HaG: Hash Graph Based Key Predistribution Scheme for Multiphase Wireless Sensor Networks

Salim Sarımurat and Albert Levi
Computer Science and Engineering
Sabanci University
Istanbul, Turkey
{sarimurat, levi}@sabanciuniv.edu

Abstract—**Wireless Sensor Networks (WSN) consist of small sensor nodes which operate until their energy reserve is depleted. These nodes are generally deployed to the environments where network lifespan is much longer than the lifetime of a node. Therefore, WSN are typically operated in a multiphase fashion, as in [1-3, 9-10], which use different key pools for nodes deployed at different generations. In multiphase WSN, new nodes are periodically deployed to the environment to ensure constant local and global network connectivity. Also, key ring of these newly deployed nodes is selected from their deployment generation key pool to improve the resiliency of WSN. In this paper, we propose a key predistribution scheme for multiphase WSN which is resilient against permanent and temporary node capture attacks. In our Hash Graph based (HaG) scheme, every generation has its own key pool which is generated using the key pool of the previous generation. This allows nodes deployed at different generations to have the ability to establish secure channels. Likewise, a captured node can only be used to obtain keys for a limited amount of successive generations. We compare the connectivity and resiliency performance of our scheme with other multiphase key predistribution schemes and show that our scheme performs better when the attack rate is low. When the attack rate is high, our scheme still has better resiliency performance inasmuch as using less key ring size compared to the existing multiphase schemes.**

Keywords—Wireless sensor networks, security, key predistribution, generation keys, multiphase.

I. INTRODUCTION

Wireless Sensor Networks (WSN) are composed of sensor nodes which have limited amount of memory, energy and computation power. In typical application settings, sensor nodes are spread randomly over an environment and collect data that is transferred to a trusted central point for further examination [4]. Most of these application scenarios require long term sensing of the environment and energy reserve of the sensor nodes last for a very limited time. Therefore, deploying new nodes to the environment in certain intervals, called *generations*, is the only way to have stable network connectivity. Since the network lifespan is much longer than the lifetime of a sensor node, it is most likely that we have multiple generations while sensing an environment. Networks that provide this property are called *Multiphase* WSN.

This paper presents a new key predistribution scheme based on hash graphs of keys that provides secure communication between sensor nodes deployed at different generations. In our Hash Graph based (HaG) scheme, each deployment generation has its own key pool and these pools are generated using the pool of the previous generation. Key pool of the first generation is randomly generated and the subsequent generations use two consecutive keys of the preceding generation to form a key for the next generation. More specifically, two sequential keys are XORed (i.e. logical Exclusive Disjunction operation) and hashed together using a secure hash function to constitute a key of the next generation key pool. When two nodes are in the communication range, they use the generation that they have been deployed to the network in conjunction with the identification numbers to decide whether they have a common key or not. If they can find at least one common key, then nodes perform XOR operation on all common keys to generate a direct link key that is used for secure communication. With the HaG scheme, a temporary attacker can only compromise some portion of the network and right after the attack stops, scheme self-heals the keys until the compromised key ratio decreases to zero. Similarly, a permanent attacker is only able to compromise some steady fraction of the network. Compared to other multiphase schemes, HaG scheme provides better in resiliency if the attack rate is low. If the attack rate is high, we have some considerable improvements on the resiliency as well. Using a smaller amount of keys, HaG scheme delivers same connectivity rate with better resiliency performance.

The rest of this paper is organized as follows. The next section summarizes existing key predistribution methods. We provide detailed information about the scheme that we are proposing in Section III. Section IV discusses the comparative performance evaluation of our scheme and RoK scheme and finally Section V concludes the paper.

II. RELATED WORK

Depending on the application area of the WSN, security of the communication becomes an important criterion. There exist various solutions to the key predistribution problem, such as full pairwise [5], probabilistic [5, 6] and matrix-based approaches [7, 8]. Full pairwise scheme proposed by Chan et al. loads $n-1$ pairwise keys to every node of the n nodes in the network [5]. Although this scheme provides high level of security, it requires high amount of memory on the sensor nodes to store pairwise keys. Besides, addition of new nodes to the network is only possible if pairwise keys of them are preloaded to the nodes that are deployed before.

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

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

In probabilistic schemes, nodes receive a group of randomly selected keys, amount of which is enough for having a good connectivity percentage over the network. Although probabilistic schemes are less secure compared to the full pairwise scheme, they circumvent the memory overhead and require nodes to store only some predefined amount of keys in their memory. Practically all of the probabilistic schemes have three stages: key predistribution, shared key discovery and path key establishment. Eschenauer and Gligor's well-known Basic Scheme [6] is one example for the probabilistic schemes. In key predistribution phase, each sensor node is loaded with τ keys that are randomly selected from a key pool of size P where $\tau \ll P$. After deployment, sensor nodes try to discover their neighbors. When two neighboring nodes find at least one common key, then they can create a direct link to communicate securely. If no common key exists, then nodes start the path key establishment phase and they try to create a direct link with the help of their common neighbors. When we evaluate the performance of the Basic scheme, since $\tau \ll P$, majority of the keys will be loaded on multiple nodes and this decreases the resiliency. Finding neighbors with common keys, called *local connectivity*, is also an important criterion of the WSN, therefore the value of τ should be selected wisely to balance resiliency and local connectivity. Considering this weakness of the Basic Scheme, Chan et al. [5] have proposed a modification on the Basic Scheme, known as Q-Composite Scheme, which requires two nodes to have at least $q > 1$ keys in common in order to establish a secure direct link. This improvement increases the resiliency of the scheme, but decreases the connectivity of the network.

In the literature, we also have deterministic key predistribution approaches which are developed from the idea of Blom [7]. Generating one public and one private matrices and storing only $\lambda + 1$ keys from these matrices allow the nodes to generate a secure direct key with any of the nodes in the network. However, compromising more than λ nodes in the network will compromise all of the keys used in the network. Du et al. [8] propose a combination of the Basic Scheme [6] and Blom's Scheme [7] without increasing λ value. This Multiple Space Key Predistribution scheme provides very good resilience but it has higher memory requirement and communication overhead.

Up to now, all discussed key predistribution schemes are intended for single phase WSN. Even though they allow node additions to the network, it is not a stress-free and secure operation. Furthermore, modification of single phase WSN key predistribution solutions to adapt multiphase network has the weakness of continuous usage of the same key pool for multiple generations. Keys captured by an attacker at any time can be used in the course of the network's operation time. However, with multiphase WSN, we can use different generation pools that are completely different from the key pools used in other generations. This way, an attacker would only be able to compromise some portion of the network and after some time, the percentage of the compromised nodes will become stable if the attack is permanent. To the best of our knowledge, there are only a few key predistribution schemes addressing multiple deployments of the sensor nodes, i.e. multiphase WSN [1-4, 9-10].

Robust Key predistribution (RoK) scheme is a multiphase scheme proposed by Castelluccia et al. [1]. RoK has forward and backward key pools for each generation. Keys in these pools are randomly generated and they are updated in forward and backward orders by hashing. Nodes are loaded with equal number of keys having the same key identifier from forward and backward key pools. Lifetime of node is constrained by $i + G_w$ generations where i is the deployment generation of the node and G_w is the generation window. A node can only produce forward keys for generation j where $j > i$, and backward keys for generation j where $j < i + G_w - 1$. When two nodes are in communication range, they exchange their generation number and node identifier. Using these values, they calculate the identifier of the keys that are loaded on the node to be communicated and if they find at least one match, then they create the session key and start the secure communication. When an attacker captures a node from generation i, he would only be able to compromise keys that are used between generations $]i, i + G_w[$ because of the generation window boundary. Therefore, attacker should be capture at some rate permanently to have some portion of the network compromised. Even if he permanently captures nodes, he would only be able to compromise some portion of the network and as soon as he stops the captures, this percentage will start decreasing and become zero after some time. However, this scheme requires number of generations to be determined before starting the network because of the offline backward key pool generation phase. Also, sensor nodes use high computational power to update forward keys at every generation time.

Random Generation Material (RGM) scheme [2-3] is another multiphase WSN key predistribution method proposal. RGM scheme has one key pool for every generation and there is no relation between key pools of different generations. Nodes are loaded with generation keys gk_t^{gg} where g is the generation that the node is deployed and t is the identification number of the key. Communication between different generations is provided with keys that are generated by XORing the keys between the generations of two nodes. Then the XORed key is hashed and used to create a direct link between two nodes in different generations. Compared to the RoK scheme, RGM has better resilience because keys compromised from two nodes are only used in the generations that these nodes are deployed. Also, RGM has no limit on the deployment of the number of nodes to the network. However, increasing G_w value also increases the communication and computation cost of this scheme.

III. Proposed Scheme

This section describes our hash graph based key predistribution scheme proposal for multiphase wireless sensor networks.

A. *Overview*

Sensor nodes have very limited amount of energy reserve that limits their lifetime to a small period of time. Typically, this limited lifetime of sensor nodes is very short compared to the lifetime of the network. Hence, new sensor nodes need to be deployed to the network in some intervals called *generations*.

The symbols and notations we use for our scheme in the rest of the paper are listed in Table I below.

TABLE I. LIST OF SYMBOLS USED IN OUR SCHEME

Symbol	Definition
P	Key pool size
G_w	Generation window
KP^j	Key pool at generation j
KR_A^j	Key ring of node A at generation j
k_t^j	Key with index t at generation j
kg_t^j	Key group with index t at generation j
k_{AB}	Direct link key between nodes A and B
$h(\cdot)$	Secure hash function $h: \{0,1\}^* \rightarrow \{0,1\}^{160}$
$f(\cdot)$	Hash function $h: \{0,1\}^* \rightarrow \{0,1\}^{P/g}$
g	Number of key ring groups that are drawn from key pool
n	Number of key groups in the key ring of a node
m	Number of keys in the key ring of a node at the initial deployment time

Lifetime of a sensor is bounded by G_w generations, which is referred as *generation window,* as in [1]. A node deployed at generation i will drain its battery before generation $i + G_w$ and each generation period is assumed to be 1 in the rest of the paper. A node that is deployed at generation j should be able to establish a secure channel with the nodes that are deployed between $[j - G_w, j + G_w]$ generation periods.

There are three procedures for our scheme: key pool generation, key ring predistribution and pairwise key establishment. Specifics of these procedures are explained in the subsections below.

B. *Key Pool Generation*

Key pool of our scheme is updated at each generation. Our initial key pool has P randomly generated keys. When the generation period ends, two consecutive keys are XORed and hashed with a secure hash function $h: \{0,1\}^* \rightarrow \{0,1\}^{160}$, such as SHA1, to generate one key from key pool of the next generation.

Initial key pool of the network is defined as follows:

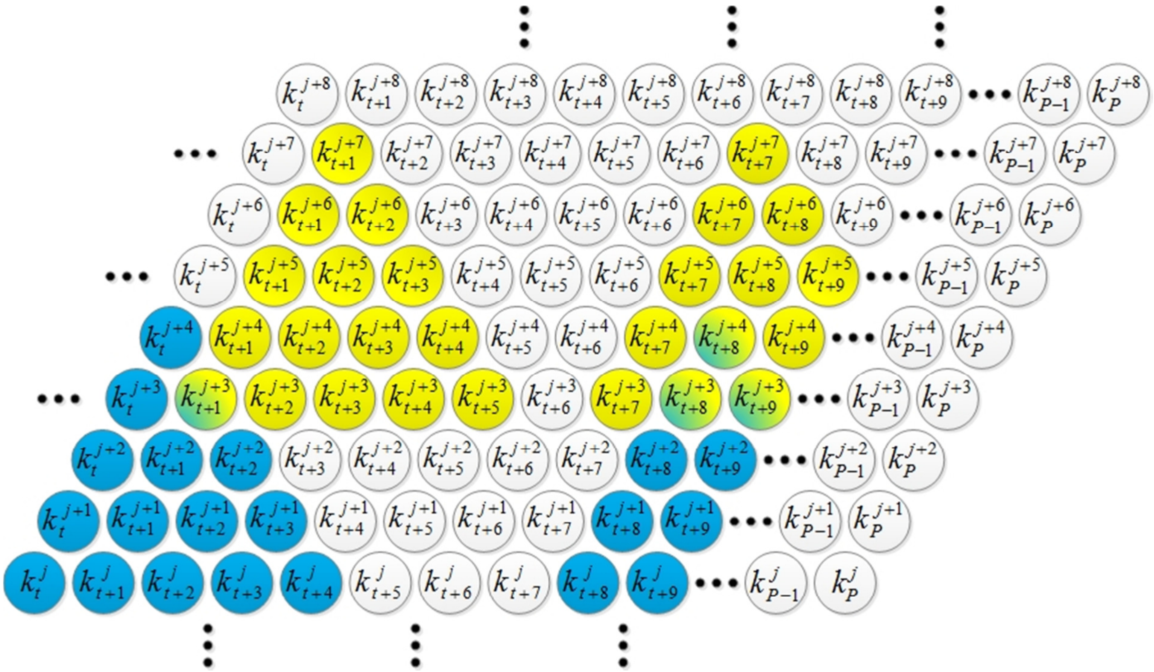
$$KP^0 = \{k_1^0, k_2^0, k_3^0, k_4^0, k_5^0, \dots, k_{P-1}^0, k_P^0\}$$

where each k_i^0 value is randomly generated.

The construction process of the generation key graph structure is depicted in Figure 1 below. To put it in more concrete terms: if the key pool at generation j is defined as $KP^j = \{k_1^j, k_2^j, k_3^j, \dots, k_{P-1}^j, k_P^j\}$, then key pool at generation $j+1$ is $KP^{j+1} = \{k_1^{j+1}, k_2^{j+1}, k_3^{j+1}, \dots, k_{P-1}^{j+1}, k_P^{j+1}\}$ where $k_t^{j+1} = h(k_t^j \oplus k_{t+1}^j)$. To reserve the key pool size P in every generation, k_P^{j+1} key is generated randomly and added to the end of KP^{j+1} key pool.

C. *Key Ring Predistribution*

In our scheme, we pre-distribute keys in groups of g keys from the generation key pool of size P. Each node has m keys that can be used to communicate with other nodes that are deployed to the environment at the same generation. Thus, nodes are loaded with $n = {}^{m}/_{g}$ different key groups from the key pool of their deployment generation. These key groups are selected using a pseudorandom function $f(\cdot)$ which does not produce consecutive numbers for the same node. For example, the first key group of the node A deployed at generation j is $f(id_A \parallel 1 \parallel j)$ which contains keys in $[f(id_A \parallel 1 \parallel j) \times g, f(id_A \parallel 1 \parallel j) \times (g+1)[$ range.



k_t^{j+8} k_{t+1}^{j+8} k_{t+2}^{j+8} k_{t+3}^{j+8} k_{t+4}^{j+8} k_{t+5}^{j+8} k_{t+6}^{j+8} k_{t+7}^{j+8} k_{t+8}^{j+8} k_{t+9}^{j+8} ••• k_{P-1}^{j+8} k_P^{j+8}
••• k_t^{j+7} k_{t+1}^{j+7} k_{t+2}^{j+7} k_{t+3}^{j+7} k_{t+4}^{j+7} k_{t+5}^{j+7} k_{t+6}^{j+7} k_{t+7}^{j+7} k_{t+8}^{j+7} k_{t+9}^{j+7} ••• k_{P-1}^{j+7} k_P^{j+7}
k_t^{j+6} k_{t+1}^{j+6} k_{t+2}^{j+6} k_{t+3}^{j+6} k_{t+4}^{j+6} k_{t+5}^{j+6} k_{t+6}^{j+6} k_{t+7}^{j+6} k_{t+8}^{j+6} k_{t+9}^{j+6} ••• k_{P-1}^{j+6} k_P^{j+6}
••• k_t^{j+5} k_{t+1}^{j+5} k_{t+2}^{j+5} k_{t+3}^{j+5} k_{t+4}^{j+5} k_{t+5}^{j+5} k_{t+6}^{j+5} k_{t+7}^{j+5} k_{t+8}^{j+5} k_{t+9}^{j+5} ••• k_{P-1}^{j+5} k_P^{j+5}
k_t^{j+4} k_{t+1}^{j+4} k_{t+2}^{j+4} k_{t+3}^{j+4} k_{t+4}^{j+4} k_{t+5}^{j+4} k_{t+6}^{j+4} k_{t+7}^{j+4} k_{t+8}^{j+4} k_{t+9}^{j+4} ••• k_{P-1}^{j+4} k_P^{j+4}
••• k_t^{j+3} k_{t+1}^{j+3} k_{t+2}^{j+3} k_{t+3}^{j+3} k_{t+4}^{j+3} k_{t+5}^{j+3} k_{t+6}^{j+3} k_{t+7}^{j+3} k_{t+8}^{j+3} k_{t+9}^{j+3} ••• k_{P-1}^{j+3} k_P^{j+3}
k_t^{j+2} k_{t+1}^{j+2} k_{t+2}^{j+2} k_{t+3}^{j+2} k_{t+4}^{j+2} k_{t+5}^{j+2} k_{t+6}^{j+2} k_{t+7}^{j+2} k_{t+8}^{j+2} k_{t+9}^{j+2} ••• k_{P-1}^{j+2} k_P^{j+2}
k_t^{j+1} k_{t+1}^{j+1} k_{t+2}^{j+1} k_{t+3}^{j+1} k_{t+4}^{j+1} k_{t+5}^{j+1} k_{t+6}^{j+1} k_{t+7}^{j+1} k_{t+8}^{j+1} k_{t+9}^{j+1} ••• k_{P-1}^{j+1} k_P^{j+1}
k_t^j k_{t+1}^j k_{t+2}^j k_{t+3}^j k_{t+4}^j k_{t+5}^j k_{t+6}^j k_{t+7}^j k_{t+8}^j k_{t+9}^j ••• k_{P-1}^j k_P^j

Figure 1. Key pool generation and pairwise key estanblishment in our scheme

More precisely, key ring of node A is constructed as:

$$KR_A^j = \{kg_t^j | t = f(id_A \parallel i \parallel j), i = 1, 2, 3, \dots, n\}$$

where $kg_t^j = \left\{k_{t\times g}^{j+1}, k_{t\times g+1}^{j+1}, k_{t\times g+2}^{j+1}, \dots, k_{(t+1)\times g-1}^{j+1}\right\}$.

Distribution of keys in groups allows nodes to have better chances of communication with nodes deployed in the future generations. We also make sure that our pseudorandom function $f(\cdot)$ does not give two consecutive group numbers for the same node; because this will give the attacker the advantage to compromise keys for more generations, and eventually reduce the resiliency of the scheme faster. For the same reason, we suggest that the number of keys in groups, g value, should be determined close to $G_w/2$; based on the observations on age distribution of the nodes provided in [1].

When the generation period ends, nodes should immediately generate the keys of the succeeding generation and delete the keys from the past generation key pool. This improves the resiliency of the network intensely because nodes that are not yet captured by an attacker will not disclose as much key as they would, if they were to store the keys of the past generations.

Our scheme has both forward and backward secrecy features. Forward secrecy, meaning the security of the future generation keys, is provided by using two sequential keys to produce a key in the next generation. If an attacker captures a node, he will only be able to compromise keys for g generations. Backward secrecy, meaning the security of past keys, is provided by the secure hash function $h(\cdot)$ and attacker is not able to recover any of the past keys even he captures all of the alive nodes in the network.

D. *Pairwise Key Establishment*

Nodes start pairwise key establishment phase right after being deployed to the environment. When a sensor node A, with node identifier id_A , is deployed to the network at generation j, it broadcast a message containing these values. Neighbor nodes can use this message to construct the key ring KR_A^j and using this key list, they can check whether they have at least one key in common or not.

If node A is deployed at generation j and node B is deployed at generation i where $i \leq j$, then they can find a common key in $[j, i + G_w[$ generation interval. If they find at least one common key, then they XOR all common keys and then hash them to generate k_{AB} which will be used to secure the communication between nodes A and B.

E. *Example*

In this section, we provide an example for the pairwise key establishment phase. As seen in Figure 1, we have two nodes, A and B, that are deployed at generations j and $j+3$ consecutively, with a generation window $G_w = 5$. Key rings of these nodes are as follows:

$$KR_A^j = \{\dots, k_t^j, k_{t+1}^j, k_{t+2}^j, k_{t+3}^j, k_{t+4}^j, k_{t+8}^j, k_{t+9}^j, k_{t+10}^j, \dots\}$$

$$KR_B^j = \{\dots, k_{t+1}^{j+3}, k_{t+2}^{j+3}, k_{t+3}^{j+3}, k_{t+4}^{j+3}, k_{t+5}^{j+3}, k_{t+7}^{j+3}, k_{t+8}^{j+3}, k_{t+9}^{j+3}, \dots\}$$

Using these keys, node A and B can only communicate in generations $j+3$ and $j+4$ using the set of $\{k_{t+1}^{j+3}, k_{t+8}^{j+3}, k_{t+9}^{j+3}, k_{t+8}^{j+4}\}$ keys in $k_{AB}^{j+3} = h(k_{t+1}^{j+3} \oplus k_{t+8}^{j+3} \oplus k_{t+9}^{j+3})$ and $k_{AB}^{j+3} = h(k_{t+8}^{j+4})$ manner. However, they cannot communicate in any other generation using these two key groups.

IV. Performance Evaluation

A. *Simulation Setup*

We performed several simulations and compared our scheme with RoK scheme. In these simulations, we have set the key pool size to 10,000 keys for both schemes. We have placed sensor nodes to the environment in totally random manner to have more realistic simulations. We have used 1,000 sensors on $400m\,x\,400m$ square environment. Communication range for nodes is set to $40m$. G_w is set to 10 and sensor nodes have a random lifetime that is determined using a Normal distribution function with mean $G_w/2$ and standard deviation $G_w/6$ as in [1]. As explained before, g value is set to be 5 which is $G_w/2$. We have also assumed that each generation consists of 10 small time units called *rounds*. Dead nodes are replaced with new randomly placed nodes at the beginning of each generation. Simulations are run for 30 generations. Also, we have run all of our simulations for 25 times and took their average values.

B. *Connectivity Analysis*

Simulations on connectivity analysis of RoK and HaG schemes are done using key ring sizes of 200, 220 and 250 keys. Nodes are moved using random walk mobility model and no changes observed for different network topologies.

Global Connectivity of the network stands for the percentage of the largest key sharing graph members over the size of the network. With the specified key ring sizes, both RoK and our scheme have 100% global connectivity.

Local Connectivity stands for the probability that any two neighbor sensor nodes share at least one common key in their ring. Figure 2 shows the Local Connectivity values for both RoK and HaG schemes using different key ring sizes. As seen from this figure, nodes in both schemes have 0.8 Local Connectivity value when using 220 keys for HaG scheme and 250 keys for RoK scheme. For a WSN, having 80% Local Connectivity can be considered as more than enough for secure communication in the network.

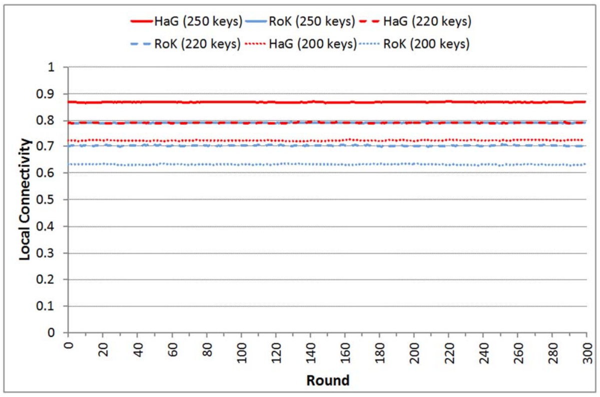


Figure 2. Local Connectivity of Rok and Our Scheme

C. *Resiliency Analysis*

In our simulations, attacker actively captures 1, 3 and 5 random nodes per round and compromises all of the keys available in their memory. Key ring size is set to 220 for HaG scheme and 250 for RoK scheme in order to have the same local connectivity value, which is around 0.8 as seen in Figure 2. Figure 3 and 4 shows the resiliency comparison of RoK scheme and our scheme; the lower the resiliency, the better. In summary, our simulations have shown that HaG scheme outperforms RoK scheme in resiliency by using smaller key ring size.

Active resiliency ratio is calculated using nodes that are currently alive and has some keys compromised because attacker has captured some other nodes that are able to communicate. As it can be seen in Figure 3, active resiliency ratio reaches its highest value in around 10th generation when most of the nodes that are deployed at the 5th generation are still alive. After 10th generation, nodes that are deployed at 5th generation start to die because their lifetime is determined with the aforementioned Normal distribution. Our results show that our scheme performs nearly 50% better when the attack rate is low, i.e. attacker captures 1 node per round. Although increasing attack rate affects the performance of our scheme negatively and it increases in a faster way, our results are still better than RoK scheme with attack rate of 5 nodes per round.

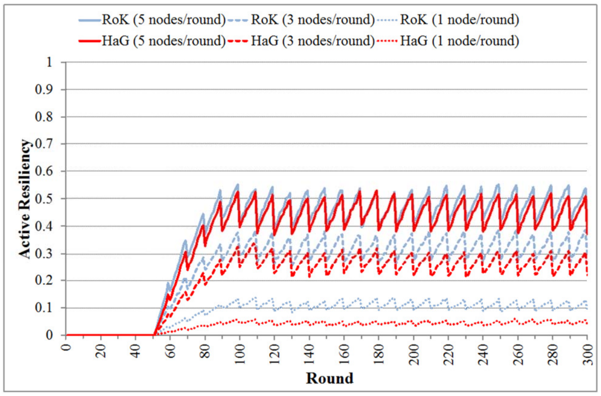


Figure 3. Active Resiliency of RoK and Our Scheme with an eager attacker having capture rates of 1, 3 and 5 nodes per round

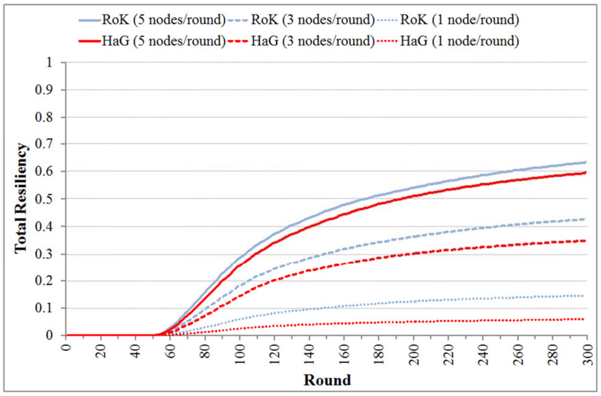


Figure 4. Total Resiliency of RoK and Our Scheme with a temporary attacker having capture rates of 1, 3 and 5 nodes per round

Total resiliency is calculated by considering all dead (i.e. captured) or alive links that are established over the course of the network. Our simulations have shown that total resiliency of HaG scheme also outperforms the RoK scheme as it can be seen in Figure 4. Similar to the active resiliency, HaG scheme has nearly 50% better results when the attack rate is low. When the attack rate increases, HaG scheme still has lower total resiliency rate compared to the RoK scheme.

V. Conclusion

In this paper, we propose a new key predistribution scheme that is designed for multiphase wireless sensor networks. Our scheme starts with an initial set of random key pool that evolves over time, in a graph fashion, to generate key pools for the subsequent generations. Sensors deployed at different generations start with a key ring that is randomly selected from the key pool of their deployment generation in groups. Deploying keys in groups increase connectivity and decreases resiliency. An attacker capturing a node can only compromise keys for generations bounded by the key group size.

Our simulations have shown that after anchoring the local connectivity value to 0.8 for both our scheme and RoK scheme, resiliency performance of our scheme is 50% better when the attack rate is small. When the attack rate increases, our scheme performs at a rate that is close to the performance of RoK but still better.

References

[1] C. Castelluccia and A. Spognardi, “RoK: A robust key pre-distribution protocol for multi-phase wireless sensor networks,” in *Proceedings of the 3rd International Conference on Security and Privacy in Communications Networks*, 2007, pp. 351–360.

[2] M. Ergun, A. Levi and E. Savas, “A resilient key pre-distribution scheme for multiphase wireless sensor networks,” in *Proceedings of the 24th International Symposium on Computer and Information Sciences*, IEEE Computer Society, Washington, DC, USA, 2009, pp. 375–380.

[3] M. Ergun, A. Levi and E. Savas, “Increasing Resiliency in Multi-phase Wireless Sensor Networks: Generationwise Key Predistribution Approach,” in *The Computer Journal*, vol. 54 (4), pp. 602–616, 2011.

[4] M. A. Simplício, Jr., P. S. L. M. Barreto, C. B. Margi and T. C. M. B. Carvalho, “A survey on key management mechanisms for distributed Wireless Sensor Networks,” in *Computer Networks*, vol. 54 (15), pp. 2591-2612, October, 2010.

[5] 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,* Washington, DC, USA, 2003, pp. 197–213.

[6] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002, pp. 41–47.

[7] R. Blom, “An optimal class of symmetric key generation systems,” in *Proceedings of the EUROCRYPT 84 Workshop on Advances in Cryptology*, Springer, Berlin, 1985, pp. 335–338.

[8] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz and A. Khalili, “A pairwise key pre-distribution scheme for wireless sensor networks,” *ACM Transactions on Information and System Security* vol. 8 (2), pp. 228–258, May, 2005.

[9] O. Z. Yilmaz, A. Levi, and E. Savas, “Multiphase deployment models for fast self healing in wireless sensor networks”, in *Proceedings of International Conference on Security and Cryptography*, 2008, pp. 136–144.

[10] K. Kalkan, S. Yilmaz, O. Z. Yilmaz, A. Levi, “A highly resilient and zone-based key predistribution protocol for multiphase wireless sensor networks,” in *Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks*, NY, USA, pp. 29–36.