A Highly Resilient and Zone-based Key Predistribution Protocol for Multiphase Wireless Sensor Networks

Kubra Kalkan Sinem Yilmaz
Sabanci University
Istanbul, Turkey
kubrakalkan@su.sabanciuniv.edu
sinemyilmaz@su.sabanciuniv.edu

Omer Zekvan Yilmaz
TUBITAK UEKAE
Gebze, Kocaeli, Turkey
zekvan@uekae.tubitak.gov.tr

Albert Levi
Sabanci University
Istanbul, Turkey
levi@sabanciuniv.edu

ABSTRACT
Pairwise key distribution among the sensor nodes is an essential problem for providing security in Wireless Sensor Networks (WSNs). The common approach for this problem is random key predistribution, which suffers from resiliency issues in case of node captures by adversaries. In the literature, the resiliency problem is addressed by zone-based deployment models that use prior deployment knowledge. Another remedy in the literature, which is for multiphase WSNs, aims to provide self-healing property via periodic deployments of sensor nodes with fresh keys over the sensor field. However, to the best of our knowledge, these two approaches have never been combined before in the literature. In this paper, we propose a zone-based key predistribution approach for multiphase WSNs. Our approach combines the best parts of these approaches and provides self-healing property with up to 6-fold more resiliency as compared to an existing scheme. Moreover, our scheme ensures almost 100% secure connectivity, which means a sensor node shares at least one key with almost all of its neighbors.

Categories and Subject Descriptors
C.2.0 [**Computer-Communication Networks**]: General – *Security and protection (e.g., firewalls).*

General Terms
Security

Keywords
Multiphase Sensor Networks; Sensor Network Security; Node Capture Attacks; Key Distribution; Resiliency

1. INTRODUCTION

Wireless Sensor Networks (WSNs) [1] consist of small, battery-operated, limited memory and limited computational power devices called *sensor nodes*. The main task of a WSN is to sense some events and carry these readings to a base station, called the *sink*. The application areas of WSNs vary from military applications to agriculture, habitat monitoring and healthcare. In several applications, security of the data transferred within the WSN is of utmost importance. When WSN deployed in hostile areas, there is a possibility that the nodes can be captured by adversaries. Each captured node gives away information about the network. Thus, the security mechanisms must be designed by considering such attack scenarios.

A key requirement of a security design for a WSN is pairwise key distribution among the neighboring sensor nodes. In the literature, the most cited approach is Eschenauer and Gligor's (EG) Random Key Predistribution approach [4]. In this approach, a set of random keys selected from a global key pool is predistributed into sensor nodes' key rings in a redundant way. After the deployment, neighboring nodes share at least one common key with a certain probability. This probability is called *secure connectivity*. More keys predistributed into sensor nodes increase the chance of having a common key between neighboring nodes (i.e. increase secure connectivity). However, in case of a capture of a node, more keys are revealed to the adversary. Therefore, there is tradeoff between secure connectivity and resiliency against node captures.

In order to increase the resiliency of EG scheme without reducing secure connectivity, Du et al. proposed Zone-based Random Key Predistribution (Zo-RKP) using deployment knowledge in key predistribution [2]. In this method, sensor field is divided into zones and nodes that are to be deployed over these zones are grouped in batches. Each zone has its own key pool and the key pools of neighboring zones share keys. Before the deployment, the nodes of each group are stored random keys that are selected from the corresponding zone's key pool. Since the nodes of a particular zone are likely to be neighbors after the deployment, same level of secure connectivity is achieved by using less number of keys per node as compared to EG scheme. Since the nodes need to store less number of keys in their key rings, less information is revealed to an attacker in case of node captures. Therefore, the resiliency increases.

Another important work in the literature that aims increase the network resiliency without reducing secure connectivity is proposed by Castelluccia and Spognardi [3] for multiphase sensor networks. In multiphase sensor networks, the sensor nodes are periodically redeployed as their batteries are depleted. In the scheme proposed in [3], called robust key pre-distribution (RoK), the keys are refreshed in each redeployment so that the keys that are compromised by the adversary become useless in time. In this way, the network heals itself.

In this paper, we propose Zo-RoK, Zone-based Robust Key Distribution. In Zo-RoK, we combine RoK [3] and Zo-RKP [2]

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

Q2SWinet'09, October 28–29, 2009, Tenerife, Canary Islands, Spain.

schemes in order to boost up the resiliency performance of WSNs while keeping the network securely connected using less number of keys per node. We adapt the zone-based random key predistribution model of Zo-RKP scheme into the multiphase deployment model of RoK scheme. In this way, 100% secure connectivity is achieved with small amount of keys per node. This, in turn, reduces the number of secure link keys to be compromised by the adversary. Moreover, due to key refresh at each redeployment, the usefulness of the compromised keys becomes limited in time. Our performance analyses show that our Zo-RoK scheme is as much as 6 times more resilient to node capture attacks as compared to RoK with the same level of secure connectivity. Moreover, Zo-RoK saves 70% of key memory space as compared to RoK scheme.

The rest of the paper is organized as follows. We explain the proposed model in Section 2. Performance evaluation is given in Section 3. In section 4, related work is presented and finally Section 5 concludes the paper.

2. THE PROPOSED MODEL

In this section, we explain the proposed Zo-RoK (Zone-based Robust Key Distribution) scheme. First, we give some background information about Du et al.'s zone-based deployment model [2] and the RoK scheme [3] on which our proposed scheme is built on. After that, we explain the random key predistribution and the pairwise key generation phases of our Zo-RoK scheme.

2.1 Background Information

In our Zo-RoK scheme, the main purpose is increased resiliency with while keeping (i) the self-healing property of RoK, and (ii) high secure connectivity rates as in RoK again. Resiliency is the endurance of the system to node capture attacks. Secure connectivity is the probability of two neighboring nodes to share a common key. In random key predistribution schemes, such as [2], [3] and [4], in order to reach high secure connectivity, nodes must be equipped with more keys prior to deployment. This, on the other hand, is a security risk. If an adversary attacks to the network and captures some nodes, it will learn keys that might also be used another part of the network. Thus, not only the captured nodes' secure links, but also some innocent secure links between uncaptured nodes become compromised. This means, resiliency is undesirably declined. In order to improve resiliency, less keys should be stored into the nodes prior to deployment, but this should be done without sacrificing from secure connectivity. The zone-based deployment model of Du et al.'s Zo-RKP scheme [2] uses deployment knowledge during key predistribution. In this way, high secure connectivity values can be reached with less number of keys per node. This, in turn, causes increased resiliency.

In RoK model [3], sensor nodes are redeployed periodically. The time period in which new nodes are deployed is called *generation*. In each generation, some nodes are redeployed. Some nodes die due to battery depletion in each generation as well. A refreshed key ring is stored for each redeployed node. The key rings are generation-specific so that the attacker cannot make use of a captured key after a certain amount of generations. Similarly, the attacker can make use of the keys of a captured node for the messages sent by the nodes that belong to a couple of previous generations only. To provide this facility, each node keeps *backward* and *forward* key rings. The keys in these key rings are randomly selected from forward and backward key pools, respectively. In each generation, the keys in forward key pool are hashed. For the backward key pool, a hash chain is generated for each key. In each generation, each key is replaced with the next value in the corresponding hash chain. Nodes that have common key indices can calculate each other's generation versions with some hash calculations. In this way, they establish pairwise key. Similarly, the attacker can calculate some link keys when it captures a node. However, these link keys are limited with the lifetimes of the sensor nodes. When a node dies and is replaced with a redeployed one, the adversary cannot make use of the keys of the dead node to compromise the links of the newly deployed one. Therefore, the effect of a node capture lasts certain amount of time. This property is called self-healing property. The amount of time that the attacker makes use of a captured key ring is related to the average lifetime of a node. In RoK model, the average lifetime is taken as 5 generations. We also use the same value in Zo-RoK. The readers may refer to [3] for more detailed discussion on RoK.

2.2 Our Contribution

In our Zo-RoK scheme, we adapt the zone-based deployment model of Zo-RKP scheme [2] into the RoK model [3] in order to have increased resiliency with self healing property. We divide the sensor field into contingent zones and employ forward and backward key pools for each zone. The nodes of particular zones obtain their key rings from their zone key pools. For each generation, zone key pools are updated (via hash and hash chains) as in RoK. The main novelty in our scheme lies in the adaptation of the Du et al.'s zone based deployment model into RoK. This is explained in the next section. As will be discussed in Section 3, with the proposed adaptation, Zo-RoK tremendously improves the resiliency of the network against node capture attacks as compared to RoK without any decrease in the secure connectivity.

Table 1. Notations used in the paper

FKP	Global forward key pool
BKP	Global backward key pool
FKP_z^j	Forward key pool for region z at generation j
$fk_{\eta,\xi,i}^j$	i^{th} key of the forward sub key pool shared between zones (regions) ξ and η at generation j
$fk_{z,0,i}^j$	i^{th} key of non-shared forward sub key pool of zone (region) z at generation j
BKP_z^j	Backward key pool for region z at generation j
$bk_{\eta,\xi,i}^j$	i^{th} key of the backward sub key pool shared between zones (regions) ξ and η at generation j
$bk_{z,0,i}^j$	i^{th} key of non-shared backward sub key pool of zone (region) z at generation j
$H(\cdot)$	Irreversible Hash function
$f_1(\cdot,\cdot,\cdot)$	Three-factor ordering function
$f_2(\cdot)$	Pseudorandom number sequence generator function
$r_{id_A,z,j,i}$	i^{th} element of the pseudorandom number sequence generated for node A of zone z at generation j
$\eta \leftrightarrow \xi$	Zone η and zone ξ are horizontally neighboring zones
$\eta \updownarrow \xi$	Zone η and zone ξ are vertically neighboring zones
$\eta \nearrow \xi$	Zone η and zone ξ are diagonally neighboring zones
m	Last generation
GW	Generation window
P	Global key pool size (half is for forward, half is for backward key pools)
S	Regional key pool size (half is for forward, half is for backward key pools)
k	Key ring size (half is for forward, half is for backward key rings)
d	Diagonal neighbor key sharing constant
n	Vertical / horizontal neighbor key sharing constant
Z	Number of zones, $Z = t \times t$

2.3 Key Predistribution in Zo-RoK

In our Zo-RoK scheme, the sensor field is divided into a two dimensional grid of zones/regions[1] as in [2]. Each zone has its own forward and backward key pools. The forward and backward key pools of each zone are selected from global forward and backward key pools. Moreover, the neighboring zones' forward and backward key pools share keys.

The notations used in the explanation of our model are given in Table 1.

2.3.1 Generation of Forward and Backward Regional Key Pools

As in [2], our scheme uses different sharing factors for the key pools of horizontal/vertical and diagonal neighboring zones. As shown in Figure 1, vertical and horizontal neighbor zones share $n \cdot S$ keys, diagonal neighbor zones share $d \cdot S$ keys, where $4(n + d) = 1$. This is the original method of Du et al. [2]; we adopt this to multiphase networks in Zo-RoK as will be detailed below.

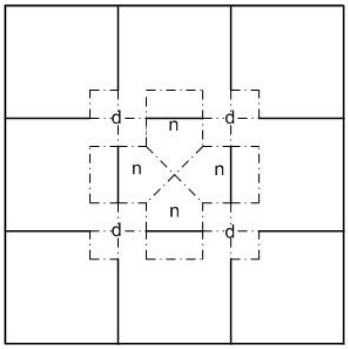

Figure 1. Key sharing among neighboring zones

The sizes of the global forward and backward key pools are $P/2$ each. Similarly, for each region, backward pool size and forward key pool size is $S/2$.

Let us now generalize key pool sharing and regional key pool generation mechanisms for a square sensor field with $t \times t$ zones. In such a field, for each row, $t - 1$ horizontal forward sub key pools are shared between neighboring zones. Similarly, for each column, $t - 1$ vertical forward sub key pools are shared between neighboring zones. In this way, the total number of horizontally and vertically shared forward sub key pools becomes $2 \cdot t(t - 1)$, each has distinct $n \cdot S/2$ forward keys drawn from the global forward key pool. The same analysis directly applies for backward sub key pools; the total number of horizontally and vertically shared backward sub key pools is $2 \cdot t(t - 1)$, each has distinct $n \cdot S/2$ backward keys drawn from the global backward key pool.

There are also distinct shared diagonal forward and backward sub key pools in this setting. The total number of diagonally shared forward sub key pools is $2 \cdot (t - 1)^2$, each has $d \cdot S/2$ distinct forward keys drawn from global forward key pool. Similarly, the total number of diagonally shared backward sub key pools is $2 \cdot (t - 1)^2$, each has $d \cdot S/2$ distinct backward keys drawn from global backward key pool.

[1] From this point on, *zone* and *region* will be used interchangeably.

Each shared sub key pool has distinct keys drawn from the corresponding global key pool (backward or forward). When a key is assigned to a shared key pool, it is deleted from the global one so that it is not reused in another shared pool.

Each zone establishes its key pool using the abovementioned horizontally, vertically and diagonally shared sub key pools. For each horizontally neighboring zone pair, the keys in a horizontally shared forward sub key pool are assigned to the forward key pools of these neighboring zones. A shared key pool used for a zone pair is not used again for other neighboring pairs. The same procedure is applied for the backward keys. Similarly, the vertical and diagonal neighbor pairs undergo the same process using vertically and diagonally shared sub key pools. In this way, all shared sub key pools are used. Identities of the individual keys are assigned during the assignment of shared sub key pools to zone key pools. More formally, a forward key or a backward key is identified using three tuples as $fk_{x,y,i}$ and $bk_{x,y,i}$, where x and y are the indices of two neighboring zones. The index i is the order of the key in the shared forward or backward sub key pool, where $i = 1, 2, \dots, n \cdot S/2$ for horizontal and vertical neighbors, $i = 1, 2, \dots, d \cdot S/2$ for diagonal neighbors.

The abovementioned process of zone key pool establishment assigns $S/2$ keys for non-boundary zones; thus, the key pool establishment for these zones is completed. However, this process puts less than $S/2$ keys in the key pools of the boundary zones since they do not have enough neighbors to share keys. In order to equalize the key pool sizes for all zones, boundary zones should fill up the remaining keys from the backward and forward global key pools. The total number of missing forward keys for each of the four corner zones is $(1 - 2n - d) \cdot S/2$. The total number of missing backward keys is also the same. Other than these four corner zones, there are $4 \cdot (t - 1)$ side zones. The number of missing keys of the forward key pool for each of these side zones is $(1 - 3n - 2d) \cdot S/2$. The number of missing backward keys is also the same. The identities of those non-shared keys are assigned after their assignments to the regional key pools. For the sake of standardization, again a 3-tuple identification is used, however second zone index is set to 0, meaning that this key is not shared between zones. More formally, such non-shared forward and backward keys are denoted as $fk_{x,0,i}$ and $bk_{x,0,i}$, where x is the index of the owning zone and i is the order of drawing from the corresponding global key pool. The range of i depends on whether x is a corner or side zone; $i = 1, 2, \dots, (1 - 2n - d) \cdot S/2$ for corner zones, and $i = 1, 2, \dots, (1 - 3n - 2d) \cdot S/2$ for side zones.

As can be seen from the above analysis, the backward and forward global key pools must be spent for the shared sub key pools and for the missing keys of the boundary zones. Moreover, there are same amount of keys for both backward and forward keys in each category. Thus, the size of regional backward/forward key pools, $S/2$, is calculated as follows.

$$S/2 = \frac{P/2}{2nt(t-1) + 2d(t-1)^2 + 4(1-2n-d) + 4(t-1)(1-3n-2d)} \quad [1]$$

As in RoK [3] scheme, we use the *generation* concept in Zo-RoK. Therefore, the regional backward and forward key pools are to be created for each generation. For forward key pools, initial generation is 0. For generation-0 forward key pool of each zone,

$S/2$ keys are selected from the global forward key pool as described in this subsection. For region z, initial forward key pool is formally shown as follows.

$$FKP_z^0 = \left\{ fk_{\eta,\xi,i}^0 \middle| \begin{array}{c} (\eta = z \vee \xi = z) \wedge (\eta \leftrightarrow \xi \vee \eta \updownarrow \xi), \\ i = 1,2,\dots,n \cdot S/2,\ \eta = 1,2,\dots,Z,\ \xi = 1,2,\dots,Z \end{array} \right\} \bigcup$$

$$\left\{ fk_{\eta,\xi,i}^0 \middle| \begin{array}{c} (\eta = z \vee \xi = z) \wedge (\eta \nearrow \xi), \\ i = 1,2,\dots,d \cdot S/2,\ \eta = 1,2,\dots,Z,\ \xi = 1,2,\dots,Z \end{array} \right\} \bigcup$$

$$\left\{ fk_{z,0,i}^0 \middle| \begin{array}{c} i = 1,2,\dots,(1-2n-d) \cdot S/2, \text{ if } z \text{ is corner zone} \\ i = 1,2,\dots,(1-3n-2d) \cdot S/2, \text{ if } z \text{ is a side zone} \\ \emptyset, \text{ if } z \text{ is a non-boundary zone} \end{array} \right\},$$

where $z = 1,2,\dots,Z$ [2]

In order to update the keys for the other generations, we use the same approach employed in RoK. At each generation, the keys are updated by the help of an irreversible hash function. Each key of the forward key pool is hashed to generate the key pool of the next generation. More formally, the forward key pool of zone z in generation j is shown as follows.

$$FKP_z^j = \left\{ fk_{\eta,\xi,i}^j \middle| fk_{\eta,\xi,i}^j = H\left(fk_{\eta,\xi,i}^{j-1}\right),\ \forall\ fk_{\eta,\xi,i}^{j-1} \in FKP_z^{j-1} \right\},$$

where $z = 1,2,\dots,Z$ and $j = 1,2,\dots,m$ [3]

Backward key pools for different generations are prepared similar to forward key pools with one difference such that the preparations should start with the last generation, m. The reason is that one-way hash chains [11] are used for each of the backward keys and they have to be utilized from the end. Thus, the first regional backward key pools are the generation-m pools, which are formally shown as follows.

$$BKP_z^m = \left\{ bk_{\eta,\xi,i}^m \middle| \begin{array}{c} (\eta = z \vee \xi = z) \wedge (\eta \leftrightarrow \xi \vee \eta \updownarrow \xi), \\ i = 1,2,\dots,n \cdot S/2,\ \eta = 1,2,\dots,Z,\ \xi = 1,2,\dots,Z \end{array} \right\} \bigcup$$

$$\left\{ bk_{\eta,\xi,i}^m \middle| \begin{array}{c} (\eta = z \vee \xi = z) \wedge (\eta \nearrow \xi), \\ i = 1,2,\dots,d \cdot S/2,\ \eta = 1,2,\dots,Z,\ \xi = 1,2,\dots,Z \end{array} \right\} \bigcup$$

$$\left\{ bk_{z,0,i}^m \middle| \begin{array}{c} i = 1,2,\dots,(1-2n-d) \cdot S/2, \text{ if } z \text{ is corner zone} \\ i = 1,2,\dots,(1-3n-2d) \cdot S/2, \text{ if } z \text{ is a side zone} \\ \emptyset, \text{ if } z \text{ is a non-boundary zone} \end{array} \right\},$$

where $z = 1,2,\dots,Z$ [4]

Each key of these key pools is the first element (seed) of a one-way hash chain. The keys of generation $m-1$ are the second elements of the chains, and so on. More formally, the backward key pool of zone z in generation j is represented as follows.

$$BKP_z^j = \left\{ bk_{\eta,\xi,i}^j \middle| bk_{\eta,\xi,i}^j = H\left(bk_{\eta,\xi,i}^{j+1}\right),\ \forall\ bk_{\eta,\xi,i}^{j+1} \in BKP_z^{j+1} \right\},$$

where $z = 1,2,\dots,Z$ and $j = m-1, m-2, \dots, 0$ [5]

Here one should notice that the subscript triplets of a forward key pool are exactly the same as the subscript triplets of the corresponding backward key pool. This is particularly important for key ring generations and session key establishment that will be discussed in subsequent sections.

2.3.2 Generation of Key Rings

In this section, we describe the process of key assignments to the nodes. Each node in Zo-RoK has forward and backward key rings, as in RoK. However, different from RoK, each node picks its keys from regional key pools. The selection process is random.

In order to facilitate the description of the key ring assignment process, we assume that the keys of both forward and backward key pools of each zone are ordered by their subscript triplets and each key is assigned an implicit sequence number in the range of $[1,2,\dots,S/2]$. The ordering is done in a way that the implicit sequence number of a particular forward key $fk_{\eta,\xi,i}^j$ is the same as its backward counterpart $bk_{\eta,\xi,i}^j$. The ordering function $f_1(\eta,\xi,i)$ gets the subscript triplet as parameter and returns the implicit sequence number (in the range of $1,2,\dots,S/2$) of that key in the corresponding forward and backward key pools.

There are total of k keys in a key ring. Half of it is for forward, the other half is for backward keys. We employ a pseudorandom number generator function $f_2(.)$ that returns a nonrepeating pseudorandom sequence of $k/2$ numbers, $r_i,\ i = 1,2,\dots,k/2$ and $0 < r_i \le S/2$. These values are then used to determine the random keys selected from the regional key pools. For each node, we use this function with the generation, zone and node IDs in order to determine a unique random sequence for that node. More formally, for node A of zone z at generation j, the random index values of forward and backward key rings are determined as follows.

$$f_2(id_A \parallel z \parallel j) = \left(r_{id_A,z,j,i} \middle| i = 1,2,\dots,k/2,\ 0 < r_{id_A,z,j,i} \le S/2\right) \text{ [6]}$$

First, forward key ring of this node is determined by picking the forward keys with implicit sequence numbers of $r_{id_A,z,j,i}$ from the corresponding forward key pool FKP_z^j. More formally, the forward key ring of node A of zone z at generation j, $FKR_{id_A,z}^j$, is defined as follows.

$$FKR_{id_A,z}^j = \left\{ fk_{\eta,\xi,\delta}^j \middle| fk_{\eta,\xi,\delta}^j \in FKP_z^j \wedge f_1(\eta,\xi,\delta) \equiv r_{id_A,z,j,i}\ , i = 1,2,\dots,\frac{k}{2} \right\} \text{ [7]}$$

The pseudorandom number sequence $r_{id_A,z,j,i}\ , i = 1,2,\dots,k/2$, is determined using Equation 6.

Second, backward key ring is determined. The indexing mechanism for the backward key ring is also the same. Same $f_2(.)$ pseudorandom sequence (eq. 6) is used to determine the index of the backward keys in order to be able to match up the forward and backward keys in the pairwise key establishment phase. The only difference in backward key ring generation is that backward key ring of a node at generation j includes keys in key pools of generation $GW + j - 1$, where GW is a system parameter called *Generation Window*. The main reason behind using generation window concept is to limit the amount of generations that a particular key becomes useful in order to provide self-healing. The use of generation window concept in Zo-RoK is borrowed from RoK scheme and will be explained in the next subsection. More formally, the backward key ring of node A of zone z at generation j, $BKR_{id_A,z}^j$, is defined as follows.

$$BKR_{id_A,z}^j = \left\{ bk_{\eta,\xi,\delta}^{GW+j-1} \middle| \begin{array}{c} bk_{\eta,\xi,\delta}^{GW+j-1} \in BKP_z^{GW+j-1} \wedge f_1(\eta,\xi,\delta) \equiv r_{id_A,z,j,i} \\ i = 1,2,\dots,k/2 \end{array} \right\} \text{ [8]}$$

The pseudorandom number sequence $r_{id_A,z,j,i}$, $i = 1, 2, \dots, k/2$, is determined using Equation 6, as in forward key ring. In other words, the sequence numbers, and consequently the subscript triplets, of the forward keys are the same the ones of the backward keys. For example, if forward key ring contains $fk^{j}_{4,7,22}$, then backward key ring also contains $bk^{GW+j-1}_{4,7,22}$.

Both forward and backward key rings are stored in the memory of a sensor node before the deployment. Therefore, total of $\frac{k}{2} + \frac{k}{2} = k$ keys are stored in each sensor node.

2.4 Zone-based Deployment and Pairwise Key Generation in Zo-RoK

Zone-based grouping of nodes and key ring assignments have been performed in the previous stage. Next step is the deployment of the nodes over the sensor field. In our zone-based deployment model, the sensor field is divided into $t \times t$ zones. There is a group of nodes associated with each zone. Initially, generation-0 nodes selected from each group are deployed over the corresponding zones. As the nodes die, replacement nodes of future generations are continually deployed. In order to have a fair comparison with the RoK scheme [3], the intra-zone deployment model in Zo-RoK scheme is assumed to be similar to the deployment model of RoK. In this deployment model, the deployment points in the zones are arranged as a regular grid in which only the horizontal and vertical neighbors hear each other. However, in order to accommodate the deployment errors in Zo-RoK, a certain fraction of nodes are assumed to be displaced to neighboring zones.

After the deployment, two neighboring nodes try to establish a common pairwise key. The basic method of pairwise key generation is the same as the RoK scheme [3]; forward and backward keys are used together. Two neighboring nodes, A and B, of generations j_1 and j_2, $j_1 \leq j_2$, first check whether their generations overlap or not. According to [3], two nodes have overlapping generations if $|j_1 - j_2| < GW$. If their generations overlap, then they exchange all the $k/2$ forward/backward key subscript triplets in their key rings. In order to establish a common key, they have to have at least one common triplet in their key rings. Suppose such a triplet (η, ξ, δ) exists. The common key between these two nodes is calculated as the hash of forward key of generation j_2, $fk^{j_2}_{\eta,\xi,\delta}$ and backward key of generation $j_1 + GW - 1$, $bk^{j_1+GW-1}_{\eta,\xi,\delta}$. More formally, the common key, K, is denoted as follows.

$$K = H(fk^{j_2}_{\eta,\xi,\delta} \parallel bk^{j_1+GW-1}_{\eta,\xi,\delta}) \quad [9]$$

In order both A and B calculate the same pairwise key, $fk^{j_2}_{\eta,\xi,\delta}$ and $bk^{j_1+GW-1}_{\eta,\xi,\delta}$ must be known by both users. Node B has $fk^{j_2}_{\eta,\xi,\delta}$ in its forward key ring, but node A does not. However, A can calculate it by hashing the forward key $fk^{j_1}_{\eta,\xi,\delta}$ that it has in its forward key ring $|j_1 - j_2|$ times. Similarly, $bk^{j_1+GW-1}_{\eta,\xi,\delta}$ exists in the backward key ring of node A, but not of node B. However, B can calculate it by hashing the backward key $bk^{j_2+GW-1}_{\eta,\xi,\delta}$ that it has in its backward key ring $|j_1 - j_2|$ times.

If there is more than one common subscript triplet, then all of them are utilized in pairwise key generation in the same manner.

The use of generation window, forward and backward keys provide the established keys to be useful for a limited amount of generations. In this way, if these keys are compromised, the attacker can make use of it only a few generations. This causes the system to self-heal in time. Since this self-healing behavior is mainly due to the RoK scheme [3], we do not go into its detail in this paper. The readers may refer to [3] for more details.

3. PERFORMANCE EVALUATION

We perform simulative performance evaluation and compare the performance of RoK scheme [3] with the proposed Zo-RoK scheme. The simulation software is developed using C# in .NET 2005. Simulations are run on a computer with Intel® Celeron® M 520 processor operating at 1.6 GHz. For the sake of accuracy, each simulation is repeated 20 times and the average values are reported. In order to make sure about the correctness of the simulation software, we reproduce the results in [3] with no error.

The parameters that we used in our simulations are as follows. The deployment area is divided into $10 \times 10 = 100$ zones, i.e. $Z = 100$. Each zone has 100 nodes, therefore, total number of nodes is 10000. As discussed in Section 2.4, intra-area deployment model is a grid-based one and only neighbors hear each other. Since we fixed the neighboring relationships, the communication range and deployment area are not system parameters. In order to accommodate deployment errors, 20% of nodes are deployed in neighboring zones using a uniform random distribution. As in RoK [3], at the end of each generation, dead nodes are replaced by new nodes with fresh key rings. The lifetime determination is also the same as RoK such that the lifetime of a node is determined according to a Gaussian distribution with mean $GW/2$ and standard deviation $GW/6$. The generation window, GW, is taken as 10.

The global key pool size, P, of both RoK and Zo-RoK is taken as 20000; half of it is used for forward keys, half of it is for backward keys. In parallel to the work of Du et al. [2], horizontal/vertical key sharing constant, n, and diagonal key sharing constant, d, are taken as 0.15 and 0.10 respectively. These are the optimum values. Using these parameters, the sizes of regional forward/backward key pools, $S/2$, is calculated using Equation 1 as ~ 178.

We, first analyze *secure connectivity* metric, which is defined as the probability of two neighboring nodes being able to establish a pairwise key (i.e. having at least one common key index in their forward and backward key rings). With this analysis, we determine the key ring size necessary to have almost 1.0 secure connectivity for both RoK and Zo-RoK schemes. The results are depicted in Figure 2. Our results show that with 250 keys in forward and 250 keys in backward key rings, RoK achieves perfect secure connectivity (i.e. 1.0) in all generations. Comparable secure connectivity for Zo-RoK is achieved when the forward and backward key ring sizes are 75 each. As a result, we can say that Zo-RoK uses 70% less memory as compared to RoK, while reaching the same level of secure connectivity. The main reason behind this performance improvement in Zo-RoK lies in the zone-based deployment and key predistribution schemes. In Zo-RoK, the nodes that will be close to each other after deployment share keys, distant nodes do not. However, in RoK, distant nodes also share keys. Therefore, an increased key ring size is needed in RoK to achieve full secure connectivity as compared to proposed Zo-RoK scheme.

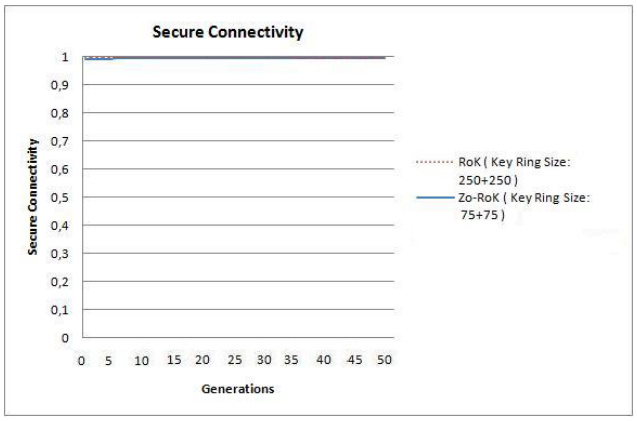

Figure 2. Secure connectivity of RoK and Zo-RoK schemes. Both are constantly 1.0.

Next, we analyze the resiliency of the network under node capture attacks. Here two different metrics are used. These metrics are the same ones used in RoK [3]:

- *Active compromise ratio*: This is the ratio of *active compromised links* / *all active links*. As the attacker captures nodes, it learns keys that are also used to secure links in other part of the network. This metric is the ratio of amount of such extra active links compromised over the amount of all active links in the network. An *active* link is defined as a link currently used by alive nodes.
- *Total compromise ratio*: This is the ratio of *total compromised links* / *all links*. In this metric, not only active links, but also dead links are taken into consideration (for both numerator and denominator). A dead link is defined as a link with one or two dead endpoints.

Moreover, two attacker models are considered, again in parallel to the models in RoK [3]. These are *temporary* and *constant* attacker models. In *temporary attacker* model, the attacker starts its node capture attack at the beginning of generation 0 and gives up at the beginning of generation 10. Within this 10 generation interval, the attacker captures nodes at each round (round is a time unit and one generation has 10 rounds). In the *constant attacker* model, the attacker starts its attack by capturing nodes at the beginning of generation 0 and never gives up.

Another attack parameter is the node capture rate of the attacker. This parameter is defined as the number of nodes that the attacker captures at each round. In our simulations, we consider 3 and 5 node captures/round (i.e. 30 and 50 nodes per generation).

Figures 3 – 6 show the resiliency performance of temporary attacker model. As shown in these figures, general trend in resiliency behavior of both schemes is similar but the harm caused by temporary attacker in Zo-RoK is significantly smaller than RoK. When the attack is at its highest stage around generation 10, RoK compromises as much as 6-fold more links as compared to proposed Zo-RoK scheme. This certainly makes Zo-RoK more resilient than RoK. When Figures 3 and 5 are compared to Figures 4 and 6 respectively, we see that compromise ratio increases as node capture rate increases, as expected. Moreover, these figures also show that the relative benefit of Zo-RoK as compared to RoK is higher with smaller node capture rates in the temporary attacker model. Figures 5 and 6 also show that the system self-heals almost at the same time (around generation 15). However, the overall effect of the attack in Zo-RoK is much smaller than RoK as discussed above. The main reason of this improved resiliency behavior of Zo-RoK is having less key in the key rings as compared to RoK. As discussed at the beginning of this section, perfect secure connectivity in Zo-RoK is achieved using 150 keys in the key ring; however, 500 keys are needed for RoK. Therefore, when a node is captured, the attacker obtains more keys in RoK as compared to Zo-RoK. This makes the attacker to compromise more links in RoK.

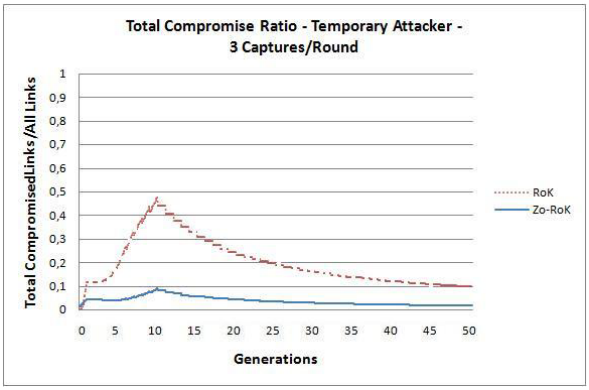

Figure 3. Total compromise ratios of RoK and Zo-RoK schemes under temporary attacker model with 3 node captures per round

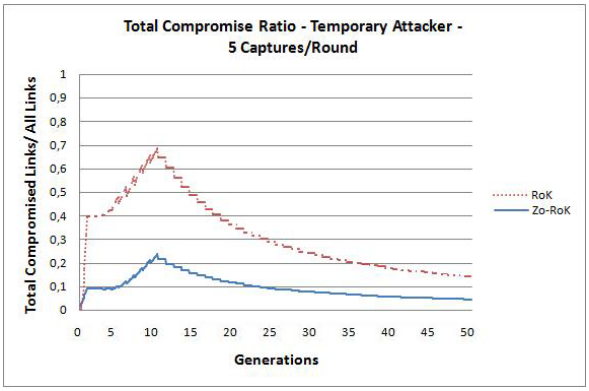

Figure 4. Total compromise ratios of RoK and Zo-RoK schemes under temporary attacker model with 5 node captures per round

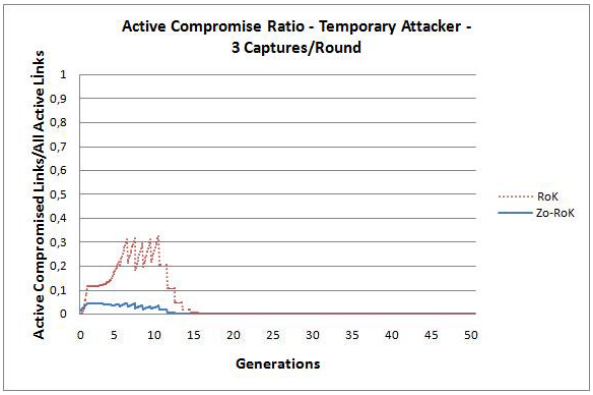

Figure 5. Active compromise ratios of RoK and Zo-RoK schemes under temporary attacker model with 3 node captures per round

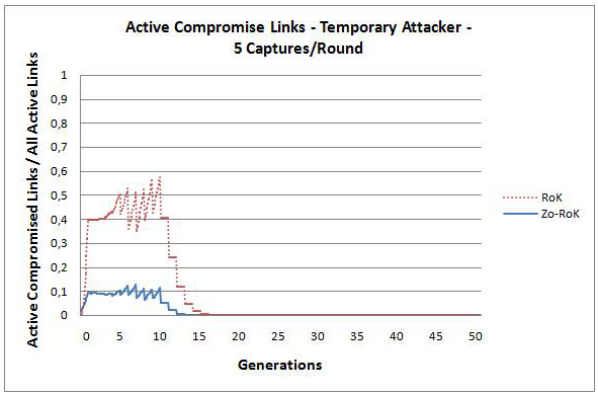

Figure 6. Active compromise ratios of RoK and Zo-RoK schemes under temporary attacker model with 5 node captures per round

The results for the *constant attacker* model, in which the attacker captures nodes forever, are shown in Figures 7 – 10. In both compromise metrics and node capture rates, Zo-RoK shows better resiliency performance than RoK does. Although total compromise ratio increases as the attack continues in both schemes (Figures 7 and 8), the performance of Zo-RoK is up to 6-fold better than RoK at the beginning of the attack. The marginal gain of Zo-RoK reduces as the attack grows in upcoming generations since the compromise ratio approaches to its maximum value of 1.0. From Figures 9 and 10, we see that RoK is able to keep the active compromise ratio within [0.2 – 0.3] for 3 nodes/round capture rate, and within [0.4 – 0.6] for 5 nodes/round capture rate. On the other hand, as seen in Figures 9 and 10, the proposed Zo-RoK scheme keeps the active compromise ratio under control around 0.05 and 0.1 for 3 and 5 nodes/round capture rates respectively. This analysis shows that the amount of active links compromised in RoK scheme becomes as much as 5-fold more as compared to Zo-RoK scheme.

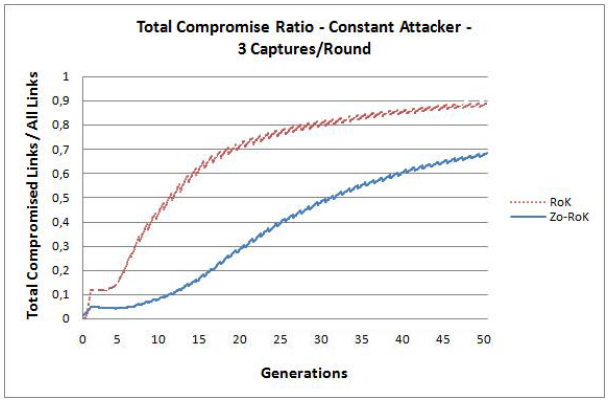

Figure 7. Total compromise ratios of RoK and Zo-RoK schemes under constant attacker model with 3 node captures per round

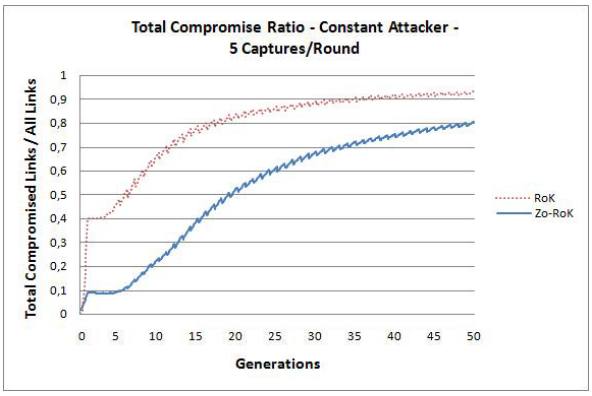

Figure 8. Total compromise ratios of RoK and Zo-RoK schemes under constant attacker model with 5 node captures per round

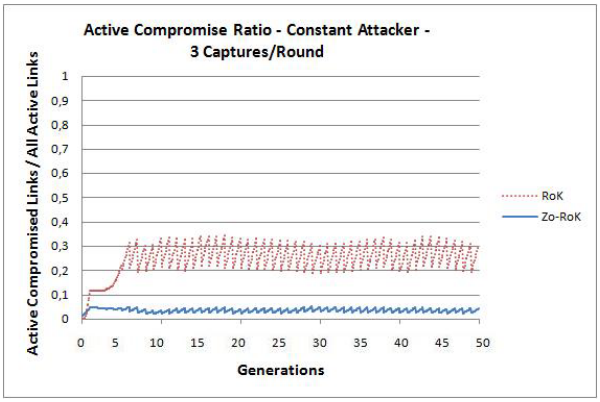

Figure 9. Active compromise ratios of RoK and Zo-RoK schemes under constant attacker model with 3 node captures per round

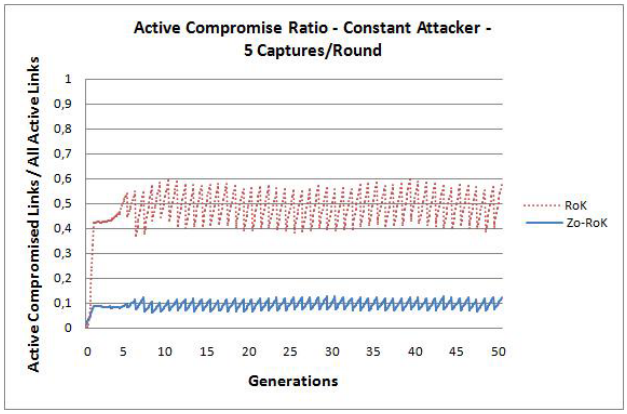

Figure 10. Active compromise ratios of RoK and Zo-RoK schemes under constant attacker model with 5 node captures per round

4. RELATED WORK

The basic model of random key predistribution in sensor networks is first proposed by Eschenauer and Gligor [4]. This model inspired several other authors. Upcoming studies, such as [5], [6] and [7], mostly focused on increasing the security and performance

of random key predistribution schemes either by using the help of the network or other cryptographic techniques.

The use deployment knowledge to improve the performance of random key predistribution schemes is first proposed by Du et al. [2]. In this scheme, the sensor field is divided into zones and nodes are grouped according to these prior zone information. In this way, after deployment, the nodes that are physically closer to each other would have more chance to share keys. This, in turn, causes less memory usage for keys and better connectivity and resiliency performance. After [2], some other schemes, such as [8] and [9], that use prior location information are proposed.

Ramkumar and Memon [12] proposed to use repeated hashing for increased efficiency in random key predistribution. Castelluccia and Spognardi proposed RoK scheme [3] for multiphase sensor networks. In this scheme, new nodes with fresh key rings replace dead ones. In this way, the network heals itself in time. Later, Yilmaz et al. [10] proposed two schemes for faster self healing.

The proposed Zo-RoK scheme also aims self healing, but by keeping the harm caused by the attacker under certain limits. Zo-RoK combines the best parts of Du et al. [2] and RoK [3] schemes and improves the resiliency of RoK up to 6-fold by using 70% less key memory for full secure connectivity.

5. CONCLUSIONS

We proposed a random key predistribution scheme for multiphase sensor networks. The proposed scheme, called Zo-RoK (Zone-based Robust Key Distribution), uses prior deployment knowledge in order to reduce the key ring size requirements. In this way, the resiliency of the network against node capture attacks also increases. Due to the multiphase property of the network, dead nodes are replaced by new nodes. The new nodes come with fresh keys in their key rings. In this way, the attacker can make use of the compromised keys only for a small period of time and, therefore, the network heals itself.

In Zo-RoK, we are inspired by [2] and [3] and combined the best parts of these schemes. To the best of our knowledge, the proposed Zo-RoK scheme is the first location-aware random key predistribution scheme proposed for *multiphase* sensor networks.

We performed simulations for comparative performance evaluation. We compared the performance of RoK with Zo-RoK schemes. We have concluded that almost 100% key sharing probability can be achieved using 70% less keys in Zo-RoK. This reduced key amount also affects the resiliency of the system since a captured node would reveal fewer keys to the adversary. In this way, the active resiliency of the system is kept within reasonable limits. Even in the worst case scenario that we tested (attacker is attacking constantly with 5 nodes/round capture rate), only 10% of the active links are compromised in Zo-RoK, whereas in RoK, the attacker could compromise half of the network.

7. REFERENCES

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. 2002. "Wireless sensor networks: a survey". Computer Networks, 38(4), pp. 393–422.

[2] Du, W., Deng, J., Han, Y. S., Chen, S., and Varshney, P. K. 2004. "A key management scheme for wireless sensor networks using deployment knowledge". INFOCOM 2004.

[3] Castelluccia, C. and Spognardi, A. (2007). "Rok: A robust key pre-distribution protocol for multi-phase wireless sensor networks". SecureComm 2007, Third International Conference on Security and Privacy in Communication Networks.

[4] Eschenauer, L. and Gligor, V. D. 2002. "A key-management scheme for distributed sensor networks". In Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 41-47.

[5] Huang, D. and Medhi, D. 2007. "Secure pairwise key establishment in large-scale sensor networks: An area partitioning and multigroup key predistribution approach". ACM Trans. Sen. Netw. 3(3), 16, Aug. 2007, DOI= http://doi.acm.org/10.1145/1267060.1267064

[6] Li, G., Ling, H., and Znati, T. 2005. "Path key establishment using multiple secured paths in wireless sensor networks". In Proceedings of CoNEXT '05 - the 2005 ACM Conference on Emerging Network Experiment and Technology, pp. 43-49. DOI= http://doi.acm.org/10.1145/1095921.1095928

[7] Lei, W., Zhi-ping, C., and Xin-hua, J. 2005. "Researches on scheme of pairwise key establishment for distributed sensor networks". WMuNeP '05 - 1st ACM Workshop on Wireless Multimedia Networking and Performance Modeling,. pp. 54-61. DOI= http://doi.acm.org/10.1145/1089737.1089747

[8] Liu, D., Ning, P., and Du, W. 2008. "Group-based key predistribution for wireless sensor networks". ACM Trans. Sen. Netw. 4(2), Mar. 2008, pp. 1-30. DOI= http://doi.acm.org/10.1145/1340771.1340777

[9] Liu, F., Rivera, J. and Cheng, X. 2006. "Location-aware key establishment in wireless sensor networks". In Proceedings of IWCMC '06 - International Conference on Wireless Communications and Mobile Computing, pp. 21-26. DOI= http://doi.acm.org/10.1145/1143549.1143556

[10] Yilmaz, O. Z., Levi, A. and Savas, E. 2008. "Multiphase Deployment Models for Fast Self Healing in Wireless Sensor Networks". In SECRYPT 2008 - International Conference on Security and Cryptography 2008.

[11] Lamport, L. 1981. "Password Authentication with Insecure Communication". Commun. of the ACM, 24(11), November 1981, pp. 770-772.

[12] Ramkumar, M. and Memon, N. 2005. "An Efficient Key Predistribution Scheme for Ad Hoc Network Security". IEEE Journal on Selected Areas of Communication, 23(3), March 2005, pp 611-621.