *g* which are used to establish secure links between nodes deployed at generations *g* and $g-i$ are calculated as follows.

$$gk_v^{g(g-i)} = H\left(gk_v^{g(g-i+1)} \,||\, \cdots \,||\, gk_v^{g(g-1)} \,||\, gk_v^{gg}\right)$$

Generation keyring of a sensor node *A* deployed at generation *j*, is split into sub-divisions. One of the sub-divisions is reserved for current generation keys as shown below.

$$GKR_A^{jj} = \left\{gk_u^{jj} \mid u = f\left(id_A \,||\, i \,||\, j \,||\, j\right), i = 1,2,...,m\right\}$$

Other sub-divisions containing random generation keys used to establish secure links with future generations are explained below. Generation keyring sub-division containing random keys used to secure communication with nodes deployed at next generation is given as follows for node *A* deployed at generation *j*.

$$GKR_A^{j(j+1)} = \left\{gk_u^{j(j+1)} \mid u = f\left(id_A \,||\, i \,||\, j \,||\, (j+1)\right), i = 1,2,...,n\right\}$$

The keyring for secure communication with nodes deployed at second next generation is as follows.

$$GKR_A^{j(j+2)} = \left\{gk_u^{j(j+2)} \mid u = f\left(id_A \,||\, i \,||\, j \,||\, (j+2)\right), i = 1,2,...,n\right\}$$

Continuing in this manner, the keyring for secure communication with nodes deployed at $G_w - 1^{\text{st}}$ next generation is given as follows.

$$GKR_A^{j(j+G_w-1)} = \left\{gk_u^{j(j+G_w-1)} \mid u = f\left(id_A \,||\, i \,||\, j \,||\, (j+G_w-1)\right), i = 1,2,...,n\right\}$$

To sum up, generation keyring of a sensor node *A* deployed at generation *j* is as follows.

$$GKR_A^{(j-1)} = \left\{\begin{array}{l} gk_{t_{0,1}}^{jj}, gk_{t_{0,2}}^{jj}, gk_{t_{0,3}}^{jj}, ..., gk_{t_{0,m}}^{jj}, \\ gk_{t_{1,1}}^{j(j+1)}, gk_{t_{1,2}}^{j(j+1)}, gk_{t_{1,3}}^{j(j+1)}, ..., gk_{t_{1,n}}^{j(j+1)}, \\ gk_{t_{2,1}}^{j(j+2)}, gk_{t_{2,2}}^{j(j+2)}, gk_{t_{2,3}}^{j(j+2)}, ..., gk_{t_{2,n}}^{j(j+2)}, \\ \vdots \\ gk_{t_{G_w-1,1}}^{j(j+G_w-1)}, gk_{t_{G_w-1,2}}^{j(j+G_w-1)}, gk_{t_{G_w-1,3}}^{j(j+G_w-1)}, ..., gk_{t_{G_w-1,n}}^{j(j+G_w-1)} \end{array}\right\}$$

In RGM, sensor nodes store *m* current generation keys and *n* future generation keys for each upcoming generation. Because a sensor node may communicate with at most $G_w - 1^{\text{st}}$ next generation, the total amount of generation keys in a keyring is $(G_w - 1) * n + m$.

C. Calculation of Link Keys

Suppose nodes *A* and *B* are deployed at generations *f* and *g* respectively, where $f < g$. Past generation key gk_v^{fg} with index *v*, which is used to secure communication between *A* and *B*, can be computed if and only if a node has current generation key gk_v^{gg} with index *v* belonging to its own generation. As a result, a sensor node, which is deployed at generation *e* where $f < e < g$, cannot compute generation key gk_v^{fg} even if it has generation key gk_v^{eg} with index *v*.

It is clear that a generation key gk_v^{fg} with index *v* may be known only by nodes of generations *f* and *g*. No such node deployed at another generation can compute the key that is unique to generation *f* and *g*. So an attacker has to waste extra effort and compromise the nodes belonging to generations *f* or *g* if she wants to acquire the link key between nodes *A* and *B*.

In our RGM scheme, as in RoK scheme, all the shared generation keys contribute to the link key. Contribution of as many keys as possible increases the resistance of link against attacks. Let's suppose, node *A* deployed at generation *f* and another node *B* deployed at generation *g* have common generation keys with indices $v_1, v_2, ..., v_z$. They compute their link key as follows:

$k_{AB} = h\left(gk_{v_1}^{fg} \,||\, gk_{v_2}^{fg} \,||\, ... \,||\, gk_{v_z}^{fg}\right)$ where $f < g$.

Node *B* can produce generation keys used generation-wise between generations *f* and *g* from current generation key gk_t^{gg} using the mechanism explained in the previous subsection. On the other hand, node *A* cannot produce generation keys, so those generation keys are preloaded to *A* before deployment again as explained in previous subsection.

IV. PERFORMANCE EVALUATION

We performed various simulations to compare the performances of RoK [1] and the proposed RGM methods. In the simulations key pool size is set to 10000 keys, and memory size is set to 1100 keys for RGM and 500 keys for RoK scheme. The reason of using less keys for RoK is that RoK reaches almost 1.0 local connectivity when 500 keys are utilized. Using more keys does not further improve the performance of RoK; on the contrary, using more keys reduces the resiliency since in case of a node capture attacker unnecessarily learns more keys.

RoK utilizes memory by dedicating half of the memory to forward keyring, and the other half to backward keyring. In our simulations, each keyring has 250 keys. RGM scheme has different *m* and *n* values for current generation sub-keyring and future generation sub-keyrings. For a memory size of 1100 keys, *m* value is set to 200 and *n* value is set to 100. There are $G_w - 1$ future generation sub-keyrings and 100 future generation keys are predistributed for each upcoming generation.

In our simulations, as in [1], sensor network is composed of a square grid of sensors, each node having exactly 4 neighbors. There are 400 sensor nodes on this square grid. Generation window (G_w) is set to 10. Sensor nodes have a random lifetime assigned according to Gaussian distribution with mean $G_w/2$, and standard deviation $G_w/6$. As in [1], dead nodes are replaced with new nodes immediately in the beginning of each generation. Simulations are run along 50 generations. All simulations are run 25 times and their average values are shown in the figures.

Our attack model assumes an attacker who can randomly capture nodes at random locations. In the simulations, the attacker's capture rate is taken as 1, 3 and 5 nodes per round. In our scheme, *round* is the time unit and one generation consists of 10 rounds.

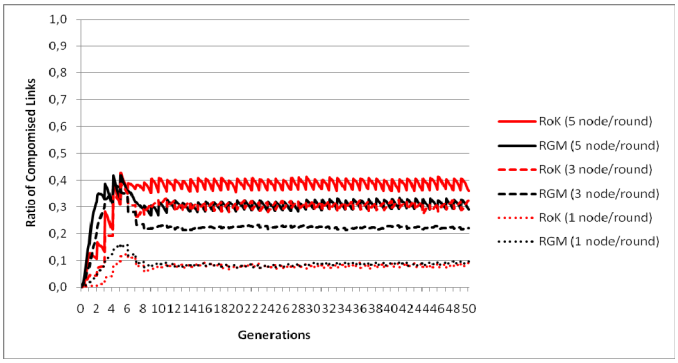

Figure 1. Active 1-resiliency of RoK and RGM in case of an eager attacker with capture rates of 1, 3, and 5 nodes per round

Figures 1 and 2 show 1-resiliency of the sensor network against an eager attacker. 1-resiliency is defined as the fraction of indirectly compromised links. That is, if this ratio is low, the network is more resilient. Indirectly compromised link is a link whose keys are known by the attacker, but none of the sensors in both ends is compromised. Eager attacker identifies an attacker who starts his/her activity from the beginning of network to the end. Active resiliency is the resiliency due to current alive links. A link is said to be alive if both ends are alive. In other words active 1-resiliency is defined as the ratio of compromised alive links to all established alive links.

As can be seen from Figure 1, active 1-resiliency reaches its highest value in around generation 5, when most of nodes deployed at the initial phase are alive. After this time on, first nodes start to die, and resiliency stabilizes together with arrival of new nodes. Our results show that in the steady state our RGM scheme performs 35% - 50% better than RoK scheme at capture rates of 3 and 5 nodes per round. At capture rate of 1 node per round, in the steady state both schemes perform equally.

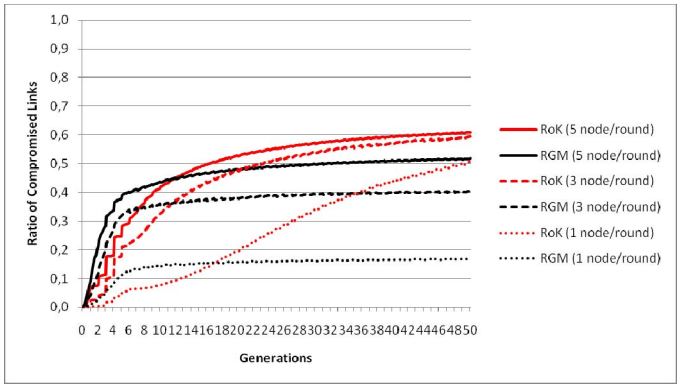

Figure 2. Total resiliency of RoK and RGM in case of an eager attacker with capture rates of 1, 3, and 5 per round

Figure 2 shows total 1-resiliency values of RoK and our RGM schemes. Total 1-resiliency is the ratio of the all indirectly compromised dead or alive links over all established links since the beginning of the network. Total 1-resiliency is important if the attacker keeps log of communications in the network. When a link is compromised even if its endpoints are dead, an attacker may have a look at the logged messages and learn the contents. Of course, the content of the message may not be as valuable as when it was sent. As can be seen from Figure 2, in steady-state (i.e. after around 10th generation) RGM performs better than RoK in all capture rates. In RoK, the 1-resiliency values tend converges to around 0.6 in all capture rates, meaning that 60% of the links are compromised. However, in our RGM scheme, even if the network is attacked with the highest rate of 5 nodes per round, the 1-resiliency value barely reaches 0.5. For lower capture rates, performance improvement of RGM over RoK is clearer. When the capture rate is 1 node per round, at the end of the simulations the number of links that the attacker captures is approximately three-fold more in RoK scheme as compared to our RGM.

Figure 3 comparatively shows 1-resiliency of RoK and RGM in case of a temporary attacker who starts his/her activity in generation 5 and ends in generation 14. When the attack starts, ratio of compromised links raises up to an upper-bound which is different for all corruption rates and after attack stops, network starts to heal itself As can be seen from Figure 3, both schemes completely heals (i.e. ratio of compromised links come to zero) almost at the same time

(generation 22). Figure 3 also shows that during the attack, RGM performs as much as 35% better than RoK for capture rates of 3 and 5 nodes per round.

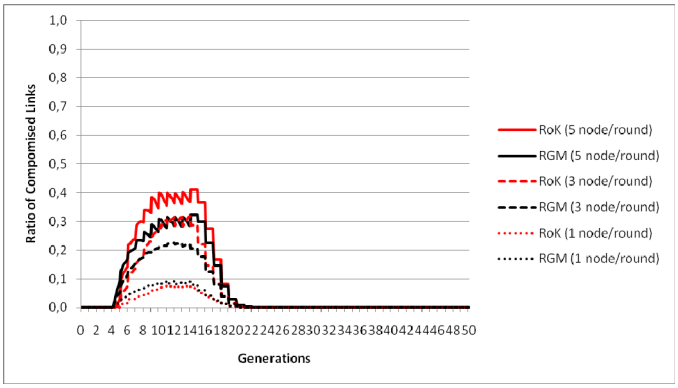

Figure 3. Active resiliency of RoK and RGM in case of a temporary attacker with capture rates of 1, 3, and 5 per round

One drawback of RGM scheme is that *local connectivity* of RGM tends to be less than in RoK. Local connectivity is defined as the probability of any two neighbors to share at least one common key. The local connectivity values of RoK and RGM for the cases that we analyze are given in Figure 4. As can be seen from this figure, in RGM neighboring nodes share at least one key with a probability of around 0.9, while in RoK this value is 1.0. That means our scheme performs 10% worse than RoK. However, this does not mean that 10% of the nodes are completely disjointed from the network. 0.9 local connectivity means 10% of all possible links are insecure. However, the end points of these insecure links have at least one other secure neighbor and via these secure neighbors they are connected to the network securely. That is why 90% local connectivity, as in RGM scheme, is considered more than enough for the secure operation of the network.

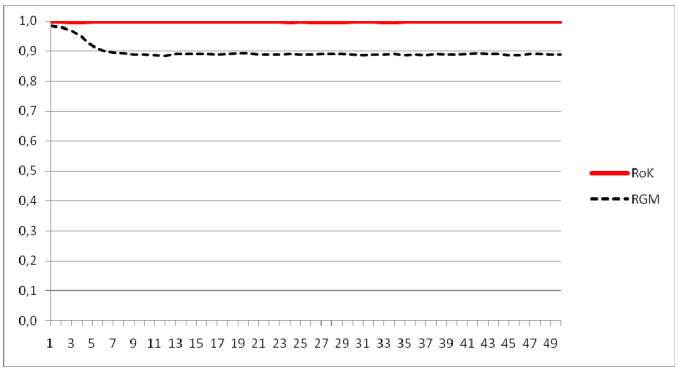

Figure 4. Local connectivity of RoK and RGM

Another relative drawback of RGM scheme is its higher memory requirements as compared to RoK scheme. However, as analyzed below, the amount of key memory that RGM uses covers only a small portion of the data memory of the state-of-the-art sensor nodes. The simulations results reported in this section require 1100 keys per node for RGM and 500 keys per node for RoK. Assuming that 128-bit keys are used and another 16 bits are employed for key identification, the amount of memory used for RGM and RoK are 19800 and 9000 bytes, respectively. In parallel with latest advances in sensor node technology, storage capabilities are increased. For example, MICAz and IRIS (http://www.xbow.com/) have 128 Kbytes of flash memory that can be used for key storage. The flash memory capacity of TinyNode584 (http://www.tinynode.com/) is 512 Kbytes. In our RGM scheme, the memory requirement for keys is less than 20 Kbytes that comprises of a small portion of the flash memory of the sensor nodes.

V. Conclusions

In this paper we proposed a random key predistribution scheme, called RGM, for multi-phase wireless sensor networks. In our RGM scheme, each redeployed node comes with a refreshed set of generation keys so that capture of a node that belong to a particular generation has minimal effect on the keys between nodes of other generations. In this way, the value of the information learned by the attacker is reduced and, therefore, the resiliency of the network is improved. Our scheme also takes care of the cryptographic connectivity of the newly deployed nodes with their physical neighbors. The simulative performance analyses show that our scheme performs approx. 50% better active resiliency under heavy attacks as compared to well-known RoK scheme [1] with the cost of 10% degradation in cryptographic connectivity. Thus, our scheme provides a good tradeoff in favor of security. It is also worthwhile to mention that the connectivity value of 0.9 does not make any node disconnected from the network. The network still operates securely with this rate.

References

[1] C. Castelluccia and A. Spognardi, "RoK: A robust key pre-distribution protocol for multi-phase wireless sensor networks," SecureComm 2007 – 3rd International Conference on Security and Privacy in Communications Networks, 2007.

[2] L. Lamport, "Password authentication with insecure communication," Commun. ACM, vol. 24, no. 11, 1981.

[3] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," In 9th ACM conference on Computer and Communications Security (ACM CCS), 2002.

[4] R. Blom, "An optimal class of symmetric key generation systems.," In Eurocrypt 84.

[5] W. Du, J. Deng, Y. Han, S. Chen and P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," In IEEE Infocom'04.

[6] W. Du, J. Deng, Y. Han and P. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," In Proceedings of the 10th ACM conference on Computer and Communications Security (CCS), 2003.

[7] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in Proceedings of the 2003 IEEE Symposium on Security and Privacy.