Wireless Personal Communications (2020) 112:2373–2391
https://doi.org/10.1007/s11277-020-07155-y

A Novel Method for Key Establishment Based on Symmetric Cryptography in Hierarchical Wireless Sensor Networks

Hamid Mirvaziri[1] · **Rahim Hosseini**[1]

Published online: 21 January 2020

Abstract
Most of recent researches carried out into wireless sensor networks are concerned with homogeneous networks in which all network nodes have the same features and key management mechanisms presented for them aim to enable as many nodes to establish shared cryptographic keys as possible. Hierarchical networks have shown a better performance because of using clustering features. This issue motivated us to present a method for key establishment in hierarchical wireless sensor networks based on symmetric cryptography. Since symmetric cryptosystems consume to have low energy, they are a good choice for sensor networks. Although symmetric cryptosystems consume to have high memory, this shortcoming can be diminished by appropriate techniques. The simulation results have shown that our proposed method can significantly reduce the memory consumption resulted from saving keys alongside reducing the energy consumption resulted from communications in all network nodes than related works while it presents a favorable level of security.

Keywords Hierarchical wireless sensor networks · Key establishment · Symmetric cryptography

1 Introduction

Wireless sensor networks (WSNs) that have various applications both in military and non-military domains are comprised of a set of small nodes used to collect data from a particular environment. WSNs don't have a specific infrastructure or topology and its constituent nodes are faced with Fundamental limitations in terms of energy, memory and processing power. The limitations of WSNs don't let us apply normal mechanisms in these networks. For example, most cryptographic mechanisms are based on asymmetric cryptography. These mechanisms, however, can't be used in WSNs due to their high energy consumptions. In WSNs, consequently, mechanisms should be presented which fit the particular conditions and limitations of these networks. WSNs can be divided into homogeneous and heterogeneous networks. In homogeneous networks, all nodes have the same features in

✉ Hamid Mirvaziri
h.vaziri@gmail.com

[1] Computer Engineering Department, Shahid Bahonar University, Kerman, Iran

terms of energy, memory and processing power. Besides, all of them play the same role in networks. These networks have their own limitations which are pointed out in some researches [1, 2].

Since exchanging data between sensors are important, especially in military applications and are exchanged through wireless media, a security mechanism should be available to make sure of integrity and confidentiality of data. One of the fundamental challenges in key management protocols based on symmetric cryptography is establishing shared secret keys between nodes, so they can encrypt data, and use basic security services including message confidentiality, data integrity and also authentication using these keys.

Symmetric or asymmetric algorithms can be applied to encrypt data in WSNs. Symmetric algorithms such as AES [3] and 3DES consume low amount of energy, but their memory consumption is naturally high. Using appropriate techniques, however, as will be stated in the following sections of this article, can counteract this drawback. In other words, it is possible to use natural characteristic of symmetric cryptography, which is low energy consumption, and also reduce memory consumption resulted from saving the keys. On the other side, although asymmetric algorithms require low memory, they consume so much energy compared to symmetric algorithms [4]. This issue causes symmetric algorithms being more favorable to be used in sensor networks since nodes of these networks aren't capable to perform asymmetric algorithms for a long period, since they have limited resources of energy. In this research, we can, secure all wireless media and encrypt the data transferred through them by saving only one key in each cluster node and two keys in each sensor node; this method can be applied in networks with any scales and numbers of nodes. Thus, the number of keys saved in different nodes does not depend on number of cluster heads and sensor nodes as it is fixed under any circumstances, and is equal to one for cluster heads and two for sensor nodes. In HWSN, nodes are divided into three groups regarding their hardware and resources; nodes with higher resources act as cluster heads, and nodes with more limited resources sense and collect data in each cluster [5–7]. Data collected by these nodes should be sent to cluster heads; then the data collected in all cluster heads should be sent to base station which is unique in the network and has the highest amount of resources among all network nodes. Cluster heads are connected to base station either directly or indirectly through other cluster heads. Sensor nodes, which should be located in each of the network clusters, are connected to cluster heads directly or through other sensor nodes which are in the same cluster. A sample of HWSNs is illustrated in Fig. 2.

The following conditions should hold in order to encrypt the exchanged data, and to ensure that all wireless media are secure: (1) if a cluster head is directly connected to the base station, a key should be shared between the base station and that cluster head. (2) If a cluster head is connected to the base station indirectly through another cluster head, a key should be shared between those two cluster heads. (3) If a sensor node is connected to a cluster head directly and without an intermediary sensor node, a key should be shared between them and (4) if a sensor node is connected to a cluster head through another sensor node, a key should be shared between those two sensor nodes. If a key establishment can satisfy these four conditions, we can be sure that all collected data in sensor nodes will be received by base station in an encrypted form and intruders won't be able to access the data.

Generally, key establishment mechanisms based on symmetric cryptography and key transfer policy which also authenticate the new nodes in HWSNs can be divided into four stages: (1) entrance of new node to network and selection of parent. (2) Sending the registration request to competent authority by new node through its parent node. Competent

authority can be different in the different protocols; in our protocol, the base station is responsible for this role. It is responsible for key generation and authentication of the new nodes. (3) Authenticating the requesting node by the competent authority. (4) Final operations of key establishment.

Some of researchers discuss about authentication and key agreement scheme using WSNs for specific purposes, for example, WSNs for healthcare through body sensor networks [8–10], WSNs for military [9] or multimedia [11] or agriculture monitoring [12].

As mentioned above, the base station, in our proposed method, has been used as the only competent authority to authenticate new nodes and generate keys. At first, maybe it seems that under such circumstances the base station will be a performance-bottleneck for the network, and the whole key establishment mechanism will be destructed upon losing base station, but it is completely wrong because of the following two reasons: Firstly, regarding the features of HWSNs mentioned in [13], base station is trusted and its energy is not supplied by battery. In addition, it is not located in unattended areas like cluster heads and sensor nodes; it is not suffering from limited resources like cluster heads and sensor nodes. As it is mentioned in [2], base station has rich resources which can be used for data processing when it is necessary. Secondly, since the base station is where all collected data in network are sent towards, if the base station is not accessible or it is destroyed, the whole network will be destroyed. In this situation collected data will not be able to get to the networks' owners. Therefore, using base station as competent authority can't lead to problems that using battery-powered cluster heads as competent authority can. So Cluster head considered an assistant who has all the information of network, When base station is not accessible or destroyed to keep the network in the normal condition.

The following sections of this paper are organized as follows: in Sect. 2, we will review the related works in the field of key establishment in WSNs. Section 3 provides the important concepts about key establishment in WSNs, which are important to understand our proposed method. The proposed method will be completely explained in Sect. 4. Evaluation of the proposed method in terms of authenticity and its resilience against various security threats will be studied in Sect. 5. Section 6 involves simulation and comparison with related works. Acknowledgment is in Sect. 7. Finally, conclusion will be presented in Sect. 8.

2 Related Work

In this section, a brief explanation is given about the previous methods presented in the field of key establishment in WSNs. These methods can be generally divided into two groups which will be mentioned as follows: 2.1. The first group includes methods based on asymmetric cryptography [13–16]. In these methods, asymmetric cryptography algorithms such as ECC [17] and RSA are mainly applied; these algorithms, generally, reduce the memory consumption resulted from saving keys compared to symmetric algorithms. These algorithms have a variety of applications in wired networks because of presenting a high level of security. However, they are not considered as a good choice for WSNs due to high energy consumption. Although energy consumption of ECC algorithm is less than RSA [18], this level of energy consumption is too high for WSN nodes which face a lot of resource limitations. In some other methods, a combination of asymmetric and symmetric algorithms is used [19–21].

The second group includes methods based on symmetric cryptography [22–26] in which cryptographic algorithms like AES can be used [27]; symmetric algorithms have shown a good performance [28]. One of the usual techniques for key establishment based on symmetric cryptography in sensor networks is Random Key Pre-distribution [29, 30] in which a set of keys called "key-pool" is available before network initialization, and subset of key pool should be saved in the memory of each node which is called "keyring" and the number of its keys is equal to K. Moreover, the total number of initial keys in key-pool is equal to P. Selecting appropriate number of keys for each node and the initial number of keys in "key-pool" is very influential and vital in performance of these techniques. This issue is very important because P and K have a direct impact on the possibility of existence of, at least, one shared key between adjacent nodes in WSNs. The smaller the K is, the less the probability is. It can be said, this is the main shortcoming of Random key pre distribution technique; we should save more keys in the memory of nodes to increase key connectivity while the wireless sensor network nodes are hard-pressed for available resources. On the other side, the key connectivity between adjacent nodes of network will be reduced if we would like to reduce the memory consumption resulted from saving keys. We can use various keying materials for key-pool. For example, it is proposed in [31, 32] that key pool includes P symmetric t-degree bivariate polynomials over Galois field Gf (q) where q should be a prime number and much greater than total number of nodes available in the network. In addition, it is suggested in [33, 34] that key pool includes P matrix M1, M2, …, MP with size N (t + 1) where these matrixes must be over Galois field Gf (q), and the N is equal to the number of nodes in WSNs, t is the security parameter and q is equal to a prime number which must be much greater than the total number of nodes.

As previously mentioned, in Random key pre-distribution, there is no guarantee that a shared key exists between all adjacent nodes in WSNs. Because of this reason, it is suggested in [35] that if no shared key is available between two adjacent nodes, these two nodes can generate a shared key in case of having at least one shared secret key with another node which is in Adjacency and radio transmission range of both of them; the third node is called "Trusted third party" (TTP), and should generate a new shared key for A and B as follows:

$$K_{A,B} = F\left(K_{A,C}K_{B,C}\right), C = \textit{the id of TTP} \tag{1}$$

$$E\left(K_{A,C}, K_{A,B}\right) \textit{ and send to node A}$$
$$E\left(K_{B,C}, K_{A,B}\right) \textit{ and send to node B}$$

To learn about cryptographic symbols you can refer to Table 1. Although the suggestion proposed in [35] can increase key- Connectivity level in some conditions, there is still no guarantee that, another node is available to perform as trusted third party. A method is proposed in [36] that can guarantee the existence of a shared key between any two nodes of network in random key pre-distribution. It works as follows: K unique keys must be selected for each node. Therefore, there are $\frac{P!}{K!(P-K)!}$ ways to select K keys. It means that there can't be more than $\frac{P!}{K!(P-K)!}$ nodes in the network. In case of K > P/2, this method guaranties that there is, at least, one shared key between any pair of nodes in the network, the shortcoming of this method is high memory consumption because K should be bigger than P/2.

Some other methods which are based on Random key pre-distribution technique are proposed in [37, 38]. For example, a suggestion proposed in [38] increases the level

Table 1 Notation for specifying cryptographic protocols

Symbol	Meaning
A, B	Sensor node A, B
$K_{A,B}$	Shared key between A and B
N_A, N_B	Random numbers generated by A and B
E (K, M)	Encryption of message M using key K
D (K, M)	Decryption of message M using key K
F (K)	Hash function F applied to K
||	Concatenation operator

of security in Random key pre-distribution. It is suggested that adjacent nodes A and B shouldn't use that key directly in cryptography in case of having a shared key. They should use it as a keying material to generate another key. For example, KA, B in ith communication between nodes A and B can be generated as follows:

$$Ki = F\left(ki - 1, K_{A,B}\right) \tag{2}$$

Generally, $K0$ is also available to all nodes in the network. If the intruders can access to $K_{A,B}$, so they can obtain all keys for future sessions between nodes A and B which is the disadvantage of this method. One of the simplest mechanisms in the field of key establishment in WSNs is Network-wide key in which only one key is shared between all network nodes [37–40]. Because of natural simplicity of this method, all protocols extracted from this method have a high level of flexibility. In addition, since only one key is saved in each node, memory consumption is minimal. In the networks using this method, it is so easy to add any number of nodes after the initial deployment of the network because only by saving one key, each node can enter into the network. In this method, key-connectivity between the network nodes is maximal. All protocols based on Network-wide key has basic security problem because intruders can decrypt the whole data of the network by accessing to only one of the network nodes. One of the challenges in the field of key establishment in WSNs is to share as many keys between network nodes as possible. According to the method proposed in [41], each of network nodes should have a shared key with n-1 (n is the total number of nodes) of the other nodes. Although this method presents the maximum level of Key-Connectivity, it results in so much memory consumption; this method cannot be used in networks with large scales because the limited memory of sensor nodes does not allow us to save such a large number of keys.

In some methods, key establishment is done alongside nodes authentication [13, 26]. These methods present a mechanism which is able to do the key establishment process in HWSNs with the features stated in [42], and also authenticate the nodes for which the new keys are to be generated. In [26], authors apply cluster heads as competent authority to authenticate the new nodes, and generate new keys for them; in some conditions, it results in saving a large number of keys in cluster heads and also sensor nodes. Moreover, because of using inappropriate criterion for new nodes to select their parent nodes, the level of energy consumption resulted from communications is too high.

In [13, 26], it is suggested that new nodes should select a node as parent which has the most signal strength; therefore, in normal conditions, the node which is closer to the new node is selected as the parent node. According to the scenario shown in Fig. 1, we

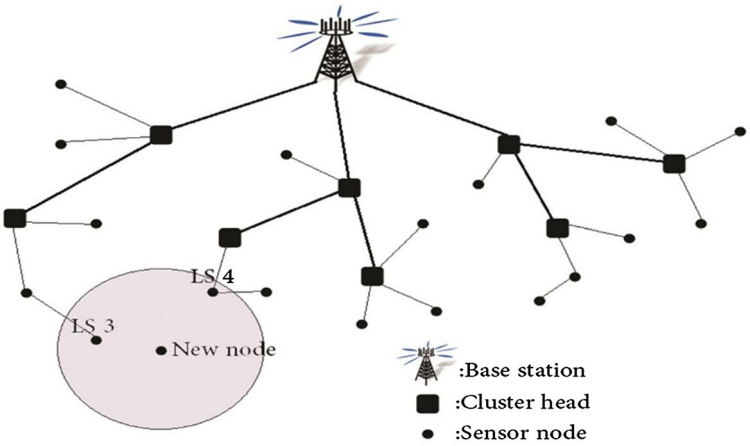

Fig. 1 The way new nodes choose their parent nodes in related works

can proof that this criterion increases energy consumption resulted from communication in many cases.

In the scenario presented in Fig. 1, when the new node enters the network, LS3 and LS4 are located in its transmission rang and based on criterion presented in [13, 26], LS3 should be selected as parent node because it is closer and therefore it has more signal strength. However, if LS4 was selected as parent node, it would lead to a condition in which more nodes are located in the route of sending data from the new node to the base station. Consequently, the average of energy consumption resulted from communication in whole network would be lesser. In these methods, it is also mentioned that to help the new sensor nodes to select their parent nodes, cluster heads must continuously and at unequal intervals send a message called "Hello" with the maximum power, so the new nodes can easily select their parents by receiving this message. This issue results in high energy consumption among cluster heads while in many sensor networks; it is possible that all sensor nodes are deployed in the target site at the beginning of network life time. After a while, either no new node enters the network or a few nodes enter the network while cluster heads should spend a large amount of their energy on sending Hello messages. Figure 1 shows the way in which new nodes select their parent nodes in related tasks.

3 Preliminaries

In this chapter, we will explain the HWSNs characteristics and their architecture. We will then study the cryptographic symbols in Table 1.

3.1 HWSNs

As previously mentioned, are consist of three different node types. The first type is sensor node which severely limited in terms of available resources. In HWSNs, sensor nodes are introduced as LS-nodes [13]. The second type is cluster-head which is in a

better condition than LS-nodes regarding resources, in spite of being battery-powered. In HWSNs, cluster heads are introduced as HS-nodes [13]. The third one is base station; in network, there is only one base station which benefits from rich resources and its energy is not supplied by battery. IN Fig. 2, a sample of HWSN is presented. HWSNs discussed earlier should satisfy the following properties according to [13]:

(a) Sensor nodes (LS-nodes) are not equipped with Tamper-resistant hardware.
(b) Cluster heads (HS-nodes) are equipped with Tamper-resistant hardware.
(c) All LS-nodes and HS-nodes should have a unique ID.
(d) Based station is trusted in HWSNs.

It should be noted that one of the characteristics of HWSNs stated in [13] is that sensor nodes are stable and are aware of their location by using GPS. However, since our method can also be used in mobile networks, it doesn't need to satisfy this property. It may seem that the stage of selecting parent nodes by new nodes is not related to key establishment and is more related to the architecture of HWSNs, but in methods such as ours that supports mobility, once a node separates from its parent node, the process of selecting parent node and registering in network plus other key establishment stages should be repeated. Consequently, to culminate key establishment methods which support mobility in HWSNs, it is better to add this stage to them. Figure 2 shows an example of hierarchical wireless sensor networks.

3.2 Notation for Cryptography

Notations which are used for cryptography and will be used in the next sections of this article are explained in Table 1.

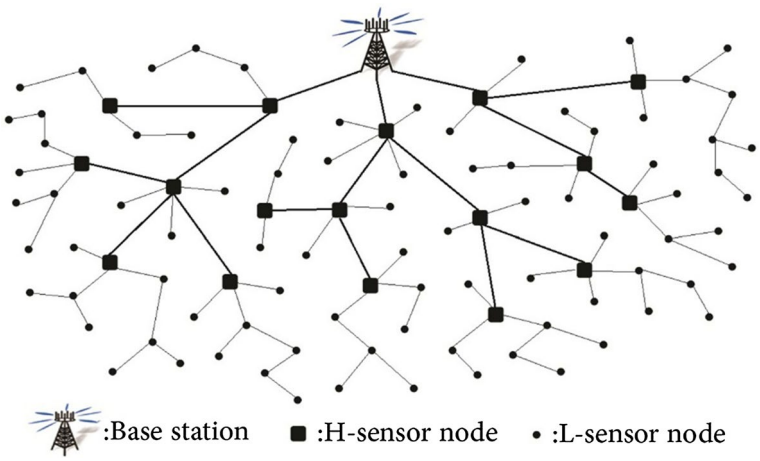

Fig. 2 An example of the hierarchical wireless sensor networks

4 Proposed Scheme

Our proposed method is based on symmetric cryptography, authenticates the new nodes and also supports mobility. We have divided our proposed method into 4 parts as follows.

4.1 Entrance of the New Nodes to Network and Selection of Parent Node

After locating in the area covered by sensor network, nodes must start to send a message called "parent find request (PFR)". Sensor node broadcast PFR message for several times during a fixed period of time, and the length of this period can be changed regarding the network circumstances. For example, if the density of nodes is high in the covered area, we can reduce the length of this period since in networks with more density, the number of neighboring nodes which receives the PFR message is more after sending PFR messages for each time. Then, sensor nodes can find a node as parent sooner. Neighboring nodes of the new node, both LS-nodes and HS-nodes, should answer PFR message upon receiving this message in such a way that a variable named "Hops to Base Station (HTB)" which is equal to the number of hops available between them and base station is sent to the new node in response. The new node saves all response messages received from neighboring nodes, and then prioritize them. The criterion for the new nodes to select parent node should be in such a way that cluster heads have the highest priority for LS-nodes, because if an LS-node connects directly to a cluster head, it doesn't waste energy of other LS-nodes which are intermediary between them. For new cluster heads, only the other cluster heads are possible to be selected as parent node since cluster heads can't connect to each other through LS-nodes. If the new LS-node receives response messages from more than one cluster head, or it doesn't receive a response from any cluster heads, it should select the parent node based on another criterion. In this case, each node with the smallest HTB variable should be selected as parent node and if the HTB variables of two or more neighboring nodes are equal, the node with the highest signal strength should be selected as parent node. In some related works like [7, 20], the only criterion for new nodes to select the parent node is signal strength which, unfortunately, severely increases the energy consumption resulted from communications during all network lifetime. Its negative results are not limited to key establishment process. Refer to Fig. 3, for more explanation. In Fig. 3, CH2 and CH4 are located in transmission rang of the new node. In this case, if the parent node is selected based on criterion presented in this article, CH2 should be selected as parent node because its HTB variable is equal to 2 and is less than HTB variable of CH4 which is equal to 3. However, if parent node is selected based on the criterion presented in [13, 26], Ch4 should be selected as the parent node as it is closer to the new node and its signal strength is higher. Consequently, more nodes are located in route from the new node to the base station. It means that more nodes should spend their precious energy to transfer data from the new node to the base station. Therefore, the absence of variable such as HTB and not using it as a criterion to select parent node, increases the energy consumption resulted from communication in network.

It should be noted that network nodes should go through this process again as well as the next three parts that will be explained in case of their or their parent's movement and being disconnected. Figure 3 shows how the new nodes select their parent nodes in the proposed method.

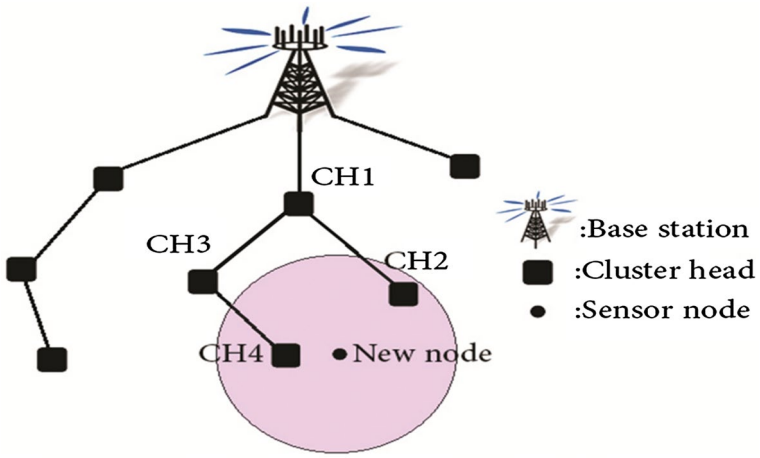

Fig. 3 How new nodes choose their parent nodes in proposed method

4.2 Sending the Registration Request to Competent Authority by New Node Through Its Parent Node

In our proposed method, base station is the competent authority to authenticate the requesting nodes. As it was mentioned in Sect. 1, because of specific characteristics of base station in HWSNs, using it as the competent authority doesn't make any concerns and will not be a performance-bottleneck like cluster heads which are battery-powered. But when overheads occur, The Base Station considered special node in the networks as the Competent Authority for key generation and authentication of new nodes, to control overheads and traffics are handled by the overloaded BS for a large network.

Moreover, we considered different key establishment mechanisms for HS-nodes and LS-nodes because of specific characteristics mentioned about HWSNs in [13] and used these characteristics for the benefit of the network interests. Due to being equipped with Tamper-resistant hardware, HS-nodes provide an advantage that makes it possible to use a key establishment mechanism with minimum memory consumption in these nodes. In our proposed method, we used a similar mechanism to Network-wide key mechanism for key establishment between cluster heads and base station. However, we don't use public shared key directly between cluster heads and base station in cryptographic operations in order to prevent birthday attack. Using a hash function and ID of the communicating nodes, we can generate a new key from the public shared key between HS-nodes and base station. For example, key required to establish an encrypted connection between two cluster nodes with ID_s of CH_1 and CH_2 can be generated as follows: $F(K_{HS}||CH_1||CH_2)$. Where K_{HS} is the public key shared between cluster heads and base station. Since K_{HS} is not directly used and hash functions have a one-way feature, we can be sure that intruders can't access to K_{HS}. As the unique ID of base station is considered zero in most of sensor networks, when a cluster head wants to send the registration request to base station, it should generate a new key as follows:

$$NEW_{KEY} = F(K_{HS}||id\ of\ CH\ node||0) \tag{3}$$

The new key is used to encrypt the message of membership request, and then membership request is sent to the base station. Membership Request message which is sent from an HS-node to the base station, should have the following items:

- Time Stamp or a Nonce to prevent replay attack.
- The unique ID of the cluster head selected as the parent node in the previous stage. If the new node is directly connected to the Base Station (its parent node is the Base Station), it must send the number zero.
- The above-mentioned message is now sent to base station.

$$\begin{aligned} K_T &= F\left(K_{HS}||id\ of\ CH\ node||0\right) \\ Mem-Req &= E_{KT}(Nonce\ or\ Time\ Stamp||Parent\ id) \end{aligned} \tag{4}$$

The conditions are different about LS-nodes because these nodes are not equipped with Tamper-resistant hardware, therefore, for LS-nodes it is not possible to employ a mechanism similar to network-wide key, which benefits a lot of advantages including increase of flexibility in network and reduction of memory consumption. Regarding the hardware conditions, LS-nodes should act in such a way that after each node is compromised, only the key of that node be enclosed to the intruders and the keys of other LS-nodes remain safe. Therefore, we can use a similar process to the one mentioned about cluster heads to generate the pre-deployment key which should be saved in memory of each LS-nodes and shared with competent authority. It means that we use a key, which is under the control of competent authority (base station), to generate LS-node's pre-deployment key in such a way that this key has a specific characteristics stated in [26]; it means that generated keys should be computable in higher-level nodes and saved in the memory of lower-level nodes. In another word, the nodes in higher levels can generate the required keys once they need them, and don't need to save them for establishing an encrypted communication with the nodes in lower levels; finally, this feature of the generated keys can reduce the memory consumption. As mentioned above, we can use a key like K_{LS} which is only under the control of competent authority (base station) for generating of the shared keys between base station and LS-nodes to enable the LS-nodes encrypt their membership request messages and send them to base station. In this stage, we can use the one-way feature of hash functions and generate a pre-deployment key for an LS-node with the ID of LS, and save it in its memory:

$$The\ pre{-}deployment\ key\ for\ LS = F(K_{LS}||LS) \tag{5}$$

Station and that LS-node will be compromised, but shared keys between the other LS-nodes and base station are still safe. Accordingly, it can be said that LS-nodes should first generate a message including the following two parts to send registration request to the base station.

- Time Stamp or Nonce to prevent replay attack.
- The unique ID of the node selected as the Parent node in the previous stage.
- This message is then encrypted by pre-deployment key of KT which is generated from K_{LS}.

$$\begin{cases} K_T = F\left(K_{LS}||L_S\right),\ K_T\ is\ generated\ by\ base\ station\ and\ stored\ in\ LS. \\ Mem-Req = E_{KT}\ (Nonce\ or\ Time\ Stamp||Praent\ id) \end{cases} \tag{6}$$

4.3 Authentication of Requesting Nodes by Competent Authority

As previously mentioned, in our proposed method, base station is the only competent authority that has shared keys with all network nodes including LS-nodes and HS-nodes, and it is responsible to authenticate the requesting nodes. Also, provides them with a key in order to encrypt and decrypt the data exchanged with their parent nodes. As it was explained in Sect. 4.2, different mechanisms are applied for HS-nodes and LS-nodes to establish initial shared key with base station. Therefore, upon receiving a request message from any node, base station should check, whether that node is HS-node or LS-node. If the message is sent by an LS node, therefore, base station should regenerate the shared key with that LS-node to decrypt this message:

$$F(K_{LS}||LSid) = K_T = Shared\ key\ between\ base\ station\ and\ LS - node \tag{7}$$

Then K_T is applied to decrypt the message of LS membership request:

$$D_{KT}(LSMem - Req) = (Nonce\ or\ Time\ Stamp||Parent\ id \tag{8}$$

After decrypting the message of LS registration request, base station can carry out the following process:

1. Authenticating LS: since requesting node, except base station, is the only node that has KT, its identity is authenticated.
2. Registering the parent node of the requesting LS-node: this ID is used for generating the new key for LS-nodes.

If the message of registration request is sent by an HS-node, base station should first generate the shared key between itself and that
HS-node:

$$KT = F(KHS||The\ id\ of\ HS{-}node||0) \tag{9}$$

Now, base station should decrypt the message of membership request sent by HS-node.

$$DKT(HSMem{-}Req) = (Nonce\ or\ Time\ Stamp||Parent\ id) \tag{10}$$

If the received message is decrypted by K_T, HS-node is authenticated.

4.4 Final Operation of Key Establishment

After authenticating the nodes, a new key should be generated to be shared between them and their parent nodes. As mentioned in Sect. 4.2, a mechanism similar to Network-wide key is used between HS-nodes and base station. Thus, no new key should be generated for HS-nodes and the process will be finished only by sending an acknowledgment message to new HS-nodes and their parent nodes. However, if the new node is an LS-node, a different measure should be taken. As previously mentioned, generated keys should have those characteristics mentioned in [26]; it means that they should be computable by parent nodes which are in a higher level, and saved in requesting nodes which are in a lower level. This characteristic can reduce memory consumption resulted from saving keys. If the new node is an LS-node, there are two possibilities: first, the requesting LS-node is directly and

without any intermediary LS-node connected to an HS-node; it means that its parent node is a cluster head. In this state, base station should first generate its shard key with that HS node and use it to generate the new key. Therefore, the new generated key should not be saved in HS-node since whenever the HS-node needs it, it can generate its shared key with the base station by using KHS, and then uses KHS in order to establish the shared key with its LS child node. Accordingly, base station should generate the new key for requesting LS-node with the id of LS which has selected a cluster head with id of HS as parent node as follows:

5 Security Analysis

The methods such as ours which establishes shared cryptographic keys in an authentic manner should be resilient against various security attacks. In this section, we show that our proposed method presents a high level of security. Note that proposed method is based on assumptions that base station is trusted and HS-nodes are equipped with tamper resistant hardware.

In the stage of sending registration request to competent authority for membership in the network, if intruders want to enter their own cluster heads to network, they should send an encrypted request using K_{HS}. In our method, the shared secret key between a cluster head with ID of HS and the base station is obtained accordingly "$F(KHS||HS||0)$". However, since only base station and other authenticated cluster heads of the network have K_{HS} and cluster heads are equipped with Tamper-resistant hardware, the intruders are not able to access to K_{HS}; as a result, they cannot enter the cluster head of their own into the network. The intruders, maybe, intend to obtain the keys applied in cryptographic processes between cluster heads and base station using birthday-attack to access the K_{HS}. However, K_{HS} is not directly applied in cryptographic operation. Also, because of benefiting from one-way feature of the hash functions, it is not possible for intruders to gain access to K_{HS}. If intruders want to enter an LS-sensor into the network, they should encrypt and send the message of membership request to base station using a shared key with the base station. Shared keys between base station and LS-nodes are obtained as follows: F ($KLS||IDoftheL-sensor$), but since only the base station has K_{LS}, there is no possibility that intruders can generate a shared key with base station. Even if intruders can access to an LS-node and the data saved in its memory such as its shared key with the base station, they are not able to gain access to K_{LS} because hash functions have a one-way feature. Also, there is no possibility to gain access to K_{LS} in case of having the shared key of an LS-node with base station. Moreover, by gaining access to LS-nodes and obtaining their shared keys with their parent nodes, the intruders will not be able to get the other keys of the parent nodes. For example, if the LS1 is the parent of LS_2, and the intruder has access to LS_2 and the shared key between LS_1 and LS_2 ($F(KLS1||LS2)$) they are not able to obtain K_{LS1} which is the shared key between LS_1 and base station.

In our proposed method, the level of key-connectivity is 100 percent between nodes which should have secure communications. It should be noted that there is no need to global-connectivity in $HWSN_s$ because the communications are hierarchical in these networks. Firstly, each node should have a shared key with competent authority to send registration request. Secondly, should have a shared key with its parent node and all its children nodes which are directly connected to it. If these conditions hold, all communications of the nodes can be secured; our proposed method observes these conditions because all

network nodes, either HS- nodes or LS-nodes, have a shared key with competent authority (base station) which is saved in their memory before joining the network; through this key, they can also generate their shared keys with all their children nodes which are connected to them directly. Moreover, once nodes are authenticated by base station, a shared key with their parent nodes is generated and sent for them.

5.1 Prevention of Attacks

In this section, we discuss how our proposed method is able to prevent some potential security attacks:

Replay attack In this attack, intruders try to resend a message which was previously sent by one of the network nodes such as membership request message to base station, and waste the bandwidth of the base station. Usually, we added a nonce or a time stamp to the messages to prevent this attack.

Sinkhole and selective forwarding attack In these attacks, intruders should first inject their desired nodes into network structure by using the attack of node injection; however, as it was mentioned at the beginning of Sect. 5, the intruders are neither able to inject a cluster head into network nor able to inject an LS- node into network because it is not possible that they can generate the shared key between them and base station to pass the authentication stage, and send the registration request in network.

Masquerade attack In this attack, the intruders try to impersonate their own nodes instead of authenticated nodes. This kind of attack is not feasible for HS-nodes since these nodes are equipped with tamper-resistance hardware, and don't allow the intruders to access to K_{HS} which is required for HS-nodes to pass the authentication step. However, in case of LS-nodes, intruders, by gaining access to one of them, can access to their shared key with base station and use it to authenticate their own node. However, they should access to one of authenticated LS- nodes in order to masquerade each LS-node of their own.

Message manipulation The intruders may intend to make some disorders in network performance by altering sent messages, but since our method supports authenticity, the intruder cannot carry out this attack as they need the key used in cryptographic operation to alter the exchanged messages without the receiving nodes figure it out.

Sybil attack In this case, the intruders introduce a compromised node to network with several IDs. It is not possible to perform this attack on HS-nodes because, based on explanations mentioned previously, they cannot be compromised. However, about LS-nodes, if the intruders can access to an LS-node, they can introduce it to the network only by ID of itself since the shared key of each LS-node with base station is generated using ID of that LS-node. Thus, if an LS-node is introduced to the network with another ID, it cannot be authenticated by base station.

6 Simulation Results

NS2 is employed for simulation of the proposed method. Base station, in center, and LS-nodes, randomly, are located in the area covered by the network. Also, HS-nodes are located in the grid form. Signal range of nodes is about 15 feet. In this simulation, our proposed method is compared with RAKE presented in [26]. We selected RAKE method because of the following reasons: firstly, RAKE method is presented for HWSNs with

features stated in [13]; secondly, this method is based on symmetric cryptography, and enables nodes to establish cryptographic keys in an authentic manner. Thirdly, based on [26], the stimulation results of RAKE compared with two other methods presented in [4, 13] have indicated that RAKE has a better performance. In Sect. 5, we analyzed the security of our method and showed that it presents a favorable level of security. However, since nodes face a lot of resource limitations in WSN_s, we compare our method with RAKE with regards to resource efficiency including memory and energy consumption. Memory consumption includes memory used for saving cryptographic keys, and Energy consumption can be divided into two parts: computations and communications. Energy consumption resulted from computations is so little compared to energy consumption resulted from communications. In addition, cryptographic algorithm and hash function used in our proposed method are the same as RAKE (cryptography algorithm is 128–bit AES, and hash function is 160–bit sha1). Thus, we made a comparison between our method and RAKE regarding energy consumption resulted from communication and memory consumption resulted from saving cryptographic keys.

6.1 Key Storage

Figure 4 provides memory consumption resulted from saving keys in LS-nodes. According to this figure, only two keys, in our proposed method, are required to be saved in LS-nodes. It should be noted that the number of keys saved in an LS-node is always equal to two. However, when the network scale is larger in RAKE method, memory consumption increases sharply in LS-nodes; it is considered as an obstacle for extension of network. Figure 5 compares RAKE with our proposed method regarding the number of keys saved in HS- nodes. As mentioned in Sect. 4.2, network-wide key is used for key establishment

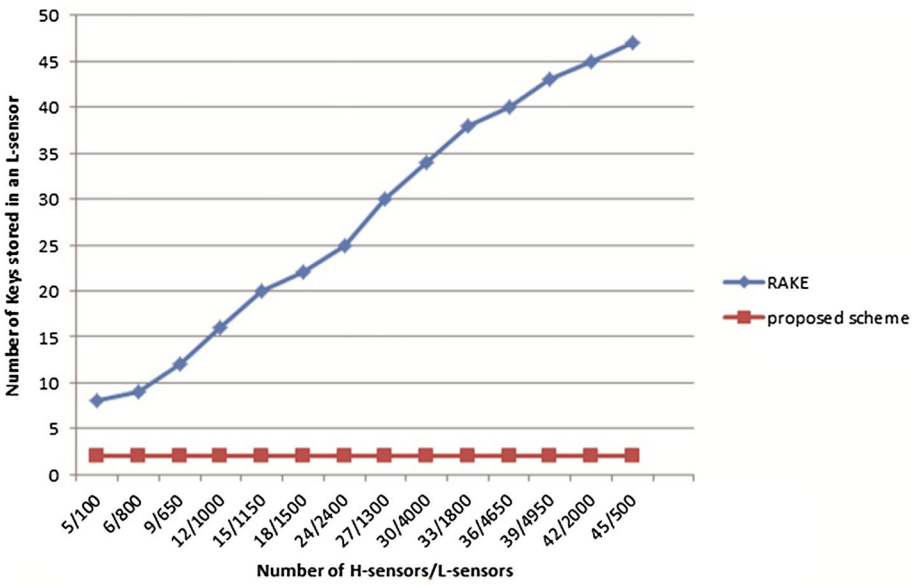

Fig. 4 Numbers of keys stored in an L-sensor with varying numbers of H-sensors and L-sensors

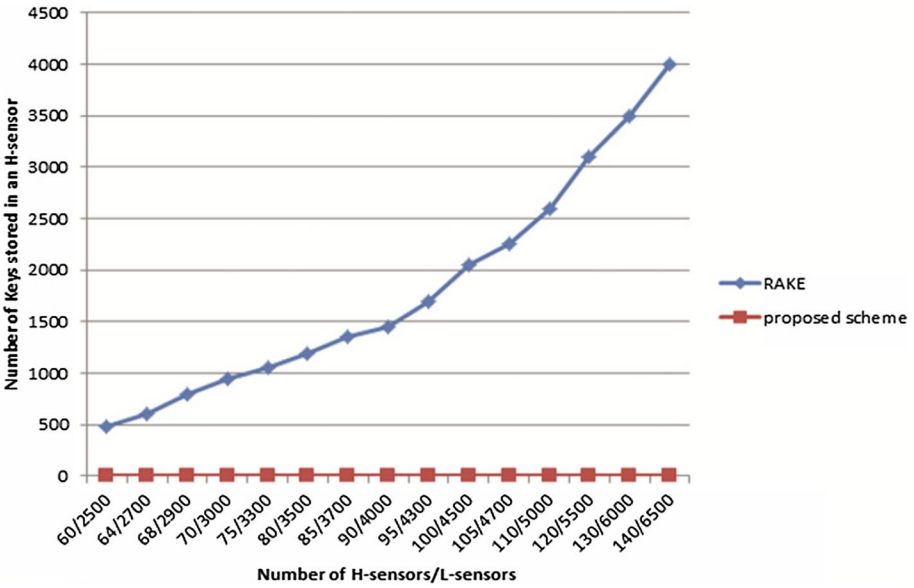

Fig. 5 Numbers of keys stored in an H-sensor with varying numbers of H-sensors and L-sensors

in proposed method, among HS-nodes and base station. Thus, the number of keys required in HS-nodes to communicate with other HS-nodes and base station is always equal to one, and it does not change by changing the network scale. On the other side, HS-nodes do not need to save their shared keys with their LS children as they can generate these shared keys once they need them. In RAKE, however, by increasing the network scale, the memory consumption in HS-nodes increases sharply. Figure 4 shows the number of keys stored in the L sensor with the number of different H sensors and L sensors. Figure 5 shows the number of keys stored in the H sensor with the number of different H sensors and L sensors.

6.2 Communication

It may seem that energy consumption resulted from communications in our method is much more than RAKE since in RAKE, LS- nodes send the request of membership to HS-nodes, but in our method, it is sent to based station. Therefore, more communications are required in our method during the key establishment process. Although more communications are required between HS-nodes in proposed method, two issues should be noted: first, HS-nodes have much more resources than LS-nodes, and by increasing energy consumption resulted from communications between HS-nodes, we could reduce the memory consumption in LS-nodes in such a way that only two keys are required to be saved in LS-nodes. Also, memory consumption in HS-nodes is reduced as well. Second, as mentioned in section 1.4, the related works such as RAKE and [13], use a method in the stage of selecting parent node for new nodes which results in more energy consumption in all nodes during the whole network lifetime than our method; since more nodes are located in the route from different nodes to base station. This energy consumption reduction is in such a way that although base station plays

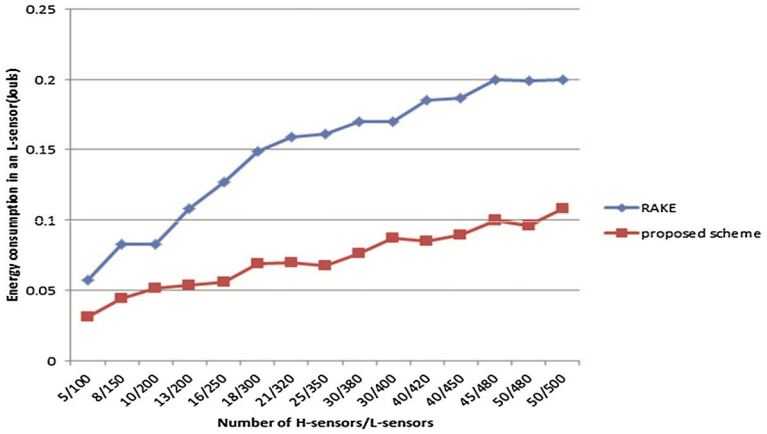

Fig. 6 Energy consumption resulted from communications in L-sensors

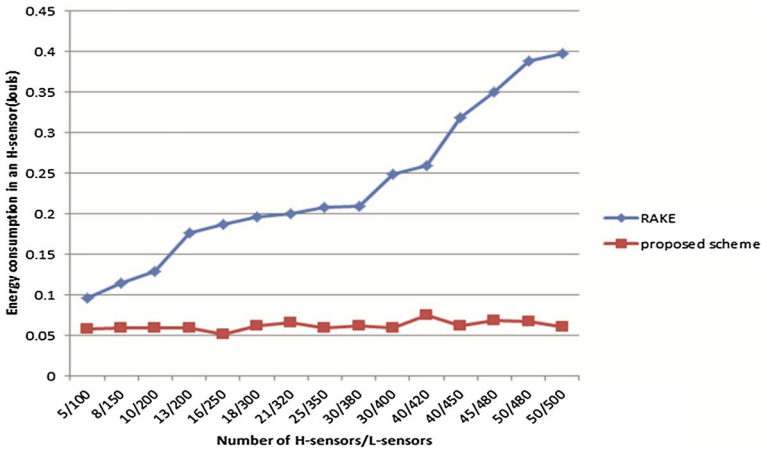

Fig. 7 Energy consumption resulted from communications in H-sensors

the role of KDC in proposed method, stimulation results have shown that the average of energy consumption resulted from communications in ours is less than RAKE, in both LS-nodes and HS-nodes. Figure 6 compares RAKE with our method regarding energy consumption resulted from communication in LS-nodes. Figure 7 shows the results of comparison between RAKE and our method in terms of energy consumptions resulted from communication in HS-nodes. Figure 6 shows the power consumption caused by the communication in the L sensors and Fig. 7 shows the power consumption caused by the communication in the H sensors.

7 Conclusions

In this article, we presented a mechanism for key establishment based on symmetric cryptography in hierarchical wireless sensor networks ($HWSN_s$); this mechanism can reduce the resource consumption significantly, and provides a favorable level of security. By

presenting appropriate solutions in proposed method, we could reduce memory consumption in such a way that by saving only two keys in LS-nodes and one key in HS-nodes, we could secure all wireless media in HWSN; communications can be vertical between nodes from various levels or horizontal between nodes available in the same level. Regarding solutions presented in this paper, energy consumption is reduced noticeably compared to related works, and stimulation results have proved this. We presented our proposed method based on symmetric cryptography since energy consumption resulted from computations is low in symmetric cryptography and its memory consumption can be even less than asymmetric cryptography by using appropriate techniques. In addition, the method presented in this article can be applied in HWSNs with any scales.

References

1. Duarte-Melo, E. J., & Liu, M. (2003). Data-gathering wireless sensor networks: Organization and capacity. *Computer Networks, 43*(4), 519–537. https://doi.org/10.1016/S1389-1286(03)00357-8.
2. Xu, K., Hong, X., & Gerla, M. (2002). An ad hoc network with mobile backbones. In *2002 IEEE international conference on communications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333)* (Vol. 5, pp. 3138–3143). IEEE. https://doi.org/10.1109/ICC.2002.997415.
3. Daemen, J., & Rijmen, V. (2002). *The Design of Rijndael, AES—The Advanced Encryption Standard* (p. 238). Springer.
4. Rifa-Pous, H., & Herrera-Joancomartí, J. (2011). Computational and energy costs of cryptographic algorithms on handheld devices. *Future internet, 3*(1), 31–48. https://doi.org/10.3390/fi3010031.
5. Mungara, R., VenkateswaraRao, K., & Pallamreddy, V. A. Routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks.
6. Yarvis, M., Kushalnagar, N., Singh, H., Rangarajan, A., Liu, Y., & Singh, S. (2005). Exploiting heterogeneity in sensor networks. In *IEEE infocom* (Vol. 2, p. 878). Institute of Electrical Engineers Inc (IEEE).
7. Girod, L., Stathopoulos, T., Ramanathan, N., Elson, J., Estrin, D., Osterweil, E., et al. (2004). A system for simulation, emulation, and deployment of heterogeneous sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 201–213). ACM. https://doi.org/10.1145/1031495.1031519.
8. Lai, B., Kim, S., & Verbauwhede, I. (2002). Scalable session key construction protocol for wireless sensor networks. In *Proceedings of the IEEE Workshop on Large Scale Real Time and Embedded Systems (LARTES)*, Washington, DC, USA, December 2002, p. 7.
9. Nemec, L., Matyas, V., Ostadal, R., Svenda, P., & Palant, P. L. (2019). Evaluating dynamic approaches to key (Re-) Establishment in wireless sensor networks. *Sensors, 19*(4), 914.
10. Chan, H., Perrig, A., & Song, D. (2003). Random key predistribution schemes for sensor networks. In *Proceedings of the 2003 Symposium on Security and Privacy*, Berkeley, CA, USA, 11–14 May 2003, pp. 197–213.
11. Álvarez Bermejo, J. A., Lodroman, A., & López-Ramos, J. A. (2016). Distributed key agreement for group communications based on elliptic curves. An application to sensor networks. *Mathematical Methods in the Applied Sciences, 39,* 4797–4809.
12. Parrilla, L., Castillo, E., López-Ramos, J. A., Álvarez Bermejo, J. A., García, A., & Morales, D. P. (2018). Unified compact ECC-AES co-processor with group-key support for IoT devices in wireless sensor networks. *Sensors, 18,* 251.
13. Rahman, S. M. M., & El-Khatib, K. (2010). Private Key agreement and secure communication for heterogeneous sensor networks. *Journal of Parallel and Distributed Computing, 70*(8), 858–870. https://doi.org/10.1016/j.jpdc.2010.03.009.
14. Azarderskhsh, R., & Reyhani-Masoleh, A. (2011). Secure clustering and symmetric key establishment in heterogeneous wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking, 2011*(1), 893592. https://doi.org/10.1155/2011/893592.
15. Boujelben, M., Youssef, H., Mzid, R., & Abid, M. (2011). IKM—An identity based key management scheme for heterogeneous sensor networks. *JCM, 6*(2), 185–197.
16. Traynor, P., Kumar, R., Choi, H., Cao, G., Zhu, S., & La Porta, T. (2007). Efficient hybrid security mechanisms for heterogeneous sensor networks. *IEEE Transactions on Mobile Computing, 6*(6), 663–677. https://doi.org/10.1109/TMC.2007.1020.

17. Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation, 48*(177), 203–209. https://doi.org/10.1090/S0025-5718-1987-0866109-5.
18. Kaur, A. (2013). Energy analysis of wireless sensor networks using rsa and ecc encryption method. *International Journal of Scientific & Engineering Research, 4*(5), 2212.
19. Landstra, T., Zawodniok, M., & Jagannathan, S. (2007). Energy-efficient hybrid key management protocol for wireless sensor networks. In *32nd IEEE conference on local computer networks (LCN 2007)* (pp. 1009–1016). IEEE. https://doi.org/10.1109/LCN.2007.135.
20. Zhang, X., He, J., & Wei, Q. (2011). EDDK: Energy-efficient distributed deterministic key management for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking, 2011*(1), 765143. https://doi.org/10.1155/2011/765143.
21. Sahingoz, O. K. (2013). Large scale wireless sensor networks with multi-level dynamic key management scheme. *Journal of Systems Architecture, 59*(9), 801–807. https://doi.org/10.1016/j.sysarc.2013.05.022.
22. Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., & Culler, D. E. (2002). SPINS: Security protocols for sensor networks. *Wireless Networks, 8*(5), 521–534. https://doi.org/10.1023/A:1016598314198.
23. Rasheed, A., & Mahapatra, R. (2010). Key predistribution schemes for establishing pairwise keys with a mobile sink in sensor networks. *IEEE Transactions on Parallel and Distributed Systems, 22*(1), 176–184. https://doi.org/10.1109/TPDS.2010.57.
24. Zhang, J., & Varadharajan, V. (2010). Wireless sensor network key management survey and taxonomy. *Journal of Network and Computer Applications, 33*(2), 63–75. https://doi.org/10.1016/j.jnca.2009.10.001.
25. Fanian, A., Berenjkoub, M., Saidi, H., & Gulliver, T. A. (2010). A new key establishment protocol for limited resource wireless sensor networks. In *2010 8th annual communication networks and services research conference* (pp. 138–145). IEEE. https://doi.org/10.1109/CNSR.2010.43.
26. Shi, Q., Zhang, N., Merabti, M., & Kifayat, K. (2013). Resource-efficient authentic key establishment in heterogeneous wireless sensor networks. *Journal of Parallel and Distributed Computing, 73*(2), 235–249. https://doi.org/10.1016/j.jpdc.2012.10.004.
27. Joan, D., & Vincent, R. (2002). The design of Rijndael: AES-the advanced encryption standard. In *Information Security and Cryptography*. Springer.
28. Singh, R., Misra, R., & Kumar, V. (2013). Analysis the impact of symmetric cryptographic algorithms on power consumption for various data types. *International Journal on Recent and Innovation Trends in Computing and Communication, 1*(4), 321–326.
29. Qian, S. (2012). A novel key pre-distribution for wireless sensor networks. *Physics Procedia, 25,* 2183–2189. https://doi.org/10.1016/j.phpro.2012.03.368.
30. Bechkit, W., Challal, Y., & Bouabdallah, A. (2013). A new class of Hash-Chain based key pre-distribution schemes for WSN. *Computer Communications, 36*(3), 243–255. https://doi.org/10.1016/j.comcom.2012.09.015.
31. Liu, D., Ning, P., & Li, R. (2005). Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security (TISSEC), 8*(1), 41–77. https://doi.org/10.1145/1053283.1053287.
32. Fanian, A., Berenjkoub, M., Saidi, H., & Gulliver, T. A. (2010). A hybrid key establishment protocol for large scale wireless sensor networks. In *2010 IEEE Wireless Communication and Networking Conference* (pp. 1–6). IEEE. https://doi.org/10.1109/WCNC.2010.5506121.
33. Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J., & Khalili, A. (2005). A pairwise key predistribution scheme for wireless sensor networks. *ACM Transactions on Information and System Security (TISSEC), 8*(2), 228–258. https://doi.org/10.1145/1065545.1065548.
34. Zhang, Y., Xu, L., Xiang, Y., & Huang, X. (2013). A matrix-based pairwise key establishment scheme for wireless mesh networks using pre deployment knowledge. *IEEE Transactions on Emerging Topics in Computing, 1*(2), 331–340. https://doi.org/10.1109/TETC.2013.2287196.
35. Sarmad, U. K., Lavagno, L., & Pastrone, C. (2010). A key management scheme supporting node mobility in heterogeneous sensor networks. In *2010 6th International Conference on Emerging Technologies (ICET)* (pp. 364–369). IEEE. https://doi.org/10.1109/ICET.2010.5638458.
36. Moharrum, M., Eltoweissy, M., & Mukkamala, R. (2006). Dynamic combinatorial key management scheme for sensor networks. *Wireless Communications and Mobile Computing, 6*(7), 1017–1035. https://doi.org/10.1002/wcm.435.
37. Lai, B., Kim, S., & Verbauwhede, I. (2002). Scalable session key construction protocol for wireless sensor networks. In *IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES)* (Vol. 7).

38. Zhu, S., Setia, S., & Jajodia, S. (2006). LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Transactions on Sensor Networks (TOSN), 2*(4), 500–528. https://doi.org/10.1145/1218556.1218559.
39. Zeng, Y., Zhao, B., Su, J., Yan, X., & Shao, Z. (2007). A loop-based key management scheme for wireless sensor networks. In *International conference on embedded and ubiquitous computing* (pp. 103–114). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-77090-9_10.
40. Dutertre, B., Cheung, S., & Levy, J. (2004). *Lightweight key management in wireless sensor networks by leveraging initial trust* (pp. 1–18). Technical Report SRI-SDL-04-02, SRI International.
41. Chan, H., Gligor, V. D., Perrig, A., & Muralidharan, G. (2005). On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Transactions on Dependable and Secure Computing, 3,* 233–247. https://doi.org/10.1109/TDSC.2005.37.
42. Mungara, R., VenkateswaraRao, K., & Pallamreddy, V. A. (2009). Routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. *IEEE Transactions on Wireless Communications, 8*(3), 371–383.

Hamid Mirvaziri has a B.Sc. from Shahid Bahonar University of Kerman, M.Sc. from Guilan University and Ph.D. from National University of Malaysia. His research area is in the field of wireless networks and computer and network security. Currently he is Assistant Professor of Computer Engineering Department Shahid Bahonar University of Kerman, Iran.

Rahim Hosseini has a B.Sc. from payam-noor University of Gachsaran, M.Sc. from Shahid Bahonar University of Kerman. His research area is in the field of wireless networks and data mining. Currently he is working on wireless sensor networks.