LIGHTWEIGHT KEY MANAGEMENT SCHEME FOR HIERARCHICAL WIRELESS SENSOR NETWORKS

Mohammed A. Al-taha[1] and Ra'ad A. Muhajjar[2]

[1]Department of Computer Science, College of Science, Basrah University, Iraq
[2]Department of Computer Science, College of Computer Science and Information Technology, Basrah University, Iraq

ABSTRACT

Wireless Sensor Networks (WSNs) are critical component in many applications that used for data collection. Since sensors have limited resource, security issues have become a critical challenge in Wireless Sensor Networks. To achieve security of communicated data in the network and to extend the WSNs lifetime; this paper proposes a new scheme called Lightweight Key Management Scheme (LKMS). LKMS used Symmetric Key Cryptography that depends only on a Hash function and XOR operation. Symmetric Key Cryptography is less computation than Asymmetric Key Cryptography. Simulation results show that the proposed scheme provides security, save the energy of sensors with low computation overhead and storage.

KEYWORDS

Wireless sensor Networks, Key Management, Symmetric Cryptography, hash function, XOR

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of hundreds, even thousands of low-cost devices called sensor nodes. These sensors have resource constraints such as limited power resource and low memory. Sensors in WSNs can communicate with each other by radio channel to transmit the data to the centric node known as the sink node (Base Station). WSNs has formed the basics for covering a different range of applications such as health care, military, environmental monitoring and other fields [1].

WSNs can be classified as flat networks and hierarchical networks. In flat networks, every sensor in the network has the same characteristics (battery lifetime, storage capacity, processor and transmitted power) and perform the same task. In flat networks sensor nodes can transmit data to its neighbour one by one to the sink. In hierarchical networks, the network divided in to several groups called clusters (each cluster has a head called cluster head and the other nodes called cluster members). Hierarchical WSNs can be implemented as homogeneous and heterogeneous. All sensor nodes have the same capabilities in homogeneous WSNs. In heterogeneous WSNs incorporate different types of sensor nodes that have different capabilities (small number of sensors with powerful characteristics elects as Cluster head and a large number of low characteristics sensors elects as Cluster members).

Natarajan Meghanathan et al. (Eds) : ICCSEA, WiMoA, SPPR, GridCom, CSIA - 2017
pp. 139– 147, 2017. © CS & IT-CSCP 2017   DOI : 10.5121/csit.2017.71111

Most applications that use WSNs [2] have sensitive data as in military applications. Due to limitation resource of sensor node and the hostile environment make it a big challenge to secure the network. Key management is the first requirements to secure communication in WSNs. Key management includes generation, distribution and installation the keys inside sensors and it should support the node addition and deletion with revocation and update the keys [3].Symmetric and Asymmetric cryptography are being used to achieve security (authentication, integrity, privacy) in WSNs. In Symmetric, each node (sender and receiver)shared the same secret key that used for encryption and decryption the data communicated in the network. In Asymmetric key, each node, the sender and receiver, have two keys which are the public key that known to all nodes in the network and secret key which is private [5]. Symmetric cryptography is less computation and consuming less energy than Asymmetric Key [4].

In this paper, Lightweight Key Management Scheme(LKMS) for heterogeneous WSNs has been proposed to generate symmetric key that use only a hash function and XOR operation to provide security in WSN.

2. RELATED WORKS

Many key management schemes have been proposed to protect the communication among sensors in the WSNs. In [6], the authors proposed symmetric key by using LU decomposition matrix of length n*n where n represent the total number of sensor nodes. Each node is pre-loaded randomly with one row of L matrix with corresponding column of U matrix, when two nodes want to communicate first they send to each other their L row then each node calculate session key by multiplying the receiving L with its U column and found that they shared the same value, the proposed scheme used Rivets Cipher (RC5) algorithm to encrypt / decrypt of transmitted data. The authors in [7], used Hybrid key management for heterogeneous WSN, they used Symmetric and Asymmetric cryptographic technique in three levels, first level used signature encryption algorithm based on Elliptic Curve Cryptography (ECC) to secure the communication between the sink and the cluster head. In second level, Diffie-Hellman key exchange algorithm based on ECC is used to generate shared key between the cluster head and sensor nodes. finally, Symmetric session key is used between two sensor nodes because the limited resource of the sensor nodes. Each cluster head selects random value r and broadcast it to all cluster members.

The scheme used shared value based on r when two nodes want to communicated. The authors of [8] proposed a key management scheme for hierarchical WSN by using Power aware routing protocol and track – sector clustering, the track sector clustering scheme is used for minimizing the transmission data redundant by means of reducing the connection between the sink node and the cluster head. Hybrid Elliptic Curve Cryptography (HECC) technique is implementing that used 80 bits key size for securing the routing. In [9] the authors proposed a new scheme used three types of keys: Network key, group key and pairwise key. The network key is used for encrypting the broadcast message and for authenticate the new node, group key is used as a shared key between all the sensor nodes in the same cluster, and the last key, the pairwise key is shared between specific pair of nodes. In the proposed scheme, they used assistant node to improve the security and reduce resource cost when cluster head is compromised. In [10], The authors proposed key management for hierarchical WSN using UAV to establish session key between two nodes. The proposed method used symmetric cryptographic key management which is depended on XOR operation and hash function. UAV used as centre unit to reduce the storage. Two nodes can’t establish session key only with the help of UAV.

3. THE PROPOSED METHOD

3.1 Network Model

The proposed scheme adopts the Heterogeneous hierarchical WSN as shown in the Figure 1.

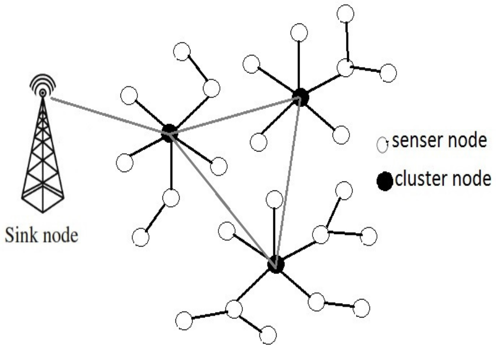

Figure1. Network Model

The proposed scheme assumes that network has the following properties:

1- The sink node has unlimited storage and sufficient energy with trusted place.
2- All sensor nodes are static.
3- If sensor node is captured by attacker, sensors can extract all information from it.
4- Cluster head is equipped with tamper resistant. If the cluster head captured by attacker, sensors cannot extract the information stored on it.

The notation of our scheme is summarized in table 1:

Table 1. Notation

Symbol	definition
K_{IN}	Initial key from sink node
K_{CN}	Shared value between cluster and its member
K_{Hi}	Shared value between cluster i and sink node
SK_{CH-BS}	Session key between sink node and cluster head i
SK_{CH-ab}	Session key between Cluster head-a and cluster head-b
SK_{AB}	Session key between Cluster head and sensor node or between sensor node-a and sensor node-b
ID_{i}	Identity of sensor node i
ID_{Hi}	Identity of cluster i
r	Random value
T	Time stamp
$hash(\,)$	Hash function
$\oplus$	XOR
$\|\|$	concatenation

3.2 Pre-distribution Phase

Before deployment of the sensor nodes, sink node selected initial key K_{IN}. Each node is assigned a uniqueID_{i} and pre-loaded with the initial K_{IN}

3.3 Key Management Phase

After the deployment, each cluster head broadcasts HELLO message which include its ID. Each sensor node may receive messages coming from more than one cluster head. Sensor node chose a cluster head that have best signal strength. Sensors are storing one more parameter which is the cluster head identity. Now each sensor node has three parameters: initial key, its identity and cluster head identity. After that each cluster head selects random value r and broadcasts it to all cluster member and to sink node encrypted by initial key K_{IN}. Each sensor node gets r by decrypting it with the initial key K_{IN} , then each sensor node calculates shared value K_{CN} with its cluster head and all cluster member as follow:

$$K_{\mathrm{CN}} = hash(r||ID_H||\,K_{IN}\,)$$

After that the node will delete r. When sink node get the encrypted r,sink node decrypt it by using K_{IN} and calculating a shared value between the sink and the cluster head CH_i as

$$K_{Hi} = hash(r||ID_{Hi})$$

The proposed scheme has three types of keys; between sink node and CH_i, between cluster head and cluster head and between cluster head and sensor member or sensor node and sensor node in same cluster

3.3.1 Sink node& cluster head

When the cluster head wants to be communicated with the Sink node, cluster head and sink useK_{Hi} as shared value between them.The session key process can be described as below:

1- First the Cluster Head send its ID to Sink node
2- After that, sink node select nonce random value R_{BS} and compute:

$$X_{BS} = R_{BS} \oplus K_{Hi}$$

$$Y_{BS} = hash(R_{BS}||K_{Hi}||\,T_1)$$

Then sink node send [$X_{BS}\,,Y_{BS}\,,T_1$] to Cluster Head

3- Cluster Head receive [$X_{BS}\,,Y_{BS}\,,T_1$] from sink node, first it verified the time stamp whether | $T_1 - T_C$ | < ΔT or not. If verification holds then computes

$$R'_{BS} = X_{BS} \oplus K_{Hi}$$
$$Y'_{BS} = hash(R'_{BS}||K_{Hi}||\,T_1)$$

If $Y'_{BS} = Y_{BS}$ then Cluster Head select a random nonce R_{ch} , otherwise send a rejection message to sink node.

Now Cluster Head computes:

$$X_{ch} = R_{ch} \oplus K_{Hi}$$

$$Y_{ch} = hash(R_{ch}||K_{Hi}||\,T_2)$$

Then Cluster Head send [X_{ch} , Y_{ch} , T_2] to sink node

4- Sink node receive [X_{ch} , Y_{ch} , T_2] from Cluster Head, first it verified the time stamp whether $|T_2 - T_C| < \Delta T$ or not. If verification holds then computes

$$R'_{ch} = X_{ch} \oplus K_{Hi}$$
$$Y'_{ch} = hash(R'_{ch} || K_{Hi} || T_2)$$

If $Y'_{ch} = Y_{ch}$proceed further, otherwise send a rejection message to sink node

5- Finally, both sink node and Cluster Head agree on same session key

$$SK_{CH-BS} = hash\ (R_{ch} \oplus R_{BS})$$

3.3.2 Cluster Head & Cluster Head

When the cluster head want to be communicated with other Cluster Head, they use K_{IN} as shared value between them.The session key process can describe as below:

1- First the Cluster Head-a send its ID to second Cluster Head-b

2- After that, Cluster Head-b select nonce random value R_{CHb} and compute:

$$X_{CHb} = R_{CHb} \oplus K_{IN}$$

$$Y_{CHb} = hash\ (R_{CHb} || K_{IN} || T_1)$$

Then Cluster Head-b send[X_{CHb} , Y_{CHb}, T_1] to Cluster Head-a

3- Cluster Head-a receive [X_{CHb} , Y_{CHb} , T_1] from Cluster Head-b, first it verified the time stamp whether $| T_1 - T_C | < \Delta T$ or not. If verification holds then computes

$$R'_{CHb} = X_{CHb} \oplus K_{IN}$$
$$Y'_{CHb} = hash\ (R'_{CHb} || K_{IN} || T_1)$$

If $Y'_{CHb} = Y_{CHb}$ then Cluster Head-a select a random nonce R_{CHa} , otherwise Cluster Head-a send a rejection message to Cluster Head-b.

Now Cluster Head-a computes:

$$X_{CHa} = R_{CHa} \oplus K_{IN}$$

$$Y_{CHa} = hash\ (R_{CHa} || K_{IN} || T_2)$$

Then Cluster Head-a send [X_{CHa} , Y_{CHa} , T_2]to Cluster Head-b.

4- Cluster Head-b receive [X_{cha} , Y_{cha} , T_2] from Cluster Head-a, first it verified the time stamp whether $| T_2 - T_C | < \Delta T$ or not. If verification holds then computes

$$R'_{CHa} = X_{CHa} \oplus K_{IN}$$
$$Y'_{CHa} = hash\ (R'_{CHa} || K_{IN} || T_2)$$

If $Y'_{CHa} = Y_{CHa}$ proceed further, otherwise send a rejection message to Cluster Head –b.

5- Finally, both Cluster Head-a and Cluster Head-b agree on same session key

$$SK_{CH-ab} = hash\ (R_{CHa} \oplus R_{CHb})$$

3.3.3 Cluster Head &Sensor ORSensor & Sensor

Suppose that two sensor nodes A and B are neighbors. Session key process is presented as bellow:

1- First node A send its ID to node B

2- Node B select nonce random value R_B and compute:

$$X_B = R_B \oplus K_{CN}$$

$$Y_B = hash\ (R_B || K_{CN} || T_1)$$

Then node B send [X_B , Y_B , T_1] to node

3- Node A receive [X_B , Y_B , T_1] from node B, first it verified the time stamp whether I $T_1 - T_C$ I < ΔT or not. If verification holds then computes

$$R'_B = X_B \oplus K_{CN}$$

$$Y'_B = hash\ (R'_B || K_{CN} || T_1)$$

If $Y'_B = Y_B$ then node A select a random nonce R_A , otherwise send a rejection message to node B.

Now node A computes:

$$X_A = R_A \oplus K_{CN}$$

$$Y_A = hash\ (R_A || K_{CN} || T_2)$$

Then node A send [X_A , Y_A , T_2] to node B

4- Node B receive [X_A , Y_A , T_2] from node A, first it verified the time stamp whether I $T_2 - T_C$ I < ΔT or not. If verification holds then computes

$$R'_A = X_A \oplus K_{CN}$$

$$Y'_A = hash\ (R'_A || K_{CN} || T_2)$$

If $Y'_A = Y_A$ proceed further, otherwise send a rejection message to node A.

5- Finally, both node A and B agree on same session key

$$SK_{AB} = hash\ (R_A \oplus R_B)$$

4. Security Analysis

The security analysis of the proposed scheme can be discussed as following:

4.1 Key updating

In the proposed scheme, each session used key different from the others. Every session key depends on shared value between two nodes. For more security, the shared value between two nodes should be updated periodically. After certain period, the update phase is started. Cluster head selects new random value $r\ \ '$ that is different from old r and it's never used before, then broadcasts it to all cluster member and to the sink node encrypted with initial key K_{IN}. When the nodes receive the new encrypted $r\ \ '$ they get it by decrypting it with initial key K_{IN}. After that,nodes calculate the new shared value $K_{\mathrm{CN}}\,'$ as fallowing:

$$K_{\mathrm{CN}}\,' = hash(r'||ID_H||\, K_{IN}\,)$$

Then nodes will delete $r\ \ '$. The sink node gets the new r' by decrypting it with initial key K_{IN}. After that calculate a new shared value between the sink node and the cluster head CH_i as

$$K_{Hi}' = hash(r\,'||K_{Hi})$$

4.2 Add New Nodes

The sensor nodes have limited energy and after some time the node will die, for that, the died node should be replaced with new node. Before deployment the new node, the sink node is known which cluster head is belong to according to location of the new node. So sink node pre-loaded the new node with cluster head ID, initial key K_{IN} and the shared value K_{CN} according to the value of r of that cluster.

4.3 Key Revocation

When a node captured, all the security information stored on it will become compromised. The shared value between the sensors in the same cluster should be updated. The cluster selects new random value $r\ \ '$ and continue with update phase.

5. PERFORMANCE EVALUATION

We evaluate the performance of our scheme from storage overhead, energy consumption and time. In our simulation environment, we used 100 sensors that randomly deploy with area 100m*100m, 9 sensors as cluster head and 91 as sensor nodes. The transmission range of cluster head is 40m and the sensor nodes is 20m with initial energy 5J and 15Jrespectively.

5.1 Storage Overhead

In [10] before the deployment, each node is preloaded with its ID and a secret key. After deployment, each sensor node need to store one more shared value$\alpha 12$. The total number of this value is fixed by its neighbors. In our scheme before deployment each node is preloaded with its ID and initial key, after deployment each node in the same cluster store one more value K_{CN}. Each sensor node is store just two values in its memory.

5.2 Energy Consumption

The energy consumption of the proposed scheme for transmit 100 packets is less than in [10] as showing in figure 2.

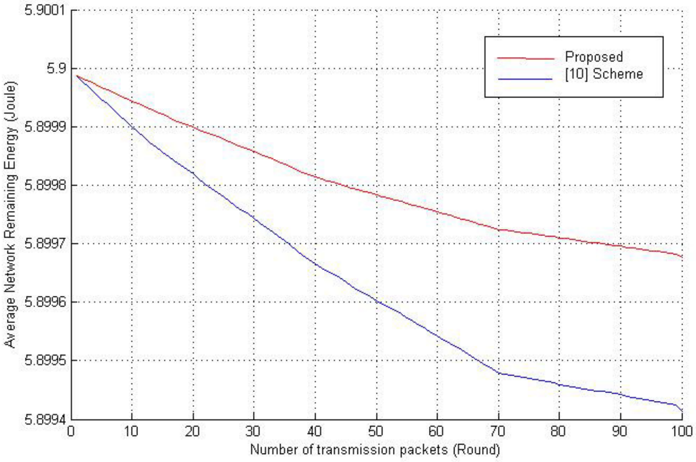

Figure 2. Energy Consumption for transmit 100 packets

Our scheme consumes less energy comparing with [10].

5.3 Time Consumption

The proposed scheme tacks less time to generate session keys and transmit the sensed date. Figure 3 shows the time for transmit 100 packets and compared with [10].

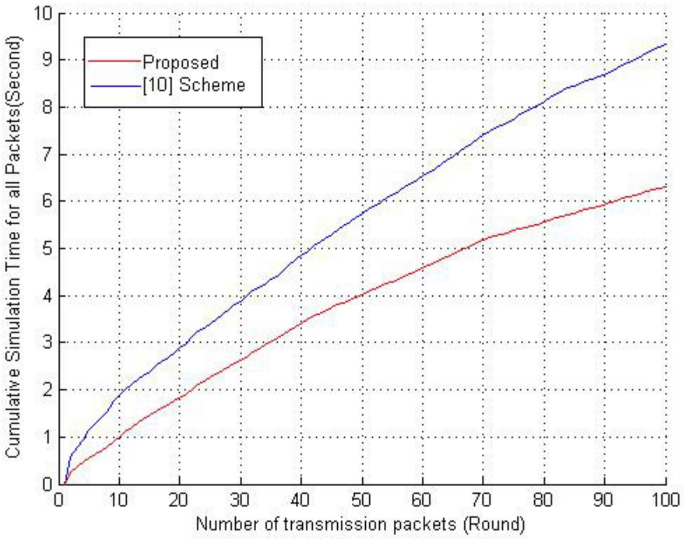

Figure 3. Time Consumption for transmit 100 packets

6. Conclusion

In this paper, a Lightweight Key Management Scheme (LKMS) for heterogeneous Wireless Sensor Networks was presented, that used symmetric cryptography only a hash function and XOR operation to establish a session key between any two nodes. Through performance evaluation, we reduce the storage overhead and extend the network lifetime by consuming less energy and take less time to establish secure communication among the nodes.

LKMS use shared value between nodes to establish symmetric session keys and update that value at regular interval to avoid node capture attack and to assure that only legal nodes can be communicated.

Simulation and analysis shows that LKMS has good energy efficient, less time consuming and low storage overhead than other similar schemes.

REFERENCES

[1] Ahmed A. Alkadhmawee and Songfeng Lu, "Prolonging the Network Lifetime Based on LPA-Star Algorithm and Fuzzy Logic in Wireless Sensor Network," World Congress on Intelligent Control and Automation (WCICA), IEEE, 2016

[2] Pawgasame, W., "A survey in adaptive hybrid wireless Sensor Network for military operations". IEEE, in Second Asian Conference Defence Technology (ACDT), pp. 78-83, 2016

[3] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," Journal of Network and Computer Applications, vol. 33, pp. 63-75, 2010.

[4] Omer K. Jasim, Safia Abbas and El-Sayed M. Horbaty, "Evolution of an Emerging Symmetric Quantum Cryptographic Algorithm", Journal of Information Security, Vol. 6, pp. 82-92, 2015

[5] Ayman T., Ayman K. and Ali C., "Authentication Schemes for Wireless Sensor Networks," in Mediterranean Electrotechnical Conference (MELECON), IEEE, pp. 367-372, 2014

[6] Khuraijam S. K andRadhika K R" A Novel Symmetric Key Encryption Algorithm Based on RC5 in Wireless Sensor Network" International Journal of Emerging Technology and Advanced Engineering , IJETAE, Volume 3, Issue 6, 2013.

[7] Ying, Z. and Pengfei, J. "An Efficient and Hybrid Key Management for Heterogeneous Wireless Sensor Networks" ,Control And Decision Conference (2014 CCDC),The 26th Chinese , ,IEEE, 2014

[8] V Vijayalakshmi, R Sharmila, and R Shalini. "Hierarchical key management scheme using hyper elliptic curve cryptography in wireless sensor networks". In Signal Processing, Communication and Networking (ICSCN), 2015 3rd International Conference on. IEEE, 2015

[9] Zhang, X., and Wsn, A. C. "An Efficient Key Management Scheme in Hierarchical Wireless Sensor Networks". 2015 International Conference on Computing, Communication And Security (ICCCS), IEEE,doi:10.1109/CCCS.2015.7374122, 2015

[10] Akansha Singh, Amit K. Awasthi and Karan Singh," Lightweight Multilevel Key Management Scheme for Large Scale Wireless Sensor Network",IEEE 2016