ORIENTAL JOURNAL OF
COMPUTER SCIENCE & TECHNOLOGY
An International Open Free Access, Peer Reviewed Research Journal
www.computerscijournal.org

ISSN: 0974-6471
December 2011,
Vol. 4, No. (2):
Pgs. 293-304

Energy Efficient Cluster Based key Management Technique for Wireless Sensor Networks

T. LALITHA[1]* and R. UMARANI[2]

[1]Research Scholar, Bharatiar University, Coimbatore, Tamilnadu (India).
[2]Research Supervisor, Bharatiar University, Coimbatore, Tamilnadu (India).
*Corresponding autor: E-mail: lalithasrilekha@rediffmail.com

(Received: October 04, 2011; Accepted: October 06, 2011)

ABSTRACT

Wireless Sensor Networks (WSN) is vulnerable to node capture attacks in which an attacker can capture one or more sensor nodes and reveal all stored security information which enables him to compromise a part of the WSN communications. Due to large number of sensor nodes and lack of information about deployment and hardware capabilities of sensor node, key management in wireless sensor networks has become a complex task. Limited memory resources and energy constraints are the other issues of key management in WSN. Hence an efficient key management scheme is necessary which reduces the impact of node capture attacks and consume less energy. In this paper, we develop a cluster based technique for key management in wireless sensor network. 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 EBS key set to every cluster head. The EBS key set contains the pairwise keys for intra-cluster and inter-cluster communication. During data transmission towards the sink, the data is made to pass through two phases of encryption thus ensuring security in the network. By simulation results, we show that our proposed technique efficiently increases packet delivery ratio with reduced energy consumption.

Key words: Wireless Sensor Networks,Key Management, Data Transmission, Attacks,Cluster.

INTRODUCTION

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 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 wireless sensor networks 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[3].

Attacks in Sensor Networks

The threats and challenges of sensor networks are

´ Spoofed, altered, or replayed routing information

´ selective forwarding

´ sinkhole attacks;

´ Sybil attacks

´ Wormholes

´ HELLO flood attacks

´ Acknowledgement spoofing.[6]

Network Security in Sensor Networks

In wireless channels, the communication is not completely secure and is subjected to security hazard. In the wireless channels, the possible security threat can be divided into two threats: inside threat and outside threat. In case of outside threat in the sensor network, the attacker does not possess control over the cryptographic materials. Whereas in case of the inside threat, the attacker will be possess some key materials and trust of some sensor nodes.

Compromising the sensor nodes is an easy task due to the absence of the expensive tampering resistant hardware. Even if it possesses the tampering resistant hardware, it may be very reliant. Modification, forging and discarding the messages is possible in case of a compromised node[7].

In vulnerable locations, maintaining the security of the sensor nodes is a major task. In WSN, the encoding and the authentication of the communication carried out is necessary, to ensure security. For communication between the sensor nodes, few solutions have been developed to attain stability in communication. Distribution key method, dissymmetric encryption method, and key predisposition method are the three kinds of key management techniques[4]. The attacks like jamming and spoofing are very destructive to the sensor networks. Whenever the cluster heads are responsible for the transmission and reception of the data, this nature of the Cluster Hierarchy distribution networks makes it susceptible to destructive networks. So, the network will get destructed if a hacker tries to become the cluster head of the cluster. Examples of this type of attack are the selective forwarding and the sinkhole attacks[5].

Key Management in Wireless Sensor Networks

Use of the pairwise keys between sensor nodes is the necessary requirement of the WSN for ensuring security. The trusted-server scheme, the self-enforcing scheme, and the key pre distribution scheme are the three classes of the key agreement schemes. A trusted server is assumed to exist in the case of trusted-server scheme for the establishment of keys between the nodes. But in case of distributed sensor networks, trusted server scheme is not appropriate due to the difficulty in developing a trusted network. Asymmetric cryptography, like that of public key certificate is utilized in the self enforcing scheme. But for sensor networks, use of the public key algorithm is inappropriate due to the restricted amount of power and resources for computation in the minute sensor node. In the key pre-distribution schemes, loading of the keying materials takes place at a prior basis in the sensor nodes[8].

In a wireless sensor network, the computation and communication capacity of every node is limited to a particular level. Node groups can be used for executing in network data aggregation and analysis. For instance, a vehicle can be tracked by a node group jointly via network. The nodes belonging to a group will keep varying repeatedly and at a faster rate in the network. In the wireless sensor network, most of the key services are executed by the groups. Hence, for admission of the new members to the group and to support group communication at a secure level, it is necessary to have a secure protocol for group management. After the computation within the group, the result is transferred to the base station. In order to ensure the transmission from a legitimate group, the result must be authenticated[9].

Issues in Key Management

More often the usage of the sensor network is in environment which is open and not well monitored. Key management has become a challenging task due to the numerous sensor nodes used and the reduced knowledge about the sensor node deployment abilities[3].

Impracticality of public key cryptosystems

The usage of the public-key algorithms, like that of Diffie-Hellman key agreement or RSA signatures is not desired due to the restricted ability of computation and restricted availability of the power resources in the sensor nodes. At present, the operations are executed by the sensor nodes over a time interval of seconds to minutes thus making it more prone to t he threats like denial of service (DoS) attacks in the network.

Limited memory resources

Due to the limited memory of the sensor nodes, the key storage memory is also limited. Hence it is not possible to assign unique keys to each node in this network[10].

Problem Identification and Solution

The main issues in the cluster based key management approach in WSN are

´ In a cluster based network, once the cluster head is compromised then the entire cluster can be broken by the simple DOS attack.

´ If the selection of the cluster head is not dynamic then the sensor nodes that are far off from the CH will exhaust its energy while trying to communicate and this leads to the formation of blind spots in the WSN.

´ Finally, WSN are vulnerable to physical attackers. An attacker can capture one or more sensor nodes and reveal all stored security information (particularly stored keys) which enables him to compromise a part of the WSN communications. For all these reasons, an efficient key management scheme should be implemented in the sensor before its deployment.

Hence this scheme must answer the following requirements:

´ Low processing/energy consumption.

´ Low memory usage.

´ High attack resistance level.

As we can see, these are conflicting requirements, and it is hard to answer them at once.

During these last years, several key management schemes have been proposed in the literature. But the all the requirement for an efficient WSN is not completely met.

In this paper, we develop an efficient cluster based key management technique for wireless sensor networks which optimizes the overhead and number of keys used.

For cluster formation, a node with high energy level and communication range broadcasts itself to all the other surrounding nodes within its coverage area. Based on the replies from the remaining nodes which include the id of the corresponding node, the broadcasted node is selected as the cluster head.

After the cluster head (CH) formation, the sink allots a cluster key to every cluster head in the network. After getting the cluster key from the sink, each CH receives the Exclusion Basis System (EBS) [14] based key set from the sink which contains the pairwise keys between the CH and its members, encrypted by the cluster key. The pairwise keys for the communication between the cluster heads are also supplied by the sink which is encrypted by the cluster key. Using the pairwise keys between the CHs, inter-cluster communication, i.e, communication between the clusters, is also allowed.

Energy efficient cluster based key management technique

Cluster Formation

In the wireless sensor network, 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, 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}.

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

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 contains the ID and the response content Ack.

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

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.

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

Cluster Communication

Fig.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 to 7, CH2 contains members 8 to 14 and CH3 contains members 15 to 21.

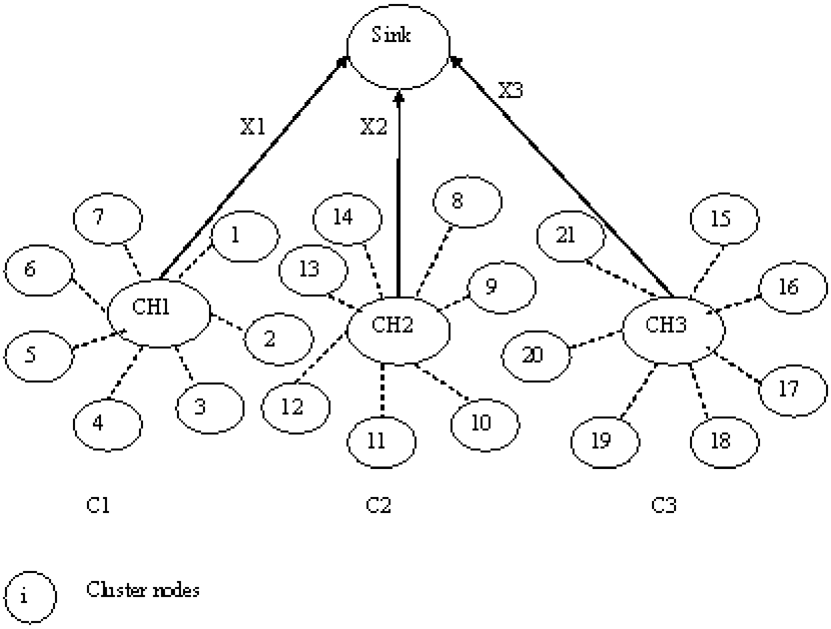

Fig.1: Clustering architecture

After the clusters are formed in the network, the CH sends the information of its members like <cluster id, member id> to the sink.

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

X1 = {<C1,1>, <C1,2>,………….. , <C1,7>}
X2 = {<C2,8>, <C2,9>,…………..., <C2,14>}
X3 = {<C3,15>, <C3,16>,………..., <C3,21>}

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.

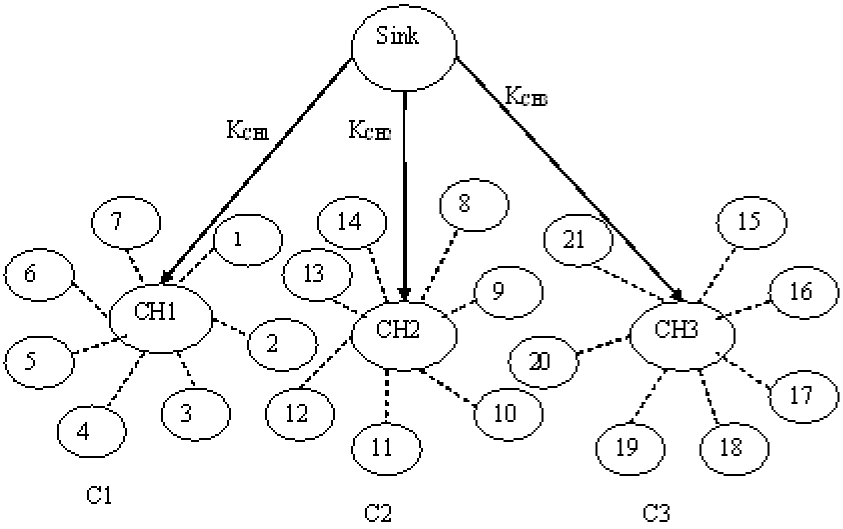

Fig. 2: Cluster key transmission from the sink to the cluster head

After getting the cluster key from the sink, each CH receives the pairwise key set which is based on Exclusion Basis System (EBS) [14]. (which will be explained in section 4)

$$\text{Sink} \xrightarrow{K_{CHi}\{\text{EBS key set}\}} CH_i$$

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

$$\text{Sink} \xrightarrow{K_{CHi}\{P_{ij} \| PH_{ii'}\}} CH_i$$

where i = 1,2,3

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.

$$CH_i \xrightarrow{P_{ij}} \{CM\}_j$$

where
i = 1→j = 1 to 7
i = 2→j = 8 to 14
i = 3→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.

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

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

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

where
i = 1 →j = 1 to 7
i = 2 →j = 8 to 14
i = 3 →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.

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

$$CH_1 \overset{K_{15}}{\rule{2cm}{0.4pt}} \{CM\}_5$$

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.

2.2.2 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.

$$CH_i \xrightarrow{PH_{ii'}} CH_{i'}$$

where i = 1,2,3; i'= 1,2,3 and i ≠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 ≠ i'

$$CH_i \xleftarrow{\text{Ack message}} CH_{i'}$$

where i = 1,2,3; i' = 1,2,3 and i ≠ i'

$$CH_i \xleftrightarrow{\quad\quad} CH_{i'}$$

where i = 1,2,3; i'= 1,2,3 and i ≠ 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.

K_{23}

In order to establish a secure channel

between C2 and C3, the following steps are followed:

$$CH_2 \xrightarrow[K_{23}]{} CH_3$$

Next the hello message is sent by C2 to C3

$$CH_2 \xrightarrow{\text{Hello message}} CH_3$$

$$CH_2 \xleftarrow{\text{Ack message}} CH_3$$

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

$$CH_2 \xleftrightarrow{\text{Secure channel}} CH_3$$

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

$$\{CM\}_{10} \leftarrow \text{Secure channel} \rightarrow \{CM\}_{15}$$

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 packet to be transmitted are encrypted with the pairwise 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.

$$\{CM\}_{11} \xrightarrow{K_{211}\{D\}} CH_2$$

Phase 2

At the $CH_{2,}$ the data is decrypted using the pairwise key. Then CH_2 encrypts the data with the cluster key, K_{CH2} and transmits it to the sink.

$$CH_2 \xrightarrow{K_{CH2}\{D\}} \text{sink}$$

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

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 ζ [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. 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

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. 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] = T1 U T2,
[1,8] – [2] = T1 U T3,
[1,8] – [3] = 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[14].

Advantages of the proposed work

´ Since the data is always encrypted twice prior transmission, the security of the data in the network is ensured.

´ If a member node or the cluster head in the network is compromised, then only the nodes of that particular cluster get affected whereas the nodes of other clusters are unaffected.

´ Since the cluster head is selected based on the energy cost and processing capacity, the energy consumed by the cluster head is minimized.

Table 1: Summarizes the simulation parameters used

No. of Nodes	100
Area Size	500 × 500
Mac	802.11
Routing protocol	EECBKM
Simulation Time	50 sec
Traffic Source	CBR
Packet Size	512 bytes
Rate	250kb
Transmission Range	250m
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 Joules
No. of Attackers	2,4,6,8 and 10

Simulation results

The proposed Energy Efficient Cluster Based Key Management (EECBKM) technique is evaluated through NS2 [18] simulation. We consider a random network of 100 sensor nodes deployed in an area of 500 × 500m. Two sink nodes are assumed to be situated 100 meters 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.

Performance Metrics

The performance of EECBKM technique is compared with the SecLEACH[17] 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.

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.

RESULTS

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. Figure 3 and 4 give the packets drop and packet delivery ratio when the attackers are increased. It shows that our proposed EECBKM technique achieves good packet delivery ratio with less packet drop when compared to SecLEACH scheme.

Since the cluster heads are selected based on the energy cost, the overall energy consumption is less in EECBKM. Figure 5 gives the energy consumption when the number of attackers is increased. It shows that our proposed EECBKM technique utilizes lower energy when compared to SecLEACH.

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.

Figure 6 and 7 give the packets drop and packet delivery ratio when the cluster size is increased. It shows that our proposed EECBKM technique achieves good packet delivery ratio with less packet drop when compared to SecLEACH scheme.

Figure 8 gives the energy consumption when the number of clusters is increased. It shows that our proposed EECBKM technique utilizes lower energy when compared to SecLEACH.

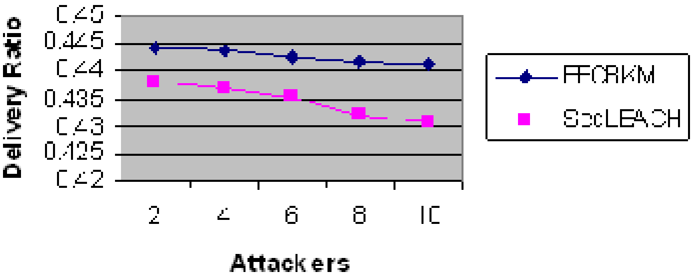

Fig. 3: Attackers Vs Delivery Ratio

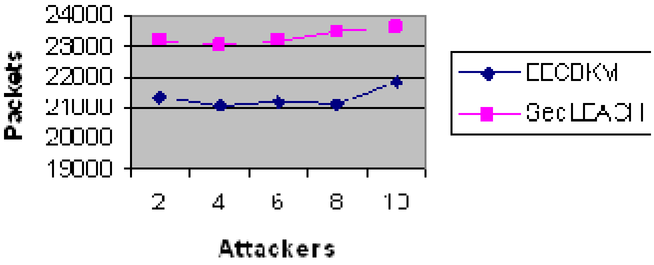

Fig. 4: Attackers Vs Packet Drop

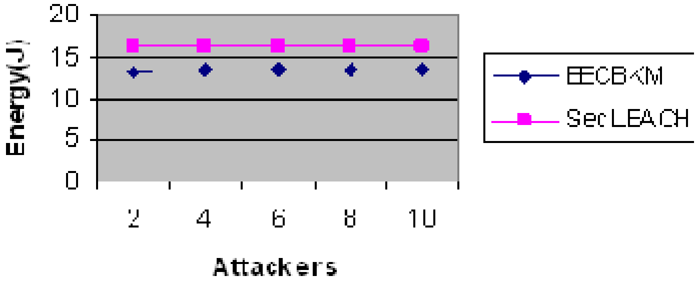

Fig. 5: Attackers Vs Energy

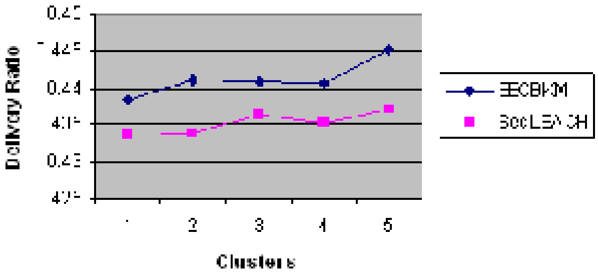

Fig. 6: No. of Clusters Vs Delivery Ratio

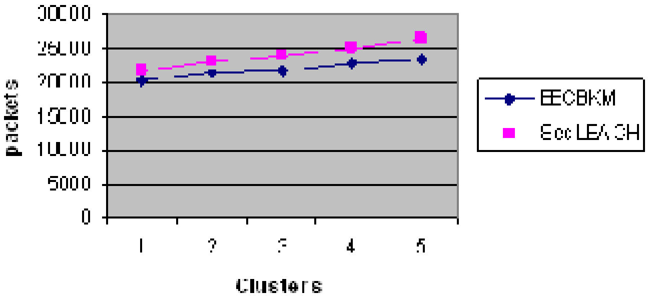

Fig. 7: No. of Clusters Vs Packet Drop

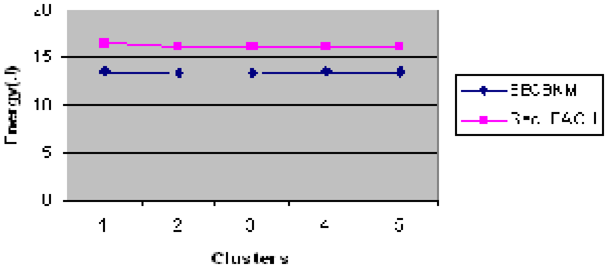

Fig. 8: No. of Clusters Vs Energy

CONCLUSION

In this paper, we have developed an efficient technique for key management in the wireless sensor network. During the formation of a cluster, initially a cluster head is selected based on eligibility criteria such as energy cost, coverage and processing capacity. After the cluster head selection, the information about all the members of the cluster is sent to the sink by the cluster head. The sink then provides the cluster head with the cluster key and the EBS key set required for the communication between the nodes. These keys are distributed to the nodes by the cluster head prior communication. After the key distribution, secure channel is established between the nodes and the clusterhead. During the data transmission from the cluster members to the sink, the data passes two phases. In the first phase the data is encrypted and transmitted to the clusterhead. In the second phase, the data is encrypted by another key by the clusterhead and then transmitted to the sink. Thus this technique allows inter cluster as well as intra cluster communication in a very efficient manner with high security. By simulation results, we have shown that our proposed technique efficiently increases packet delivery ratio with reduced energy consumption.

REFERENCES

1. Lina M. Pestana Leão de Brito and Laura M. Rodríguez Peralta, “An Analysis of Localization Problems and Solutions in Wireless Sensor Networks”, Polytechnical Studies Review, **6**: ISSN: 1645-9911 (2008).
2. Huang Lee and Hamid Aghajan, “Collaborative Self-Localization Techniques for Wireless Image Sensor Networks”, In Proc. of Asilomar Conf. on Signals, Systems and Computers, (2005).
3. D.Saravanan , D.Rajalakshmi and D.Maheswari “DYCRASEN: A Dynamic Cryptographic Asymmetric Key Management for Sensor Network using Hash Function”,

International Journal of Computer Applications (0975 – 8887) **18**(8): (2011).

4. Yoon-Su Jeong, and Sang-Ho Lee "Secure Key Management Protocol in the Wireless Sensor Network", *International Journal of Information Processing Systems*, **2**(1): (2006).
5. Mohammed A. Abuhelaleh and Khaled M. Elleithy "Security in Wireless Sensor Networks: Key Management Module in Sooawsn", *International Journal of Network Security & Its Applications* (IJNSA), **2**(4): (2010).
6. John A. Clark, John Murdoch, John A. McDermid, Sevil Sen, Howard R. Chivers, Olwen Worthington and Pankaj Rohatgi "Threat Modelling for Mobile Ad Hoc and Sensor Networks", In Annual Conference of ITA (2007).
7. Yingpeng Sang and Hong Shen "Secure Data Aggregation in Wireless Sensor Networks: A Survey", PDCAT (2006).
8. Jiyong Jang, Taekyoung Kwon and Jooseok Song "A Time-Based Key Management Protocol for Wireless Sensor Networks", ISPEC, 314-328 (2007).
9. Adrian Perrig, John Stankovic, *and* David Wagner "Security in Wireless Sensor Networks", Communications of the ACM **47**(6): (2004).
10. Haowen Chan Adrian Perrig Dawn Song "Random Key Predistribution Schemes for Sensor Networks", Proceedings. 2003 Symposium 197-213 (2003).
11. Jeffrey Dwoskin Dahai Xu Jianwei Huang Mung Chiang Ruby Lee "Secure Key Management Architecture against Sensor-node Fabrication Attacks", In the Proceedings of IEEE GLOBECOM (2007).
12. Yogendra Kumar Jain, Vismay Jain "An Efficient Key Management Scheme for Wireless Network", *International Journal of Scientific & Engineering Research*, **2**(2): ISSN 2229-5518 (2011).