An Efficient and Secure Key Management Scheme for Hierarchical Wireless Sensor Networks

Abdoulaye Diop, Yue Qi, Qin Wang, and Shariq Hussain

***Abstract*—Key management in wireless sensor network is a complex task due to its nature of environment. Wireless sensor network comprise of large number of sensor nodes with different hardware abilities and functions. Due to the limited memory resources and energy constraints, complex security algorithms cannot be used in sensor networks. Therefore, an energy efficient key management scheme is necessary to mitigate the security risks. In this paper, we present an Efficient and Secure Key Management Scheme for Hierarchical Wireless Sensor Network (ESKMS). The proposed technique distributes the keys within a cluster efficiently and updates the pre-deployed keys to mitigate the node compromise attack. We also provide a detailed security analysis of our ESKMS protocol and show its advantages in avoiding different type of attacks from malicious nodes. Finally, using NS-2 simulator, the results shows that ESKMS is more energy efficient and provides a longer network lifetime compared to the existing key management schemes.**

***Index Terms*—Wireless sensor network, key management, security, attacks, cluster.**

I. Introduction

The tremendous development in the electronics technology lead the way to development of micro-electronics thus enabling production of small chips and micro devices. The communication technology is being reformed due the design and development of micro devices and hence enabled the design and development of wireless sensor networks (WSNs) with low cost, low energy consumption and high utilization. WSNs have lot of applications in military, health and other industries. Because of the characteristics of WSNs, sensor nodes are usually characterized by limited power, low bandwidth, memory size and limited energy [1].

Due to the scalability and energy efficiency characteristics, many routing protocols for cluster-based WSNs proposed by researchers [2]. In cluster base networks, clusters are formed by organizing nodes. Further, cluster heads (CHs) are responsible for relaying of messages from ordinary nodes to the Base Station (BS). CHs can communicate directly with the BS, can be anywhere in the network, and change per interval, which also improves network's energy efficiency [2].

Most routing protocols for WSNs are vulnerable to a number of security threats [3]. Attacks involving CHs are the most damaging. As WSNs are typically composed of sensor nodes, so capturing a sensor node can enable the intruder to become a CH and further propagate attacks such as sinkhole and selective forwarding. This could result in disruption of entire network. Hence, it is essential to establish encryption keys among sensor nodes, thus restricting the security impact of a node compromise [4]. The area of key management is one security aspect that receives a great deal of attention in cluster based WSNs. An overview of different key management techniques for different types of network architecture are presented in the study [1]. Keys which are necessary for security and efficiency requirements of WSNs are listed in Table I.

TABLE I: Design Requirement Of Key Pre-Distribution Scheme.

S. No	Requirement Type	Requirements
1.	Security Requirement	Authentication Secrecy Resilience against node capture Resistance against node replication Compromised node revocation Fresh node addition
2.	Efficiency Requirement	Network connectivity Maximum supported network size Minimum memory storage Low computational overhead Low communication overhead

However, due to the resource constraints of wireless sensors, public-key based cryptographic algorithms like RSA and Diffie-Hellman are too complicated and energy-consuming for WSNs.

In this paper, we proposed an Efficient and Secure Key Management Scheme (ESKMS) for Hierarchical Wireless Sensor Networks (HWSNs) to distribute the keys within a cluster and update the keys at regular interval to avoid node capturing problem. We use one way hash function, data encryption and message authentication code (mac) to authenticates the communicating nodes and update the pre-deployed network keys.

In fact, if an intruder manages to capture a node, then a encryption mechanism should be present to restrict the access of intruder to the message history of node. Therefore, after key pre-distribution and sensor deployment, a key updating scheme should be used to update pre-deployed keys regularly [5]. This procedure ensures that intruders cannot acquire the keys easily, and also avoid a different type of attacks from malicious nodes.

The rest of the paper is organized as follows. Section II describes the related work of security for WSN. Section III explains the network model used in this work and some assumptions about security. Section IV explains the proposed

Manuscript received July 25, 2012; revised September 5, 2012. This work was supported by the High-tech Research and Development Program of China (Grant No. 2011AA040101-3).

The authors are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China (e-mail: b20100556@xs.ustb.edu.cn, qiyuee@ustb.edu.cn, wangqin@ies.ustb.edu.cn, b20100557@xs.ustb.edu.cn).

key management scheme in details. Section V presents the security analysis and the simulation results of the proposed key management scheme. Finally in section VI, we present our concluding remarks and future work.

II. Related Work

HWSNs is widely one of the main research areas in wireless sensor networks and behave better in performance and reliability than traditional flat wireless sensor networks (FSNs). Fig. 1 shows two kinds of architectures for WSNs.

Various key distribution and management schemes have been proposed in wireless sensor networks. The first key pre-distribution scheme was investigated by Eschenauer and Gligor [6]. They suggested a probabilistic key pre-distribution technique to bootstrap the initial trust between sensor nodes. In this approach, a large size symmetric key pool P is generated first. Before deployment, each sensor node's memory is preloaded with a set of randomly selected keys from the key pool P. Then, in order to establish a pair-wise key, two sensor nodes only need to identify the common keys that they share. The main problem of this scheme is it cannot provide sufficient security when the number of compromised nodes increases. Because of the low-cost hardware, wireless sensors are not tamper resistant devices. If a sensor node is captured, all its stored cryptographic information can be easily extracted by the adversary. To improve the network resilience against node capture attacks, Chan et al. further extended this idea and propose the q-composite key pre-distribution [7]. This approach allows two sensors to setup a pair-wise key only when they share at least q common keys. Chan et al. also developed a random pair-wise keys scheme to defeat node capture attacks. Basagni et al. [8] presented 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. Zhu et al. [9] give Localized Encryption and Authentication Protocol (LEAP), a proposed scheme based on local distribution of keys among nodes in a neighborhood.

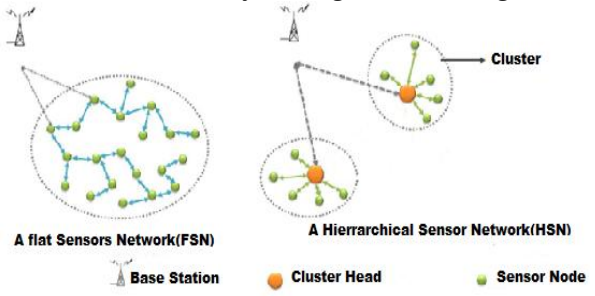

Fig. 1.Wireless sensors networks architectures.

LEAP establishes four types of keys that must be stored in each sensor, and is rather efficient for flat networks where nodes interact with a rather static set of neighbors. The main drawback of this proposition is compromise of the initial key allows an adversary to deduce all the pair-wise keys installed in the network.

In Improved Key Distribution Mechanism (IKDM) [10], the bivariate polynomial key pre-distribution scheme has been introduced. Only two pair-wise keys are pre-loaded in each sensor node to reduce the key storage overhead. One is for secure communication with the sink node, randomly initialized by the KDS and the second is for communication with the physical cluster head. IKDM scheme assume fixed cluster heads. One weakness of this approach is that once a cluster head is captured, all the keys stored in sensor nodes in that cluster will be compromised. Therefore, it is required either to replace the sensor nodes in a cluster or replace a compromised cluster head in that cluster.

There are some secure routing protocols based on LEACH (Low-Energy Adaptive Clustering Hierarchy) protocol [2]. Addition of security to LEACH protocol is presented by Heinzelman et al. [2]. Some secure routing protocols proposed, such as SecLEACH [11] and GS-LEACH [12]. SecLEACH show how a random key pre-distribution can be used for secure communication in hierarchical (cluster-based) protocols, such as LEACH.

However, SecLEACH and GS-LEACH still have security vulnerabilities caused by random key pre-distribution scheme and nature characteristics of LEACH. Thus, we found that SecLEACH and GS-LEACH are vulnerable to key collision attacks and do not provide full connectivity.

III. Network Model

In this section, we focus on hierarchical structure of sensor network as illustrated in Fig. 2. In our network model, we consider that there are:

- The BS is a control center used to connect the WSN with external network and for processing the sensed data. Further, it is assumed that the base station has unlimited computational, communication, and memory resources and it is considered trustworthy and it can also transmit directly to every sensor node.
- Sensors nodes collect information of surrounding environment and transmit them to the cluster.
- Cluster heads responsible for the coordination, the data retransfer and the management of all the nodes in the cluster.

We assume that WSNs are homogeneous and symmetric. Node position is random in sensor field. Sensor nodes keep stationary after deployment during the network operation.

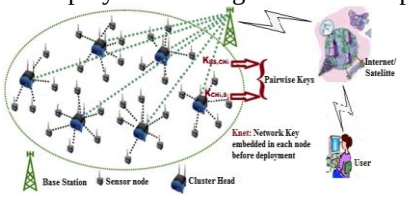

Fig. 2. Hierarchical wireless sensor network architecture.

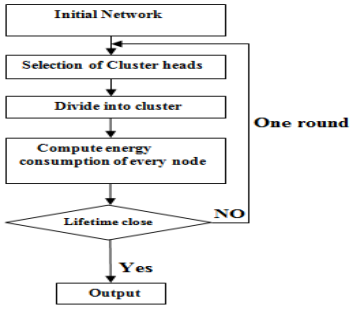

Fig. 3. Flow chart for LEACH protocol.

Note that, CH is responsible for processing of the data in cluster and transmission to BS, therefore it has relatively large energy consumption and must be replaced periodically to balance the energy cost. In our scheme, we use Low-Energy Adaptive Clustering Hierarchy (LEACH) [2] to randomly choose CHs. Sensors nodes choose their leader based on some parameters such as the strongest signal received from a CH [2]. As shown in Fig. 2, there is no communication between sensors nodes. After certain time interval new nodes are selected as CH to reduce the energy consumption of a CH. Rotating CHs have the advantage of averaging energy consumption among sensor nodes [2]. Fig. 3 shows the entire process of Leach protocol.

A. Security Assumptions

We make the following reasonable assumptions as already employed in most of the current sensor network security schemes:

- Each sensor has a unique id with enough length to distinguish between them.
- BS has a node member table of node id. If a node adds to network, its id adds to node member table.
- BS has authentication system for any node in the network [13].
- All CH in the network can reach the BS.
- We assume that an adversary need at least time Tcapture to compromise a node.
- Each exchanged message has a timestamp called "N" that guarantee the freshness of information.

Descriptions of the notations used in the proposed key management technique are listed in Table II.

TABLE II: NOTATION DESCRIPTION

S. No	Notation	Description
1.	idSNi:	Identification Number of node i
2.	idCHi :	Identification Cluster Head i
3.	idBS :	Identification Base Station
4.	K_{Net}:	Network-Key, embedded in each sensor node before deployment
5.	$K_{BS,CHi}$:	Pair-wise key shared between the Base Station (BS) and the Cluster Heads (CH)
6.	$K_{CHi,S}$:	Pair-wise key shared between sensor nodes and the CH that form the same cluster
7.	EK(M) :	Encryption of message M with key K
8.	V :	An array of node ids
9.	H() :	One-way hash function
10.	mac K (M):	The message authentication code of message M using key K
11.	Tcapture	Time need to capture a node
12.	$\oplus$	Bit wise XOR operation

IV. The Proposed Hierarchical Key Management Scheme

In our key management scheme, the level of security uses two kinds of keys:

Network Key (K_{Net}): This is a globally shared key that is used by all nodes and BS for encrypting messages that are broadcasted to all nodes in the sensor network. All messages transmitted by the base station are encrypted through the network key. This key is also used in cluster formation. Further, the base station refreshes the network key periodically.

Pair-wise keys: Our scheme guarantees that two communicating parties can establish unique pair-wise keys between them:

- KCHi,S shared between cluster heads and cluster member from the same cluster, and is used to authenticate and secure the communication between them. KCHi,S provides confidential communication between a cluster member and its cluster head.
- KBS,CHi shared between the BS and the CHs, is used to authenticate and secure the communication between BS and CH.

During the initialization phase and the cluster formation, pair-wise keys are set dynamically. In fact, Network Keys (K_{Net}) are programmed into the memory of the sensor nodes just before they are being deployed. Note that the current network key is valid only for a limited period. So it is essential to update the network key K_{Net} periodically. The proposed model is divided into five phases: (1) Key pre-distribution phase (2) Pair-wise keys establishment (3) Data transmission phase (4) Key updating phase and (5) Re-clustering phase.

A. Key Pre-distribution Phase

Due to the resource constraints of wireless sensors, the best key distribution method is preloading the secret keys into sensors before they are deployed [5]. Similarly, some secret information needs to be pre-loaded into sensor nodes before they are deployed. In our proposed scheme, sensor nodes are preloaded each with one unique secret key, shared with the BS. Sensor nodes must authenticate themselves with the BS using their corresponding unique keys. During this phase, the BS generates K_{Net} and loads each node with this key. The K_{Net} can be seen as the network key and will be used during the cluster formation phase. Note that all members should prove their validity to the sink. So for each node, a unique key Ku is used to authenticate the own node, shared with the sink and is deleted after the first round.

B. Pair-wise Key Establishment

Shared pair-wise key ($K_{BS,CHi}$) between CH and BS: After the deployment, the BS needs to establish pair-wise keys with each CHs to secure the communication between them. The BS generates an array V of all sensors nodes idSNi in the network. After deployment, some nodes are randomly selected as CH. The BS first using the network key K_{Net} encrypts a threshold value T(n), generates a mac and broadcasts these information and a nonce (number used once) to all sensor nodes. Node generates a random number R between 0 and 1. If R is less than a given threshold T(n), the node acts as a cluster head. When a node SNi become CH first time, it sends an authentication packet to BS by inserting its id and encrypting message using its network key K_{Net}.

idCHi , idBS	E K_{Net}(M|N)	mac K_{Net} (M|N)

where mac is generated using K_{Net}, N is the timestamp and M is the message of cluster head.

$$M= (idCHi|idBS| K_{Net})$$

Upon receiving the CH information M, the BS authenticates M and computes a new key KBS,CHi by using a keyed one-way hash function H_K(val).

$$K_{BS,CHi} = HK_{Net} (V[idCHi] + V[idBS])$$

After that, BS encrypts M and $K_{BS,CHi}$ using network key K_{Net} and sends it to CHi.

idCHi , idBS	EK_{Net}(M|N| KBS,CHi)	mac$K_{BS,CHi}$(M|N)

Intra-cluster pair-wise key establishment ($K_{CHi,Si}$): Each CH need to establish a pair-wise key with its cluster member SNi. All the communication between CH and SNi is encrypted by the established pair-wise key $K_{CHi,Si}$ to achieve communication security. The cluster pair-wise key establishment phase can be briefly described as follows:

First, Each CH broadcasts an advertisement message M using its network key K_{Net}, its idCHi and a timestamp N to avoid replay attack.

idCHi	EK_{Net}(M|N)	mac K_{Net} (M|N)

Node SNi authenticates the CH by verifying the mac, using the network key KNet. A node SNi joins a cluster based on the received signal strength. The node chooses the CH which has the best received signal strength. Then, for membership of this cluster, a node generates a message M as follows:

$$M= idSNi|idCHi|K_{Net}$$

Now node Ai encrypt the message M using network key K_{Net}, include the timestamp N and sends the encrypted message to the selected CHi.

idSNi , idCHi	E K_{Net} (M|N)	mac K_{Net} (M|N)

After that, CHi sends the identity list (idList) of each simple node member of the cluster to the BS.

idCHi , idBS	$EK_{BS,CHi}$ (M|N | idList)	$K_{BS,CHi}$ (M|N)

where idList = {idSN1,idSN2,…… ,idSNk-1}, k is the number of node in the cluster and M is the cluster head message. Finally, the BS computes the cluster's key $K_{CHi,Si}$ using a one-way hash function and the pair-wise key $K_{BS,CHi}$. The BS sends the cluster's pair-wise key to the CH.

idBS,idCHi	$EK_{BS,CHi}$ (M|N| KCHi,Si)	mac$K_{BS,CHi}$(M|N)

The CH transmits the intra-cluster pair-wise key to sensors nodes SNs.

idCHi	EK_{Net} (M|N | KCHi,Si)	mac K_{Net} (M|N)

C. Data Transmission Phase

This phase mainly consist of two distinct steps in hierarchical model of sensor network. In first step, member nodes send their sense data to their CH. A member node encrypts data packets using $K_{CHi,Si}$. The data packets format is as follows:

idSNi,idCH	E $K_{CH,Si}$ (M)	mac $K_{CHi,Si}$ (M)

where M is the sense data, $EK_{CH,Si}$(M) is the encrypted message and mac$K_{CH,Si}$ (M) is the authenticate message. When CH sends data packets to BS for processing, it encrypts the message using the pair-wise key $K_{BS,CHi}$ and insert its idCH and encrypted message into the data packet.

idCH	$EK_{BS,CHi}$ (H(M1, Mj , ….. , Mn))

D. Key Updating Phase

To reduce the risk of node capture attacks, it is essential to update the network key K_{Net} [7]. Hence the network key K_{Net} of a node is updated periodically. The network key is valid only for a limited time period that is less than the predicted time required for node compromise (Tcapture). That period of time is dependent on the network environment. After that period, BS generates a new network key K_{Net} and broadcasts it by encrypting with the current network key K_{Net}. The nodes in the network receive the broadcast message, decrypt it using the current network key and get the new network key.

E. Re-Clustering of Sensor Network

In this proposed model of key management technique, we consider that cluster heads are rotated after certain time interval [2], and all nodes get a chance to be a cluster head equal number of times. This approach allows balancing the energy consumption among all nodes in the network.

BS broadcast a packet to all CHs at the end of cluster duration to erase its member table. New cluster head make a table of its member node when a new cluster goes on, as described in intra pair-wise key establishment, and forward it to the BS and continue its operation.

V. Security Analysis and Simulation and Results

In this section, we evaluate the security properties and network performance of our ESKMS and we compare it to some of existing solutions.

A. Security Analysis

In our ESKMS approach, we show that our key management technique provides different types of security services for the communications. Before transmission of the message, encryption is performed to secure the transmission with the help of hash function. ESKMS also provide freshness using time interval, time-stamps and nonce.

During the initialization phase, the BS encrypts the threshold value T(n) with the network key K_{Net}. Only legitimate nodes that own the network key can decrypt this message. Note that 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 compromised keys. ESKMS provides secure cluster formation process and prevents the malicious nodes to join the network. It also provides CH to authenticate members by verifying the mac calculated using $K_{CHi,Si}$ by cluster members in the join

request message. To prevent a malicious node to attempt pair-wise key establishment, message encryption and mac are used, then malicious nodes will not be authenticated by the BS. The mac provides authentication of the BS and the integrity of the received key. This secure mechanism enables only the participation of safe nodes during the clusters formation.

The sensors nodes authentication is achieved by periodically updating the network key; this feature allows every entity in the network to be confirmed or authenticated continuously and reduces the chances of compromise. Since the encrypted message and the mac include the nonce, we argue that all messages are not out to date. We guarantee a freshness of messages exchanged in the network. ESKMS as compared to other key pre-distribution schemes like GS-LEACH and SecLEACH, based on LEACH and random key pre-distribution provide efficient security. SecLEACH and GS-LEACH have security vulnerabilities caused by random key pre-distribution. Note that in these schemes, when cluster head broadcasts a message, these protocols do not provide broadcast authentication. Also, SecLEACH and GS-LEACH are vulnerable to some attacks caused by node compromising and do not provide full connectivity among sensor nodes. The ESKMS approach provides authentication of not only a cluster head, but also all cluster members. Simultaneously, the proposed scheme provides broadcast authentication when selecting a cluster head among sensor nodes and full connectivity among sensor nodes. Also, in the approaches based on probabilistic key distribution like SecLEACH, the number of keys follows the number of nodes. Then generate a lot of messages and require much more memory space. Contrary to this, in our scheme the number of keys does not follow the number of sensor nodes; therefore it is suitable for large WSNs. Security comparison is presented in Table III.

TABLE III: SECURITY COMPARISON

Protocol	Connectivity	Prevention of Node Compromise	Energy efficiency
GS-LEACH	Medium	X	Good
Sec-LEACH	Medium	X	Medium
ESKMS	Full	△	Good

Further, ESKMS also provides confidentiality, freshness, integrity and almost full connectivity during clustering. ESKMS satisfies general security requirements, such as confidentiality with encryption, message integrity with mac, node authentication as mentioned before, and message freshness with nonce. Besides providing security, our scheme has high energy efficiency.

B. Simulation Results

The proposed ESKMS is evaluated through the network simulator NS-2 [14]. We consider a random network of 250 sensor nodes deployed in a 100m × 100m area, with a fixed BS located near the sensing. The BS has unlimited energy. The number of chosen CH is fixed to 10% for one interval [2]. Simulation time for every simulation was 10 minutes and the number of attackers from 10 to 20 attackers, 512 bytes for the packet size. In order, to evaluate the performance of the security overhead in our protocol, we consider two metrics: the energy consumption and the memory storage.

1) Energy Consumption: We compare the proposed protocol with LEACH [2] to determine the benefits of ESKMS in terms of energy consumption. We measured the average energy when the rekeying protocol was performed for the periodic key update. Fig. 4 shows the average energy consumption of sensor nodes following different network size.

It is observed that with the increase number of nodes from 50 nodes to 250 nodes and the number of attackers from 10 to 20, the energy consumption of LEACH protocol is slightly lower by 1.55% to 1.05% when compared with ESKMS because of the communication overhead. We can notice that the gap between ESKMS and LEACH is extremely low and practically identical.

Therefore, it could provide energy efficient technique of establishing shared-key, and could prolong the lifetime of the network, thus will increase the security performance. Fig. 4 also shows that in LEACH as well as in ESKMS, the energy of sensor nodes remains almost unchanged for all network sizes. This result was expected because in our model, cluster members communicate only with the cluster head, each ordinary node sends one message and receives one message.

Fig. 5 shows the average energy consumption of cluster head over cluster density. We can see that with the increase of the cluster size from 25 to 50 nodes in a network of 250 sensors, the average energy consumption of CH increases too.

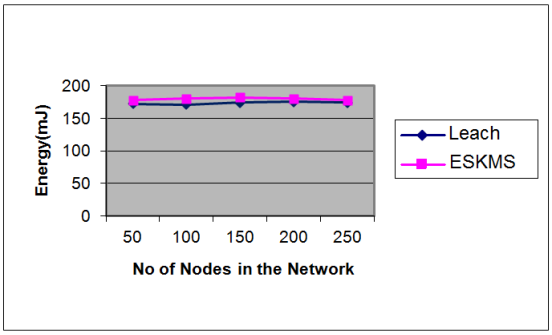

Fig. 4. Energy consumption vs. number of nodes.

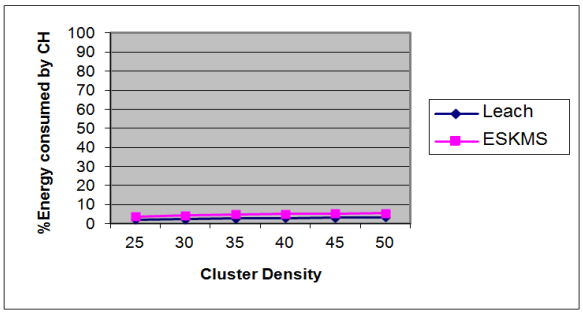

Fig. 5. Average energy consumption of cluster head over cluster density.

Because increasing the number of messages add significant cost to the cluster energy. Both models show that the energy consumption of CH increases with cluster size. In a network of 250 sensors and cluster density of 25 nodes, the average energy consumption of CH in ESKMS is more than 2.65% compared to LEACH, because of the computation overhead.

2) Storage Overhead: For the memory overhead, the normal nodes store only two keys K_{Net} and $K_{CHi,S}$. Apart from this, if the node is CH, it needs to store a pairewise key $K_{BS,CHi}$ as well as the two above mentioned keys. The storage overhead is expressed as for any normal sensor node:

$$\{\text{Size of } (K_{Net}) + \text{Size of } (K_{CHi,S})\}$$

The storage overhead for CH would be:

$$\{\text{Size of } (K_{Net}) + \text{Size of } (K_{CHi,S}) + \text{Size of } (K_{BS,CHi})\}$$

Thus, it obviously increases a little bit storage overhead. Note that, by assuming 128 bits default key size, the storage overhead for an ordinary node would be 256 bits (32 bytes) and for a cluster head it would be at most 48 bytes which is less than 1 KB.

Therefore we can notice that ESKMS does not require an important storage space and the memory requirement for our scheme is very less as compared with other random key pre-distribution approaches based on key pools [11]–[12].

VI. Conclusion

In this paper, we proposed an Efficient and Secure Key Management Scheme (ESKMS) for Hierarchical Wireless Sensor Network. Through performance evaluation, we find that the overhead which the ESKMS protocol leads to is acceptable, and reduces the memory overhead.

ESKMS distribute the keys within a cluster and update the pre-deployed keys at regular interval to avoid node-capturing problem and assure that only legitimate nodes send data for processing. Hence, provides continuous authentication of nodes in the network.

Simulation and analysis has shown that our ESKMS approach is more advantageous in energy-efficient, communication, and storage than other similar schemes.

Our next step is to develop a complete security protocol for hierarchical sensor network including trust establishment and trust management in sensors to deal malicious nodes.

Reference

[1] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 63-75, 2010.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for WSNs," in *Proc. of the 33rd Hawaii International Conference on System Sciences*, Washington, 2000.

[3] A. Modirkhazeni, N. Ithnin, and O. Ibrahim, "Empirical Study on Secure Routing Protocols in Wireless Sensor Networks," *International Journal of Advancements in Computing Technology*, vol. 2, no. 5, pp. 25-41, 2010.

[4] J. Lee, V. Leung, K. Wong, J. Cao, and H. Chan, "Key management issues in wireless sensor networks: current proposals and future developments," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 76 –84, 2007.

[5] S. Zhu, S. Setia, and S. Jajodia, "Leap+: efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500–528, 2006.

[6] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks," in *Proc. of the 9th ACM conference on Computer and communications security*, New York, 2002, pp. 41-47.

[7] H. Chan, A. Perrig, and D. Song, "Random key pre-distribution schemes for sensor networks," in *Proc. of the 2003 IEEE Symposium on Security and Privacy*, Washington, 2003, pp. 197–213.

[8] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti, "Secure pebblenets," in *Proc. of the 2nd ACM international symposium on Mobile ad hoc networking and computing*, New York, 2001, pp. 156–163.

[9] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," in *Proceedings of the 10th ACM conference on Computer and communications security*, New York, 2003, pp. 62–72.

[10] Y. Cheng and D. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," *Ad Hoc Networks (Elsevier)*, vol. 5, no. 1, pp. 35–48, 2007.

[11] L. B. Oliveira, A. Ferreira, M. A. Vilaca, *et al*., "SecLEACH-on the security of clustered sensor networks," *Signal Processing*, vol. 87, no. 12, pp. 2882-2895, 2007.

[12] P. Banerjee, D. Jacobson, and S. N. Lahiri, "Security and performance analysis of a secure clustering protocol for sensor networks," in *Proc. 6th IEEE Intl. Symposium on Network Computing and Applications*, 2007, pp.145-152.

[13] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," in *Proc. of the 10th Annual Network and Distributed System Security Symposium*, California, 2003.

[14] NS-2 web site. [Online]. Available: http://www.isi.edu/nsnam/ns.