An Authenticated Key Management Scheme for Hierarchical Wireless Sensor Networks

Firdous Kausar[1], Ashraf Masood[1], and Sajid Hussain[2]

[1] College of Signals, National University of Science and Technology. Rawalpindi, Pakistan. firdous.imam@gmail.com, mashraf@comsat.net.pk

[2] Acadia University, Jodrey School of Computer Science, Nova Scotia, Canada sajid.hussain@acadiau.ca

Summary. Key Management is a critical security service in wireless sensor networks (WSNs). It is an essential cryptographic primitive upon which other security primitives are built. The most critical security requirements in WSNs include authentication and confidentiality. These security requirements can be provided by a key management but it is difficult due to the ad hoc nature, intermittent connectivity, and resource limitations of the sensor networks. In this paper we propose an authenticated key management (AKM) scheme for hierarchical networks based on the random key pre-distribution. Further, a secure cluster formation algorithm is proposed. The base station periodically refreshes the network key, which provides the following: a) the authenticated network communication, and b) a global and continuous authentication of each network entity. Multiple level of encryption is provided by using two keys: 1) a pair-wise shared key between nodes, and 2) a network key. The AKM scheme is more resilient to node capture as compared to other random key pre-distribution schemes. The proposed key management scheme can be applied for different routing and energy efficient data dissemination techniques for sensor networks.

Keywords: random key distribution, security, wireless sensor networks, authentication

1 Introduction

Due to recent advances in electronic industry, wireless sensors can be used in various ubiqutous and pervasive applications such as military, security, health-care[1, 2], industry automation, environmental and habitat monitoring [3, 4]. Wireless sensor networks (WSNs) consist of large number of low power nodes, with limited processing, communication, and storage resources [5]. Due to limited resources of WSNs, it is challenging to incorporate basic security functions, such as authentication, access control, data integrity, privacy, and key distribution. For instance, asymmetric cryptography such as RSA or Elliptic Curve cryptography (ECC) is unsuitable for most sensor architectures due to high energy consumption and increased code storage requirements. To avoid the use of asymmetric cryptography, several alternative approaches have been developed to perform key management on resource-constrained sensor networks, such as random key pre-distribution schemes, plain text key exchange schemes, and transitory master key schemes.

In WSNs, hierarchical clustering provides scalability, self-organization, and energy efficient data dissemination [6]. A number of cluster formation protocols have been proposed but most existing protocols assume benign environments, and are vulnerable to attacks from malicious nodes. In this chapter, we use an existing ring structure energy efficient clustering architecture (RECA) [7] to divide nodes into clusters. However, we modify its initial cluster formation algorithm by providing an authentication mechanism, so that no malicious node can take part into cluster formation process. The proposed authenticated key management (AKM) scheme enables only legitimate nodes to join the network.

The rest of the chapter is organized as follows: Section 2 discusses key distribution schemes, which are relevant to the proposed scheme. In Section 3, the AKM scheme is described in detail, such as nodes authentication prior to join a network, secure cluster-formation along with details about shared key discovery, and new node addition. Section 4 focuses on the performance and security analysis of the AKM scheme. Section 5 summarizes the major contributions of this work.

2 Related Work

There are many key management protocols which are proposed for WSN. Eschenauer and Gligor [8] propose a probabilistic key pre-distribution technique to bootstrap the initial trust between sensor nodes. The main idea is to have each sensor randomly pick a set of keys from a key pool before deployment. Then, in order to establish a pairwise key, two sensor nodes only need to identify the common keys that they share. Chan et al. further extended this idea and propose the q-composite key pre-distribution [9]. This approach allows two sensors to setup a pairwise key only when they share at least q common keys. Chan et al. also developed a random pairwise keys scheme to defeat node capture attacks.

Carman et al. [5] study the performance of a number of key management approaches in sensor network on different hardware platform. Perrig et al. [10] develop a security architecture for sensor networks (SPINS), where each sensor node shares a secret key with the base station. As two sensor nodes cannot directly establish a secret key, they can use base station as a trusted third party to setup the secret key.

In [11], Basagni et al. present a key management scheme to secure the communication by periodically updating the symmetric keys shared by all sensor nodes. However, this scheme assumes a tamper-resistant device to protect the key, which is not always available in sensor networks.

Blundo et al.[12] proposed several schemes which allow any group of t parties to compute a common key, while being secure against collusion between some of them. These schemes focus on saving communication costs while memory constraints are not placed on group members. When $t = 2$, one of these schemes is actually a special case of Blom's scheme [13].

Availability of some information on the sensor distribution in the field helps to improve the security of the key pre-distribution schemes. Some location-aware schemes are proposed in [14] and [15]. These techniques divide the target field into non-overlapping square areas and randomly deploy the sensors in every area. The exact location of a sensor in any area is unknown, but there is knowledge about the identity of sensors in every area. This information helps to eliminate the dependency of keys between nonadjacent cells.

Karlof and Wagner [16] identify security goals for routing in sensor networks and analyzed the vulnerabilities as well as the countermeasures for a number of existing routing protocols. Zhu et al. [17] give Localized Encryption and Authentication Protocol (LEAP), which is a complete key management framework for static WSNs that includes mechanisms for securing node-to-base station traffic, base station-to-nodes traffic, local broadcasts as well as node-to-node (pairwise) communications.

There is an active research for hierarchical clusters. For instance, Hu et al. [18] give a Secure Power-Efficient Clustered-Topology Routing Algorithm (SPECTRA), which integrates routing and key management to provide an energy efficient security and routing solution. Chan et al. [19] describe the problem of designing a clustered distributed sensor network (DSN) for specific scenarios, where the probability of node compromise in different deployment regions is known apriori. They use the apriori probability to design a variant of random key pre-distribution method that improves the resilience and reduces the compromised communications as compared to other related schmes. Oliveira et al. [20] propose SecLEACH that shows how a random key pre-distribution can be used for secure communication in hierarchical (cluster-based) protocols, such as LEACH [21].

3 The AKM Scheme

We propose authentication and key management scheme for hierarchical clusters in wireless sensor networks.

3.1 Terms and Assumptions

Some of the terms and assumptions that are needed for the proposed scheme are as follows:

Network Model

The WSN consists of a 3-tier architecture consisting of a base station, cluster heads, and cluster members. Base station is a secure and not prone to failure node. It has virtually unlimited computational, communication, and memory resources. Further, it is assumed that base station can transmit directly to every sensor node.

Sensor nodes, however, are battery-constrained and inexpensive nodes. They have limited communication, processing and memory storage resources. Each sensor node can act as a cluster head (CH) or a cluster member. The CH is chosen based on a RECA algorithm given below. A cluster member communicates directly with its cluster head (CH); there is no communication between sensors. In other words, there is 1-hop communication between a cluster member and the CH. Further, cluster heads can communicate with each other directly and to the base station. Figure 1 shows a WSN with three clusters. The cluster heads transmit directly to the base station and cluster members communicate with their cluster heads only. Further, there are a few isolated sensors that do not belong to any cluster and they communicate directly with the base station.

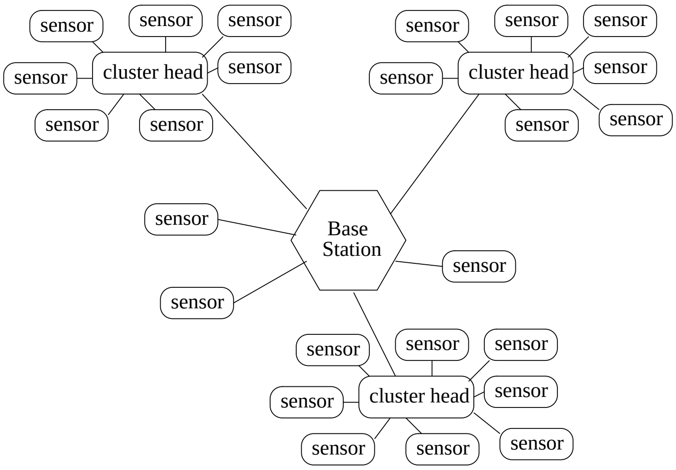

Fig. 1. Wireless Sensor Network

Threat Model

We assume that the adversary can access all the data (including keys) of a compromised sensor node. Further, an adversary can eavesdrop on the traffic, inject new messages, replay and change old messages, or spoof other identities.

Authentication Key (K_{Auth_i})

The authentication key is a unique pair-wise shared key between a sensor node and the base station. During initial setup phase (i.e. just after sensor deployment), sensor nodes and CHs must authenticate themselves with the BS using their corresponding authentication keys. The authentication key can only be used once in the lifetime of a node, and the key is deleted after authentication from base station.

Network Key (K_{Net})

The base station uses a network key to provide a secure broadcast for WSN. All messages transmitted by the base station are encrypted by the network key. Further, the base station refreshes the network key periodically.

Cluster Key ($K_{CH_i,A}$)

Each cluster member uses a unique pair-wise shared key with its cluster head. The cluster key provides confidential communication between a cluster member and its cluster head.

Security Level

Security level gives the probability that a randomly chosen link is not compromised when a node, which is not the either end of the link, is compromised.

Key Pool (P)

A key pool is a large pool of random symmetric keys which is stored at the base station. The base station assigns a particular number of keys to each sensor node, where keys are randomly chosen from the key pool without replacement. S gives the number of keys in the key pool.

Key Ring (R)

The keys which are assigned to each node from the key pool are called key ring. The number of keys in the ring R is given by m.

Unsupervised Node

The nodes which do not belong to any cluster are called unsupervised nodes. These nodes communicate directly with the base station.

For convenience, a summary of notations and symbols used in the paper are given in Table 1.

Notation	**Definition**
BS	Base Station
CH	Cluster Head
K_{Auth}	Authentication Key
K_{Net}	Network Key
P	A pool of keys
S	A number of keys in a key pool
R_X	Set of the keys in node X's key ring
RID_X	Set of key IDs in node X's key ring
m	A number of keys in a key ring
$K_{X,Y}$	A shared key between X and Y
$K_{CurrNet}$	Current Network Key
$K_{RefreshNet}$	Refreshed Network Key
id_A	Identity of node A
$E_K(m)$	An encryption of message m with key K
$nonce$	A random number String
$mac_K(msg)$	MAC calculated using key K
d_A	Data from node A
$F(d_i, d_j, \cdots, d_n)$	Data aggregation function
adv	Advertisement message announced by CH

Table 1. Symbol Definition

3.2 Key Pre-deployment Phase

First, base station generates a large pool of S keys and their ids prior to network deployment. Each node is then assigned a ring of m number of keys, which are drawn from the pool pseudorandomly [**?**] without replacement.

For each node X, we use a pseudorandom function (PRF) to generate its unique id called id_X, which is then used to seed a pseudorandom number generator (PRNG) of a large enough period to produce a sequence of m numbers $(n_1, n_2, \cdots, n_m)$. RID_X is the set of key ids assigned to X, which can be obtained by mapping each number in the sequence to its corresponding value modulus N, i.e. $n_i \ mod \ N$, where $i \in [1, m]$). Also prior to deployment, each node is assigned a unique pairwise key (K_{Auth_i}) shared with the BS.

3.3 Initial Setup Phase

During the initial setup phase, all sensor nodes and CHs are authenticated. A node is authenticated by using the latest network key. In order to get the latest network key, node A_i sends a request to the BS. The request consists of node's id, nonce, and MAC, where MAC is calculated by using that node's authentication key K_{Auth_i}, as shown in Figure 2. The BS authenticates the node by verifying the MAC using that node's authentication key associated with it's node ID. The BS replies to the node with the latest network key K_{Net}, which is encrypted with K_{Auth_i} of the requesting node and mark the

K_{Auth_i} of that node as *used* in its database. The node receives the message from BS and uses it K_{Auth_i} to decrypt the network key. Also, the node A_i deletes K_{Auth_i} from its memory after joining the network.

$A_i \Rightarrow$ BS:	$Msg\left(id_{A_i}, nonce, MAC_{K_{Auth_i}}(id_{A_i} \| nonce)\right)$
$BS \Rightarrow A_i$:	$E_{K_{Auth_i}}(K_Net)$

Fig. 2. Node authentication

3.4 Cluster Organization Phase

After authentication and receiving K_{Net} from base station, nodes enter into the cluster organization phase. Figure 3 shows the algorithm for initial cluster formation. The algorithm is executed on every node in the network. The *head* and *member* variables determine the cluster head and cluster member states respectively. Table 2 shows the possible states for a node. For a cluster head, the values of *head* and *member* are true and false respectively. However, for uninitialized node, both *head* and *member* are false. Further, if both *head* and *member* are true, it shows an erroneous state which is not possible in a typical case.

head	member	status
true	false	cluster head
false	true	cluster member
false	false	uninitialized
true	true	erroneous

Table 2. Possible nodes states

The γ determines the estimated number of nodes in a cluster, which can be estimated as $\gamma = \frac{N \times \pi \times r^2}{A}$, where N is the total number of nodes in a network, A is the network area, and r is the minimum transmission range.

If a node is uninitialized, it generates a random number α between 0 and 1. If α is less than a given threshold, $\min\left(\frac{2^i}{\gamma}, 1\right)$, the node acts as a cluster head and broadcasts an advertisement message consisting: id_{CH}, nonce, adv, $\text{mac}_{K_{Net}}(\text{id}_{CH} \| \text{nonce})$, where MAC is generated using latest K_{Net}. Upon receiving the CH advertisement message *adv*, the BS authenticates *adv* and generates a new key K_{BS,CH_i}, which is transmitted to the CH encrypting it with K_{Net}.

The nodes listen to the cluster head advertisements for a period t_{slot}. Upon receiving cluster head advertisement message *adv*, the node A_i does the following: authenticates the CH by verifying the MAC, using the latest K_{Net},

```
/* Initial State */
$\gamma$: the expected number of nodes in one cluster.
t$_{slot}$: duration of one time slot.
$head$: TRUE if the node is a cluster head; otherwise FALSE.
$member$: TRUE if the node is a cluster member; otherwise FALSE.

procedure InitCluster()
for i=0 to $\lceil log_2\gamma \rceil$ do
  if (!$head$) AND (!$member$) then
    generate $\alpha \in [0,1]$
    if $\alpha$ < min $\left(\frac{2^i}{\gamma}, 1\right)$ then
      advertise itself as a cluster head
      broadcast id$_{CH}$, nonce, adv, mac$_{K_{Net}}$(id$_{CH}\|$nonce)
      return
    end if
  end if
  listen for a duration of t$_{slot}$
  if (!$head$) AND ($adv_{rcvd} > 0$) then
    choose $id_{K_{CH,Ai}}, \forall\ id_{K_{CH,Ai}} \in (RID_{CH} \cap RID_{A_i})$
    send id$_{A_i}$,id$_{CH}$,$id_{K_{CH,Ai}}$,join_req,$mac_{K_s}(id_{A_i}\|id_{CH}\|id_{K_{CH,Ai}}\|nonce)$ to CH
    join the cluster using suitable RSSI and LQI values
  end if
end for
```

Fig. 3. Initial Cluster Formation Algorithm

b) computes the set of CH key ids (using the pseudorandom scheme described above), and c) identifies whether it shares a key $K_{CH,i}$ with that CH.

At the end of t_{slot}, the node chooses the CH with whom it shares a key and has the best received signal strength and link quality. Then, it sends a join request message to the selected CH; the join request is protected by MAC, using $K_{CH,i}$ and include the nonce from CH broadcast (to prevent replay attack),as well as the id of shared key ($K_{CH,i}$) chosen to protect this link (so that the receiving CH knows which key to use to verify the MAC).

Figure 4 shows the messages transferred between sensor nodes, CHs, and the BS. First, a CH broadcasts the advertisement. Second, a node A_i joins a cluster based on the received signal strength and shared key. Third, a node transmits its data to the CH. Finally, the CH sends the aggregated data to the BS.

Node to CH communications are protected using two keys: latest K_{Net} and $K_{CH,i}$. First, the data is encrypted using the latest K_{Net}. Second, the encrypted data is again encrypted using $K_{CH,i}$. Then, the data is forwarded to the CH. The CHs decrypt the received data, perform data aggregation, and sends the aggregated result to BS, where the result is first encrypted with K_{Net}, and then with K_{BS,CH_i}.

```
1: $CH \Rightarrow * : id_{CH}, nonce, adv, mac_{K_{Net}}(id_{CH} \| nonce)$
   /* $A_i$ chooses $id_{K_{CH,Ai}}$, $\forall\ id_{K_{CH,Ai}} \in (RID_{CH} \cap RID_{A_i})$ */
2: $A_i \Rightarrow CH : id_{A_i}, id_{CH}, id_{K_{CH,A_i}}, join\_req, mac_{K_{CH,Ai}}(id_{A_i} \| id_{CH} \| id_{K_{CH,Ai}} \| nonce)$
3: $A_i \Rightarrow CH : id_{A_i}, id_{CH}, E_{K_{Net}}(E_{K_{CH,Ai}}(d_{A_i}))$
4: $CH \Rightarrow BS : id_{CH}, E_{K_{Net}}(E_{K_{BS,CH}}(F(d_i, d_j, \cdots, d_n)))$
```

Fig. 4. Messages Transferred between sensor nodes, CHs, and the BS.

Unsupervised Nodes

At the end of clustering process, it is expected that a fraction of the ordinary nodes will not be matched with a CH because of key sharing constraints; these nodes are called unsupervised nodes. The unsupervised nodes will communicate directly with the BS. The unsupervised node X sends a special message $(id_X, req, nonce, mac_{K_{Net}}(id_X \| req \| nonce))$ to BS requesting a pair-wise shared key. The BS generates a key $K_{BS,X}$ and sends it to the corresponding unsupervised node encrypting it with K_{Net}. The node X decrypts the message and will use the key $K_{BS,X}$ for future communication with the BS. Further, every message sent to the BS is first encrypted with K_{Net} and then with $K_{BS,X}$. The number of the unsupervised nodes depends on the following: the size of key pool, the size of the key ring, and the number of CHs.

Refreshing Network Key

All nodes in the network are continuously and periodically authenticated. This is achieved through the periodic refreshing of the K_{Net}, which is refreshed periodically. The network key is valid only for an epoch. An epoch is defined as a period of time that is less than the predicted time required for node compromise. The epoch time is dependent on the network environment. After that epoch, BS generates a new network key $K_{RefreshNet}$ and broadcasts it by encrypting with the current network key $K_{CurrNet}$. The nodes in the network receive the broadcast message, decrypt it using the current network key and get the refreshed network key. Now the refreshed network key will be used as the current network key, i.e. as a K_{Net} in the network.

3.5 New Node Addition

When a new node say u joins a network, first it gets current network key from BS after going through the authentication process. Then, it broadcasts hello message consisting of its id, a nonce and MAC calculated on these two values using K_{Net}. The CHs, which receive hello message, authenticate the node by verifying the MAC, and then send the advertisement message *adv*. The node u verifies the MAC of receiving advertisement messages and computes the set of CH key ids (using the pseudo random scheme described above), chooses the closest CH with whom it share a key $K_{CH,u}$, and sends a join request message, which is protected by MAC using $K_{CH,u}$, and include the nonce from CH broadcast (to prevent replay attack), as well as the id of shared key ($\mathrm{id}_{K_{CH,u}}$) chosen to protect this link as described above.

4 Analysis

This section analyzes the proposed scheme and compares it with the other related approaches. Given a WSN, the amount of storage allocated for keys in each node is likely to be a preset constraint, which makes the size of the key ring m a fixed parameter. Once m is set, the choice of S will impact the security level and probability of key sharing among nodes in the network.

Given a (S,m)-network, where each node is assigned m keys from a key pool of size S, the security level sl can be defined as follows:

$$sl = 1 - \frac{m}{S} \tag{1}$$

which gives the probability that a randomly chosen link is not compromised, when a node that is not either end of the link is compromised. For a fixed m, sl is proportional to S, as shown in Figure 5.

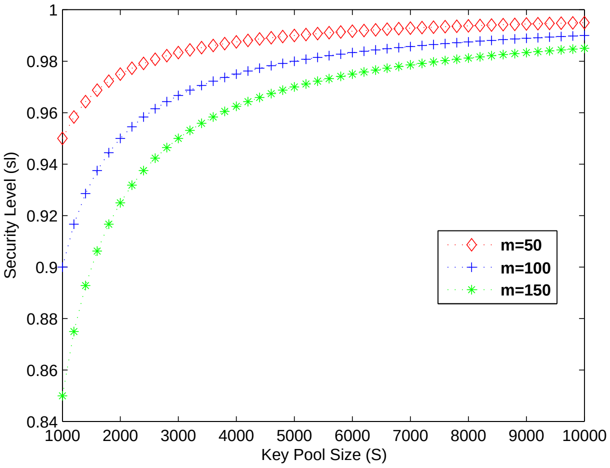

Fig. 5. Security Level

Given any two nodes in a (S,m)-network, the probability P_s that they will share a key is given by:

$$P_s = 1 - \overline{P_s} \tag{2}$$

where $\overline{P_s}$ is the probability that they will not share a key, is given by:

$$\overline{P_s} = \frac{[(S-m)!]^2}{S!(S-2m)!} \tag{3}$$

Figure 6 shows the variation in probability of key sharing with respect to key pool size S and the ring size m. For a fixed m, P_s decreases as the value of S is increased. However, for fixed S, the P_s increases as the value of m increases.

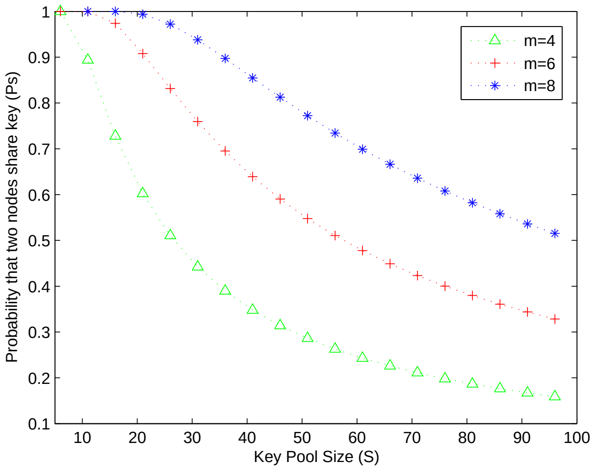

Fig. 6. Probability of key sharing

In our scheme, only a fraction of CHs is probabilistically accessible by an ordinary node. P_s and the number of CHs h in the network can also determine the expected number of unsupervised nodes, i.e. the probability that an ordinary node will be unsupervised. Given P_s and h, the probability of the number of unsupervised nodes is given by:

$$P_0 = (1 - P_s)^h \tag{4}$$

In a network with N nodes, it is then expected that $N \times P_0$ nodes will be unsupervised. Figure 7 shows fraction of unsupervised nodes as a function of h under different values of sl. As h increases, the number of unsupervised nodes decrease rapidly. Further, as security level sl increases, the number of unsupervised nodes also increase.

Table 3 gives probability P_s of key sharing as a function of security level sl and key pool size S. For a fixed value of S, as sl increases, the value of P_s decreases accordingly. However, for fixed sl, as the value of S increases, the value of P_s also increases.

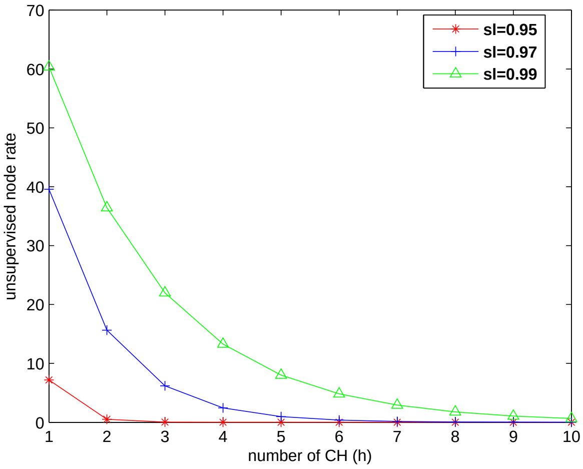

Fig. 7. Unsupervised Node Rate

	Security Level (**sl**)		
S	**0.95**	**0.97**	**0.99**
1000	0.9280	0.6044	0.0960
2000	0.9948	0.8436	0.1828
5000	0.9999	0.9903	0.3964

Table 3. Probability $P_s$of key sharing as a function of security level *sl* and key pool size S.

4.1 Security Analysis

In key distribution schemes, resiliency against node capture measures how much of the network (its communication links) is compromised when a node is compromised. It is a critical performance measure that gauges the robustness of a solution. Our scheme as compared to other random key pre-distribution schemes is more resilient against node capture provided that compromised node re-enter into the network after the epoch in which network key has been refreshed. If a node gets compromised, it is possible for the adversary to know all the keys stored at that node. If we expect that the attacker requires a fixed amount of time to compromise the node, the network key would have changed to new one before the attacker could use the compromise keys.

The tampered node cannot rejoin the network, because K_{Auth_i} is used only once during the life time of a node at the time of joining the network, in order to get the current network key. The advantage of deleting K_{Auth_i} (after authentication) is that if a node is being captured by an adversary, and adversary wants to re-enter the node into the network after the time period in which the K_{Net} has been refreshed, it will fail to get the current K_{Net} because its K_{Auth} has been deleted. But if adversary re-enters the compromised node into the network before refreshing the current K_{Net}, the resiliency of our proposed schemed will be same as given in Eschenauer et al. [8].

Our scheme provides secure cluster formation process and prevents the malicious nodes to join the network. As shows in initial cluster formation algorithm, node to CH authentication is achieved by verifying the MAC calculated using current K_{Net} by the CH in the advertisement message. It also provides CH to member authentication by verifying the MAC calculated using $K_{CH,i}$ by cluster members in the join_req message.

The global authentication is achieved by periodically refreshing the network key; this feature allows every entity in the network to be confirmed or authenticated continuously and reduces the chances of compromise because compromise has to occur before the system is authenticated again.

Confidentiality and authentication at every step in the network is provided by using two keys. First, any message passed within the network is encrypted with the network key, and then by the shared key. Therefore, in order to participate within the network, the current network key must be known. If a node is lacking the current network key, no information can be sent or received. Multiple keys are important in the network because they make compromise exponentially difficult. Not only must a compromised node have knowledge of two different keys (i.e., network key and shared key), but it must also know exactly when to use them.

5 Conclusion

Key establishment, a fundamental building block in cryptography, is defined to be any process whereby a shared secret key becomes available to two or more parties, for subsequent cryptographic use. This paper proposes a variant of random key pre-distribution scheme for bootstrapping the clustered WSNs. Clustering approaches have been found useful in providing scalable data aggregation, security and coding for large scale WSNs. In our proposed scheme, we show that it is more resilient to node capture as compared to other random key distribution schemes provided that compromised node rejoin the network after the epoch in which network key has been refreshed. It also provides global and continuous authentication of nodes in the network by periodically refreshing the network key. Finally, it provides multiple level of encryption by using more then one encryption key and also secure cluster formation algorithm.

References

1. Gao, T., Greenspan, D., Welsh, M., Juang, R.R., Alm, A.: Vital signs monitoring and patient tracking over a wireless network. In: The 27th Annual International Conference of the IEEE EMBS, Shanghai, China (2005)
2. Gu, L., Jia, D., Vicaire, P., Yan, T., Luo, L., Tirumala, A., Cao, Q., He, T., Stankovic, J.A., Abdelzaher, T., Krogh, B.H.: Lightweight detection and classification for wireless sensor networks in realistic environments. In: The 3rd ACM Conference on Embedded Networked Sensor Systems, San Diego, USA (2005)
3. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine (2002)
4. Kahn, J., Katz, R., Pister, K.: Next century challenges: Mobile networking for smart dust. In: The ACM International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, USA (1999)
5. Carman, D.W., Kruus, P.S., Matt, B.J.: Constrains and approaches for distributed sensor network security. In: Technical report, NAI Labs. (2000)
6. Zhao, F., Guibas, L.: Wireless sensor networks. Elsevier Inc (2004) 23–24
7. Li, G., Znati, T.: Reca: a ring-structured energy efficient cluster architecture for wireless sensor networks. International Journal of Sensor Networks **2** (2007) 34–43
8. Eschenauer, L., Gligor, V.D.: A key management scheme for distributed sensor networks. In: ACM CCS. (2002)
9. Chan, H., Perrig, A., Song, D.: Random key pre-distribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy. (2003) 197–213
10. Perrig, A., Szewczyk, R., Tygar, J., Victorwen, Culler, D.E.: Spins: Security protocols for sensor networks. In: Seventh Annual Int'l Conf. on Mobile Computing and Networks. (2001)
11. Basagni, S., Herrin, K., Bruschi, D., Rosti, E.: Secure pebblenets. In: ACM Int'l Symp. on Mobile ad hoc networking and computing. (2001) 156–163
12. Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, London, UK, Springer-Verlag (1993) 471–486
13. Blom, R.: An optimal class of symmetric key generation systems. In: Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques, New York, NY, USA, Springer-Verlag New York, Inc. (1985) 335–338
14. Liu, D., Ning, P.: Location-based pairwise key establishments for static sensor networks. In: SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, New York, NY, USA, ACM Press (2003) 72–82
15. Wadaa, A., Olariu, S., Wilson, L., Eltoweissy, M.: Scalable cryptographic key management in wireless sensor networks. In: ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04), Washington, DC, USA, IEEE Computer Society (2004) 796–802
16. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. In: IEEE Int'l Workshop on Sensor Network Protocols and Applications. (2003)

17. Zhu, S., Setia, S., Jajodia, S.: Leap: Efficient security mechanisms for large-scale distributed sensor networks. In: ACM Conference on Computer and Communications Security. (2003) 62–72
18. Hu, F., Siddiqui, W., Cao, X.: Spectra: Secure power-efficient clustered-topology routing algorithm in large-scale wireless micro-sensor networks. International Journal of Information Technology **11** (2005)
19. Chan, S.P., Poovendran, R., Sun, M.T.: A key management scheme in distributed sensor networks using attack probabilities. In: IEEE Globecom. (2005)
20. Oliveira, L.B., Wong, H.C., Bern, M., Dahab, R., Loureiro, A.A.F.: Sec leach: A random key distribution solution for securing clustered sensor networks. In: 5th IEEE international symposium on network computing and applications. (2006) 145–154
21. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: IEEE Hawaii Int. Conf. on System Sciences. (2000) 4–7