

A new key management scheme in heterogeneous wireless sensor networks

Saber Banihashemian*, Abbas Ghaemi Bafghi*

*Department of Computer Engineering

Ferdowsi University of Mashhad (FUM), Mashhad, Iran

saberbanihashemi@yahoo.com, ghaemib@ferdowsi.um.ac.ir

Abstract— Key management is considered as the fundamental part of any secure communication. A secure wireless sensor network communication protocol relies on an efficient key management system. In sensor networks some random pre-distribution schemes are proposed for key management protocols. Connectivity and resiliency are two important criteria in key management. There is a trade-off between connectivity and resiliency in random key pre-distribution since increasing one decreases the other one and vice versa. In this paper we propose a new key management scheme in heterogeneous wireless sensor networks in which tradeoff between these criteria is lower than other schemes. This scheme consists of four stage key pre-distribution, localization and seed assignment, deriving new keys and shared key discovery. Simulation and analytical results show that proposed scheme has higher connectivity and resiliency and needs lower storage space compared to AP and EG key management schemes.

Keywords— security, key management, wireless sensor networks, heterogeneous.

I. INTRODUCTION

A wireless sensor network (WSN) consists of a large number of tiny sensor nodes which have limited computing capability and energy resources which can be deployed anywhere, and work without any assistance [1]. These characteristics introduced critical security issues like network access control, authentication, confidentiality and compromising nodes. Key management is crucial to the secure operation of Wireless Sensor Networks.

In recent years, some proposed scheme use a few number of powerful sensors along with large number of ordinary sensors. They utilize powerful sensors in key management for reducing storage overhead on sensors [3] and [4]. These schemes, take advantage of the powerful high-end sensors (H-sensors) in a Heterogeneous Sensor Network (HSN). They utilize the large storage of H-sensors and pre-load each H-sensor with a relatively large number of keys. They also reduce the total storage space for key pre-distribution while achieving significant reduction on sensor storage. They can load fewer keys in each sensor that doesn't have tamper resistant hardware. Hence, when a node is compromised,

fewer keys are revealed and as result smaller fraction of links in the network are compromised due to key exposure resulting from node capture.

In this paper, we propose a new key distribution scheme based on random key distribution for heterogeneous sensor networks. This scheme is based on [3] and use heterogeneity for key management. New aspect of our work is that we use separate keys in different clusters and take into account distance of sensors from their cluster head. Some base keys are preloaded in sensors and after deployment new keys are derived concerning that sensor belong to which cluster and how far is from cluster head. We consider two criteria, connectivity and resiliency, for comparing our scheme with [2,3]. Connectivity in this paper is fraction of physical links that we can establish a secure link. We define physical link as links between two neighbor sensors that stand in the same cluster and difference of their *levels* is equal or smaller than one. We explain *level* in section four.

In resiliency criterion, we consider the fraction of links in the network that are compromised due to key exposure resulted from node capture.

The proposed scheme reduces the storage requirements along with increasing connectivity and resiliency. The rest of paper is organized as follows: Section 2 provides the related work and Section 3 describes the proposed scheme. Section 4 gives the results and performance evaluation. Finally, Section 5 concludes the paper.

II. RELATED WORKS

Eschenauer and Gligor [2] propose a probabilistic key pre-distribution technique to bootstrap the initial trust between sensor nodes. It generates a large pool of random symmetric keys and then preconfigured each node with a number of keys randomly selected from the key pool. Neighboring nodes use their preconfigured keys to set up their pairwise keys. A communication channel secured between two nodes using pairwise keys is called a key path. To protect confidentiality, every key is usually assigned an index, and during shared key discovery, nodes exchange the index of keys with neighbors to ultimately determine their shared pairwise keys. Finally, during path-key establishment phase, pairs of neighboring nodes that do not share a key can set up their own keys, as long as they are connected by two or more key path at the end of shared key

discovery. If the network density, the size of the key pool, and the number of keys preconfigured in each sensor node are carefully chosen, it is highly likely that all nodes in the network will be connected via key paths.

Chan et al. [5] propose the q -composite key pre-distribution, which allows two sensors to setup a pairwise key only when they share at least q common keys for improving resiliency against node capture. Chan and Perrig [6] also develop a protocol named PIKE for key establishment by using peer sensor nodes as trusted intermediaries.

Some location-aware schemes which improve the security of the key pre-distribution schemes are proposed in [7, 8]. The idea of threshold key pre-distribution schemes is proposed in [9] and further studied in [10].

Du et al. [3] proposed the Asymmetric Pre-distribution (AP) key management scheme. Its main idea is to pre-load a relatively large number of keys in each one of a small number of powerful nodes (H-sensor), while only a small number of keys are stored in each one of nodes (L-sensor) which have very limited space of storage and capacity of communication. Due to the usage of these two types of nodes, two different types of key rings have been used to achieve a high probability that each two nodes share at least one shared key. Indeed, AP scheme is more scalable than the basic scheme and it reduces the number of pre-loaded keys compared to the basic scheme, but it is still sometimes unsuitable for some types of sensors due to memory constraints. Also, in order to reduce the storage requirements and to maintain the same security strength of the basic scheme and AP scheme [2, 3], Hussain et al. [11] proposed a key generation process to reduce the key storage requirement.

Liu et al. [12] propose a framework for key management schemes in distributed wireless sensor networks with heterogeneous sensor nodes. Traynor et al. [4] demonstrate that a probabilistic unbalanced distribution of keys throughout the network that leverages the existence of a small percentage of more capable sensor nodes cannot only provide an equal level of security, but also reduce the consequences of node compromise.

Traynor et al. [13] characterize the effects of the unbalanced key management system, and design a complementary suite of key establishment protocols known as LIGER. Using their pre-deployed keys, nodes operating in isolation from external networks can securely and efficiently establish keys with each other.

III. NETWORK MODEL

Base Station is assumed secure and have unlimited resources such as energy, memory, and processing power. Compared to L-sensors, H-sensors have more memory and processing capability but they are limited. These nodes communicate directly with the base station.

A. Assumptions

The following assumptions exist:

- H-sensors and L-sensors are assumed to be uniformly and randomly distributed in the field. H-sensors act as cluster heads and L-sensors are part of clusters.
- We assume that adversaries will not be able to compromise a node for a small interval initially after the node is deployed. After this initial interval an adversary might be able to compromise any node. A similar assumption is also made in [14].
- Due to cost constraints, L-sensors are not equipped with tamper-resistant hardware, so an adversary can extract all key material, data, and code stored on compromised L-sensor.
- H-sensors are equipped with tamper-resistant hardware.
- Each H-sensor and L-sensor is static.
- Base stations are trusted.
- H-sensors have large transmission range so that most L-sensors can receive Hello messages from one or more H-sensors.
- We don't have any assumption on number of sensor nodes in a cluster and cluster's scope.
- Each H-sensor is equipped with GPS and knows its location.

B. Notations

We use the following notations to describe our key management protocol and the involved cryptographic operations in this paper.

BS: is Base Station

adv_i : is advertisement message by i th cluster head

CH_i : cluster head of i th cluster

K_{BS-i} : pairwise key between BS and node with ID i

$seed_{i,j}$: seed related to i th cluster and j th level

S : total number of seeds used in entire network

S_p : minimum number of seeds needed by protocol

S_d : additional seeds that need after cluster formation

$E_K(M)$: encrypt message M by key K

$D_K(M)$: decrypt message M by key K

$hash(K,seed)$: hash key K with $seed$

$dist$: distance between adjacent levels

BK_i : i th base key

$DK_{i,j}$: i th key hashed by seed j

K_N : common key used by all nodes in network

K_{Cm} : cluster key used by all nodes in cluster m

K_{BS-CHi} : pairwise key between CH_i and BS

IV. PROPOSED PROTOCOL

Our scheme is based on [3]. The main idea is utilization of cluster information and also node distance from cluster head in key management. We use a concept called level. Nodes belong to a level based on their distance from cluster head. This distance can obtain based on RSSI [15] during clustering phase.

Each level has separate seed used for deriving new keys that are only used in that level and neighbor level. Therefore, network is partitioned to sections that have different keys from each other. Our scheme consists of four stages: key pre-distribution, localization and assigning seeds, deriving new keys and shared key discovery. Key pool consists of base keys

and derived keys. Derived key are hash of base keys with different seeds. Number of seeds is large enough for satisfying key management requirements. There are minimum number of seeds, S_b that we set $s_b = \text{number of H-sensors}$. Based on desired key pool size, we obtain number of base keys using division of desired key pool size on S_b . We use S_b to make a connection between our scheme to AP and EG scheme for comparison purpose. In fact, we don't need this parameter in the proposed scheme. We use additional seeds, S_d , that are used after node deployment. Table 1 shows structure of key pool.

A. Pre-distribution phase

In the first stage, we generate a key pool composed of two types of keys as mentioned above. We just store base keys into sensor nodes and don't use derived keys in this phase. For this, we store k base keys that will be randomly selected, in each sensor node. We store c base keys into each H-sensor in which $c \gg k$. Also K_{BS-CH_i} used as pairwise key between CH_i and BS. In addition to base keys in L-sensors, a pairwise key between sensor and BS will be stored into L-sensors. This key is used for authentication by BS. There is a key in network, K_N , which is stored into each L-sensor and each H-sensor. This key is used for generating cluster key.

B. computing number of seeds needed for each cluster

We can estimate the area in which nodes are deployed but there is no assumption as to where nodes are placed. For this reason we don't know how many seed we can use. But since we use separate keys for each cluster, the number of clusters is the minimum seeds that we can have. Each H-sensor becomes a cluster head. After node deployment, H-sensors obtain their location by GPS and send it back to BS. When BS receives this locations information, it knows the location of each cluster head and can estimate maximum distance a point in cluster can have. For estimating maximum distance of a point in a cluster, from cluster head, we make a grid of deployment area. For example in a $400m \times 400m$ area, we make cells $2m \times 2m$ and therefore we have 40000 cells. For each cell in grid we assign a virtual coordinate x_{cell} and y_{cell} . Then we specify that each cell belong to which cluster, based on virtual grid and the nearest cluster head. We use maximum distance of a point for each cluster to determine seed count. At first, maximum distance of a point for cluster is 0. For each cell, we compare the distance of the cell to its cluster head's cell with maximum distance of a point to that cluster head. If the distance of the cell to its cluster head's cell was bigger, then we replace this distance with 20 of

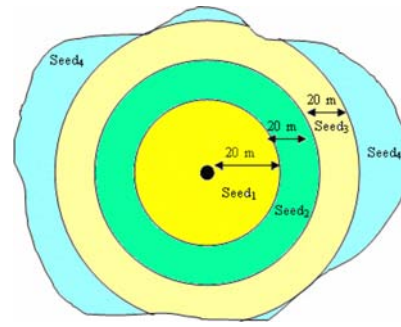


Figure 1. cluster and level formation

cluster maximum distance of a point to that cluster head. After this procedure, with respect to maximum distance of a point to each cluster head and distance that a seed is used, BS send some seeds to each cluster head.

In the proposed scheme, minimum distance of utilizing a seed is equal to propagation distance of an L-sensor. BS specifies some seeds for each cluster based on maximum distance of a point to each cluster head and distance of seed utilization. For example BS decides to use each seed in distance 20, namely, $seed_1$ is used by nodes in distance 0 to 20 of cluster head, $seed_2$ is used by nodes in distance 21 to 40 of cluster head and so on. It is needed to be mentioned that the smaller the distance of seed utilization, the more the number of seeds and the more resilient the proposed scheme. Distance of seed utilization in each level can vary for balancing purpose. An example of this is shown in figure 1.

C. Computing new keys by seed

After deploying nodes and clustering, some seeds are sent to cluster's nodes by their cluster head. In addition to separating keys of different clusters, we can divide clusters to different levels. For computing new keys, nodes must know their distance from their cluster head so that we can use localization techniques for this goal. Cluster formation and specifying distance to cluster head could be done together as mentioned before. After this stage, each cluster head take seeds from BS with respect to maximum distance of a point to that cluster head and the distance of seed utilization. After deploying nodes, each cluster head propagate a Hello message and nodes compute their distance from cluster heads based on RSSI and join to the cluster that has minimum distance to its cluster head. Then nodes compute cluster key by $K_{Cm} = \text{hash}(K_N || ID_{CHm})$. At the same time, BS sends the related seeds to each cluster head.

$$BS \rightarrow CH_i : E_{BS-CH_i} ([seed_{i,1}, dist_1], [seed_{i,2}, dist_2], \dots, [seed_{i,z}, dist_z])$$

Then the cluster head computes new keys by the received seeds and forwards seeds to nodes in the cluster and encrypts this message by cluster keys.

$$CH_i \rightarrow Node : K_{C_i} ([seed_{i,1}, dist_1], [seed_{i,2}, dist_2], \dots, [seed_{i,z}, dist_z])$$

Then each node computes its distance from cluster head and uses the related seeds to produce new Derived keys. Each node,

Table 1. key pool of proposed scheme

Seed _s	DK _{1-s}	DK _{2-s}	DK _{3-s}	DK _{n-s}
Seed ₃	DK ₁₋₃	DK ₂₋₃	DK ₃₋₃	DK _{n-3}
Seed ₂	DK ₁₋₂	DK ₂₋₂	DK ₃₋₂	DK _{n-2}
Seed ₁	DK ₁₋₁	DK ₂₋₁	DK ₃₋₁	DK _{n-1}
Base Keys	BK ₁	BK ₂	BK ₃	BK _n

in addition to producing its level key, produces keys of next level for connectivity between adjacent levels.

$$\begin{array}{ll} K_{j1} = \text{hash}(K_{j1}, \text{seed}_{i,z}) & K_{j1} = \text{hash}(K_{j1}, \text{seed}_{i,z+1}) \\ K_{j2} = \text{hash}(K_{j2}, \text{seed}_{i,z}) & K_{j2} = \text{hash}(K_{j2}, \text{seed}_{i,z+1}) \\ \vdots & \vdots \\ K_{jk} = \text{hash}(K_{jk}, \text{seed}_{i,z}) & K_{jk} = \text{hash}(K_{jk}, \text{seed}_{i,z+1}) \end{array}$$

After producing new keys, nodes delete seed, Base keys and cluster keys for preventing the revealing seeds and base keys after compromising a node.

D. Shared key discovery

In the next stage, adjacent nodes find common keys with each other. Each node propagates list of its keys and its level, and encrypts this message with cluster key.

Neighbor nodes establish their common keys directly. If there are multiple common key between two nodes, we select one common key randomly. If a common key doesn't exist between two nodes X and Y, each of them send a request message including its ID, its level, List of its keys and ID of another node to cluster head. Cluster head that have more keys, have more chance of having a common key with each node. Cluster head select a key randomly and forwards it to X and Y, provided that it has common key with two nodes. If cluster head didn't have common key, it forwards messages of X and Y to BS, and BS sends a share key to X and Y by cluster head.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our key management scheme.

A. Connectivity and key pool size calculation

In this section we obtain probability of establishing a secure link between two L-sensors. Then for specified probability, and consideration of maximum storage space in L-sensors and H-sensors that we can allocate for key management, we can estimate maximum key pool size in scheme. Probability of establishing a secure link between two L-sensors computes as follow:

$$P = [(two\ L\text{-sensors}\ have\ common\ key\ with\ each\ other)] + [(1\text{-two}\ L\text{-sensors}\ have\ common\ key\ with\ each\ other) * (probability\ that\ two\ L\text{-sensors}\ have\ one\ common\ key\ with\ cluster\ head)] \quad (1)$$

We first consider the probability that any two L-sensors, say n_i and n_j , share at least one key,

$$P_1 = 1 - (probability\ of\ that\ two\ L\text{-sensors}\ don't\ have\ common\ key) \quad (2)$$

Probability of key sharing between two L-sensors is the same as [2]:

$$P_1 = 1 - \frac{((b-k)!)^2}{b!(b-2k)!} \quad (3)$$

If there is not common key between two L-sensors, they communicate with the cluster head that have larger key ring.

For establishing a secure link between two L-sensors and cluster head, a common key must exist between each of L-sensors and cluster head. We name probability of existence a common key between an L-sensor and cluster head as P_2 . Therefore, probability of establishing a secure link between two L-sensor through a cluster head is $P_2 * P_2$. Obtaining P_2 is mentioned in [3]. Here, we have P_2 as follow:

$$P_2 = 1 - \frac{(b-c)!(b-k)!}{b!(b-c-k)!} \quad (4)$$

So we can write probability of establishing secure link between two L-sensors in proposed scheme as follows:

$$P = P_1 - [(1-P_1) * (P_2)^2] \quad (5)$$

Using equation (5) we can determine for a specified key pool size the number of keys we must store in each L-sensor and H-sensor that obtain desired probability of existence a secure link. We compare our scheme with AP scheme based on storage requirement in L-sensor when the number of keys in H-sensors is the same. We have a key pool with the size 10000. Minimum number of seeds, S_b , is 10 and number of base keys are 1000; therefore we have 1000 derived keys. Of course, in the proposed scheme S_b doesn't have any effect on connectivity and only the number of base keys determines connectivity. We compare two schemes when probability of key sharing is greater than 0.5. Result of this comparison is shown in figure 2. As we can see, for the same probability of key sharing and same key ring size in H-sensors, our scheme has significant storage saving compared to AP scheme. In this comparison, our scheme improves storage saving 6 times more than AP scheme.

For comparing probability of key sharing of our scheme with AP scheme, we draw 3D graph of probability of key sharing for key pools $P=5000$ for AP scheme according to the initial number of preloaded keys. In this comparison, resiliency of two schemes is equal for the same c and k . The number of clusters in scheme is 10 and therefore we set $S_b=10$. We compute b based on P and S_b , namely, from division of P on S_b , we obtain b . therefore we have $b=500$. In this method, probability of key sharing is based on (4).

For the same c and k , results are shown in figure 3 and as it can be seen our scheme has better probability of key sharing than AP scheme.

B. Security analysis

In this Section, we analyze the resilience of our key management scheme against node compromise attack. We attempt to find out the effect of compromising t L-sensors on the rest of the network. Each L-sensor is preloaded with k key. From these k base keys, we obtain $2k$ derived keys. The probability of a given key l belonging to an L-sensor's key ring is $2k/(b*(S_b+S_d))$, where b is the number of base keys and S_b is the number of base seeds and S_d is the number of the remained seeds that compute as $S_d = total\ number\ of\ seeds - S_b$. We show total number of seeds with S .

Therefore, the probability of l not in an L-sensor's key ring is $1-2k/(b*s)$. Furthermore, the probability of l not in any of

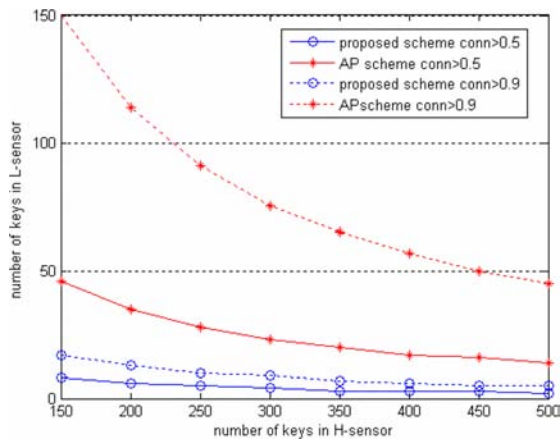


Figure 2. minimum number of keys in L-sensors that needed by proposed scheme and AP scheme for connectivity greater than 0.5 and 0.9

the key rings of the t L-sensors is $(1 - 2k/(b*s))^t$. Thus, the probability of a given key l in any of the key rings of the t L-sensors is: $P_{reveal} = 1 - (1 - 2k/(b*s))^t$.

We evaluate our proposed scheme by simulation and compare it with AP scheme and EG scheme. We evaluate the proportion of total established links that an adversary can compromise based on the key information retrieved from c captured nodes. Here, we assume that an adversary can eavesdrop the entire network on the first moment. In AP scheme and in proposed scheme, we consider two keys for a secure link if established by cluster head. For a secure indirect link which is established by two keys, we say the link is compromised if one of these keys is revealed. When a node in network is compromised, the adversary can compromise links that are established by keys stored in the compromised node. We measure proportion of the compromised link to total secure established links and name this criterion as *proportion of compromised links*. When a node is compromised, the links of the compromised node is established with its neighbor nodes, will be compromised. Ideally the compromised links only belong to compromised nodes. In the other words, nodes establish pairwise keys. In *proportion of compromised links* we use this ideal as base criterion of *proportion of compromised links* and also for determining efficiency of scheme that is under consideration.

We compare our scheme with AP and EG scheme. We simulate these schemes in Matlab and run each of them 100 times. We employ 1000 L-sensors and 10 H-sensors as cluster head randomly in 400m*400m area. We set 40m for the distance that a seed is used and also 40m for propagation range of an L-sensor. For our scheme and AP scheme, we store 400 base key in each H-sensor. The size of key pool in AP scheme and EG scheme is 10000. In the proposed scheme, the number of base keys is 1000 and total number of seeds are $S=30$, therefore, we have 30000 derived key.

We compared these three schemes two times. In the first comparison, we stored the same number of keys in each L-sensor. In another one, we considered the effect of node

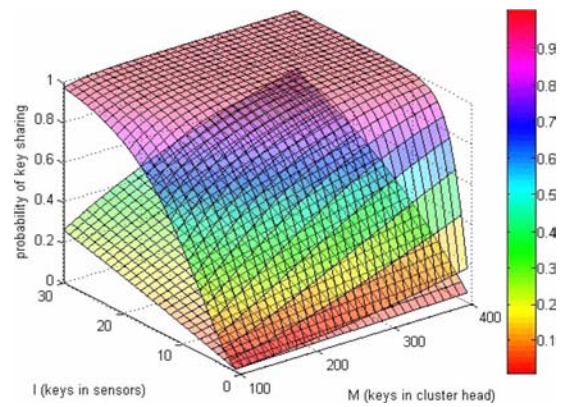


Figure 3. comparing probability of key sharing between AP and our scheme- parameter for AP scheme $P=10000$ and parameter for proposed scheme $S_b=10, b=500$

compromised when these schemes have the same probability of key sharing.

In the first study, we stored 20 keys in each L-sensor in three schemes. We compared these schemes on the basis of *proportion of compromised links*. Result of study is shown in figure 4.

Our scheme improves on proportion of compromised links whereas in this situation, the probability of key sharing in the proposed scheme is 0.9999, in AP scheme is 0.3369 and in EG scheme is 0.0393. In our scheme and AP scheme, we take into account keys in H-sensors for computing probability of key sharing.

In the second study, we consider the effect of node compromise when these schemes have the same probability of key sharing 0.8. For this purpose, in the proposed scheme we store 5 base keys in each L-sensor to obtain probability of key sharing 0.8552; in AP scheme we store 40 keys in each L-sensor for obtaining probability of key sharing 0.8053; and in EG scheme we store 127 keys in each L-sensor for obtaining probability of key sharing 0.8048. Result is shown in figure 5.

As we see in figure 5, when connectivity in the three schemes is greater than 0.8, our scheme decreases the effect of compromising L-sensors on secure links.

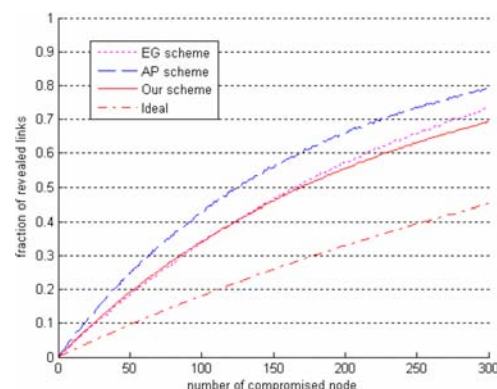


Figure 4. fraction of compromised links when the same number of keys preloaded in each L-sensor

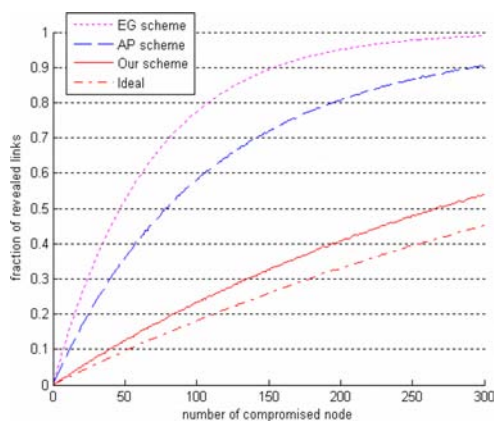


Figure 5. fraction of compromised links when probability of key sharing greater than 0.8

VI. CONCLUSION

In this paper, we propose a new key management scheme based on random key pre-distribution. We use a large number of L-sensors and a small number H-sensors as used in [3]. H-sensors act as cluster head. In our scheme we use some base keys that are preloaded in L-sensors and H-sensors and after deployment we assign two seeds to each L-sensor according to their clusters and distance from their cluster heads. In our scheme, the number of base keys has effect on connectivity between nodes and number of seeds has effect on resiliency against node capture. In proposed scheme we can increase probability of key sharing and resiliency against node capture at the same time. At the end, we analyzed the proposed scheme based on resiliency and connectivity and compare it with previous works [2,3] and results show that our scheme is improved.

ACKNOWLEDGMENT

This method has been sponsored by Iran Telecommunication Research center (ITRC), Tehran, Iran. This support is gratefully acknowledged.

REFERENCES

- [1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y.: Wireless sensor networks: a survey. *Computer Networks* 38, 2002 .
- [2] L. Eschenauer, V.D. Gligor, A key management scheme for distributed sensor networks. In: *Proceedings of (CCS '02.)*, pp. 41-47, Washington, 2002.
- [3] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen. An effective key management scheme for heterogeneous sensor networks, *Ad Hoc Networks*, 5(1):24–34, 2007.
- [4] P. Traynor, P. Kumar, H. Bin Saad, G. Cao, T. La Porta, Establishing pair-wise keys in heterogeneous sensor networks, In: *INFOCOM 2006, 25th IEEE international conference on computer communications. Proceedings*, 2006.
- [5] H. Chan, A. Perrig, and D. Song, Random key pre distribution schemes for sensor networks, In *IEEE Symposium on Research in Security and Privacy*, 2003.
- [6] H. Chan H, A. Perrig, Pike: peer intermediaries for key establishment in sensor networks. In: *INFOCOM 2005, 24th annual joint conference of the IEEE computer and communications societies*, pp 524–535, 2005..
- [7] D. Liu D, P. Ning, Location-based pairwise key establishments for static sensor networks. In: *SASN '03: proceedings of the 1st ACM workshop on security of ad hoc and sensor networks*, ACM, Press, New York, pp 72–82, 2003.
- [8] Y. Zhang , W. Liu, W. Lou, Y. Fang, Securing sensor networks with location-based keys, In: *Wireless Communications and Networking Conference*, 2005.
- [9] R. Blom R, Non-public key distribution. In: *Advances in cryptology—CRYPTO 82*, pp 231–236, 1982.
- [10] C. Blundo C, A. De Santis, Herzberg A, Kuttan S, Vaccaro U, Yung M, Perfectly-secure key distribution for dynamic conferences. In: *CRYPTO '92: proceedings of the 12th annual international cryptology conference on advances in cryptology*, London, UK, 1993. Springer, New York, pp 471-486,1993.
- [11] S. Hussain., F. Kausar, A. Masood, An efficient key distribution scheme for heterogeneous sensor networks. In: *the international conference on wireless communications and mobile computing*, pp. 388-392.Hawaii(2007)
- [12] K. Lu, Y. Qian, J. Hu, A framework for distributed key management schemes in heterogeneous wireless sensor networks. In: *IEEE international performance computing and communications conference*, pp 513–519,2006.
- [13] P. Traynor, R. Kumar, H. Bin Saad, G. Cao, T. La Porta, Efficient hybrid security mechanisms for heterogeneous sensor networks. *IEEE Trans Mobile Comput* 6(6):663–677, 2007.
- [14] S. Zhu, S. Setia, and S. Jajodia. Leap: Efficient security mechanisms for large-scale distributed sensor networks. In *Proc. of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, Oct 2003.
- [15] G. Mao, B. Fidan, B.D.O. Anderson, Wireless sensor network localization techniques. *Computer Networks*, Volume 51, Issue 10, 2007.