Wireless Pers Commun (2016) 87:349–367
DOI 10.1007/s11277-015-2872-6

Mobility Based Key Management Security Scheme for Wireless Sensor Networks

T. Lalitha[1] · S. Jayaprabha[2]

Published online: 8 July 2015

Abstract In wireless sensor networks (WSNs) handling mobility of nodes in key management is a challenging task. In this paper, a mobility management technique for keying scheme of WSNs is presented. The technique selects nodes with high energy resources, wide communication range and high processing capacity as cluster heads. Cluster keys for cluster heads and pairwise keys for nodes are generated by the sink through exclusion basis systems. Whenever a node moves from currently connected cluster to another in the network, the mobility based key management scheme is triggered. The sink verifies the authenticity of roaming node and allocates it to a nearby cluster. New pairwise keys are generated and transmitted to the roaming node through newly connected cluster head. Further, a key organization technique is presented to ensure the forward and backward secrecy of nodes. The proposed technique is simulated in NS2 and simulation results show the efficiency of our technique.

Keywords Mobility · Wireless sensor network · Cluster · Key management · Authentication

1 Introduction

1.1 Wireless Sensor Network

A network comprising of several minute wireless sensor nodes which are organized in a dense manner is called as a wireless sensor network (WSN). Every node estimates the state of its surroundings in this network. The estimated results are then converted into the signal

✉ T. Lalitha
lalithasrilekha31@gmail.com

[1] Sona College of Technology, Salem, Tamil Nadu 636005, India

[2] Jayam College of Engineering and Technology, Dharmapuri, Tamil Nadu 636813, India

form in order to determine the features related to this technique after the processing of the signals.

Based on the multi hop technique, the entire data that is accumulated is directed towards the special nodes which are considered as the sink nodes or the base station (BS). The user at the destination receives the data through the internet or the satellite via gateway. The use of the gateway is not very necessary as it is reliant on the distance between the user at the destination and the network [1].

For supervising the physical world, the WSNs are the promising technology. In order to collect the data from the surrounding in a sensor network application, several minute sensor nodes are organized and collaborated. Sensing modals like image sensors are placed in every node and this possess the ability to communicate in the wireless environment [2]. Military sensing and tracking, environment monitoring, patient monitoring and tracking are the fields where the sensor networks are utilized. Several low power sensors are distributed across the location that is to be monitored in the sensor network.

1.2 Problem and Overview

We have proposed a cluster based key management technique for authentication in WSNs. Initially, clusters are formed in the network and the cluster heads are selected based on the energy cost, coverage and processing capacity. The sink assigns cluster key to every cluster and an exclusion basis systems (EBS) key set to every cluster head. The EBS key set contains the pair wise keys for intra-cluster and inter-cluster communication. The cluster head upon detecting a compromised node in its cluster sends a request to sink to perform re-keying operation [3]. The re-keying process utilizes the hashing function for authentication and nodes are recovered in a secured manner. During data transmission towards the sink, the data is made to pass through two phases of encryption thus ensuring security in the network.

However, the cluster based key management technique [4] has not considered mobility of nodes. The secure key management scheme did not assure forward and backward secrecy. Furthermore, key life time and node computation capability are some other issues that were not discussed in existing system. In order to address the above described issues; in this paper we intend to presents a key management system that considered these issues.

In this paper, a mobility management technique for keying scheme of WSNs is presented. Initially, after the deployment of nodes in the network, nodes with high energy resources, wide communication range and high processing capacity as cluster heads. Cluster keys for cluster heads and pairwise keys for nodes are generated by the sink through EBS [3]. Whenever a node moves from currently connected cluster to another in the network, the mobility based key management scheme is triggered. The roaming node transmits a request message to the sink. The sink verifies the authenticity of roaming node and allocates it to a nearby cluster. New pairwise keys are generated and transmitted to the roaming node through newly connected cluster head. Further, a key organization technique is presented to ensure the forward and backward secrecy of nodes. According to this scheme, the sink and every cluster head maintains a key table (K-Table), where keys of every node are maintained along with lifetime of keys. Thus, keys are changed periodically to add up more security in the network.

1.3 Attacks in Sensor Networks

Low energy adaptive clustering hierarchy (LEACH) and secLEACH is the first and most popular energy-efficient hierarchical clustering algorithm for WSNs reducing power consumption. In LEACH, the clustering task is rotated among the nodes, based on duration. Direct communication is used by each cluster head (CH) to forward the data to the BS. LEACH is completely distributed and requires no global knowledge of network. Also, LEACH clustering terminates in a finite number of iterations, but does not guarantee good CH distribution and assumes uniform energy consumption for CHs.

2 Related Work

Jeong and Lee [5] have proposed a new cryptographic key management protocol, which is based on the clustering scheme but does not depend on the probabilistic key. The protocol can increase the efficiency to manage keys since, before distributing the keys by bootstrap, the use of public keys shared among nodes can eliminate the processes to send or to receive keys among the sensors. Also, to find any compromised nodes safely on the network, it solves safety problems by applying the functions of a lightweight attack-detection mechanism.

Dwoskin et al. [6] have proposed two low-cost secure-architecture-based techniques to improve the security against such node fabrication attacks. Their new architectures, specifically targeted at the sensor-node platform, protect long-term keys using a root of trust embedded in the hardware System-on-a-Chip (SoC). This prevents an adversary from extracting these protected long-term keys from a captured node to fabricate new nodes.

Jain and Jain [7] have presented a security framework Wireless Sensor Networks Security Framework (WSNSF) to provide a comprehensive security solution against the known attacks in sensor networks. The proposed framework consists of four interacting components: A Secure Triple-Key (STKS) scheme, secure routing algorithms (SRAs), a secure localization technique (SLT) and a malicious node detection mechanism. Singly, each of these components can achieve certain level of security. However, when deployed as a framework, a high degree of security is achievable. WSNSF takes into consideration the communication and computation limitations of sensor networks.

Maala et al. [8] have presented a Two Level Architecture key management scheme for wireless sensor networks (TLA). Our scheme combines efficiently different key management techniques in each architecture level. This combination gives TLA good performances in terms of key storage overhead as well as in terms of resistance degree against node capture.

Shen and Shi [9] in this study have presented a lightweight key management approach. A dynamic key management protocol is proposed to satisfactorily resolve the key distribution issues of WSN. The protocol assumes that the wireless sensor system has already been equipped with effective security detection mechanisms, which can decide if a sensor node is compromised or has used up its energy. Its analysis shows that this approach is an effective solution to the key management of hierarchical clustered WSNs. This protocol assumes that each sensor node is able to get its location information, which is currently a major restriction to its application.

Kim et al. [10] in this study proposed a key distribution scheme which improves the resilience against node capture and reduces communication cost. This key establishment model is devised comparing the benefits and weaknesses of the EG scheme and LEAP. As a result, this scheme inherits the security of the EG scheme during key setup phase and the improved security of LEAP after that phase. Also, this scheme does not require the assumption in LEAP that no nodes are captured during that phase, meaning this scheme is more practical than LEAP. In addition, this scheme has low communication overhead.

Shaikh et al. [11] have proposed two new identity, route and location privacy algorithms and data privacy mechanism that addresses the privacy problem. The proposed solutions provide additional trustworthiness and reliability at modest cost of memory and energy. Also, they proved that their proposed solutions provide protection against various privacy disclosure attacks, such as eavesdropping and hop-by-hop trace back attacks.

Abuhelaleh and Elleithy [3] have proposed a special kind of architecture to the cluster hierarchy of WSNs. The most interesting protocol that has been proposed for this kind of architecture is LEACH. This proposal is a module of a complete solution that is developed to cover all the aspects of WSNs communication which is labeled Secure Object Oriented Architecture for Wireless Sensor Networks (SOOAWSN).

3 Energy Efficient Cluster Based Key Management Technique

3.1 Cluster Based Key Management Technique

3.1.1 EBS Construction

An EBS consists of several subsets of the member set collection. In the EBS, every subset is analogous to a particular key and the nodes which possess the key become the element of the subset.

The dimension of the EBS is represented by (N, K, M) and it depicts a condition of a N membered secure group with numbering from 1 to N and a separate key is maintained for every subset by the key server. In EBS, if there exists a subset A_i, then every member of this subset will have knowledge about the key K_i. In EMS, there are M elements for every t € [1, N] and its union is equal to [1, N] − {t}. Hence, any member t can be ejected by the key server. Then re-keying is performed to enable every member to know the replacement keys for the K keys. To perform this, the M messages are multicast after encrypting them with the keys which correspond to the M elements, which has a union equal to [1, N] − {t}. To restrict decipherability to selected members, encryption of every key is performed by its predecessor.

A canonical enumeration technique is made use of, for the construction of EBS subsets. In the formation of subset of K objects out of K + M object set, every feasible method is taken into consideration.

Matrix A is formed in order to develop a bit string sequence in its canonical (K, M), in which the K and M are already known, C (K + M, K) columns indicate the successive bit strings of which has a length of K + M objects, where K ones are present in each. For EBS (N, K, M), "A" is known as the canonical matrix.

For instance, the canonical matrix A for EBS(8, 3, 2) enclose the enumeration of all C(5, 3) ways to form a subset of 3 keys from 5 keys, as shown in Table 1.

Every row in the table corresponds to a subset Ti after the construction of the matrix A, where an entry 1 in the row indicates that the corresponding node is present in the subset.

Table 1 Enumeration matrix for EBS(8,3,2)

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
T1	0	0	0	0	1	1	1	1	1	1
T2	0	1	1	1	0	0	0	1	1	1
T3	1	0	1	1	0	1	1	0	0	1
T4	1	1	0	1	1	0	1	0	1	0
T5	1	1	1	0	1	1	0	1	0	0

Since N = 8, M9 and M10 are not useful, in Table 1, T1 = [5, 6, 7, 8], T2 = [2, 3, 4, 8], T3 = [1, 3, 4, 6, 7], T4 = [1, 2, 4, 5, 7], and T5 = [1, 2, 3, 5, 6, 8]. It is easy to prove:

$$[1,8]-[1] = \mathrm{T1\,U\,T2},$$
$$[1,8]-[2] = \mathrm{T1\,U\,T3},$$
$$[1,8]-[3] = \mathrm{T1\,U\,T4}$$

Hence, on the exit of any node in the network information about the keys will be updated only by two node subsets. In this protocol, only five management keys are necessary whereas 15 keys are necessary in case of LKH. This in turn minimizes the key computation and also saves space for storage.

During the construction of the EBS (N, K, M) model in this protocol, the values of the parameters N, K and M are raised in order to facilitate the production of larger number of management keys. Later on, the spare keys are used for the new nodes of the cluster.

3.2 Cluster Formation and Communication

In the WSN, after the nodes are deployed in the physical environment, they first report to the base station their physical locations and then the network starts to select cluster heads.

According to the cluster head selection algorithm [4], each node decides if it is capable of serving as a cluster head based on the following selection criteria:

- High energy resources
- Wide communication range
- High processing capacity

For the authentication process, the encryption mechanism is carried on.

When the selection criteria are satisfied by a particular node, it is capable of being the cluster head. So, this node, N_i broadcasts a Cluster head beacon (CH_BEACON) packet. The CH_BEACON packet is encrypted with a key called as the primary key, K_{pri}:

$$N_i \xrightarrow{K_{pri}(CH\_BEACON)} \text{broadcast}$$

When the neighboring nodes S_i receive this message, a cluster head reply (CH_REPLY) message is sent to the node, N_i by the nodes which intend to join the cluster. The reply message sent fron the sink that contains the ID and the response content Ack:

$$N_i \xrightarrow[K_{pri}(ID\{S_i\}||Ack)]{CH\_REPLY} S_i$$

If the number of reply messages received by N_i is greater than a threshold R_{th}, then N_i can be selected as the cluster head, CH.

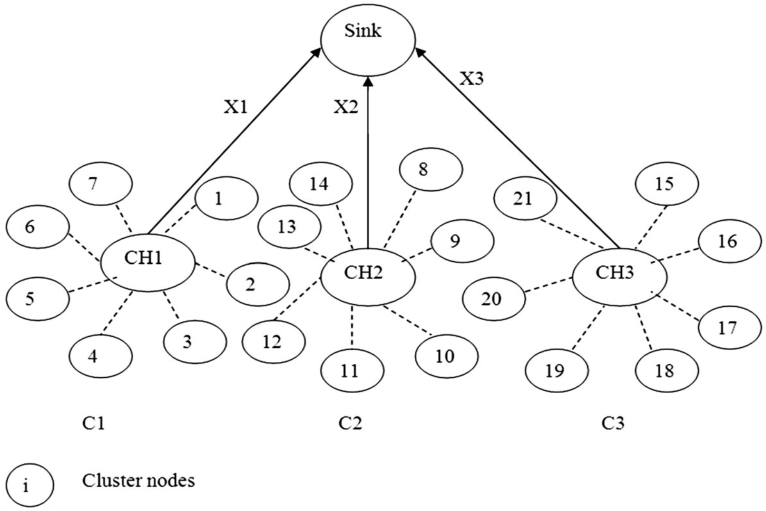


Fig. 1 Clustering architecture

Finally, the cluster head assigns IDs to all its member nodes that intend to join the cluster.

Figure 1 shows the architecture of the clustering system with every CH connected to the sink. In this figure, the network possesses three clusters. Each cluster possess a cluster head i.e., CH1, CH2 and CH3 are the cluster heads of clusters C1, C2 and C3, respectively. CH1 contains the members 1–7, CH2 contains members 8–14 and CH3 contains members 15–21.

After the clusters are formed in the network, the CH sends the information of its members like $\langle \text{cluster id, member id} \rangle$ to the sink.

X1, X2 and X3 are the cluster information sent by CH1, CH2 and CH3 towards the sink, given by:

$$\begin{aligned} X1 &= \{\langle C1, 1\rangle, \langle C1, 2\rangle, \ldots, \langle C1, 7\rangle\} \\ X2 &= \{\langle C2, 8\rangle, \langle C2, 9\rangle, \ldots, \langle C2, 14\rangle\} \\ X3 &= \{\langle C3, 15\rangle, \langle C3, 16\rangle, \ldots, \langle C3, 21\rangle\} \end{aligned}$$

The sink allots a cluster key, K_{CH} to every cluster in the network. In Fig. 2, the cluster keys obtained by the cluster heads CH1, CH2 and CH3 are K_{CH1}, K_{CH2} and K_{CH3}, respectively.

After getting the cluster key from the sink, each CH receives the pairwise key set which is based on EBS [9].

$$\text{Sink} \xrightarrow{K_{CHi}\{\text{EBS key set}\}} CH_i \quad \text{where } i = 1, 2, 3$$

The EBS key set includes the pairwise keys, P_{ij} for communication between the CH and its member and also the pairwise keys, $PH_{ii'}$ for communication between the CHs, encrypted by the cluster key. Hence EBS key set transmission can also be given as:

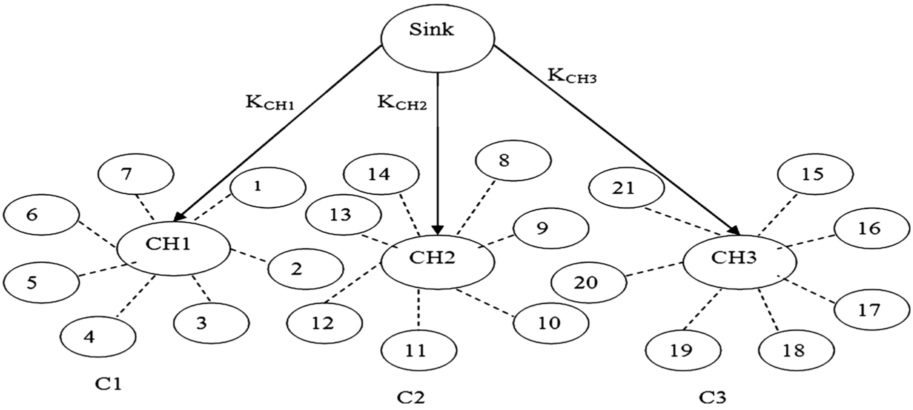


Fig. 2 Cluster key transmission from the sink to the cluster head

$$\mathrm{Sink} \xrightarrow{K_{CHi}\{P_{ij}||PH_{ii'}\}} CH_i \quad \text{where } i = 1, 2, 3$$

3.2.1 Intra Cluster Communication

The CH decrypts the pairwise keys sent by the sink, with its cluster key K_{CH} and distributes them to its cluster members:

$$CH_i \rightarrow \{CM\}_j$$

where $i = 1 \rightarrow j = 1$ to 7, $i = 2 \rightarrow j = 8$ to 14, $i = 3 \rightarrow j = 15$ to 21, where, CM are the cluster members.

After the pairwise keys are distributed by the CH to its members, for the establishment of the secure channels between the CH and the cluster members, the CH sends a hello message to the cluster members. Based on the reception of the Acknowledgement message from its members, the CH establishes a channel between itself and its members:

$$CH_i \xrightarrow{\text{Hello message}} \{CM\}_j$$

$$CH_i \xrightarrow{\text{Ack message}} \{CM\}_j$$

$$CH_i \xrightarrow{\text{Secure channel}} \{CM\}_j$$

where $i = 1 \rightarrow j = 1$ to 7, $i = 2 \rightarrow j = 8$ to 14, $i = 3 \rightarrow j = 15$ to 21.

For example, in Fig. 2, if node1 of C1 wants to communicate with node5 of the same cluster, then CH1 distributes a pairwise key to node 1 and node 5:

$$CH_1 \xrightarrow{K_{11}} \{CM\}_1$$

$$CH_1 \xrightarrow{K_{15}} \{CM\}_5$$

Next a secure path is established between the two nodes; node 1 and node 5 after the exchange of hello message and acknowledgement message:

$$CH_1 \xrightarrow{\text{Hello message}} \{CM\}_1$$

$$CH_1 \xrightarrow{\text{Hello message}} \{CM\}_5$$

$$CH_1 \xleftarrow{\text{Ack message}} \{CM\}_1$$

$$CH_1 \xleftarrow{\text{Ack message}} \{CM\}_5$$

After receiving the acknowledgement message, a secure channel is set up between the node and the CH. Thus through the CH, a continuous path is established between the two nodes that need to communicate with each other:

$$CH_1 \xleftrightarrow{\text{Secure channel}} \{CM\}_1$$

$$CH_1 \xleftrightarrow{\text{Secure channel}} \{CM\}_5$$

$$\{CM\}_1 \xleftrightarrow{\text{Secure channel}} \{CM\}_5$$

This technique allows secure communication between intra cluster nodes as well as inters cluster nodes.

3.2.2 Inter Cluster Communication

Whenever a node within a cluster wants to communicate with a node belonging to another cluster then the inter cluster communication takes place in the network. For communication between two clusters, the CH uses the pairwise keys, $PH_{ii'}$ obtained from the EBS key set:

$$CH_i \xrightarrow{H_{ii'}} CH_i$$

where, $i = 1, 2, 3$; $i' = 1, 2, 3$ and $i \neq i'$

After the distribution of the pairwise keys between the CHs, the secure channels are established between the CHs. Initially the source CH sends a hello message to the CH with which the former wants to communicate. On reception of the Acknowledgement message from the target CH, the source CH establishes a channel between itself and the target CH:

$$CH_i \xrightarrow{\text{Hello message}} CH_i$$

where, $i = 1, 2, 3$; $i' = 1, 2, 3$ and $i \neq i$:

$$CH_i \xleftarrow{\text{Ack message}} CH_i$$

where, $i = 1, 2, 3$; $i' = 1, 2, 3$ and $i \neq i$:

$$CH_i \longleftrightarrow CH_i$$

where, $i = 1, 2, 3$; $i' = 1, 2, 3$ and $i \neq i$.

For example, in Fig. 2, if node 10 of C2 wants to communicate with node 15 of C3, then the following sequence of steps will take place. Initially the CH2 distributes the pairwise key K_{210} to the node10 and CH3 distributes the pairwise key K_{315} to node 15 and then a secure channel is established in C2 between CH2 and node10 and in C3 between CH3 and node15.

In order to establish a secure channel between C2 and C3, the following steps are followed:

$$\mathrm{CH}_2 \xrightarrow{\mathrm{K}_{23}} \mathrm{CH}_3$$

Next the hello message is sent by C2 to C3:

$$\mathrm{CH}_2 \xleftarrow{\text{Ack message}} \mathrm{CH}_3$$

$$\mathrm{CH}_2 \xleftarrow{\text{Ack message}} \mathrm{CH}_3$$

On receiving the acknowledgement message, a secure channel is established between the C2 and C3:

$$\mathrm{CH}_2 \xleftrightarrow{\text{Secure channel}} \mathrm{CH}_3$$

Then through CH2 and CH3, the node10 of C2 and node15 of C3 are connected to each other to form a secure path:

$$\{\mathrm{CM}\}_{10} \xleftrightarrow{\text{Secure channel}} \{\mathrm{CM}\}_{15}$$

3.2.3 Data Transmission to the Sink

When a sensor node wants to transfer its data securely to the sink, the data transmission takes place in two phases.

In the first phase, the data packets to be transmitted are encrypted with the pair wise key by the member node and then transmitted to the corresponding CH. On reaching the CH, the data packet is decrypted by the CH and original data is retrieved.

In the second phase, the data packet is encrypted with the cluster key by the CH and then transmitted to the sink. At the sink, the data packet is decrypted with the cluster key and the original data is retrieved.

In Fig. 2, if node 11 of C2 wants to transmit the data to the sink, then the following steps are carried out.

Phase 1: The data, D at node 11 is encrypted by the pairwise key, K_{211} and then transmitted to CH2.

$$\{\mathrm{CM}\}_{11} \xrightarrow{\mathrm{K}_{211}\{\mathrm{D}\}} \mathrm{CH}_2$$

Phase 2: At the CH_2, the data is decrypted using the pairwise key. Then CH_2 encrypts the data with the cluster key, $\mathrm{K}_{\mathrm{CH2}}$ and transmits it to the sink.

$$\mathrm{CH}_2 \xrightarrow{\mathrm{K}_{\mathrm{CH2}}\{\mathrm{D}\}} \text{sink}$$

At the sink the data packet is decrypted and the original data is retrieved by the sink.

3.3 Authentication Technique

When the cluster head (CH) has found that one of the members in its cluster is compromised or captured, it requests the sink to implement the re-keying operation. The steps involved in the re-keying process are as follows.

1. CH retrieves the ID of the node (say v) requiring re-keying.
2. CH XORs the ID, its own secret K_{CH}, the request for re-keying message G_{req} and the time T to obtain the message G. It is represented as as $(ID \otimes K_{CH} \otimes G_{req} \otimes T)$
3. CH computes hash value of the XORed data represented as d.

$$\text{i.e. } d = H(ID \otimes K_{CH} \otimes G_{req} \otimes T)$$

4. CH sends the computed hash value d, node ID, G_{req} and T to the sink.
5. The sink retrieves K_{CH} and re-computes the hash value which is given as f. The computed hash value is compared with d.
6. Upon comparison, if it is found that d is equivalent to f, and the time T is not utilized in a re-keying process previously, then the sink performs the subsequent steps.
7. The sink XORs the pairwise key P_{ij}, the re-keying message G_{req} and time T to obtain G. i.e. $(P_{ij} \otimes G_{req} \otimes T)$
8. The sink then computes the hash value of P_{ij}, G_{req} and T. i.e.$H(P_{ij} \otimes G_{req} \otimes T) = p$.
9. The sink sends p, G_{req} and T to CH.
10. The CH re-computes hash value of XOR of v with P_{ij}, G_{req} and T which is represented as b and compares it with p. Upon comparison, if it is found that $b = p$ and the time T has not been utilized in a re-keying process previously, the re-key P_{ij} is sent to the affected node and it is re-keyed.
11. The node updates its secret to $P_{ij} = G$ and informs the sink that it has successfully re-keyed and the sink then updates the node's secret in its table.

For example consider Fig. 3. In this, the node 7 in CH1 is compromised and requires re-keying.

CH1 computes H $(ID7 \otimes K_{CH1} \otimes G_{req} \otimes T)$ which is represented as d7. It then sends d7, ID7, G_{req} and T to the sink. The sink retrieves the K_{CH1} and re-computes the hash value given as f7. If $f7 = d7$ and T is not been used in earlier re-keying process, then sink computes H $(P_{17} \otimes G_{req} \otimes T)$ which is represented as p7. The sink sends p7, G_{req} and T to CH1. The CH1 retrieves P_{17} from p7 and re-computes hash value given as b7 and compares it with p7. If $b7 = p7$ and the time T has not been utilized in a re-keying process previously, the re-key P_{17} is sent to node 7 and it is re-keyed. The node updates its secret to $P_{17} = G$ and informs the sink that it has successfully re-keyed and the sink then updates the node's secret in its table.

4 Overall Algorithm

The entire steps involved in clustering, key management and authentication are summarized in the following algorithm.

Step 1 The nodes are deployed in the physical environment and they report the base station about their physical locations.

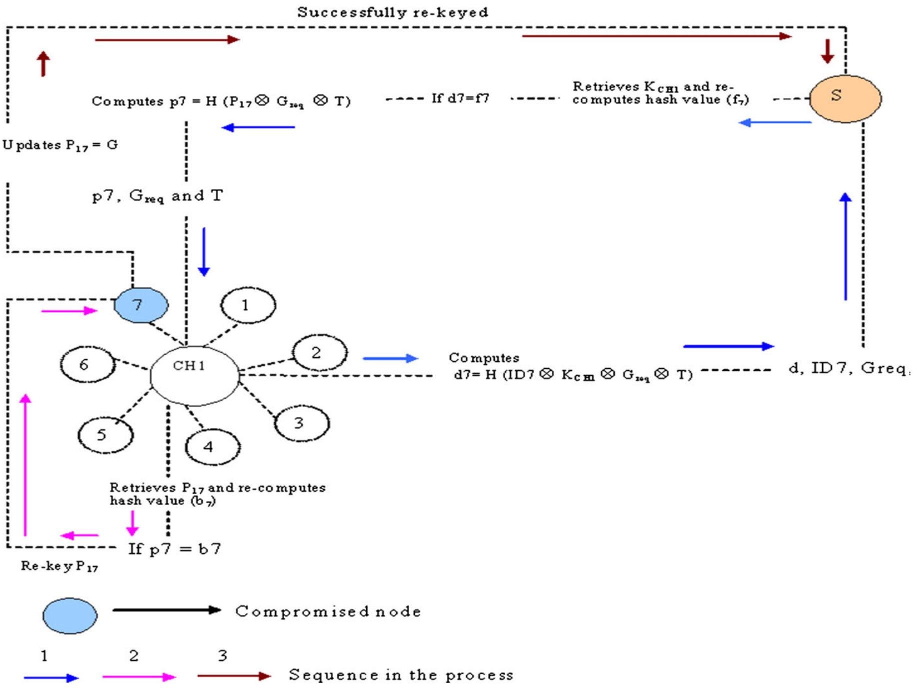

Successfully re-keyed
Computes p7 = H (P17 ⊗ Greq ⊗ T)
If d7=f7
Retrieves KCH1 and re-computes hash value (f7)
S
Updates P17 = G
p7, Greq and T
7
1
2
6
CH1
5
4
3
Computes d7= H (ID7 ⊗ KCH1 ⊗ Greq ⊗ T)
d, ID7, Greq
Retrieves P17 and re-computes hash value (b7)
If p7 = b7
Re-key P17
Compromised node
1
2
3
Sequence in the process

Fig. 3 Authentication technique

Step 2 Every node decides its potential of being a cluster head based on the selection criteria such as high energy resources, wide communication range and high processing capacity.

Step 3 When selection criteria are satisfied by a particular node, it is chosen as cluster head.

Step 4 The chosen cluster head assigns IDs to all its member nodes that intend to join the cluster.

Step 5 After the clusters are formed in the network, the CH sends the cluster id and the member id of its members to the sink and sink allots a cluster key to every cluster in the network.

Step 6 After getting the cluster key from the sink, each CH receives the pairwise key set for inter-cluster and intra-cluster communication

Step 7 After the pairwise keys are distributed by the CH to its members, for the establishment of the secure channels between the CH and the cluster members, the CH sends a hello message to the cluster members. Based on the reception of the Acknowledgement message from its members, the CH establishes a channel between itself and its members.

Step 8 When CH detects that one of the member nodes in its cluster is compromised or captured, it requests the sink to implement the re-keying operation.

Step 9 CH retrieves the ID of compromised node that requires re-keying and computes the hash value $\mathrm{H}(\mathrm{ID} \otimes \mathrm{K}_{\mathrm{CH}} \otimes \mathrm{G}_{\mathrm{req}} \otimes \mathrm{T})$ and forwards it to sink.

Step 10 The sink retrieves the K_{CH} and re-computes the hash value. Upon comparison, when the hash values computed at CH and sink are similar, the sink then computes the hash value. $H(P_{ij} \otimes G_{req} \otimes T)$, where P_{ij} is the new pairwise key for the compromised node. It forwards the computed hash value to the CH.

Step 11 The CH retrieves P_{ij} and re-computes hash value. When the hash value computed at the sink and CH are similar, the re-key P_{ij} is sent to the affected node and it is re-keyed.

Step 12 The node updates its secret key and informs the sink that it has successfully re-keyed and the sink then updates the node's secret in its table

5 Simulation Results

The proposed Energy Efficient Cluster Based Key Management (EECBKM) technique is evaluated through NS2 simulation. In the Table 2, we consider a random network of 100 sensor nodes deployed in an area of 500 × 500 m. Two sink nodes are assumed to be situated 100 m away from the above specified area. In the simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. The simulated traffic is CBR with UDP. The number of clusters formed is 9. Out of which, we transmit data from 4 cluster heads to the sink. 3 sensor nodes in each cluster are sending data to their cluster head. The attacker nodes are varied from 2 to 10.

5.1 Performance Metrics

The performance of EECBKM technique is compared with the SecLEACH [3] scheme. The performance is evaluated mainly, according to the following metrics:

- Average Packet Drop: The number of packets dropped due to various attacks is averaged over all surviving data packets at the destination

Table 2 Summarizes the simulation parameters used

No. of nodes	100
Area size	500 × 500
Mac	802.11
Routing protocol	EECBKMA
Simulation time	50 s
Traffic source	CBR
Packet size	512 bytes
Rate	250 kb
Transmission range	250 m
No of clusters sending data	1,2,3 and 4
No. of nodes per cluster sending data	3
Transmit power	0.395 W
Receiving power	0.660 W
Idle power	0.035 W
Initial energy	17.1 J
No. of attackers	2,4,6,8 and 10

- Average Packet Delivery Ratio: It is the ratio of the number of packets received successfully and the total number of packets transmitted
- Energy: It is the average energy consumed for the data transmission.

5.2 Results

5.2.1 *Based on Attackers*

In our initial experiment, we vary the number of attackers as 2, 4, 6, 8 and 10 from various clusters performing node capture attacks.

When the number of attackers is increased, naturally the packet drop will increase there by reducing the packet delivery ratio.

Since EECBKM reduces node capture attacks, the amount of packet drop is less, when compared with the existing schemes. Figures 4 and 5 give the packets drop and packet delivery ratio when the attackers are increased. Figure 6 gives the energy consumption when the number of attackers is increased.It shows that our proposed EECBKM technique achieves good packet delivery ratio with less packet drop when compared to SecLEACH scheme.

5.2.2 *Based on Various Cluster Sizes*

In this experiment we vary the cluster size from 1 to 4. 3 sensor nodes in each cluster are sending data to their cluster head, which are forwarded to the sink. The attacker nodes are kept as 2.

Figures 7 and 8 give the packets drop and packet delivery ratio when the cluster size is increased. It shows that the proposed EECBKM technique achieves good packet delivery ratio with less packet drop when compared to SecLEACH scheme.

Fig. 4 Attackers versus delivery ratio

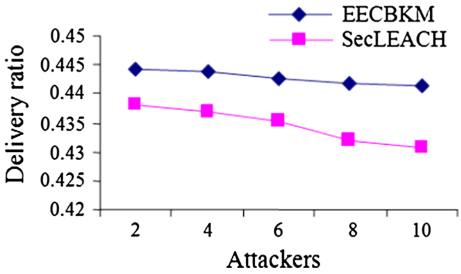


Fig. 5 Attackers versus packet drop

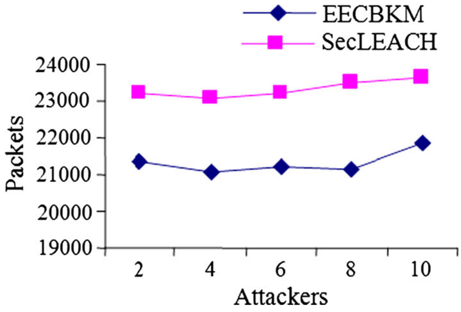
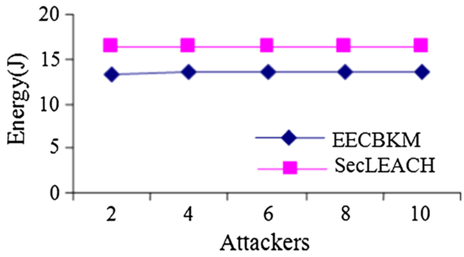


Fig. 6 Attackers versus energy



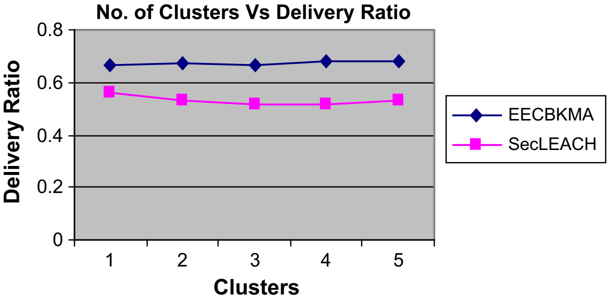


Fig. 7 Number of clusters versus delivery ratio

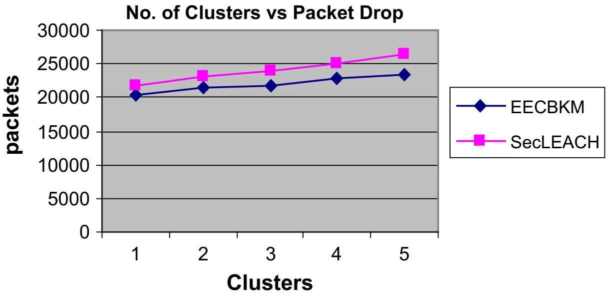


Fig. 8 Number of clusters versus packet drop

Figure 9 gives the energy consumption when the number of clusters is increased. It shows that the proposed EECBKM technique utilizes lower energy when compared to SecLEACH.

6 Results: Comparison with LEACH Protocol

LEACH(Low Energy Adaptive Clustering Hierarchy) and it is the first and most popular energy-efficient hierarchical clustering algorithm for WSNs reducing power consumption. In LEACH, the clustering task is rotated among the nodes, based on duration. Direct communication is used by each cluster head (CH) to forward the data to the BS. LEACH is

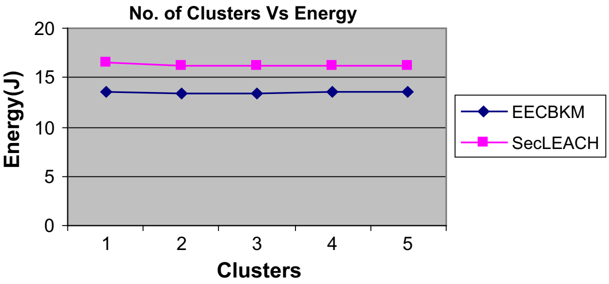


Fig. 9 Number of clusters versus energy

completely distributed and requires no global knowledge of network. Also, LEACH clustering terminates in a finite number of iterations, but does not guarantee good CH distribution and assumes uniform energy consumption for CHs.

Our experiment shows that the variation of energy consumption is very significant under the simulation network (i.e. 100 sensors). With a 0.5 J energy initially given to each node in both protocols, we can tell that nodes in the EECBKM tend to consume their energy at a slower rate if compared to the LEACH nodes.

According to the energy saving analysis, we can easily figure out that the number of alive nodes that may appear in EECBKM will be much higher than the number in LEACH. This can be depicted in Fig. 10.

7 Mobility Based Key Management

We consider a sensor network with a large number of sensor nodes that are randomly moves within the network. Hospital environment and nuclear power plants are some examples for this kind of distributed sensor network. When a sensor node moves from the transmission range of one cluster to another, then it invokes the mobility management scheme. This scheme function as follows,

1. Let $\mathrm{MS_i}$ be the sensor node that moves from cluster $\mathrm{C_i}$ of the network. It first forwards the C-REQ (cluster request) message to the sink. (Fig. 9) The C-REQ message includes the ID of $\mathrm{MS_i}$, sink ID, old cluster head id (O-CH-ID) and the random number ($\mathrm{R_n}$) produced by $\mathrm{MS_i}$. This can be denoted as,

$$MS_i \xrightarrow{C-REQ} \text{The sink} \left\{MS_i,\ SinkID,\ \left|O\text{-}CH\text{-}ID||R_n||MAC\left(K_{pri}, MS_i||SinkID||R_n\right)\right|\right\}$$

   The Message Authentication Code (MAC) is generated to the content of C-REQ message using the key $\mathrm{K_{pri}}$. Here, $\mathrm{K_{pri}}$ is the global key.

2. On receiving C-REQ message, the sink checks cluster information of $\mathrm{MS_{i-}}$ to ensure its legitimacy. If $\mathrm{MS_i}$ is proved to be the valid sensor then the sink verifies the MAC using $\mathrm{K_{pri}}$. On successful verification of MAC, the sink generates the pair wise keys ($\mathrm{P_{ij}}$ and $\mathrm{PH_{ii}}$) using EBS. $\mathrm{P_{ij}}$ and $\mathrm{PH_{ii}}$ keys are used for inter and intra cluster communication respectively.
3. The sink generates a random number $_{\mathrm{Rn+1}}$ and constructs a support message to the cluster nearby $\mathrm{MS_i}$. This support message comprises of MS_ID, pairwise keys, cluster ID and random number ($\mathrm{R_n}$ and $\mathrm{R_{n+1}}$) generated by $\mathrm{MS_i}$ and the sink respectively.

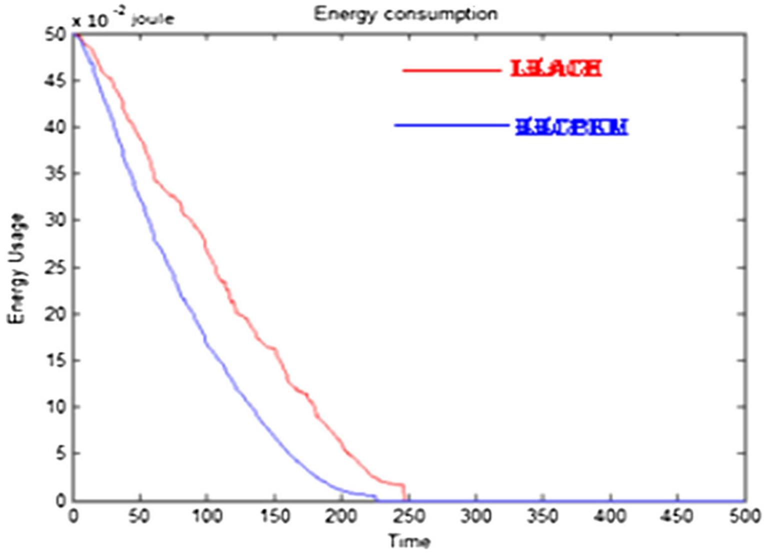


Fig. 10 Alive nodes

The content of support message is encrypted using cluster key (K_{CHi}). The support message is transmitted to the nearby cluster head (CH_{i+1}).

$$\text{The Sink} \xrightarrow{\text{Support Message}} \text{The Cluster Head}$$

$$\text{Support Message: } \left\{ SinkID, MS_iID, CH, E_{K_{CH_{i+1}}}\left(P_{ij,}PH_{ii}, R_n, R_{n+1}\right) \right\}$$

4. Apart from transmitting support message to the new cluster head, an intimation message is transmitted to the old cluster head of MS_i. The intimation message informs the cluster head about the removal of node MS_i to another cluster head.
5. While receiving the support message, the CH_{i+1} decrypts it using its cluster key and retrieves the pairwise keys. And then it forwards an approval message to MS_i. It includes random keys generated by MS_i and the sink, pairwise keys, node ID and cluster head ID.

$$\text{Cluster Head} \xrightarrow{\text{Approval}} MS_i$$

$$\text{Approval: } \{MS_iID, CH, R_n, R_{n+1}, MAC(P_{ij,}PH_{ii})\}$$

6. Now, MS_i verifies the random number R_n with its original random number and verifies the MAC using K_{pri} and retrieves the pairwise keys. Finally, the MS_i joins with the new cluster by transmitting back ACK message to the new cluster head.

Consider the mobility based key management illustration given in Fig. 11 In that, consider node CM_8 of Cluster C_2 roams away from the transmission range of CH2. At first, CM_8 transmits a C-REQ message to the sink. By receiving C-REQ, the sink checks the

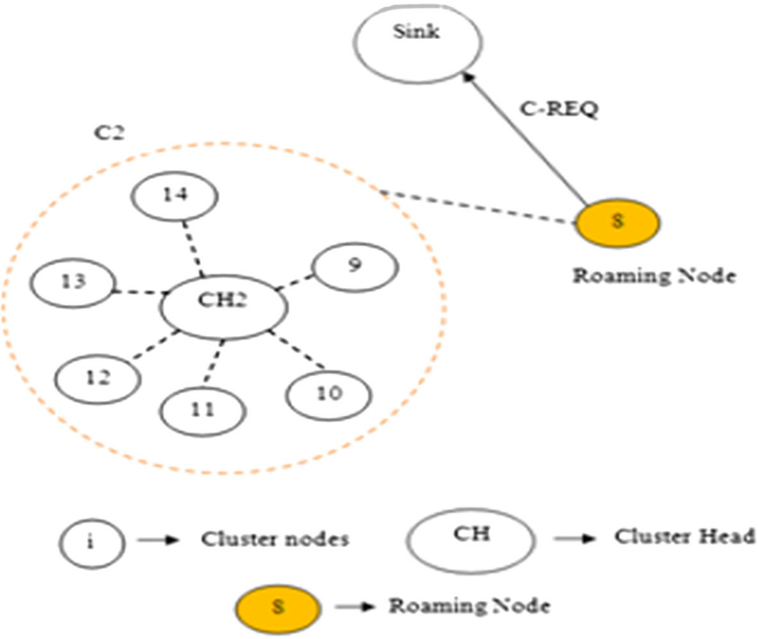


Fig. 11 Mobility management

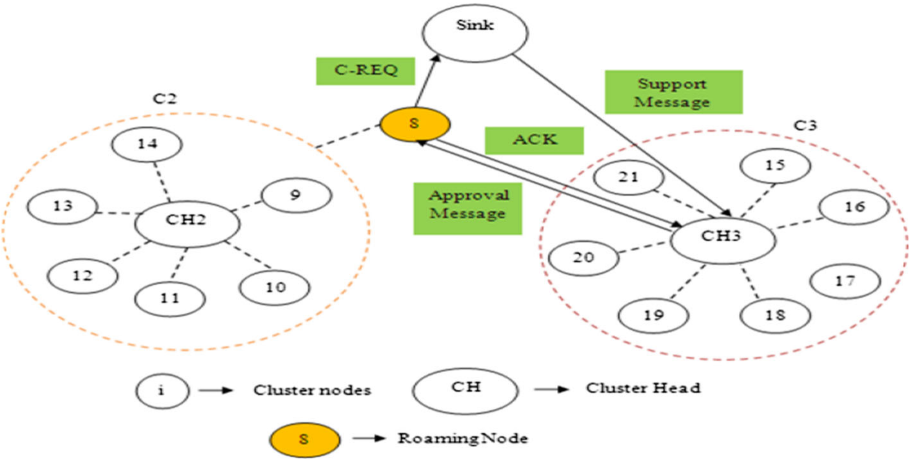


Fig. 12 Mobility based key management

authenticity of that node and generates a pairwise keys for inter and intra cluster communications. The generated keys are included in the support message and forwarded to the nearby cluster of CM_8. In Fig. 12, the nearby cluster is C3 (i.e) CH3. Now, CH3 transmits an approval message to CM_8 and an ACK message is forwarded back to CH3 by CM_8. Finally, CM_8 joins C3 and it became a member of CH3.

8 Conclusion

In this paper, a mobility management technique for keying scheme of WSNs is presented. The technique selects nodes with high energy resources, wide communication range and high processing capacity as cluster heads. Cluster keys for cluster heads and pairwise keys for nodes are generated by the sink through EBS. Whenever a node moves from currently connected cluster to another in the network, the mobility based key management scheme is triggered. The sink verifies the authenticity of roaming node and allocates it to a nearby cluster. New pairwise keys are generated and transmitted to the roaming node through newly connected cluster head. Further, a key organization technique is presented to ensure the forward and backward secrecy of nodes. The proposed technique is simulated in NS2 and simulation results show the efficiency of our technique.

References

1. de Brito, L. M. P. L., & Peralta, L. M. R. (2008). An analysis of localization problems and solutions in wireless sensor networks. *Polytechnical Studies Review, 6*(9), 1–27.
2. Lee, H., & Aghajan, H. (2005). Collaborative self-localization techniques for wireless image sensor networks. In *Proceedings of the asilomar conference on signals, systems and computers*.
3. Abuhelaleh, M. A., & Elleithy, K. M. (2010). Security in wireless sensor networks: Key management module in SOOAWSN. *International Journal of Network Security & Its Applications (IJNSA), 4*, 67–78.
4. Lalitha, T. (2012). Energy efficient cluster based key management & authentication technique for wireless sensor networks. *European Journal of Scientific Research, 76*(3), 403–410.
5. Jeong, Y. S., & Lee, S. H. (2006). Secure key management protocol in the wireless sensor network. Inter J. Inform. Proc. Syst. 2.
6. Dwoskin, J., Xu, D., Huang, J., Chiang, M., & Lee, R. (2007). Secure key management architecture against sensor-node fabrication attacks. In *Proceedings of the IEEE global telecommunications conference*, Nov. 26–30 (pp. 166–171). Washington, DC: IEEE Xplore Press. doi:10.1109/GLOCOM.2007.39
7. Jain, Y. K., & Jain, V. (2011). An efficient key management scheme for wireless network. *International Journal of Scientific and Engineering, 2*(2), 1–7.
8. Maala, B., Bettahar, H., & Bouabdallah, A. (2008). *Performances of key management schemes in wireless sensor networks*. Singapore: World Scientific Rev.
9. Shen, L., & Shi, X. (2008). A dynamic cluster-based key management protocol in wireless sensor networks. *International Journal of Intelligent Control and Systems, 13*, 146–151.
10. Kim, Y. H., Lee, H., & Lee, D. H. (2007). A secure and efficient key management scheme for wireless sensor networks. In *Third International Conference on Security and Privacy in Communications Networks and the Workshops*, Sept. 17–21 (pp. 162–167). Nice, France: IEEE Xplore Press. doi:10.1109/SECCOM.2007.4550324
11. Shaikh, R. A., Jameel, H., Auriol, B. J., Lee, H., Lee, S., & Song et al., Y. J. (2010). Achieving network level privacy in wireless sensor networks. *Sensors, 10*(3), 1447–1472.

Dr. T. Lalitha is an Associate professor at the department of Master of Computer Application (MCA), Salem. She have a 15 years experience in academic field. She completed Master of Computer Application in 2000 and completed M.Phil in Bharathidasan University in 2004. She completed Ph.D. Computer Sciencein 2013 at Bharathiar Uiversity, Coimbatore. Totally she had 30 International publication and these publications are scopus indexed and high impact factor. She had 31 International Conference publication. She published a book such as "Problem Solving Techniques", "Open Source System" and "Computer Concepts". She delivered a topic such as "Computer Algorithms", Open Source System", "Digital Communication" and "Network Security" in various Organizations. She is a Lifetime member of ISTE.

S. Jayaprabha is an Assistant Professor at the department of Master of Computer Application. She have 8 Years experience in academic field. She completed MCA in 2008. She had two Journal Publication and four conference publications.