Indian Journal of Science and Technology, *Vol 8(19), DOI: 10.17485/ijst/2015/v8i19/75986, August 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Suggesting a Method to Improve Encryption Key Management in Wireless Sensor Networks

Mehrnoush Toghian* **and Matei Ciobanu Morogan**

Department of Computer and System Sciences (DSV), Royal Institute of Technology (KTH), Stockholm, Sweden;
mehrnoush_toghian@yahoo.com, Matei@dsv.su.se

Abstract

Background/Objectives: Recent technology achievements in electronics and wireless communication makes it possible to provide sensor nodes in low cost, low power, small size, and ability to communicate with other nodes in vicinity, which results in design of wireless sensor networks. Nowadays there are many different applications for wireless sensor networks such as in medicine, military, business, etc. **Methods/Statistical Analysis:** Considering the structure of such networks that is based on wireless communication, also the variety of usage in different critical domains such as military domain, information and communication security is one of the major aspects that must be exactly considered to design and implement wireless sensor networks. Proposing a method to provide security in wireless sensor networks is depended on structure and properties of each node, thus it is more difficult and more complex than in other types of networks. **Findings:** Considering the vulnerability of this type of networks against different attacks, finding out a low cost method to guarantee a reasonable level of security in wireless sensor networks has become an issue of interest these days. There are several methods to provide security in such environments. One of them is data encryption using encryption keys. When it comes to wireless sensor networks, the issue of key management is different because of the different structure of these networks. In other words, considering mobility of nodes, structure of wireless communication and message broadcasting, limited energy resources, variety of application domains, and widespread nodes, providing security in such networks is totally different from other similar environments. **Application/Improvements:** A lot of research has been done in this area and different methods have been proposed. However there still are lots of aspects in this area that need to be studied more. In this study we are going to find a new expansive method to distribute the encryption keys in wireless sensor networks based on some of special properties of such network.

Keywords: Encryption Key Management, Network Security, Wireless Sensor Networks

1. Introduction

Recent developments in wireless communications, electronics and radio technology have caused the creation of cheap and low power sensor nodes, in small size and capability to communicate with other near nodes. This development on the one hand has speeded up many activities and also reduced the costs in contrast to use of large and expensive pieces and with accurate wirings in typical networks, and on the other hand, using sensor networks applications has been made possible which could not be done in existing networks. In fact the sensor networks have been considered as a new field in subject related to IT that try to present an image of embedded internet, in which the processing pieces are embedded inside a physical environment and are connected to each other. The sensor nodes have limited sensing and processing parts and components to establish wireless connection[1]. These parts altogether create sensor networks. As a matter of fact sensor networks are a development of old ones. A sensor network is made up of numerous sensor nodes. These sensors, as network nodes, are placed in or near an environment, to monitor or collect data of that environment and react to events according to the given program[1].

**Author for correspondence*

Finding proper location for nodes does not require a precise engineering calculations or pre-determination of the location. In fact sensors are distributed randomly in the environment. Hence the algorithms related to such networks should be self-configurable. As mentioned, sensor networks have variety of applications. Some typical applications can be seen in military, medical sciences and commercial fields. In military industries for instance, specifications like quick setup, automatic organizing and tolerance to fault have changed sensor networks to a quite useful solution in control, communication, and targeting. A patient, too, could be under control by utilizing the sensors in medical sciences; also medical research could be performed easily by applying them in different parts of the body. Other applications of sensor networks in business systems could be warehouse management, forecasting systems, quality control, and monitoring hazardous places/environments.

The sensor networks have special limitations, capabilities, peculiarities, complexities and operational environment, which distinguishes them from similar matters such as Ad-Hoc networks and forces the software and hardware and algorithm designers to have special considerations in their designs and proposed solutions. Some concerns with sensor networks are interaction with environment, complexity of environment, mobility of nodes in the environment and reacting to an event and the dynamic entity of environment[1].

The sensor nodes are being produced in research and industrial centers and this is a good base for sensor networks application in a wider and more general area, but hardware alone cannot respond all demands. In addition, each technology has some limitations and challenges that should be recognized and faded away. For this purpose, a software system is required which is compatible with characteristics of these pieces, limitations, abilities and the operating environment in order to expand and easily develop applications on them.

As a whole, the information security in this type of network is a vital parameter due to wireless communication structure and variable applications of this type of network in different fields, especially military applications which needs particular attention. In contrast with other types of networks, the methods for security of wireless sensor networks have an important relation with the nodes structures and also their special features. Hence, the researchers are trying to find inexpensive methods to increase the security of sensor networks, due to its breakable entity.

Since security is an important aspect in authenticating and authorizing network members and controlling their access to network resources, choosing a suitable strategy is quite important.

There have been numerous methods to establish a secure network. One of the most fundamental security mechanisms is encryption of transmitting messages through key management[1]. This is done by introducing and distributing encryption keys in the network and applying them to the messages. But here the difference is the different structure of wireless sensor networks. Altogether, in sensor networks, the security issue is quite different to other similar environments because of nodes mobility, power constraints, and wireless characteristics (message broadcasting)[1]. Also network members have limited power sources and different applications along with nodes dispersion.

As a whole the subject of encryption key management consists of encryption key establishment, encryption key exchange, encryption key distribution, encryption key structure, key storage, and digital signatures. There have been plenty of studies and researches done in this field and various methods have been proposed[1]. But despite this, due to existing challenges, the necessity of more research is quite obvious. In this study, we try to propose an expandable approach for encryption key distribution in wireless sensor networks based on some of special peculiarities of the networks.

1.1 Wireless Sensor Networks

Ever increasing progress in production of small pieces with capability of processing, sensing, and wireless communication has put forward the subject of sensor networks as a new field of research. The idea of sensor networks is due to the fact that the development in digital circuits has provided the possibility of integrating sensing units, processing units, and wireless communication units into a single chip. Indeed, creating sensor networks is an expansion to old sensors. Allocating the nodes, in the above-mentioned network does not need special calculations or initializations. Sensors are often allocated in an environment, randomly[2,3].

Limitations, capabilities, specifications, complexities, and special operating environment of sensor networks have created a lot of challenges in various fields for this new technology and it is predicted to be used in future as a modern information infrastructure. Because of special characteristic of this technology and its forming elements, it will have many applications. We should be quite familiar with characteristics of the sensor nodes, sensor networks, and their differences with other networks. Hence, we are going to introduce the sensor network, fully in following:

1.2 Sensor Node

A sensor node consists of the following 3 main units:

- Simple equipment for wireless communication with other nodes or a central node (Sink)
- A central processor which should be able to run the required commands
- Special unit in each sensor node capable of sensing its vicinity.

A sensor can be a camera, an audio device, or one of the infra-red, humidity, light, temperature, or pressure sensors. It also can be a Vibrating, radioactive, or a magnetic sensor. Sensor networks consist of a couple of sensor nodes which are distributed compactly close to the target.

1.3 Security Monitoring

The second application of sensor networks is security monitoring. This network is formed by nodes which are located in stationary places in environment and monitor one or more sensors continuously, in order to report an event, so the main difference of this type of application with environment monitoring is that this kind of application does not collect data, and tries indeed to report the data in case of observing problems or unnatural affairs. Hence considering the aim of this application, it does not have important effects on the architecture of sensor networks. The network in, this kind of application, should be formed such that the nodes confirm the situation of another one. One method is connecting a node to a nearby one. So when it fails for any reason, the other node reports it. In environment monitoring application, each node should transmit the data of child nodes. Hence it's optimal to have a broader tree with less height. But in security monitoring, the optimum configuration is to have a linear topology which changes the network to a Hamilton cycle. The rate of energy consumption is distributed between the nodes. Within this model the main rate of energy consumption is when we want to report a data to the sink, with the minimum delay. In fact, the delay between the considered node and the sink should be minimized- Minimizing causes increase of energy consumption, since the transmitting nodes should monitor the radio channel more times.

2. Encryption Key Management in Wireless Sensor Networks

2.1 Fundamental Concepts of Security in Wireless Sensor Networks

Sensor networks have a different structure compared with other types of networks. In this type of network, we have a heterogeneous system consisting of a number of sensors and drivers and the sensors are controlled by a coordinator and all have a unique processing aim. Examples of these networks are the battlefield trackers, environment supervision, and tracking jungle fires. These kinds of networks usually have limitations in using resources like energy, bandwidth and memory, have a wide spread working space, and are vulnerable to unauthorized access.

Hence, providing security in this environment is one of the most important concerns. But the important thing is providing a suitable solution to be able to create a secure environment with minimum cost.

The existing problems regarding safety and security are divided into the following 3 categories[4,5]:

- **Intrusion:** Any action that threatens the data security. Intrusions include all types of attacks that are performed by any illegal action on data. Intrusions are divided into two groups of <u>*active*</u> and <u>*passive*</u>, based on data manipulation type. The active intrusions change or replace all or parts of the data.

  Active intrusions include the four following types:
- **Masquerade:** A behavior that the intruder impersonates another person- or machine[6].
- **Replay:** Receiving the data (through the path) and retransmitting it for illegal access[5].
- **Modification:** Changing the received data illegally[6].
- **Denial of Service:** Any attempt to make a service or network resource unavailable to intended users[6].

  Passive intrusions are used to monitor flow of data and do not modify data contents. These intrusions are usually a step towards performing an active intrusion.
- **Security Mechanism:** The mechanism designed to recognize and prevent an intrusion, is called the security mechanism. Each mechanism is a solution to

security requirements. Security mechanisms explain methods and ways to confront intrusions and their effects. Encryption mechanism is indeed the most principle and the most important of these mechanisms. In addition, digital signatures and access control could be named.

- **Security Service:** It is a service used to improve data security. Implementation of these services is possible by using one or some of the security mechanisms. The security services could be considered as the main goal of data security.

2.2 Kinds of Attacks in Wireless Sensor Networks

There are many potential attacks in wireless sensor networks. Some of them are considered hereafter[4]:

- Collecting Passive Data: A malicious node can easily listen to transmitted data across a specific route and add some invalid data to attack a network. The effective solution for this threat is using a strong cryptography algorithm.
- Destruction of a Node: What should be done if a node is in the hands of an unauthorized person, or if its stored data is revealed by him/her, or even if he/she benefits from the node? There are several solutions for this case that are going to be discussed later.
- False Node and Wrong Information: In this case, the malicious node broadcasts wrong data and distributes it in the network to produce wrong decisions or wrong information in network. In addition, such a node consumes network resources like bandwidth or power (Sleep Deprivation Torture), or strongly increases network traffic by broadcasting dummy information. Using a strong authentication technique can prevent a fake node to collapse the network.
- Sinkhole Attacks: In this type of attack, the attacker tries to create a heavy traffic in the network. In this case the energy is consumed more and more rapidly and this increases the probability of error in data collection. These traffics usually are created near the base station. For example, if the attacker announces a path to neighbor nodes to use for transmitting the messages to the base station, and they do that, the rate of traffic increases rapidly in a route. Not a lot of activities are done for this subject, but some actions could be considered by authentication.

3. Proposed Method

3.1 Limitations of the Existing Methods

Almost in all of the existing methods there is a lot of overhead in distributing and generating encryption keys. Since the communication overhead in wireless sensor networks is rather higher than computational overhead,[7] introducing a solution to reduce it is quite important. In preceding methods the distribution of keys is done anyway. But the high overhead and the lack of a suitable method to manage it made us to introduce a method-which is believed to be more appropriate- in order to reduce the working overhead in addition to distribute the encryption keys in secure and invulnerable way. Another important point is that due to small memory of the nodes, it's vital to strongly manage memory consumption to save as much memory as possible.

Moreover we propose a method to reduce a great deal of working overhead, reduce nodes required memory to store the encryption keys, and distribute the encryption keys in a more secure way in the network.

3.2 Introducing the Proposed Method

The proposed method is called *Key Establishment Protocol for Wireless Sensor Networks (KEPS)* and is presented in two different network structures named Hierarchical WSN and Distributed WSN. These two structures are shown in Figure 1.

In the distributed WSN the nodes are distributed randomly in the network and we have no default calculations to distribute and setup them.

But in hierarchical WSN the network is divided into definite parts named *clusters*. Each cluster is controlled and managed by a manager node called *cluster head*. Each cluster head, in addition, is responsible to com-

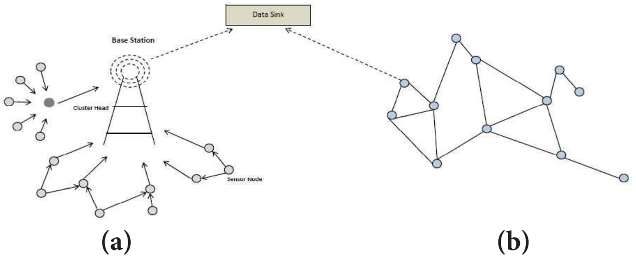


Figure 1. Wireless Sensor Networks in two Distributed and Hierarchical Structures. **(a)** Hierarchical WSN and **(b)** Distributed WSN.

municate with the base station and the nodes inside the clusters. Also they are directly connected to the base station in the form of single-hop. These connections include either requests that are issued in the base station and then assigned to a group of nodes, or reports that are provided by a group of nodes (cluster nodes) and are supposed to be sent to the base station.

3.3 Proposed Solution in Hierarchical WSN Structure

Before considering the proposed method it is good to explain some assumptions regarding the type of the nodes.

3.4 Boundaries and Assumptions

Before stating assumptions about storing information in each node of the network, in this section we divide the memory space of each node in three completely distinct parts:

- Temporary memory RAM
- Memory used by executable codes
- Permanent memory

According to the above divisions, in the case of physical access to the existing sensor nodes, the attacker can just access the permanent memory and executable codes and if he tries to access the stored information in RAM the captured node resets and hence the attacker cannot obtain the information inside RAM. But the main aim is that the network nodes are never captured by any intruder or attacker.

Regarding the memory division method, the following information will be kept by each node:

- A unique ID that is assigned by the manufacturer.
- A pseudo-random function (F)[8,9] for generating encryption keys according to a definite algorithm – This function is assigned to each node before distributing the nodes in the network. It generates some strings as symmetrical encryption keys by using the existing parameters in the network. The size of this encryption function is L units of memory. In the proposed solution the function F is used to generate encryption key using the previous-generated key. Using this function and the version of encryption key, the next key is generated as follows:

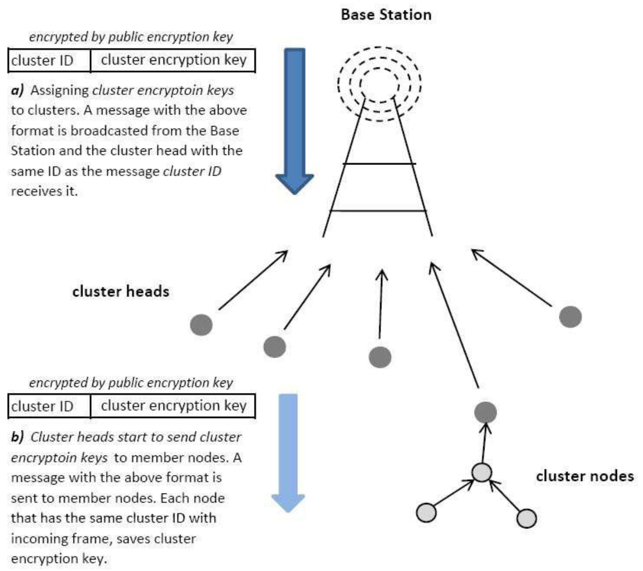
$$K_{vi} \longleftarrow F_{vi}\left(K_{vi-1}\right) \qquad \textit{Relation } 1$$

- In hierarchical structure the clustering is defined before turning on the network, and the nodes are divided into some definite groups. So before the network comes up, a unique number is assigned to each node that clarifies which cluster it belongs to. Obviously the nodes which have the same number belong to the same cluster. So a unique number is assigned to each cluster to be distinguished from other clusters. This unique number is stored in a single unit of memory.
- A public encryption key is transmitted to nodes to be used for broadcasting. This key is dedicated to nodes before distribution and occupies a single unit of memory.

3.5 Details of the Proposed Method in Hierarchical Structure

Now we have a hierarchical clustered network in which the nodes are in separate groups. Each node of the cluster uses relation 1 to generate encryption key. In this structure any cluster encryption key has a number called V_i.

According to previous explanations, the base station sends a cluster encryption key to all nodes which are in the same radio frequency with it using a start message. This message is encrypted by the primary public key that is known to all nodes in the network. Then all cluster heads that receive the message decrypt it using the public key. Each cluster head obtains its own cluster primary key by comparing the cluster ID inside the message and that of its own. It then broadcasts the cluster primary key to other nodes in the cluster. So the base station has a list of cluster IDs and beside each one, the primary cluster encryption key. Each cluster head then distributes the primary cluster encryption key by broadcasting a message that is encrypted by the public encryption key. As mentioned earlier, this public encryption key is built up of a version number V_1 and cluster ID. In this case three different situations might occur. First, the primary cluster encryption key that the base station has assigned to the cluster heads has been received by all member nodes in the cluster, which is the best situation and very optimistic indeed. The second situation is that the cluster primary encryption key is not received by all member nodes in clusters. In this case the node that does not have the key sends a request message to its neighbors. This request is sent after a period which is determined by the



a) Assigning *cluster encryptoin keys* to clusters. A message with the above format is broadcasted from the Base Station and the cluster head with the same ID as the message *cluster ID* receives it.

b) *Cluster heads start to send cluster encryptoin keys* to member nodes. A message with the above format is sent to member nodes. Each node that has the same cluster ID with incoming frame, saves cluster encryption key.

Figure 2. Distribution of Cluster Encryption Keys in Hierarchical WSN. **(a)** Transmitting Encryption Keys to Cluster Heads by the Base Station and **(b)**

operator. This period is longer than network loading and basic configuration time. At this time each neighbor node that receives the message and has the same cluster ID as the requesting node, sends the key to it by a message encrypted by the public key. The third situation is that in distributing the cluster encryption key, the member node in a cluster is inside the radio frequency of another cluster head and obtains the key of that cluster, too. In this case since the key does not belong to the corresponding cluster, it is dropped.

Till now we have introduced cluster encryption key distribution. (With ordinal number '1') As it was stated, this can also be done before distributing the nodes in the network. (factory-built) But the point is that if the operation is done in the factory, the complication of network will be proportional to total number of network nodes [O(N)], which in a hierarchical implementation decreases to [O(log N)][10,11]. So how are the encryption keys refreshed securely after configuration and distribution of nodes in the network? Are there any new keys distributed in the network? Who is responsible for distribution of the keys?

the message encrypted by new key K_{vi}	cluster ID	key K_{vi} Version number
message trailer	message header	

Figure 3. Key refreshing packet in hierarchical structure.

In answer to the above questions, it should be noticed that transmitted message has two distinct parts: *message header* and *message trailer* which can be seen in Figure 3.

In the message header that is encrypted by the public encryption key, version of new cluster key and also cluster ID of the message receiver is included. The message encrypted by new cluster key is included in message trailer. In this case the base station generates a new cluster encryption key using function F and the previous cluster key and inserts the encrypted message into the trailer of the packet. Then it includes cluster ID and version of the new cluster key in its header. After that encrypts it with public encryption key and sends it to the considered cluster. The receiver, (cluster heads or ordinary nodes) then decrypts the message header using public key. If the cluster ID of node and the cluster ID included in the message are identical, the node accepts the message and generates a new cluster encryption key using the version number of the new cluster key included in the message and function F. Finally the node uses it to decrypt the packet trailer that includes the message content. There are some very important points to be issued:

- With this method nothing will be exposed to the attacker, except sequence number of keys which is not useful at all.
- The instruction of changing cluster key is issued by the base station and its time is defined by the operator, e.g. for any transmitted message.
- The public key and even the function F could be changed and sent to all of the nodes in the cluster at any time if necessary.
- In previous methods, the cluster encryption key is sent to the nodes by messages. Hence it was possible for the attacker to reveal the encryption key and read the message. Also the encrypted message was liable to be transmitted to clusters by new cluster encryption key. But in the proposed method the encryption key is not transmitted to the network lonely.

4. Proposed Solution for Distributed Structure

Unlike hierarchical structure, in this structure the nodes are not grouped in different clusters. In fact the nodes are distributed quite randomly in the network. Hence in distributed structure there is no cluster IDs in the nodes.

In this structure each node in pre-distribution state must have the following items:

- Function F to generate keys
- Unique identifier of node
- Public encryption key

4.1 The Details of the Proposed Method in Distributed Structure

For distributed WSN, KEPS method operates in such a way that all the member nodes in the network communicate with each other by the public encryption key. Mainly the distributed structure is used when operations does not need any special classification. It means that in hierarchical structure it's possible to classify data, instructions and processes by placing a series of nodes in a cluster. Certainly the clustering results higher overhead, but operating in a clustered network has advantages like parallel processing, information classification, more scalability, increased automatic control, reduction of energy (power) consumption, and increasing network performance[12].

With regards to the above description, all the nodes in distributed structure are collaborating with each other to achieve a definite goal. Each node communicates with other nodes by encrypting messages using public encryption key. After definite intervals defined by the operator, the nodes start to refresh public encryption key.

This task is similar to refreshing cluster encryption key in hierarchical structure. But transmitted message has a different format.

In this case each node tries to communicate by sending messages encrypted by the public key and then in specific times distributes the new encryption key in the network. The refreshing of encryption key is not done by a specific node in the network. This means that we are not going to introduce a specific node responsible for distributing the encryption key as a server. Each of the member nodes in the network can generate a new encryption key in definite time intervals using the encryption key generator function F. The node then generates the new public key by sending it to the nodes which it is connected to. So each node that receives this message generates a new encryption key for itself using the mentioned function.

the message encrypted by new public key	new public key version number
message body	message header

Figure 4. The Structure of the public key refreshing message in distributed structure.

4.2 Presenting a Clustering Method in Distribution of Encryption Key

In this method all the assumptions about the sensor nodes in the network are identical to those of distributed structure. We introduced this method in sensor networks with the nodes that are communicating with each other in single-hopform. The base station at first sends a hello message to the entire nodes within its radio frequency. All receivers of the hello message start collecting the IDs of multi-hop neighbor nodes. Then the nodes transmit the ID and the number of the neighbors to the base station using a message encrypted by the public key. Then the base station selects the best nodes as the cluster heads using the collected data. Since the nodes have similar processing power, calculations, and connections, the key item for a node to be selected as cluster head is supporting more nodes within its radio frequency. The more nodes covered the higher probability to be selected as cluster head by the base station.

In the next step, the primary cluster encryption key and cluster ID is encrypted by the public key and sent to nodes. Then the cluster heads encrypt the received message using their specific encryption key and send it to the entire nodes within their radio frequency.

Generally Nodes That Receive This Message From The Cluster Head Are Divided Into Three Groups:

- The nodes that have received only one cluster ID. These nodes (like the nodes 3, 5, 6 in Figure 5) are in radio frequency range of just one head. So they will join the corresponding cluster.

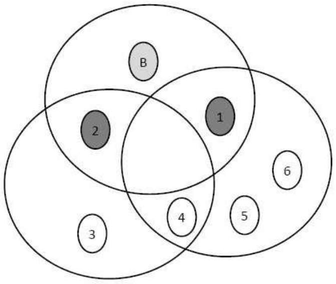


Figure 5. Proposed solution in distributed wireless sensor networks.

- The nodes that have not received a cluster ID. Though it's rare since the nodes are in radio frequency of cluster heads that have announced their ID. If a node did not get the message (e.g. losing message or being down) it requests the cluster ID and primary key of the cluster from the base station by sending a message that is encrypted by public key.
- The nodes that have received more than one cluster ID. These nodes (like node 4 in Figure 5) are in frequency range of more than one cluster head. These nodes select one of the received messages and join the corresponding cluster.

Hence the proposed method creates a clustered network, in which each node has a cluster ID and a primary cluster encryption key. There are considerable points in this structure in addition to the points in hierarchical structure:

- Processing and calculation power of the existing nodes in a distributed wireless sensor network affects their selection as cluster heads. It's desired that nodes declare their processing power to the base station.
- If the cluster head has more storage space than the other nodes, it can have a list of cluster members and other parameters in its cluster; similar to the base station that has a list of all the member nodes in the network and their encryption keys. This could be achieved by introducing a small change in the solution. Each cluster head has a list of its neighbors (the nodes covered by its radio frequency) and hence it can act as a cluster head for other nodes in the next levels. So the introduced problem for clustering a widespread distributed network could be solved in some levels.

5. Evaluating the Efficiency of Proposed Method

5.1 Basic Subject in Evaluating the Efficiency of the Proposed Method

5.1.1 The Nodes Connectivity

Connectivity in its local form is the possibility of sharing at least one encryption key between 2 sensor nodes inside the same radio frequency. In its global form, it is the quotient of dividing amount of nodes that obtain their key in distribution step by total amount of nodes in the network.

To analyze the local connectivity of the nodes, considering that public encryption key is shared between all the nodes and the definition of local connectivity[13] in both different structures of the network, existence of at least one key between the nodes is certain. Therefore regardless of nodes location for secure data transmission, it is enough to transmit a message containing the node ID after encrypting it by public encryption key.

But for global connectivity it should be mentioned that the purpose of evaluating this parameter is investigating amount of nodes that obtain their key using our proposed solution. Hence by simulating the proposed solution using Visual Sense software, we've obtained these parameters[14]. The performed experiments in both structures indicate that all the existing nodes in the network are capable of obtaining their encryption key in any step via the proposed method.

5.2 Connectivity of the Nodes in Distributed Structure

In 60 seconds, we have distributed encryption keys having 30 sensor nodes that have the radio range of 100m and are distributed in an environment with 500*500m dimensions. The key distribution in the proposed method is performed by transmitting messages in the network and the nodes start generating it by obtaining the new encryption key version number. Then any node that can connect to at least one node in the network and receive a message from it, updates the encryption key by considering version of new key and regardless of location and delay in receiving the messages.

The results obtained from this experiment as shown in Figure below indicates that all the member nodes in the network obtain the new key.

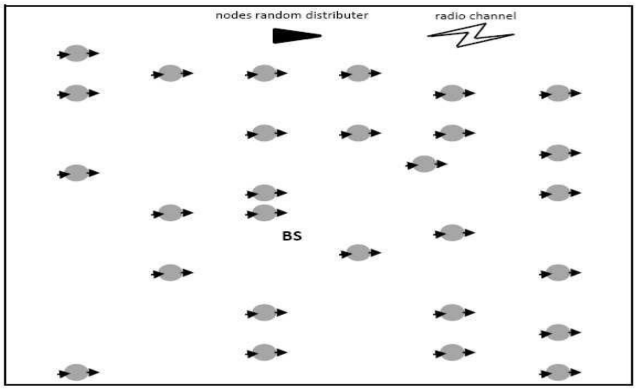


Figure 6. Distribution of the primary encryption key in simulation environment.

In this experiment the nodes obtaining the primary key are shown in grey.

Then the base station issues the instruction for updating the key in the network by sending the version number of the new key.

As it has been depicted in Figure 6, the nodes show receiving of a message by turning to white.

+After the base station broadcasted the version number of the second encryption key by transmitting a message in the network, and the nodes their key and also by transmitting the mentioned message informed the others of updating their key, the base station broadcasts the third key in the network The black nodes in Figure 8 indicate this point.

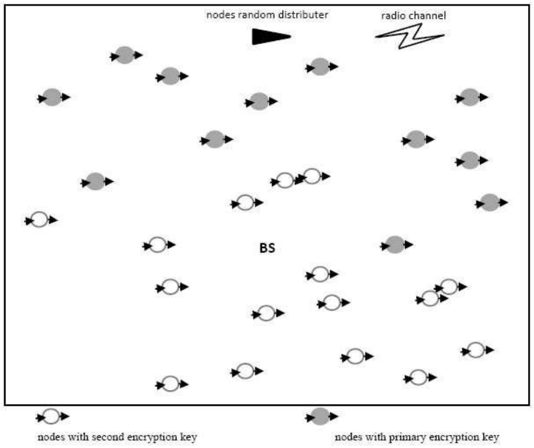

nodes random distributer
radio channel
BS
nodes with second encryption key
nodes with primary encryption key

Figure 7. Distribution of the second encryption key in the network.

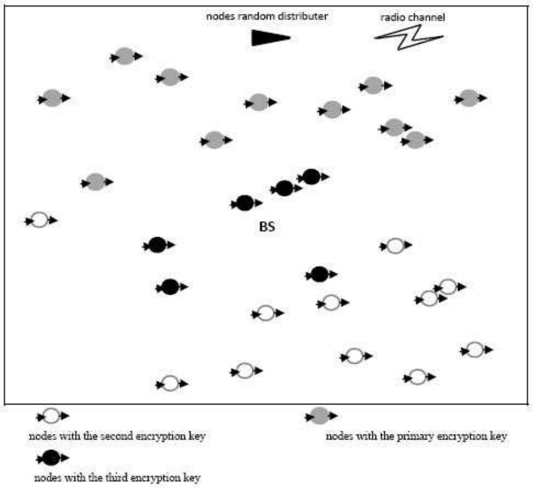

nodes random distributer
radio channel
BS
nodes with the second encryption key
nodes with the primary encryption key
nodes with the third encryption key

Figure 8. Distribution of the third encryption key in the network.

By performing this experiment on 16 and 32 nodes, the same results were obtained. Regarding the obtained results, it was observed that the global connectivity is established in the proposed solution completely. So we presume that the member nodes in the network will be able to generate an encryption key proportional to the existing one in the network by the use of encryption key generator function and after receiving the version number of the new key. Therefore the global connectivity is established in the network. But if for any reason, no update message is received, the node will be completely useless, since the mentioned node cannot communicate with other nodes of the network. This highly depends on used routing algorithms and the nodes characteristics, such as their coverage range.

5.3 Connectivity in Hierarchical Structure of the Proposed Method

The connectivity parameter is also considered in the hierarchical structure by some experiments. Hence, as it is shown in Figure 9, the experiment was carried out by distributing 32 sensor nodes, each have the radio range of 100m, in an environment with 500*500m dimensions and 4 clusters, that each has a cluster head. In this experiment, the base station was placed in the center of the environment and the cluster heads communicate with it.

After placement of the nodes in clusters, the base station starts distributing the primary encryption key in the network. This is shown in Figure 10.

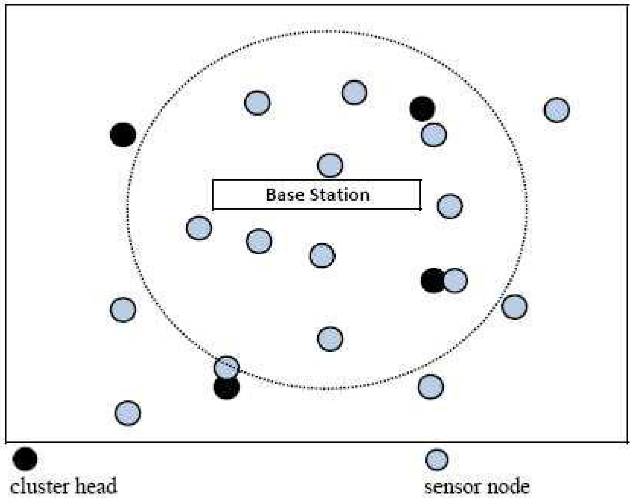


Figure 9. Clustered Nodes in simulated environment before distribution of the encryption key.

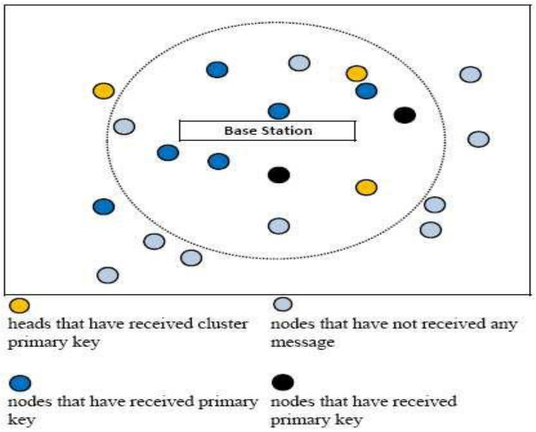

Base Station
heads that have received cluster primary key
nodes that have not received any message
nodes that have received primary key
nodes that have received primary key

Figure 10. Receiving the primary cluster encryption key by the nodes.

As shown in Figure 5-5, the cluster head nodes distribute the clustered primary key in cluster level, after receiving it (the nodes with black spot in the center of Figure 10). Also the ordinary nodes of the network will obtain the clustered key by comparing their cluster ID and the existing cluster ID in the message that contains the primary cluster key. (Grey nodes in Figure 10)

But the cluster heads that are not connected to the base station in single-hop form, obtain their key by the help of the node or nodes that are within the radio range of the base station or another cluster.

Figure 11 indicates this situation. In this case, the node that has obtained its primary cluster key, by checking the cluster ID, takes it to the cluster head.

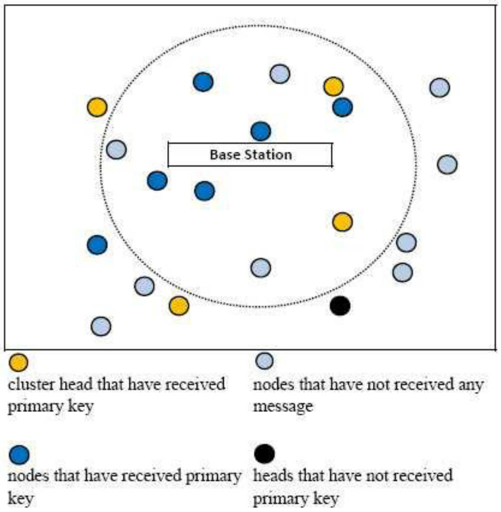

Base Station
cluster head that have received primary key
nodes that have not received any message
nodes that have received primary key
heads that have not received primary key

Figure 11. Receiving the primary cluster encryption key by the cluster head by multi-hop connection and distributing the second cluster encryption key.

As it can be seen in Figure 11, each cluster head that obtains the cluster encryption key, starts to distribute a new cluster key in the cluster. The dark grey nodes have received the new key from their cluster head.

In this experiment we found that all the network nodes, regardless of their location in clusters, can obtain the encryption key and hence the global connectivity is fully established in this structure for the proposed solution.

We believe that if the existing nodes in the network can at least establish a connection with a node that has received its new encryption key, they will be able to update the encryption key regarding to the version number, the function of key distribution, and also the cluster primary encryption key. In these experiments, as in distributed structure, all nodes update their encryption key, except the nodes that do not have their primary cluster key and/or are disconnected from the nodes which have updated their cluster key. By investigating both the above situations, the reason of the occurred problem was that the node hadn't received the update message. Similar to the distributed structure, the nodes that belong to higher groups and cannot connect to their neighbors cannot play role in network operations.

6. Unauthorized Access to the Information

In the proposed method, the space of the memory of the sensor nodes are divided into 3 parts of temporary memory RAM, memory used for executing codes, and permanent memory. In this case all the information needed by the nodes are stored and kept in these three parts. The attacker, by capturing the sensor nodes could only access the existing information in the memories used for executing codes and the permanent memory. Therefore, since the information of sensor nodes such as the physical characteristics of the node and the manufacturer are in the permanent memory, and also application and parameters used to generate new encryption key (like information used to generate new encryption key) are stored in RAM. Physical information is not useful for the attacker. Generally, by capturing the sensor nodes, the attacker can find information from the physical situation of the nodes and also the information which are not important.

This information also can be obtained by access to the transmitted message. Hence, the important or secret information of nodes including encryption keys, and encryption key generator function is placed in temporary memory. (RAM) If the attacker wants to access the information in this memory, the sensor node will be turned off. Therefore, access to temporary memory of the nodes in the proposed method, is impossible and hence access of the attacker to secret information of the encryption keys in RAM of sensor nodes is impossible.

Of course, in the existing solutions, we can see similar assumptions, too. For example in BROSK method, it is assumed that the main common key between the nodes can never be revealed[15] or the LEAP method assumes that the function of generating encryption key and the primary key is deleted from the nodes memory[16].

7. Scalability in the Proposed Method

Another important thing in evaluating the efficiency of encryption key management in wireless sensor networks is the scalability of the proposed solution in the network. Scalability is the possibility to increase the sensor nodes in network without losing the existing arrangements. In our proposal, the identity of a sensor node in the network depends on the information that at the time of joining the network has been allocated to it. This information includes the function of generating the encryption key and the primary encryption key relative to the introduced structure. Generally this information is assigned to node considering the defined application of the network at the time of manufacturing in the factory, or by the base station, at the time of distribution. Then due to both different structures in our proposal and also the programming of the nodes, the added node will be able to declare its existence as a new member, in the network and update the encryption key using the received information. Since adding a node to the network has no time limitation and also due to above descriptions, the scalability of the wireless sensor network will not face problems in our proposal. It should be mentioned that the scalability in the existing solutions performs the same as in our proposal. In the LEAP method, the nodes that have the function of generating encryption key and the primary key can be added to the network[16]. In BROSK method too, the nodes have the common primary key (set in the factory), can be added to the network

8. The Rate of Consumed Memory in Sensor Nodes to Store Encryption Key

Sensor nodes can't have large-size memories due to their structure. So one of the important factors in distributing the encryption keys in wireless sensor networks, is reducing the size of the consuming memory required to store the encryption keys. By comparing the rate of consumed memory in the proposed method and LEAP method, which is a frequently used and well-known method in hierarchical structure, we pretend that our proposed method in this field has a better performance. Also in distributed structure, we compare our proposed method, KEPS and BROSK method.

As explained in other sections, the LEAP method establishes security in transmitting information in this network by distributing four types of keys of individual, paired, cluster, and group in the wireless sensor networks. If we assume that we need one memory unit to store each key and also L units of memory for the key generator function, F, and also considering that each node in each cluster has D neighbors, then for storing the key we need K units of memory according to the following relation:

$$K(\text{LEAP}) = 1 + 1 + D + 2D + L \qquad \textit{Relation } 2$$

The memory consumed in the following way to store the encryption keys in each sensor node:

- Individual encryption key; 1 unit
- Group encryption key ;1 unit
- Cluster encryption key, as much as neighbors of (D); D units in each cluster.
- Paired encryption key, twice the number of neighbors of each node (D); 2D units in each cluster
- Encryption key generator function; L units

Now if we compare it with the consumed memory in each node by the BROSK method, we will end up the following results:

$$K(\text{BROSK}) = 1 + 1 + 1 + D + D \qquad \textit{Relation } 3$$

In BROSK method the obtained amount in relation 5-2 will be used as follows:

- The main common encryption key; 1 unit
- Node ID number (ID_A);1 unit

- One-digit random number (N_A); 1 unit
- Encryption key for each neighbor; D units (D could be total number of network nodes)
- ID of neighbor nodes; D units (D could be total number of network nodes)
- By considering the memory parameter in hierarchical and distributed structures, we are going to analyze the consumed memory for each node.

8.1 Hierarchical Structure of KEPS Method

As explained before each node in this structure stores the following items in its memory. The sensor nodes secure the transmitted messages by storing public and cluster key. Therefore regarding the descriptions, each node needs M units of memory in KEPS method.

$$M = 1 + 1 + 1 + 1 + L \qquad \textit{Relation } 4$$

- This amount of memory is used to store encryption keys in each sensor node in the hierarchical structure in the following way:
- Cluster encryption key; 1 unit
- Public encryption key; 1 unit
- Encryption key generator function; L units
- Cluster ID and ID of each member node in the network; 1 unit
- Sequence number of encryption key; 1 unit (due to small value of this number, in conditions that the number of changes of encryption keys are too many, 1 unit is assigned to it)

8.2 Distributed Structure of the Proposed Method

As explained earlier our proposal was presented for two different structures.

In distributed structure, similar to the hierarchical structure, each node requires information for security and protecting the context of the messages. As in part 4-2-2, this case could be considered in two ways; firstly, by considering a public encryption key for each node, and secondly, by changing the distributed structure to the hierarchical. Here we've tested both ways. In the first solution, each node needs to have a public key, encryption key generator function, and its own ID number. Hence the rate of consumed memory needed for each node in this structure is as follows:

$$M = 1 + 1 + 1 + L \quad \text{Relation 5}$$

In this case the memory to store the encryption keys in each node is used as follows:

- Public encryption key; 1 unit
- Encryption key generator function; L units
- ID number of each node; 1 unit
- The sequence number of encryption keys; 1 unit (due to small value of this number, in conditions that the number of changes of encryption keys are too many, 1 unit is assigned to it)

In the second solution in this structure each node needs the public key, cluster key, encryption key generator function, and the list of Id numbers of its neighbors. According to this the connection 5-4 for distributed structure will be as follows:

$$M = L + E + 1 + 1 + 1 + 1 \qquad \text{Relation 6}$$

In this case the memory is used to store the encryption keys in each node as follows:

- Public encryption key; 1unit
- Cluster encryption key; 1 unit
- Cluster ID and ID of each node in the network; 1 unit
- Encryption key generator function; L units
- List of ID numbers of neighbors of each node; E units (Certainly the size of the ID of each node is less than an encryption key. So we can say E<d.)
- Sequence number of the encryption keys; 1 unit (due to small value of this number, in conditions that the number of changes of encryption keys are too many, 1 unit is assigned to it)

With regards to the mentioned statements, we showed that the required memory for each sensor node in both structures of the proposed method is reduced as compared with the existing methods of LEAP and BROSK.

8.3 Communication Overhead

As explained in other sections, in present key distribution methods, the key distribution server node distributes a new key in network and then sends messages using this encryption key[17]. It means that all the members of the network should receive the new key in each update step of encryption key.

But in KEPS method the new keys that are generated by network nodes, are transmitted by messages which are going to be encrypted with new keys. Hence regarding the descriptions a new encryption key is generated when a refreshing is required. In this case the nodes are engaged with changing and updating the encryption keys that are sending a new message in the network.

But the most important point is that in the proposed method in contrast with all the existing ones, the overhead of the distributing encryption keys is very light compared with overhead of communication of sensor nodes. In this method we do not transmit any messages just for distributing the encryption keys. We use typical messages for communication and processing operations in the network.

Hence, in result, in addition to reducing the number of nodes engaged in updating encryption keys, the new key and also the context of the message in the network is transmitted by a single message. This is done in two steps: first by transmitting a new encryption key and second transmitted message context, separated and in a distinct massage. To present more officially, the transmitted bits in normal case will equal n × L + K × n, by assuming transmission of K messages with n bits length, and L times of distributing the encryption key in the network. If we analyze this in the proposed method and assume that s bits are added to each message as the message header, we'll have:

$$(n+s)\times L+n\times(K-L)=L\times s+K\times n \qquad \textit{Relation 7}$$

Considering s<n, the rate of transmitted bytes will reduce and hence the overhead will be effected. For example, we transmit a 16-bit message per minute in a network, as a typical message. Also assume that we change and send the encryption key 10 times per hour. Then the working overhead of the transmitted bits in the network, will equal 16 × 60 + 16 × 10 = 1120. Now if according to the proposed method, the transmission of the encryption key is done in the network and it is assumed that 8 bits are added to the header, this number will be reduced to 10 × (8 + 16) + 50 × 16 = 1040.

Another important note in KEPS method is that the security of distributing the encryption key is higher than existing methods, because of automatic, alternative, and flexible changes of encryption key and also fewer transmissions of messages containing encryption key in the wireless sensor network.

9. Energy Consumption

After simulation of the proposed method from the viewpoint of the nodes connectivity, using Visual Sense software, we decided to measure the energy consumption of the network in the process of distributing the encryption key and analyze KEPS method by comparing this parameter with that of similar existing methods. But apart from the graphical interface of Visual Sense simulator, that helped us in the analysis and regarding the lack of an accurate energy consumption model, we tried to use another simulator to evaluate the consumed energy and performance of the proposed method compared with different solutions. JiST/SWANS[18] is a network and application layer simulator which uses simple and easy to understand method. This tool is scalable. We use an energy model in this tool, which was the closest to reality. This consuming energy model considers 660, 390, and 22 milliwatt consequently for transmitting, receiving, and in idle time of nodes[1,19,20]. The criterion of evaluation is the energy consumption. In the simulation model for the two introduced structures, the evaluation of energy consumption was performed as follows:

9.1 The Rate of Energy Consumption in Distributed Structure of KEPS Method

In this case, the experiment was setup in two steps for 50 distributed sensor nodes, in an environment with 200× 200m area. Each node has a 40m radio range and a primary energy for the nodes is assumed 10 joules. The simulation is done for 100 seconds. In this experiment we assume that the sink is located in the middle of the nodes distributing border. In the simulation that its results are shown in Figure 12, after configuration time that took 5 seconds, the base station distributed the encryption keys in the network by transmitting the updating instruction of 2 encryption keys in different times. One of the most similar existing methods with distributed structure of the proposed method is the method of BROSK. So we compared our method with BROSK. In result we observed that KEPS method has less energy consumption. The BROSK method, as one of the most similar methods to our proposed method was analyzed. In this method two messages were needed for distributing an encryption key and each key was generated at the end- node of each communication route. If the base station wants to distribute a key to establish a secure connection, this key should be created for nodes as destination and for tye base station as distributor, regardless of single-hop or multi-hop form of connection of nodes to the base station. But in the proposed methods, only one encryption key is updated for each network node. The base station performs key distribution by transmitting only one message.

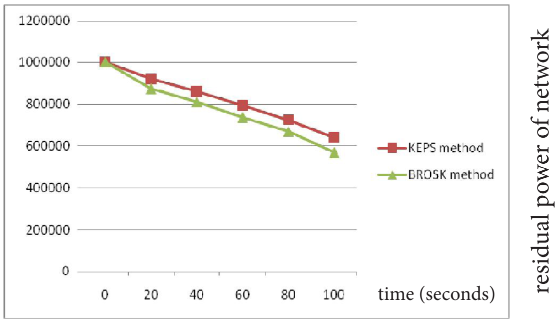


Figure 12. The residual energy of the network in KEPS method and BROSK method for distributing two encryption keys.

According to the explanations, the working overhead for distributing the encryption key in KEPS method is somehow lower than the BROSK method. It's obvious that if the number of distributed encryption keys increases, our method transmits less messages compared with the method of BROSK and the performed operations for distributing the key is less in the proposed method than the BROSK method. So as it is indicated in Figure 13, we conclude by repeating the experiment for 5 encryption keys that our solution is much more secure.

Therefore if the number of distributing keys increases, the efficiency of KEPS method increases.

9.2 The Rate of Energy Consumption in Hierarchical Structure of KEPS Method

After considering the energy consumption of our proposed method in distributed structure, and showing that the proposed method reduces the energy consumption by decreasing the number of transmitted messages we tried to analyze the obtained results in the hierarchical structure of the proposed method.

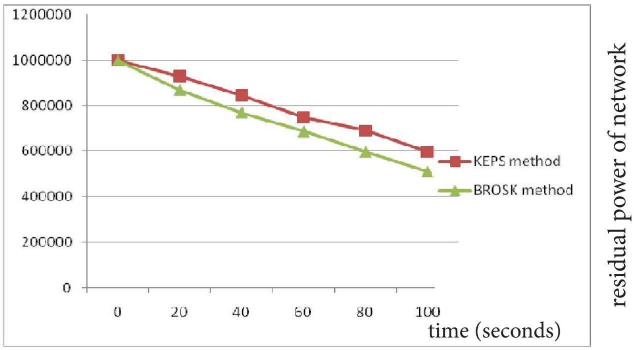


Figure 13. The residual energy of the network in the KEPS method and the method of BROSK for distributing five encryption keys.

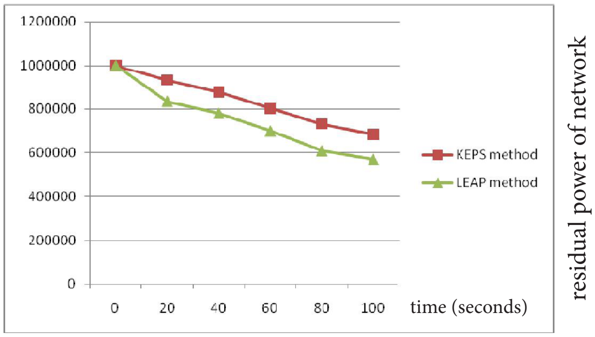


Figure 14. Residual Energy of the Network in the proposed method and the LEAP method for distributing two encryption keys.

So we did our experiment again in two steps on 50 distributed sensor nodes in an environment with 200×200 area. Each node has a radio range of 40 meters and the primary energy of the nodes is assumed to be 10 joules. In this structure the nodes are randomly distributed in the environment by dividing the simulated environment into 4 addressable regions and we did the experiment by locating them in 4 clusters.

The simulation is done in 100 seconds and the configuration time is 5 seconds. In this section we assume that the sink is situated in the middle of the nodes distribution border. In the first step, we did the experiment by distributing 2 encryption keys.

With respect to the considerations we selected the method of LEAP as a similar method for the hierarchical structure of KEPS method. As the results from the experiment show in Figure 14, the LEAP method consumes a lot of energy to generate pair-wise keys in each cluster after starting and during the configuration. But in our method such amount of energy is not consumed. Therefore energy los is a cost that the LEAP method pays for it. After that the network comes up and encryption key is distributed in every cluster, each header distributes only the version number of the new key in the cluster. But in each step of distributing the cluster key in the LEAP method, each head distributes new cluster key using a message that has been encrypted by pair-wise key of each node in the cluster. With regards to the analysis, our method in this structure consumes less energy compared with similar method.

In the second experiment, by increasing the new distributed key to 5 encryption keys, we tried to engage more nodes in distributing the encryption keys. Figure 14 indicates the results in this experiment. It can be observed that both experimented methods consume more energy by increasing the keys. But the important point is that this amount in LEAP method is rather more than our proposed method. We can conclude that in LEAP method by increasing the number of distributed encryption keys, we need to transmit D messages in the header for each distribution. But in our proposed solution this is done by only one message in the cluster.

In Table 2 this conclusion has been stated for hierarchical structure. As it can be seen according to the existing parameters, the proposed solution is rather better than the similar method in hierarchical structure. (LEAP)

Table 1. Conclusion of the comparison between KEPS method and the existing similar method in distributed structure

Proposed method (KEPS) with distributed structure	Method of BROSK	
All the nodes	The nodes covered by the key server	Global connectivity
With public key	With the main common key	Local connectively
It's assumed that RAM is inaccessible	It's assumed that the main common key is not revealed	Possibility of being captured
Nodes have a primary key and a key generating function	Nodes have a common main key	Scalability
3 + L	3 + 2D	Consumed memory
One message for pair-wise key and one message for public key	Sending 2 messages For pair-wise key and 2N Messages for public key	Communication overhead
2 keys: Each node 3.7 joules approx. 5 Keys: Each node 4.1 joules approx.	2 keys: Each node 4.6 joules approx. 5 keys: Each node 5.1 joules approx.	Consumed energy

D: number of neighbors of key distributing node
L: size of the required memory to store the encryption key generator function N: total number of the network nodes

According to the obtained results from the experiments it can be observed that our proposed solution can improve the effective parameters in encryption key distribution in wireless sensor network in both structures. By reducing 15 to 20% of the consuming energy, reducing the number of transmitted messages for distributing the encryption key, and also the consumed memory to store the keys, (according to Tables 1 and 2) our method can distribute the encryption key without violating or threatening the data security of network, nodes connectivity, and the network scalability.

Table 2. Conclusion of comparison between KEPS Method and the existing similar method in hierarchical structure

Proposed method with hierarchical structure	LEAP method	
All the nodes	All the nodes	Global connectivity
With pubic key	With group key	Local connectivity
It's assumed that RAM is Inaccessible	It's assumed that generator function and primary key are removed	Possibility of being captured
The nodes have primary key and the key generating Function	The nodes have generating function and primary key	Scalability
4 + L	L + 2 + 3D	Consumed memory
One message for the pair- wise key and one message for the cluster key	Sending 2 messages for the pair-wise key and D messages of the cluster key	Communication overhead
2 keys: Each node 3.5 joules approx. 5 keys: Each node 4 joules approx.	2 keys: Each node 4.5 joules approx. 5 keys: Each node 5 joules approx.	Consumed energy

key distributing node
D: number of neighbor nodes
L: size of the required memory to store the encryption key generator function

10. Conclusion

There are many different methods to establish a secure network in wireless sensor networks among them encryption of messages using the encryption keys is an important one. Because of different structure of wireless sensor networks to other type of networks there are some more factors that have to be considered in order to establishment and distributing of encryption keys in these type of networks, such as mobility of nodes, wireless characteristics, and limited power resources.

In our proposed method in this study (KEPS) we tried to introduce a key management protocol with the reduced amount of key transferring and communication, and with less storage space. We tried to improve the security in this method by applying a refreshing method.

What was achieved by this evaluation indicates that the proposed strategy, in cases which the data security is important and more keys in the network are required, performs much better and more efficient than the existing methods.

This research could be continued in future for exactor the following matters:

- Using the proposed method in pervasive computing environment-since in pervasive compiling environments the operators have less confrontations with thesis and most of the operations are done by the system, in another words the system is self- organizing[47,48] this aim could be achieved by introducing an automatic system with no interferes by the operator. Since in the proposed method the process of distributing encryption key is done automatically and by transmitting messages in the network, and these operations need no human operators, this method can be used to distribute the encryption key.
- If we change this method in order to have the sequence number of the encryption key as a parameter of time measurement, then it can be used to synchronize wireless sensor nodes.
- The proposed method has made some assumptions, including inaccessibility of RAM information. (not revealing the information in captured nodes by the attacker) By introducing a solution for preventing the capture of nodes, other methods can be introduced.
- In this method all the important information of the nodes are in RAM. Hence if a node for any reason loses the information of its memory-including the primary cluster encryption key and the node ID, etc, how can the nodes continue their activities? This could be achieved in future by transferring an strongly encrypted copy of information to permanent memory.
- In distributed structure of the proposed method that needed cluster encryption keys, the base station node was connected to cluster head nodes and also the headers were connected to the member nodes in the cluster in single-hop form. We can have nested clusters in future by introducing better solutions.
- Considering the reduction of the number of transmitted messages and also the energy consumption in the proposed method of distributing the encryption key in wireless sensor networks, the idea could be used for distributing encryption keys in smart dust network in which the parameters of security and energy consumption with regards to the nodes structures and the utilization of this type of network are of great importance.
- The issue that can threaten the global connectivity of a group of the nodes is the failure of network critical nodes, like cluster heads or nodes which connect clusters to each other, in distributing encryption keys. Introducing a method for replacing such node to continue the process, can be considered in the future, though we could somehow solve the problem by introducing an appropriate solution.

11. References

1. Ivan Stojmenovic. Handbook of Sensor Networks Algorithms and Architectures. University of Ottawa: John Wiley; 2005. p. 25–153.
2. Akyldiz IF, Su W, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. IEEE Communications magazine. 2002 Aug; 40(8):102–14.
3. Xu N. A Survey of Sensor Networks Applications. 2006 Sep; Dept of Comp Sci, Univ of Alberta. Technical Report CSC 333.
4. Onel T, Onur E, Ersoy C, Delic H. Wireless Sensor Networks for Security: Issues and Challengaes. Springer; 2006. p. 95–119.
5. Undercoffer J, Avancha S, Joshi A, Pinkston J. Security for Sensor Networks. CADIP Research Symposium; 2002.
6. Erdal C, Rong C. Security in Wireless Ad Hoc and Sensor Networks. John Wiley; 2009. p. 109–16.
7. Camtepe SA, Yener B. Key Distribution Mechanisms for Wireless Sensor Networks: a Survey. Rensselaer Polytechnic Institute, Computer Science Department; 2005 Mar.

8. Goldreich O, Goldwasser S, Micali S. How to construct random functions. Journal of the ACM. 1986; 33(4): 792–807.
9. Bishop M. Computer Security. Addison Wesley; 2003.
10. Lee J, Sinston D. A Combinatorial Approach to Key Pre-Distributed Sensor Networks. Second European Workshop on Security in Ad hoc and Sensor Networks; 2005; Budapest, Hungary. Available form: http://www.cacr.math.uwaterloo.ca/~dsinston/publist.html
11. Hwang J, Kim Y. Revisiting random key pre-distribution for sensor networks. ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04); 2004; Washington DC. USA.
12. Kun S, Peng P, Peng N, Wang C. Secure Distributed Cluster Formation in Wireless Sensor Networks. The 22nd Annual Computer Security Applications Conference (ACSA'06); 2006 Dec; Shanghai, China. p. 131–40.
13. Mei SD, Bing H. Review of key management mechanisms in wireless sensor networks. Acta Automatica Sinica. 2006 Nov.
14. VisualSense [cited 2009-2010]. Available from: http://ptolemy.berkeley.edu/visualsense
15. Lai B, Kim S, Verbauwhede I. Scalable session key construction protocol for wireless sensor networks. 1st International Workshop on Peer-to-Peer Systems (IPTPS'02); 2002 Mar; Cambridge, MA, USA.
16. Zhu S, Setia S, Jajodia S. LEAP: efficient security mechanisms for large-scale distributed sensor networks. 10th ACM conference on Computed Communications Security. 2003; Washington DC, U.S.A. p. 500–28.
17. Deng J, Han R, Mishra S. Enhancing Base Station Security in Wireless Sensor Networks. 2003. Department of Computer Science, University of Colorado. Tech Rep. CU-CS-951-03.
18. Swans J [cited 2009–2010]. Available from: http://jist.ece.cornell.edu/
19. Intanagonwiwat CH, Govindan R, Estrin D. Directed Diffusion: A scalable and robust communication paradigm for sensor networks. The Sixth Annual International Conference on Mobile Computing and Networks (MOBICOM); 2000 Aug; Boston, Massachusetts, ACM. p. 56–67.
20. Delicato FC, Pires PF, Costa CLFR, Pirmez L, Rezende JF. Reflective Middleware for Wireless Sensor Networks. 20th Annual ACM Symposium on Applied Computing, ACM SAC'2005; 2005 Mar; Santa Fe, New Mexico, USA. p. 1155–9.