The ISC Int'l Journal of Information Security

ISeCure

January 2017, Volume 9, Number 1 (pp. 27–39)

http://www.isecure-journal.org

LPKP: Location-based Probabilistic Key Pre-distribution Scheme for Large-Scale Wireless Sensor Networks Using Graph Coloring☆

Alireza Ahadipour [1,*], and Alireza Keshavarz-Haddad [1]

[1] *School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran*

ARTICLE INFO.

Article history:
Received: 10 May 2016
Revised: 1 January 2017
Accepted: 18 January 2017
Published Online: 22 January 2017

Keywords:
Random Key Pre-distribution, Symmetric Key Management, Probabilistic Key Sharing, Random Graph, Graph Coloring, Wireless Sensor Network.

ABSTRACT

Communication security of wireless sensor networks is achieved using cryptographic keys assigned to the nodes. Due to resource constraints in such networks, random key pre-distribution schemes are of high interest. Although in most of these schemes no location information is considered, there are scenarios that location information can be obtained by nodes after their deployment. In this paper, we propose a novel probabilistic key pre-distribution scheme, for large-scale wireless sensor networks which utilizes location information in order to improve the performance of random key pre-distribution substantially. In order to apply the location information of the nodes in key distribution process, we partition the network into some regions and use graph coloring techniques to efficiently assign the random keys. The proposed scheme has a superior scalability by supporting larger number of nodes and also increasing the probability of existence of a shared exclusive key among the nearby nodes, i.e., the probability of having an isolated node is significantly reduced in comparison with the existing random key pre-distribution schemes. Our simulation results verify these terms.

1 Introduction

Achieving high level of security in networks which communicate over an open media such as wireless networks is an important issue. In order to check confidentiality and integrity of data messages in such networks, the messages need to be encrypted. It is necessary to distribute a number of secret keys among the network nodes for establishing secure encrypted communications. However, providing a key management mechanism for a large-scale network without an additional infrastructure is a very challenging task.

In this paper, we study large-scale stationary wireless sensor networks in which the nodes have constraints on communication capability, computation power, storage space. Also, there are constraints on network features like lack of a prior knowledge of post-deployment configuration, and vulnerability of nodes to physical capture. Notice that it is not sensible to entrust the responsibility of agreement on keys to ordinary nodes, e.g., Diffie-Hellman algorithm, in an extensive network; because, in addition to the restrictions of nodes, high traffic congestion is imposed to the network. Furthermore, in this case nodes need to somehow authenticate themselves before establishing a secure link, which itself is a challenging matter.

☆ This article is an extended version of an ISCISC'12 paper.

* Corresponding author.

Email addresses: ahadipour.alireza@shirazu.ac.ir (A. Ahadipour), keshavarz@shirazu.ac.ir (A. Keshavarz-Haddad)

ISSN: 2008-2045

One solution for key management in large-scale networks is to assign keys in early stages such as during manufacturing or deploying stage of the network nodes. The proposed methods in the literature based on this idea consist of three phases: (i) *key pre-distribution*, (ii) *shared-key discovery*, and (iii) *path-key establishment*. In the first phase, keys are produced offline and are stored in the memory of the nodes. In the next phase, each node discovers its neighbors with whom it has a common key and establishes secure links with those neighbors. Finally, in the third phase, pair of neighbors which possess no shared-key, establish a path with the aid of those nodes that have a common key after the second phase and communicate with each other over that secure path. They can also use that secure path to establish a shared-key with each other [4].

In this paper, we propose a new *location-based probabilistic key pre-distribution (LPKP)* scheme in which a novel key assignment technique and a secret key establishment method are used in the first and second phases. In the presented scheme, the third phase is done similar to the existing methods. Before describing the idea of the LPKP key pre-distribution approach, we review some preliminaries on dividing the network into zones, concept of graph coloring, and different classes of assigned keys.

- First, we divide the area of the network into a number of zones, which are called *cells*. The size of each cell is set based on the radio range of nodes such that every two nodes in two adjacent cells be able to communicate with each other; as illustrated in Figure 1. It should be stated that these cells just define regions in which the location of nodes is important and they should not be confused with the concept of cells in cellular networks. Cells can be square, hexagon, etc. in shape in order to fit together and evenly cover the area of the network. However, in general, we can use different shapes to cover the network. In this paper, without loss of generality, we consider square cells to simplify illustrations of our key pre-distribution scheme and the analysis.
- We put upon the idea of graph coloring in our approach. Graph coloring is the assignment of colors to the elements of a graph subject to defined constraints. Simply, colors are assigned to either the vertices (vertex coloring) or the edges (edge coloring) such that no two adjacent vertices (or edges) share the same color [1–3]. In the LPKP scheme, a color is assigned to each cell in such a way that no two adjacent cells have the same color. For a network with square cells, we can use 9 distinct colors to fully color the network cells under this constraint. Figure 2 depicts a proper coloring (here, instead of different colors, we use various hachures) for the square cells in an area. Here, the colors are assigned deterministically, but this can be done in various forms with 4 or more number of colors. We assume that nodes are stationary (fixed in their location) after deployment; accordingly, we can use GPS coordinates in deployment phase to figure out the color of the cell in which each node is settled.
- Eventually, after formation of cells and definition of the colors, LPKP scheme assigns two classes of keys to each node based on its location and the color of its cell: *primary keys* and *secondary keys*.

  I Primary keys are assigned to the nodes randomly; and by using these primary keys, nodes can establish secure links with their neighbors. Since the allocation of primary keys to each node is done by chance, it is probable that some keys are used as shared-keys in different locations in the network. Therefore, a vulnerability is imposed to the network against node capture; which is common in most of pre-shared key methods. In order to address this susceptibility, secondary keys are defined in LPKP.

  II Secondary keys are assigned to the nodes based on the cell in which they are settled. Secondary keys are merged with primary keys in such a way that all primary keys that are common among nodes remain common after being combined with the secondary ones. Secondary keys are incorporated with primary keys in order to decrease the probability that a key is used as a shared-key more than one time in the whole network.

The scheme is described in details in Section 3. Note that from now on, we use the term "keys" for primary keys, unless it is mentioned secondary keys.

In the LPKP scheme, the size of key-pool only depends on the average number of neighbors for each node (which is proportional to the average number of nodes in each cell). This means that the size of key-pool does not grow by increasing the area of the network. Therefore, our approach possesses a superior

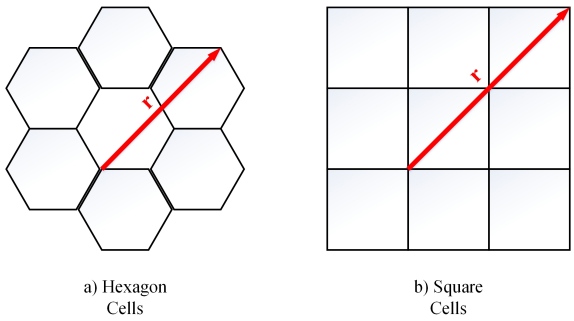


Figure 1. Relation between the size of each cell and the radio range of nodes.

scalability and probability of existence of a shared-key among neighbor nodes compared to the existing methods.

We analyze, evaluate and compare the performance of LPKP scheme in terms of the total number of keys in the networks, the size of key-rings for each node, connectivity degree of secure links, and etc. with the existing related schemes. Our simulation results verify the superiority of our proposed scheme in large-scale stationary wireless sensor networks. For instance, compared to other pre-shared key approaches, our presented scheme has a higher probability that the graph of the network become connected and it supports larger number of nodes. As a basic example, in LPKP, the probability that the graph of the network become connected is two times the basic scheme; or under the same conditions, our proposed method could support seven times larger networks than the basic scheme. On the other hand, in order to acquire the aforementioned advantages, LPKP approach needs to decrease the size of the key-pool which results in the fact that keys been used as primary keys more repeatedly. To overcome this shortcoming, in this work we present the idea of employing the secondary keys by the nodes in each cell.

Notably, the present work completes our previous conference paper [5] and provides more technical details and some new ideas for key pre-distribution.

The remainder of this paper is organized as follows: In Section 2, we briefly discuss related work. In Section 3, we describe our proposed LPKP scheme for key pre-distribution. Section 4 presents the analysis of LPKP scheme. In Section 5, the evaluation and simulations results of LPKP scheme are presented. Finally, we conclude the paper in Section 6.

2 Related Work

In this section we briefly review various probabilistic key pre-distribution schemes that can be employed in multi-hop wireless sensor networks.

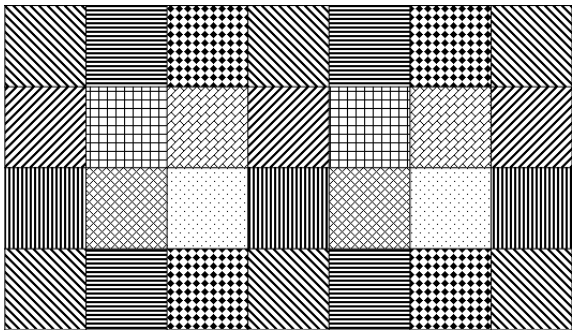


Figure 2. An example of partitioning the area of a network with square cells and definite color (hachure) assignment for each cell.

L. Eschenauer and V. D. Gligor proposed a random key pre-distribution scheme for wireless sensor networks in [4]. In this approach, a *key-pool* of KP distinct keys is generated randomly and a unique identification (ID) is assigned to each key. For each node, kr keys are randomly chosen from the key-pool and in accompany with the ID of each key is stored in the memory as the node's *key-ring*. After the network deployment phase, each node broadcasts a message to its neighbors which contains a list of its key-IDs. The neighbor nodes receive and compare the IDs with their own key-IDs. If there is a common ID for a neighbor node; it means that the neighbor and the node have a shared symmetric key. Our proposed scheme has been developed based the idea of this work, from now on, we refer to it as *basic scheme*.

The basic scheme has been used as a framework of posterior key pre-distribution schemes in wireless networks. For instance, [6] and [7] combined Blom's method [8], and Blundo's method [9], with the basic scheme respectively and presented a new scheme. [10] changed the basic scheme in such a way to gain a higher resilience against node capture. In this work, in order to establish a secure link between two neighbor nodes, the nodes require to have at least q keys (instead of at least one) in common among their key-rings.

Du *et al.* presented an approach in [11] which utilizes the basic scheme in accompany with the deployment knowledge to improve the performance of the basic scheme. It is assumed that the deployment of nodes is non-uniform. Hence, knowing that which nodes have the probability to settle near each other will benefit the key pre-distribution scheme. The idea beneath this approach is to divide the key-pool into smaller groups of key-pools so that each group of key-pools corresponds to a specific group of nodes which are more likely to be neighbors. The goal is to let key-pools corresponding to nearby groups of nodes share more keys compared to distant groups of nodes. In [12] Du *et al.* have proposed an improved scheme in which they have combined their presented scheme with their DDHV (Du-Deng-Han-Varshney) approach in [6].

Kong *et al.* were inspired by the scheme presented in [11] and proposed their own scheme. But, instead of utilization of rectangular partitions, they got use of hexagon ones and showed that under the same conditions, their presented method attains a higher probability of local connectivity and more resiliency against node compromise compared to [11].

In [14], Anjum presented a LDK (Location Dependent Key) management scheme; a location-based approach for key distribution in wireless sensor networks. This scheme not only increases the probability of connectivity of nodes, but also reinforces the resiliency

against node compromise. In this approach, a random set of keys are allocated to each node prior to deployment. After deployment phase, actual keys are chosen among those sets. Also, [16] uses the location of nodes in order to decide which keys should be assigned to each node. The author shows that this approach reduces the number of keys that is needed to be stored on the memory of each node, whilst compromise of one node will affect communications only in that neighborhood.

Faghani *et al.* used of Anjum's LDK approach and presented their own sectorized LDK scheme which does not require any information related to the deployment of nodes [15]. They contend that their presented scheme improved key resiliency and moreover, nodes could be added to the network at any time.

Gaur *et al.* proposed a scalable method in [17] which guarantees fully pairwise secure connectivity among all neighbor nodes.

There are also some key distribution schemes which utilize location information to achieve a particular goal. For instance, [18] and [19] present a location-based method which is used in hierarchical networks. [20] and [21] employ location-based approach in order to made the network resistant against insider threats and node compromise, respectively. Also, various key management schemes for wireless sensor networks are presented in [22] and [23]. A survey on key distribution schemes for wireless sensor networks has been provided in [22].

3 Proposed LPKP Scheme

In LPKP key management scheme, we consider a large-scale wireless sensor networks; in which, the nodes are stationary and distributed uniformly random over a large area.

The LPKP scheme is founded on Eschenauer and Gligor's method [4] (which we referred to as "basic scheme" henceforth). The main objective is to determine the size of key-pools for achieving a full and secure connectivity among nodes *with high probability (w.h.p.)*.

Our goal is to have a shared key w.h.p. between every two adjacent nodes to establish a secure link, and use these links we establish secure paths among the network nodes. Due to limited processing power and memory space of the nodes, we should introduce a distributed key pre-distribution scheme which requires low processing power and memory at each node. Foremost, as the radio range of the nodes is given, we have to consider the size of the cells in such a way that every two nodes in contiguous cells be within the radio range of each other. According to Figure 1, the relation between the radio range of each node and the length of each square cell is following,

$$r = 2(\sqrt{2} \times L) \tag{1}$$

where r is the radio range and L in the length of each cell.

3.1 Primary and Secondary Key Distribution

In LPKP scheme, key-rings that are stored in each node's memory depend on the color of the cell in which every node is going to reside. As mentioned in Section 1, the colors are assigned to the cells in such a way that every two adjacent cells have distinct colors. Obviously, there will be some neighbor nodes which are settled in two adjacent cells with different colors and they need to have a shared-key. To have this, primary keys are randomly selected from some key-pools which are correspondent to the colors of nearby cells and are stored in the corresponding key-rings.

For example, with assumption of square cells with 9 definite colors, we create 9 distinct key-rings for each node in order to communicate with nodes within its own cell or other 8 nearby cells. Each key-ring is selected from its corresponding key-pool. Note that totally, we have 45 disjoint key-pools; 36 cases are for choosing 2 cells out of 9 cells (i.e., $\binom{9}{2}$) when two nodes are settled in two different cells and 9 remaining cases are for choosing 1 cell out of 9 cells when two nodes belong to the same cell.

As the nodes are scattered in the network in different ways, it is possible that a node does not reside in the expected cell with the expected color. In this case, again, all the key-rings in the node's memory are usable; because, the latter cell is still neighbor to all 8 remaining cells. As a result, the misplaced node could find common keys with its neighborhood yet.

An alternative scenario for assigning keys to each node is to allocate all the 45 distinct key-rings to each node, while these key-rings are chosen from corresponding key-pools. In this scenario, it is assumed that nodes could reside in every cell with every color. So, it is not certain in prior that each node is going to be settled in which cell with the specified color. As a result, by assigning all possible key-rings, each node can find common keys with its neighbors and finally establish a secure link with them. In this paper, we base our assumptions on the first scenario and evaluate the performance this scheme.

In the start-up phase, each node figures out that with which neighbors it has common primary keys. If two neighbor nodes find just one key in common, they establish a secure link using that key; otherwise, if they find more than one common key in their key-

rings, they have to build a single shared-key in terms of those common keys to communicate with. Thus, they incorporate those keys to acquire a new key. This incorporation could be the output of a hash function which takes the common keys as the input or the XOR of those keys.

Furthermore, there is a chance that the shared primary key of two neighbor nodes are being used elsewhere in the network. This means if a node is compromised, it is probable that its keys are being used as shared-keys in other places in the network. This fact jeopardizes the security of other links and empowers the adversary to eavesdrop other communications.

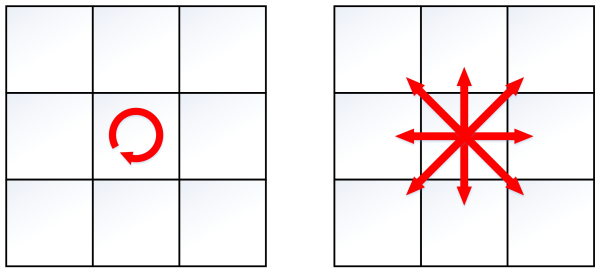
This made us to present *secondary keys* in order to increase the level of security in the presented scheme. As stated before, secondary keys are assigned to nodes regard to their locations. As a result, by utilization of secondary keys and combination of them with primary ones, the resulted keys would not appear elsewhere in the network. Consequently, after a node is captured, its key-rings are not functional for eavesdropping in other parts of the network.

In the LPKP scheme, after the start-up of the network, and after all those aforementioned phases, i.e. key pre-distribution, shared-key discovery, and path-key establishment, have passed, some secondary keys are needed to be assigned to nodes and be merged with primary ones in key-rings in order to alter the shared-keys in such a way that shared-keys will not be repeated elsewhere in the network. These secondary keys should be allocated in a way that after these keys are combined with primary keys, those adjacent nodes which have keys in common (established secure links), still have keys in common. In other words, by deployment of secondary keys the connectivity of the secure links should not be changed; neither any secure link should be removed nor new secure link should be added to the network.

For instance, one way to incorporate primary and secondary keys is to compute $hash(\text{Primary} \parallel \text{Secondary})$, where $\parallel$ denotes concatenation; or, $\text{Primary} \oplus \text{Secondary}$, where $\oplus$ denotes XOR operation. It is clear that if a primary key is common between two nodes, after being combined with a secondary key, the result would remain common between those nodes.

The following procedure is considered in the LPKP scheme in each cell in order to assign secondary keys to nodes:

1) A node is selected as a cluster-head in every cell. The selection of a cluster-head can be done randomly or based on a specific criteria [24–27].
2) Cluster heads pick randomly one key which is not used as a shared-key out of the key-ring corresponding to the cell in which they are settled, i.e. $kr_{i,i}$.
3) Thereupon, they send the chosen key to the neighbors through the formed secure links; including nodes in their own cell and in the adjacent cells; as depicted in Figure 3. As it is expected that the graph of the network be connected by utilization of primary keys, all the neighbors would receive the sent secondary key with one or more hop transmissions.
4) All the neighbors amalgamate the received secondary key with their shared-keys they possess with nodes in that cell and construct a new key. Hereinafter, those nodes get use of this new key to communicate with nodes within the cell they obtained the secondary key from.



a) Intra Cell Secondary Key Sharing b) Inter Cell Secondary Key Sharing

Figure 3. Employing a secondary key by the nodes in each cell.

It is not far from reality that by the start-up of the network, no node is compromised. As a result, by dispatching secondary keys through secure links that are constructed by primary keys, nodes could collect secondary keys securely and no secondary key would be disclosed to the adversary.

It should be stated that as each cell dispatches its own secondary key to its adjacent cells, shared-keys for transmission and reception differ from each other. For instance, suppose that nodes a and b reside in two adjacent cells and have a common primary key in their key-rings. Without loss of generality, both nodes a and b are considered as cluster-heads of their own cells. Node a issues its secondary key to all adjacent cells including the cell containing node b; and vice versa. Thus, the sent key from a to b varies from the key that b sends to a. As a result, after primary and secondary keys been combined, node a calculates a new secret key in order to encrypt sent messages to b, while b computes another key for encrypting the sent messages to a. Indeed, nodes a and b exploit different secret keys to communicate with each other.

Lets denote the secondary key issued by node a as S_a, the secondary key sent by node b as S_b, and the shared primary key between node a and b as $P_{a,b}$

which is obviously equal to $P_{b,a}$. Suppose that each node compute Primary $\oplus$ Secondary to calculate a new secret key. Consequently, node a computes the secret key for communication to b as:

$$K_{a,b} = P_{a,b} \oplus S_b \tag{2}$$

where $k_{a,b}$ denotes the secret key that node a uses for encrypting sending messages to node b.

Similarly, node b computes the secret key for communication to a as:

$$K_{b,a} = P_{a,b} \oplus S_a \tag{3}$$

It is evident that $K_{a,b}$ and $K_{b,a}$ differs from each other. In other words, in the communication between two adjacent cells, the ingoing traffic is encrypted with a distinct key from the outgoing traffic. This fact hardened the fact for the adversary who sniffs the communications between every two node that are residing in different cells as they send and receive messages that are encrypted with distinct keys.

Furthermore, according to the explanations regarding to the assignment of secondary keys, the number of allocated secondary keys is proportional to the number of cells in a vicinity. Here, for square cells, each node needs to store 17 secondary keys in its memory; 1 key for send and receive in its own cell, 8 keys for transmitting messages to adjacent cells, and 8 keys for receiving messages from nearby cells.

3.2 Key Revocation

As soon as a node is compromised, it is indispensable that all its neighbor nodes revoke all the keys that exist in the compromised node's key-rings. Various methods have been proposed which enable nodes to identify compromised nodes pursuant to their misbehaviors [28–30]. After being ascertained that a node is compromised, all its adjacent nodes remove their shared-keys and thus their secure links with that node. To attain a higher level of security, after key revocation, LPKP performs secondary key distribution once more.

3.3 Re-keying

Whenever it is needed that re-keying be accomplished, in LPKP scheme, one way is that the procedure of distribution of secondary keys takes place. Another way is to allocate 9 different hash functions to the nodes in each cell to be used for communicating with nodes located in 8 adjacent cells and also with nodes in their own cell. Each hash function is used to vary the keys in the corresponding key-ring. For instance, among these 9 hash functions, one is used to change keys in $kr_{1,1}$, another is used for keys in $kr_{1,2}$, and so on. As a result, analogous to the key-pools, at least 45 distinct hash functions are required in the network. These hash functions can be used for creating temporary secret keys between neighbor nodes which vary over time. Hash functions take the original key and time value as their input and generate a new key. Since the output of hash functions is completely irrelevant to its input, the adversary will not be able to distinguish the keys after they are passed through the hash functions. Although this mechanism reinforce the security of the network against cryptanalysis attack, it is still susceptible to node compromise.

4 Analysis of the LPKP Mechanism

4.1 Key Distribution Analysis

In LPKP approach, the size of each key-ring is $kr_{i,j}$ which is randomly chosen from a key-pool of size $KP_{i,j}$. An important question that arises here is that what is the suitable size for $KP_{i,j}$ and $kr_{i,j}$ to acquire a desired *connectivity probability* of p_c?

Note that in LBC scheme, the sizes of key-pools and key-rings only depend on the average number of neighbors nodes for each node (which is proportional to the density of nodes) while in the former schemes they are computed in terms of the whole number of nodes in the network. This means our scheme shows a superior scalability as the number of nodes grows in the network compared with the previous approaches.

Memory of each node restricts the number of keys that could be stored. So, in practice, the size of key-rings must be determined by the size of memory. To figure out the size of key-pools, a theorem from random graph theory is applied here, [31] and [32].

$$p_c = \lim_{n\to\infty} \Pr\left[G(n, p_{ij}) \text{ be connected}\right] = e^{-e^{-c}} \tag{4}$$

where $G(n,p)$ is a random graph with n nodes and $p_{i,j}$ is the probability of existence of an edge between two nodes. Also, c is a positive constant which is related to the value of $p_{i,j}$ by the following formula (as n grows to infinity):

$$p_{i,j} = \frac{\ln n}{n} + \frac{c}{n} \tag{5}$$

To have a connected graph, it is sufficient that the graph which is made up of nodes from any two adjacent cells be connected and also, to have connectivity inside each cell.

First we compute $KP_{i,j}$ for the case that two nodes are in different cells. Constant c is obtained from (4) as follows:

$$c = -\ln(-\ln p_c) \tag{6}$$

Moreover, $p_{i,j}$ is obtained from (5) by setting n to be approximately the number nodes in two adjacent

cell.

To tie in $p_{i,j}$ to the size of $KP_{i,j}$, we have,

$$p_{i,j} = 1 - \Pr[\text{two nodes do not have a shared key}] \tag{7}$$

In order to have two nodes with no key in common, the first key-ring is arbitrarily drawn out of $KP_{i,j}$ keys, while the second key-ring must be chosen among the $KP_{i,j} - kr_{i,j}$ remaining keys. Therefore, the number of cases for the first and the second key-rings are as follows respectively,

$$\binom{KP_{i,j}}{kr_{i,j}} = \frac{KP_{i,j}!}{kr_{i,j}!(KP_{i,j} - kr_{i,j})!} \tag{8}$$

$$\binom{KP_{i,j} - kr_{i,j}}{kr_{i,j}} = \frac{(KP_{i,j} - kr_{i,j})!}{kr_{i,j}!(KP_{i,j} - 2kr_{i,j})!} \tag{9}$$

So the probability that two nodes do not share any common key can be calculated as following:

$$\frac{\binom{KP_{i,j}}{kr_{i,j}}\binom{KP_{i,j}-kr_{i,j}}{kr_{i,j}}}{\binom{KP_{i,j}}{kr_{i,j}}\binom{KP_{i,j}}{kr_{i,j}}} = \frac{((KP_{i,j} - kr_{i,j})!)^2}{KP_{i,j}!(KP_{i,j} - 2kr_{i,j})!} \tag{10}$$

Thus,

$$p_{i,j} = 1 - \frac{((KP_{i,j} - kr_{i,j})!)^2}{KP_{i,j}!(KP_{i,j} - 2kr_{i,j})!} \tag{11}$$

As depicted in simulations, $KP_{i,j}$ is in the order of $10,000$. Hence, it is almost infeasible to compute the exact solution of (11). As a result, we use Stirling's approximation for $m!$, where m is a large number; i.e.,

$$m! = \sqrt{2\pi m}(\frac{m}{e})^m \tag{12}$$

Consequently, by using (12), $p_{i,j}$ is approximately equal to:

$$p_{i,j} \cong 1 - \frac{(1 - \frac{kr_{i,j}}{KP_{i,j}})^{2(KP_{i,j}-kr_{i,j}+0.5)}}{(1 - 2\frac{kr_{i,j}}{KP_{i,j}})^{(KP_{i,j}-2kr_{i,j}+0.5)}} \tag{13}$$

The size of $KP_{i,j}$ can be obtained based on (13). However, still computing $KP_{i,j}$ through this equation is hard and time consuming. We reformulate the equation as following:

$$\ln(1 - p_{i,j}) \cong 2(KP_{i,j} - kr_{i,j} + 0.5)\ln(1 - \frac{kr_{i,j}}{KP_{i,j}}) - (KP_{i,j} - 2kr_{i,j} + 0.5)\ln(1 - 2\frac{kr_{i,j}}{KP_{i,j}}) \tag{14}$$

Since the size of key-ring is much too smaller than the size of key-pool, the proportion of kr to KP is too small. Thus, the following approximation can be used for small δ:

$$\ln(1 - \delta) \cong -\delta \tag{15}$$

Thus, equation (14) would be written as:

$$\ln(1 - p_{i,j}) \cong 2(KP_{i,j} - kr_{i,j} + 0.5)(-\frac{kr_{i,j}}{KP_{i,j}}) - (KP_{i,j} - 2kr_{i,j} + 0.5)(-2\frac{kr_{i,j}}{KP_{i,j}}) \tag{16}$$

Finally, $KP_{i,j}$ could be computed as:

$$KP_{i,j} \cong \frac{-2kr_{i,j}^2}{\ln(1 - p_{i,j})} \tag{17}$$

Similar formulas can be derived for the case that two nodes are located in the same cell. Just in equation (5), n must be considered as number of nodes in a cell, which gives us smaller key-pool sizes. For simplicity, in the proposed scheme we assume the size of key-pool and key-ring for the latter case is equal to the former one, where two nodes are located in two adjacent cells.

4.2 Security Analysis

In this section, we investigate the possibility of leakage of keys when x nodes are compromised. As key-rings are assigned to nodes based on their location and each key-ring is chosen out of a particular key-pool, the adversary needs to compromise nodes in specific cells to gather a sufficient number of keys to be able to perform a successful attack.

Assume that the adversary compromises x nodes and obtains x number of key-rings for a given color ($kr_{l,m}$ for given l and m). The best event for the adversary is when these key-rings mutually disjoint; which is almost true according to Figure 7 and 8. As a result, the adversary would attain almost $x \times kr_{l,m}$ keys from $KP_{l,m}$.

The average number of secure links that each node in cell with color m possesses with nodes in the adjacent cell with color l is $n \times p_{m,l}$; where n is the average number of nodes in each cell and $p_{l,m}$ is the probability that every two adjacent node share at least one key. As a result, on average, the total number of secure links that nodes in cell with color l form with nodes in cell with color m is equal to $n^2 \times p_{m,l}$.

Now, we want to pursue what the hostile can acquire by compromising x nodes. First, we have to see what are the probabilities that two neighbor nodes share exactly one key, two keys, and so on. By using (14), the probability that any two adjacent node share at least one key ($p_{l,m}$) is computed. $p_{l,m}$ could be also written as:

$$p_{l,m} = p_1 + p_2 + p_3 + \dots \tag{18}$$

where p_q denotes the probability that two neighbors have exactly q keys in common; and, $p_1 \gg p_2 \gg p_3 \gg \dots$.

In order that two key-rings have exactly q shared-keys, the first key-ring should be drawn out the whole key-pool ($\binom{KP_{l,m}}{kr_{l,m}}$), then those q common keys should be chosen from the selected key-ring ($\binom{kr_{l,m}}{q}$), and eventually, the remained $kr_{l,m}-q$ keys of the second key-ring must be chosen from the remaining key-pool ($\binom{KP_{l,m}-kr_{l,m}}{kr_{l,m}-q}$); indeed,

$$\binom{KP_{l,m}}{kr_{l,m}}\left[\binom{kr_{l,m}}{q}\binom{KP_{l,m}-kr_{l,m}}{kr_{l,m}-q}\right] \tag{19}$$

Consequently, p_q could be calculated as:

$$p_q=\frac{\binom{KP_{l,m}}{kr_{l,m}}\binom{kr_{l,m}}{q}\binom{KP_{l,m}-kr_{l,m}}{kr_{l,m}-q}}{\binom{KP_{l,m}}{kr_{l,m}}\binom{KP_{l,m}}{kr_{l,m}}} \tag{20}$$

As $p_1 \gg p_2 \gg p_3 \gg \dots$, it is more probable that an adversary be able to eavesdrop the communication between two nodes with just single key in common, rather than overhearing a link among two nodes that have more than one key in common.

In order to compute p_1 in terms of $p_{i,j}$, we rewrite (11) as:

$$p_{i,j}=1-p' \tag{21}$$

where

$$p'=\frac{\binom{KP_{i,j}}{kr_{i,j}}\binom{KP_{i,j}-kr_{i,j}}{kr_{i,j}}}{\binom{KP_{i,j}}{kr_{i,j}}\binom{KP_{i,j}}{kr_{i,j}}} \tag{22}$$

By comparing (20) and (22), p_1 would be calculated as:

$$p_1=\frac{kr_{l,m}^2}{KP_{l,m}-2kr_{l,m}+1}p' \tag{23}$$

Thus,

$$p_1=\frac{kr_{l,m}^2}{KP_{l,m}-2kr_{l,m}+1}(1-p_{i,j}) \tag{24}$$

As $KP_{l,m}$ is much larger than $kr_{l,m}$, p_1 could be approximately written as:

$$p_1\cong\frac{kr_{l,m}^2}{KP_{l,m}}(1-p_{i,j}) \tag{25}$$

According to (17), the equation above would be written in the simplified form as:

$$p_1\cong\frac{-\ln(1-p_{i,j})}{2}(1-p_{i,j}) \tag{26}$$

After the adversary achieves x number of $kr_{l,m}$, the probability of a successful attack, i.e., the probability that at least one secure link, which is formed by utilization of just one shared-key, among two adjacent cells with colors l and m become vulnerable is equal to:

$$p_{SA}\cong n^2p_1\frac{x\,kr_{l,m}}{KP_{l,m}} \tag{27}$$

Thus,

$$p_{SA}\cong\frac{x\,n^2}{kr_{l,m}}\left[\frac{-\ln(1-p_{i,j})}{2}\right]^2(1-p_{i,j}) \tag{28}$$

We desire that an adversary could not be able to perform a successful attack by compromising x number of sensor nodes; i.e., $p_{SA}\ll 1$. Thus:

$$x\ll\frac{4\,kr_{l,m}}{n^2(1-p_{i,j})\,\left(\ln(1-p_{i,j})\right)^2} \tag{29}$$

Consequently, by compromising x nodes where x satisfies the above condition, the antagonist could not fulfill any successful attack.

By applying secondary keys, as stated before, the adversary is not able to perform a successful attack in a region in the network by getting use of the achieved key-rings of compromised nodes in another location of the network.

5 Simulation Results

In this section we describe the results of *MATLAB* simulations for the proposed scheme. We focus on the performance of the scheme for distributing the primary and secondary keys and compare it with the basic scheme.

5.1 Evaluation of the Presented Scheme

5.1.1 Numerical Analysis

In this section, we numerically analyze the presented scheme from different perspectives.

First, we present a simple numerical example to illustrate how the scheme works. Calculations for obtaining the size of key-pools are based on (14). Consider a network with 100 nodes which are distributed randomly in a region of $100m\times 80m$. Suppose that the radio range of the nodes is $50m$. Hence, the length of cells would be $\frac{50}{2\sqrt{2}}=17.68m$ and the average number of nodes in each cell will be $\frac{100}{17.68^2}\cong 4$.

Assume that the desired connectivity for graph is $p_c=0.99$, meaning that the network is almost certainly connected. Using (5), we obtain $c=4.6$ and from (6), we have $p_{i,j}=0.8349$. Figure 4 depicts a network with the above parameters whose nodes are distributed randomly. The lines indicate that the two neighbor nodes have at least one shared-key. It can be seen that the network graph is connected via secure links.

Next, we assume that each node can store 225 keys in the memory. So we can assign 25 keys to each key-ring ($25\times 9=225$). From (14), we set $KP_{i,j}=560$. Therefore, we need to generate $44\times 560=25200$ distinct keys to feed all key-pools. Just to give another

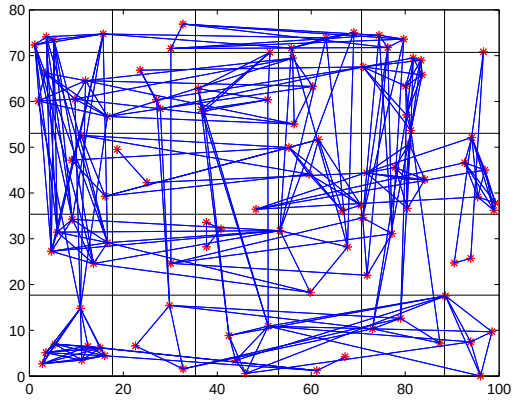


Figure 4. Graph connectivity over secure links for 100 nodes

example, under the same condition, if the size of $kr_{i,j}$ is reduced from 25 to 20, $KP_{i,j}$ decreases to 360 and thus totally 16200 keys are needed.

Figure 5 shows the effect of the average number of nodes in each cell on the probability of sharing at least one key. The probability of existence of a shared secret key decreases while the number of neighbors grows. It also shows that when the desired probability for connectivity of graph boosts, the probability of sharing a key increases as well. Note that a little change in p_c has a significant impact on $p_{i,j}$. For instance, if the average number of nodes in cells is 10, when p_c increases from 0.99 to 0.999, $p_{i,j}$ elevates approximately from 0.4 to 0.5.

Next, we study the effect of number of neighbor nodes on the total size of key-pool. Figure 6 depicts that the relation between these parameters is almost linear. While the mean number of nodes in each cell boosts, the size of key-pool elevates either. This figure also illustrates the relation between the size of key-pool and the size of key-ring.

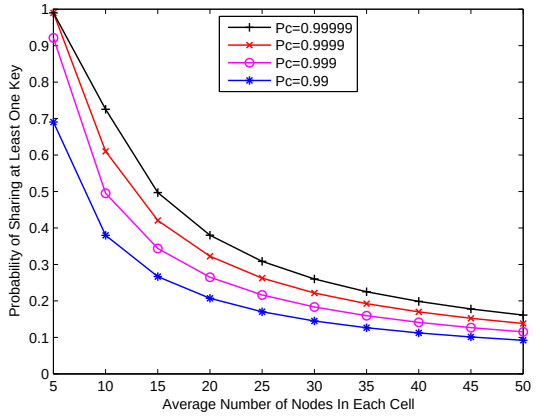


Figure 5. The probability of sharing at least one key vs. the average number nodes in cells, when $kr_{i,j} = 24$.

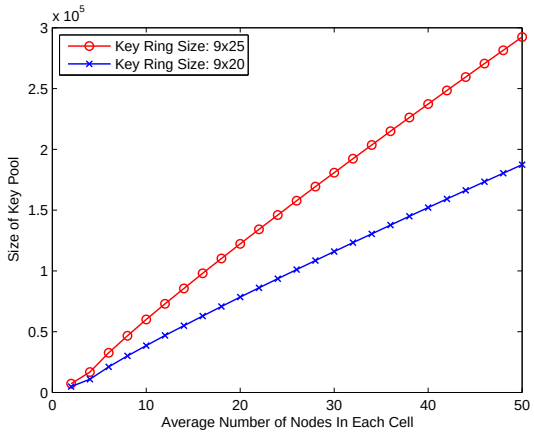


Figure 6. The size of key-pool vs. the average number of nodes in cells, when $p_c = 0.99$ and $kr_{i,j} = 20$ and 25

Finally, according to (29), the threshold for the number of compromised nodes (i.e., x) which does not enable the adversary to perform a successfull attack for the aforementioned example is 46. As stated before, each cell contains 4 sensor nodes in average; as a result, in such network, the adversary should compromise nodes in approximately 12 cells. On the other hand, by applying secondary keys, the shared-keys vary from cell to cell. Consequently, the antagonist is not able to carry out any successful attack.

5.1.2 Security Analysis

From now on, we investigate the security of our presented scheme.

Note that in our scheme only the nodes which their keys are chosen from the same key-pool have a chance to possess common keys in their key-rings. Since the key-pools are utilized based on the color of the cells, most of the nodes with the same color will be distant from each other, and this fact hardened the matter for the adversary.

Consider the first example with 100 nodes distributed randomly in the network of size $100m \times 80m$. Figure 7 shows the number of times that all of the primary keys in all key-pools are allocated to the entire network nodes. As illustrated in the figure, 10339 keys are assigned to just one key-ring (one node); 3951 keys are repeated in two key-rings (two nodes), and so on. It is obvious that the summation of the values in the figure should be equal to the size of key-pool (25200 keys). It should be stated that as key assignment is random, these results would vary in each run.

In the previous case, the repetition of keys does not mean that those keys are used as shared keys. Indeed, we just count the number of times that keys are assigned to different key-rings in distinct nodes.

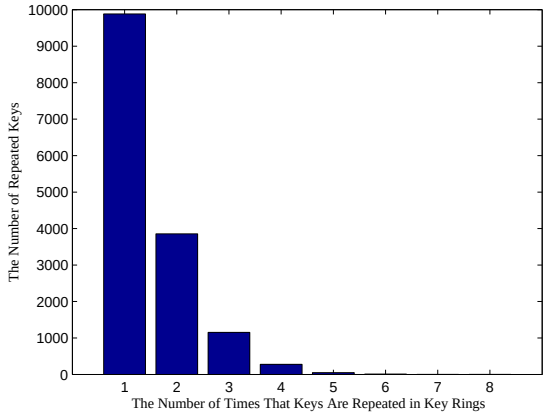


Figure 7. The number of repeated primary keys vs. the number of repetition of primary keys.

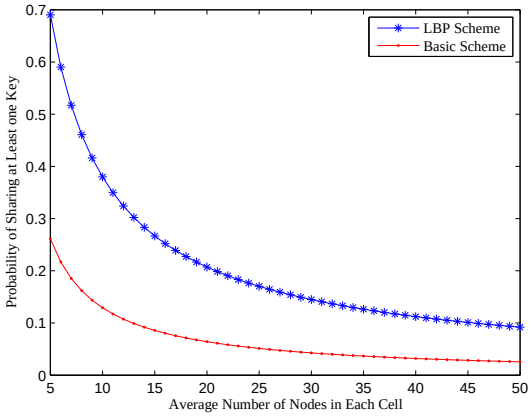


Figure 9. Difference between the LPKP scheme and the basic scheme from the perspective of $p_{i,j}$.

Here, Figure 8 depicts the number of times that those keys are used as shared-keys among neighbor nodes.

As shown in Figure 8, for the previous example, 2651 keys are used to secure just one link, 285 keys are utilized as shared-keys twice, 229 keys are used to secure three links, and etc. In other words, out of a key-pool of size 25200, 10.5%, 1.1%, and 0.9% of the keys are used to secure one, two, and three links respectively.

It is obvious that as the network size grows without any change on the average number of nodes in cells, the probability for existing at least one shared-key between two neighbors does not change; so, the size of key-pools do not vary either. But, as the total number of nodes increases, we have more key-rings that should be chosen from a fixed key-pool. Therefore, the primary keys are used more repeatedly in key-rings. For example in the case that there are 5 nodes in each cell, for total network size of 500 nodes, the average maximum number of times that a primary key

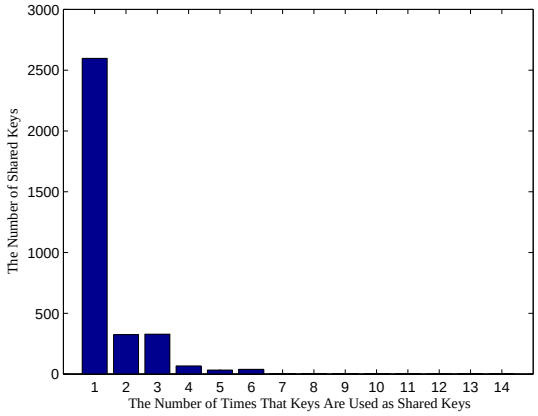


Figure 8. The number of shared-keys vs. the number of times that primary keys are used as shared-keys.

repeats is equal to 14.33. If the network size grows to 5000 nodes with average 5 nodes in cells, a key repeats 82.64 times on average.

5.2 Comparison Between the LPKP and the Basic Scheme

We compare the performance of our key distribution scheme with the basic scheme from different aspects.

First, consider the aforementioned area with fixed total number of nodes. By varying the number of neighbor nodes, we can compare the probability of existence of a shared-key. Figure 9 shows the probability for these schemes. It is assumed that the total number of nodes in the network is 1000 with desired connectivity of 0.99. This indicates that our method improves the probability of existence a link between two neighbor nodes.

Figure 9 indicates that our scheme has a higher probability of existence of secure links between two adjacent nodes. For instance when the average number of nodes in each cell is 10, $p_{i,j}$ from basic scheme is equal to 0.1292 but our approach results in 0.3798, more than two times.

The next comparison could be made in terms of key-pool size. Similarly, we consider the conditions analogous to previous example. In addition, the size of key-ring is equivalent to 225. The result in Table 1 indicates that our approach is economical in terms of number of required keys for the key-pool. For example, there are 5 nodes in each cell, on average each node has more than 45 neighbors. In this case basic method results in 167393 keys in the key-pool for totally 1000 nodes. But as analyzed earlier the total key-pool size in our approach is equal to 25110.

Finally, for the last comparison, we study the scalability of our proposed method. To do this, we again

Table 1. Comparing the key-pool size in the proposed scheme and the basic scheme as the density of nodes grows.

	Basic scheme	Proposed scheme
Number of neighbor nodes	45	-
Average number of nodes in a cell	-	5
Key-pool size	167393	25110
Number of neighbor nodes	90	-
Average number of nodes in a cell	-	10
Key-pool size	366245	59985
Number of neighbor nodes	225	-
Average number of nodes in a cell	-	25
Key-pool size	961086	151830
Number of neighbor nodes	450	-
Average number of nodes in a cell	-	50
Key-pool size	1951999	292365

Table 2. Maximum number of nodes for having a shared-key among neighbors with a certain probability.

	Basic scheme	Proposed scheme
Network Size	100	751
Probability of Sharing a key	0.1688	0.1644
Network Size	200	1471
Probability of Sharing a key	0.0929	0.0921

consider a region of $100m \times 80m$ and scatter a defined number of nodes in that region with radio range of $50m$. It is assumed that the desired probability for graph connectivity is 0.99. The probability of existence of link between two neighbor nodes is computed based on (4) and (5). As our scheme has a higher probability, we add nodes to the network till the probability of existing key between adjacent nodes in our approach become close to the basic scheme. The results are depicted in Table 2.

For instance, we spread 100 nodes in the mentioned area. Then the probability that two neighbor nodes have a shared-key in basic scheme is computed which is equal to 0.1688. The number of nodes in the network increased until the probability of sharing at least one key between two neighbors in our proposed method become close to that probability. In this case, our scheme can support 751 nodes under the same conditions.

6 Conclusion

We presented a novel location-based key management scheme for symmetric key pre-distribution based on Eschenauer and Gligor's method. Our scheme uses the location information of the nodes for choosing the key-rings which results in higher probability for existence of secure link between two neighbors. The presented scheme is designed in such a way that the places in which the same key is used are distant from each other. Therefore, if a node is compromised, it can cause only a local security issue. Because, by utilization of secondary keys those compromised keys are useless in other parts of the network. So, the network is resilient against node capture. Furthermore, unlike other key distribution methods that are dependent to the total number of nodes in the network, our proposed approach is dependent to the density of the nodes; as a result, it shows a senior scalability for large-scale wireless sensor networks.

Our simulation results verify that in large-scale wireless sensor networks with stationary nodes, our proposed scheme can randomly pre-distribute keys based on the location of the nodes with a higher probability for shared-key existence compared to the basic scheme. In addition, in our scheme it is easier to control and manage the keys if some of them compromised locally. On the other hand, as the presented method is just dependent to the density of nodes, the size of key-pools do not increase as the network size boosts. Thus, in order to address this deficiency, we introduce secondary keys.

Finally, as stated in the content, our proposed scheme does not rely on the network size; on the contrary, it is based on the density of the nodes. So, as it can be seen in the simulations, the graph of the network is connected within each cell too. As a result, the presented method could be employed in networks that each nodes has a small number of neighbors; like hierarchical wireless sensor networks in which it is sufficient that all the nodes within each cluster form a secure link with their cluster-head.

References

[1] M. Kubale, "Graph colorings," American Mathematical Society, ISBN 0-8218-3458-4, 2004.

[2] J. H. Van Lint, R. M. Wilson, "A course in combinatorics (2nd ed.)," Cambridge University Press, ISBN 0-521-80340-3, 2001.

[3] T. R. Jensen, B. Toft, "Graph coloring problems," Wiley-Interscience, New York, ISBN 0-471-02865-7, 1995.

[4] 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.

[5] A. Ahadipour, A. Keshavarz-Haddad, "A novel location-based key distribution scheme for large-

scale stationary wireless networks," IEEE 12th International Iranian Society of Cryptology Information Security and Cryptology (ISCISC), Rasht, Iran, 2015.

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

[7] D. Liu, P. Ning, "Establishing pairwise keys in distributed sensor networks," In Proceedings of the Annual ACM Computer and Communications Security (CCS), 2003.

[8] R. Blom, "An optimal class of symmetric key generation systems," In Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), 1984.

[9] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, "Perfectly-secure key distribution for dynamic conferences," In Proceedings of the 29th International Cryptology Conference (CRYPTO), 1993.

[10] H. Chan, A. Perrig, D. Song, "Random key predistribution schemes for sensor networks," In Proceedings of IEEE Symposium on Security and Privacy (SP), 2003.

[11] W. Du, J. Deng, Y. S. Han, Sh. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," 23th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2004.

[12] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A key pre-distribution scheme for sensor networks using deployment knowledge," IEEE Transactions on Dependable and Secure Computing, vol. 3, pp. 62-77, 2006.

[13] B. Kong, H. Chen, X. Tang, K. Sezaki, "Key pre-distribution schemes for large-scale wireless sensor networks using hexagon partition," IEEE Wireless Communications and Networking Conference (WCNC), 2010.

[14] F. Anjum, "Location dependent key management using random key-predistribution in sensor networks," Proceedings of the 5th ACM workshop on Wireless security, pp. 21-30, 2006.

[15] M. Faghani and S. Motahari, "Sectorized location dependent key management," Proceedings of Wireless and Mobile Computing, Networking and Communications, pp. 388-393, 2009.

[16] F. Anjum, "Location dependant key management in sensor networks without using deployment knowledge," Wireless Networks, vol. 16, no. 6, pp. 1587-1600, 2010.

[17] A. Gaur, S. Toshniwal, A. Prakash, D. P. Agrawal, "Enhanced location based key pre-distribution scheme for secure communication in Wireless Sensor Network (WSN)," IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS), 2010.

[18] H. Fakhrey, S. Boussakta, R. Tiwari, Y. Al-Mathehaji, A. Bystrov, "Location-dependent key management protocol for a WSN with a random selected cell reporter," IEEE International Conference on Communications (ICC), 2015.

[19] R. Vaid, V. Katiyar, "VLKM: Virtual location-based key management scheme for Wireless Sensor Networks," International Conference on Parallel, Distributed and Grid Computing (PDGC), 2014.

[20] J. Choi, J. Bang, L. Kim, M. Ahn, T. Kwon, "Location-based key management strong against insider threats in wireless sensor networks," IEEE System Journal, pp. 1-9, Issue: 99, 2015.

[21] M. J. Duan, J. Xu, "An efficient location-based compromise-tolerant key management scheme for sensor networks," Information Processing Letters, pp 503-507, 2011.

[22] C. Chen and H. Chao, "A survey of key distribution in wireless sensor networks," Security and Communication Networks, 7.12; pp. 24952508, 2014.

[23] Misra, Sudip and Zhang, Isaac and Misra, Subhas Chandra, "Guide to wireless Ad Hoc networks," Security in wireless Ad Hoc networks, p. 391-425, Springer London, 2009.

[24] S. H. Zanakis, A. Solomon, N. Wishart, S. Dublish, "Multi-attribute decision making: a simulation comparison of select methods," European Journal of Operational Research, vol. 107, no. 3, pp. 507529, 1998.

[25] C. Zopounidis, M. Doumpos, "Multicriteria classification and sorting methods: a literature review," European Journal of Operational Research, vol. 138, no. 2, pp. 229246, 2002.

[26] A. Jahan, F. Mustapha, M. Y. Ismail, S. M. Sapuan, M. Bahraminasab, "Acomprehensive VIKOR method for material selection," Materials and Design, vol. 32, no. 3, pp. 12151221, 2011.

[27] A. Chauhan, R. Vaish, "Pareto optimal microwave dielectric materials," Advanced Science, Engineering and Medicine, vol. 5, no. 2, pp. 149155, 2013.

[28] S. Marti, T. J. Giuli, K Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, pp. 255-265. ACM, 2000.

[29] G. Wang, W. Zhang, G. Cao, and T. La Porta, "On supporting distributed collaboration in sensor networks," In IEEE Military Communications

Conference (MILCOM'03), vol. 2, pp. 752-757, 2003.
[30] A. Seshadri, A. Perrig, L. Van Doorn, and P. Khosla, "Swatt: Software-based attestation for embedded devices," In Proceedings of IEEE Symposium on Security and Privacy, pp. 272-282, 2004.
[31] P. Erdos, A. Renyi, "On random graphs I," in Publ. Math. Debrecen 6, p. 290297, 1959.
[32] J. Spencer, "The strange logic of random graphs," Algorithms and Combinatorics 22, Springer-Verlag ISBN 3-540-41654-4, 2000.

Alireza Ahadipour received his B.S. and M.S. degrees in Electrical Engineering from Shiraz University, Shiraz, Iran, in 2012 and 2015, respectively. Currently, he is a Ph.D. student at Shiraz University. His research interests include computer and communication networks, network security, and cryptography.

Alireza Keshavarz-Haddad is an assistant professor at the School of Electrical and Computer Engineering at Shiraz University, Shiraz, Iran. He received the B.S. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2001; and the M.S. and Ph.D. degrees in electrical and computer engineering from Rice University, Houston, Texas, USA, in 2003 and 2007, respectively. His research interests are in the field of computer and communication networks with emphasis on fundamental mathematical models, architectures and protocols for wireless networks, and network security.