SECURITY IN WIRELESS SENSOR NETWORKS: KEY MANAGEMENT MODULE IN SOOAWSN

Mohammed A. Abuhelaleh and Khaled M. Elleithy

School of Engineering
University Of Bridgeport, Bridgeport, CT
{mabuhela, elleithy} @bridgeport.edu

ABSTRACT

Due to high restrictions in wireless sensor networks, where the resources are limited, clustering protocols for routing organization have been proposed in much research for increasing system throughput, decreasing system delay and saving energy. Even these algorithms have proposed some levels of security, but because of their dynamic nature of communication, most of their security solutions are not suitable. In this paper we focus on how to achieve the highest possible level of security by applying new key management technique that can be used during wireless sensor networks communications. For our proposal to be more effective and applicable to a large number of wireless sensor networks applications, we work on a special kind of architecture that have been proposed to cluster hierarchy of wireless sensor networks and we pick one of the most interesting protocols that have been proposed for this kind of architecture, which is LEACH. This proposal is a module of a complete solution that we are developing to cover all the aspects of wireless sensor networks communication which is labeled Secure Object Oriented Architecture for Wireless Sensor Networks (SOOAWSN) .

KEYWORDS

LEACH (Low Energy Adaptive Clustering Hierarchy), Sensor Networks, Network Performance, Routing, Sec-LEACH (Secure LEACH), Network security, Random KD (Key Distribution), Multi-Generation Keys.

1. INTRODUCTION

There are many advantages of using wireless sensor networks. One of these advantages is reducing the cost of the applications by having many sensors with little cost communicate with each other and with the base station providing full network function. At the same time sensor networks have some special characteristics compared to traditional networks which make it hard to deal with such kind of networks. The most important property that affects these types of network is the limitation of the available resources, especially the energy [1].

Sensor networks are self organized networks, which makes them suitable for dangerous and harmful situations, but at the same time makes them easy targets for attack. For this reason we should apply some level of security so that it will be difficult to be attacked, especially when they are used in critical applications [1, 2].

Wireless Sensor Networks (WSNs) [3, 4] are special kinds of Ad hoc networks that became one of the most interesting areas for researchers to study. Routing techniques are the most important issues for such kind of network where resources are limited. Cluster-base organization has been proposed to provide an efficient way to save energy during communication [5-9]. In this kind of organization, nodes are organized into clusters. Cluster heads (CHs) pass messages between

DOI : 10.5121/ijnsa.2010.2406

groups of nodes (group for each CH) and the base station (BS), (see figure1). This organization provides some energy saving, and that was the main idea for proposing this organization. Depending on this organization, LEACH (Low Energy Adaptive Clustering Hierarchy) [10] added another interesting issue to this kind of network, security, where the CHs are rotating from node to node in the network making it harder for intruders to know the routing elements and attack them.

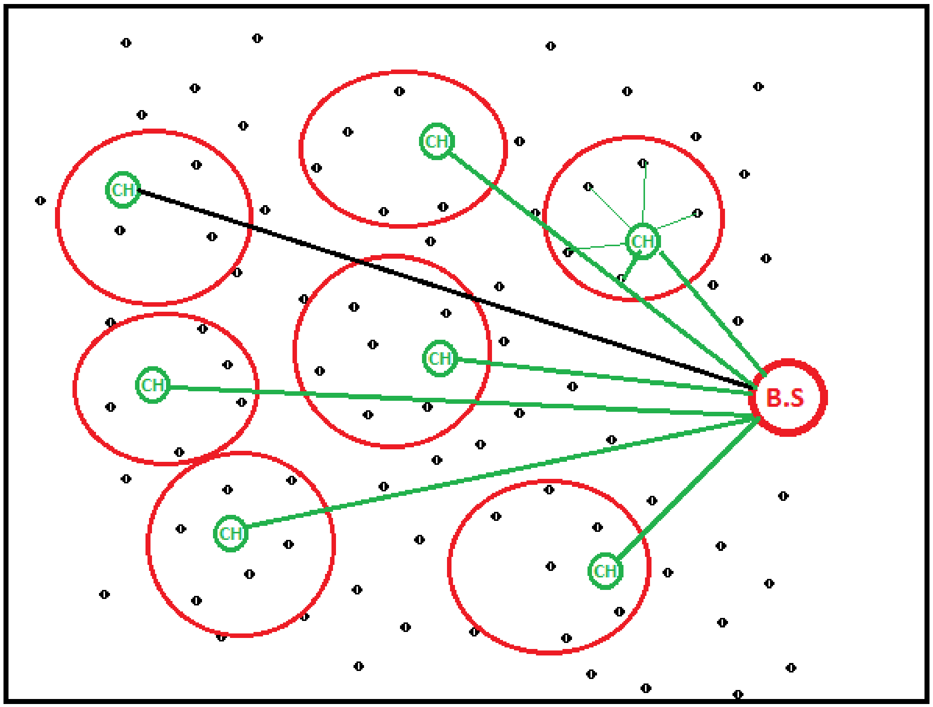


Figure1. Cluster organization for sensor networks

There are some existing works to improve the security of LEACH. One of the most interesting proposals is Sec-LEACH that provides an efficient security to pair-wise node-to-CH communication [11]. A modified version of LEACH is proposed that inherits its security from random key distribution technique.

In this paper, we discuss existing work of LEACH and we focus on how to increase its security. In section two, we discuss the original work of LEACH. In section three we discuss two of the most interesting modifications proposed for LEACH to increase the performance and the security. In section four, we propose modification to enhance the security of LEACH. In section five, we evaluate the performance and security of our solution compared to other solutions.

2 LEACH PROTOCOL

LEACH was first proposed to reduce total energy consumption in sensor networks. It assumed that every node can directly communicate with a BS using a high enough transmitting power. By applying the clustered hierarchy, we can balance the energy consumption. Sensors send their messages to specific sensors which they will be considered as cluster heads (CHs). CHs then aggregate these messages and send them to BS. This process results in energy saving for nodes that are not involved in CHs since they can transmit with less transmission power, but at the same time we consume the energy of CHs. To solve this problem, LEACH proposed a dynamic CH rotation which concluded that the CHs should change at each round. Each round, a new node will become a CH. The network chooses CHs using a distributed algorithm and then dynamically clustering the remaining nodes around CHs [10].

2.1 Description

LEACH consists of two phases implying five steps at each network round. In this section we summarize the steps for single round. The phases are: the setup phase (initial phase) and the steady state phase (real transmission phase) [10, 11]. For the setup phase, each node decides the probability that it can be a CH for the current round while considering the energy and the knowledge of the desired percentage of CHs. Let us call these Ready Nodes RNs. RNs then broadcast advertising messages for the whole of the network. When the nodes receive all advertising messages, the remaining nodes will choose a CH depending on the highest signals received from RNs and then each of these nodes will send a message to the desired CH requesting to join it. When the CHs receive the messages, they start to broadcast the confirmation for these accepted nodes by sending confirmation messages with a time slot schedule for each node in the group. This time slot informs each node in that group on time to transmit its messages.

The second phase concerns the transmission of the real data among the network. According to the time schedule provided by CHs to other nodes, each will start sending its data to the proper CHs. CHs then collect the messages from their members, analyze and handle them, then send the results to the BS.

2.2 Security in LEACH

Jamming and spoofing are kinds of attacks that could be harmful to sensor networks. The nature of Cluster Hierarchy distribution networks may lead to harmful attacks, especially when these attacks rely on CHs for sending and receiving data. If a hacker decides to become a CH, this can result in a disrupted network. Selective forwarding and sinkhole attacks are examples of these kinds of attacks [11, 12].

In LEACH, the possibility for the network to be attacked by these kinds of attack is very small, because CHs are changing in each round of communication, making it hard for the intruders to know the expected CHs for each round so that they can disrupt the critical points of the network [10].

2.3 Improving LEACH

By analyzing LEACH, we can determine the critical points of communication, and then we can focus on providing more efficient security at those points. One approach is to determine CHs in a way that it will be hard for the intruder to guess which nodes will be CHs.

The easiest, and the most efficient way, is to prevent suspicious nodes from participating in the network, and this step should be taken at the time of network setup [11].

By providing a secure way to prevent illegitimate nodes from participating in the network, we can achieve a good level of security and we can reduce the future workload of the network to support security. Some studies propose controlling access to the network for sensor networks, and most of these works are based on key distribution (KD) for cryptographic mechanisms (11, 13-19).

3 CURRENT RESEARCHES ON LEACH

As we mentioned earlier, there are many techniques proposed as new modifications for LEACH to provide more security and to reduce energy consumption. In this section we will discuss two of these works and then we will propose some modifications for these two works.

3.1 SLEACH

SLEACH [20] proposed some additions to LEACH so that it can improve protection for the network. It is suggested that each node has to have two symmetric keys: a pairwise key shared with the BS and the last key chain held by the BS. According to that, it suggested small modifications to LEACH. For the setup phase, the message sent by RNs should consist of an encrypted message that contains the ID of the node that should receive the message and the ID of CH itself as a plain text, and the encryption (ID of CH, the counter shred by CH and the BS, and the advertisement message) using the message authentication code (MAC) that is produced using the shred key between CH and the BS.

The nodes hold CHs IDs, and at the same time the BS will analyze the messages sent by CHs to authorize them. Any valid CH will then have its ID added to the list of valid nodes IDs. After that, the BS broadcasts the list with the encrypted list for all nodes in the network using µTESLA [21] broadcast authentication scheme. Now the nodes can recognize the authenticated RNs to be connected with, so these nodes send their requests to participate with CHs groups. CHs then broadcasts confirmation messages for approved nodes. Each message will contain the time slot schedule for each node.

We can see in this proposed protocol that it does not provide full authentication for node-CH where the messages to be sent from the nodes to CH are not authenticated.

Oliveria et.al propose another solution to provide some ways to pre-distribute the keys using random key pre-distribution for securing node-CH communication in LEACH [11].

3.2 Sec-LEACH- Random KD to LEACH

Sec-LEACH [11] proposes some creative modifications to LEACH protocol. It shows how to invest the key pre-distribution scheme to secure node-to-CH communications. The main idea is to generate a large pool of keys and their IDs at the time the network is deployed, and then each node is assigned a group of these keys randomly. Also each node is assigned with a pair-wise key which shares with the BS; these keys are used during node-node and node-B.S. communications.

This algorithm provides authenticity, confidentiality, and freshness for node-to-node communication. The security level is not impacted by the number of nodes; actually it depends on the size of the key group assigned for each node according to the total size of the key pool [11].

4 NEW KEY MANAGEMENT

In this section first we briefly discuss the main drawbacks of the existing solutions, the whole architecture of our proposal for key management, and then we discuss each part of it in details. For this purpose we consider the whole solution as an object and each part of it as method that can be used individually or with other method in that solution. The main idea here is to break the whole problem into small problems at the beginning, and at the same time having the ability to apply different levels of security on the application according to the application needs.

As we discussed before, we deal with objects that have methods which can be used and directed in many ways to cover the security part of the application according to its needs.

At the beginning we need to divide the security problem into its major parts to be able to deal with each part individually. First, we can divide it according to the type of communications that may appear in WSN, where we have node to node communication and node to B.S. communication. In order to have a secure network we need to secure these communications. For this purpose we need different kinds of keys to be used in securing these communications. We have three types of keys: Secret Key, sharing key, and private and public keys. Next we discuss the methods in our object for using these kinds of keys on a way that offers different levels of security.

4.1 Key Pre-distribution (KP) Method

In this method we apply the same technique that has been applied by Sec-Leach. The aim of this method is to have different levels of security on the network communication for the first generation of the network deployment (i.e. without the ability to add new sensors after the network is deployed).

The idea is to create a pool of keys at the B.S. that has specific number of keys generated randomly using a pseudorandom number generator function. At the same time, the B.S. randomly generates key ID for each generated key which is unique for each key. The second step is to provide each sensor with group of keys with equal sizes for each sensor, and these keys has to be picked randomly without removing any key from the key pool. This leads the sensors to have some sharing keys between each other which make node-node communication possible (these keys called sharing keys). Meanwhile, the B.S. provides each sensor with at least one unique key (named Master Key) which is to be used to communicate between each node and the B.S.

LEACH protocol can be securely applied as follows: After each sensor applied the self electing equation on itself and determine its ability to become a CH for the current round, the communication process starts with the setup phase. In this phase, each CH includes the IDs of the keys in its key group, a nonce in its advertising message offering its availability to become a CH. The ordinary nodes then choose an ID (r) that is shared with CH. Then each of these ordinary nodes sends the message to CH requesting to join its group. The message includes the ID of the node, ID of CH, r, join_ request message, and the encryption of node ID, CH ID, r and the nonce sent by CH) using MAC that is produced using a symmetric key associated with r. Each CH then sends a confirmation message to approved nodes containing the ID of CH and a group of pairs (ID and time slot for each node to start transmission).

In steady state phase, the nodes transmit the messages to CHs according to the time slot provided before. Each message includes the ID of the node, the ID of desired CH, sensing report from the node, and the encryption (node ID, CH ID, node sensing report and the nonce+ reporting cycle within the current node) using the same MAC used before. Finally, CH starts sending the final data for the BS, and the message includes the ID of CH, the ID of BS, the aggregation data from all nodes, and the encryption (the aggregation data and the ID of CH) using the MAC produced from the ID of CH (Master Key of CH).

As discussed by [11], this algorithm provides authenticity, confidentiality, and freshness for node-to-node communication. The security level is not impacted by the number of nodes; actually it depends on the size of the key group assigned for each node according to the total size of the key pool.

The last part of this section shows how we can determine the level of security that we need according to the application needs.

In WSN, there is a fixed space for each node to store the key group selected from the key pool. This means the size of the group (GS) is fixed at the first time the network is built. Then after GS is determined, the size of the key pool (PS) will affect the network in two ways

1. Level of security:
   Depending on the variable names provided before, the security level can be given in this formula [11]
   Security level= 1- GS/PS
   This means: increasing the PS will provide us with higher level of security.
2. Sharing keys probability
   The probability of two nodes not to share the key is given by the formula [11]
   $P= [(PS-GS)!]^2 / PS!*(PS-2GS)!$

This means: the probability for two nodes to share the key is decreased by increasing the size of the key pool.

Since we used the same technique to generate the key pool and to provide the key groups, then the issue of key sharing technique will get the same performance proposed by Sec-LEACH.

Another issue discussed by oliveria et.al is the number of CHs in the network [11]. Because all CHs use the same single hop to communicate with the B.S, then increasing the number of CH will lead to more power consumption. We follow the KD scheme used by Sec-LEACH to produce the sharing keys. As we mentioned before, increasing the size of the pool will decrease the number of CH produced, where only the nodes that received the first packet and share the same key can then proceed with the communication. On the other hand, decreasing the number of CH may results in increasing the number of nodes that joined the CHs.

Providing a suitable size of key pool leads to suitable level of security with high performance, see figure (2).

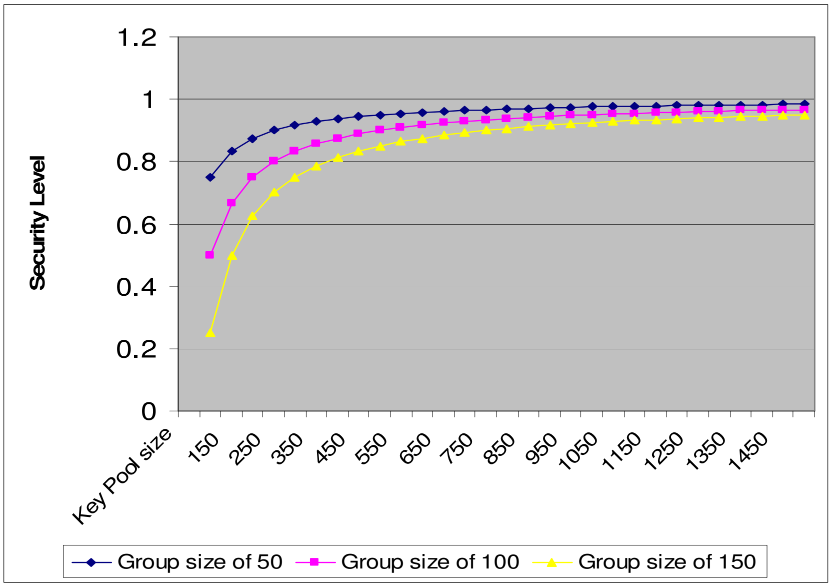


Figure2. Security level affected by key pool size and the keys group size, m represents the size of each group.

4.2 Public and Private Keys Method

In this method, each sensor use two keys for communication with other sensors, Public key and Private Key; the idea is similar to the traditional use of public and private keys in asymmetric key cryptography in traditional networks.

Each sensor generates at least one pair of keys that are related mathematically to each other. The sensor keeps one of these keys to itself as a private key and broadcasts to its neighbors the other key as a Public key. When Sensor A wants to send a message to sensor B, it can follow different procedures. The first one is to send an encrypted message to B using Bs public key (Public-key encryption). B is the only sensor that is able to decrypt this message using its private key. The other scenario is that A sends an encrypted message to B, encrypted using A’s Private key. In this case, a successfully decryption of the message by B, using A’s Public key, guarantees that A is the one who sent the message (Digital Signature). Another scenario is that A sends an encrypted message to B using B’s Public key and send as a part of this message a small part which is encrypted using A’s Private key as a signature of A. Another scenario that can be applied is using this technique for key exchanges purpose to exchange keys between sensors; in this scenario, A sends the secret key that need to share with B, this key and a signature of A is encrypted using B’s Public Key.

4.3 Multi-generations Keys Method

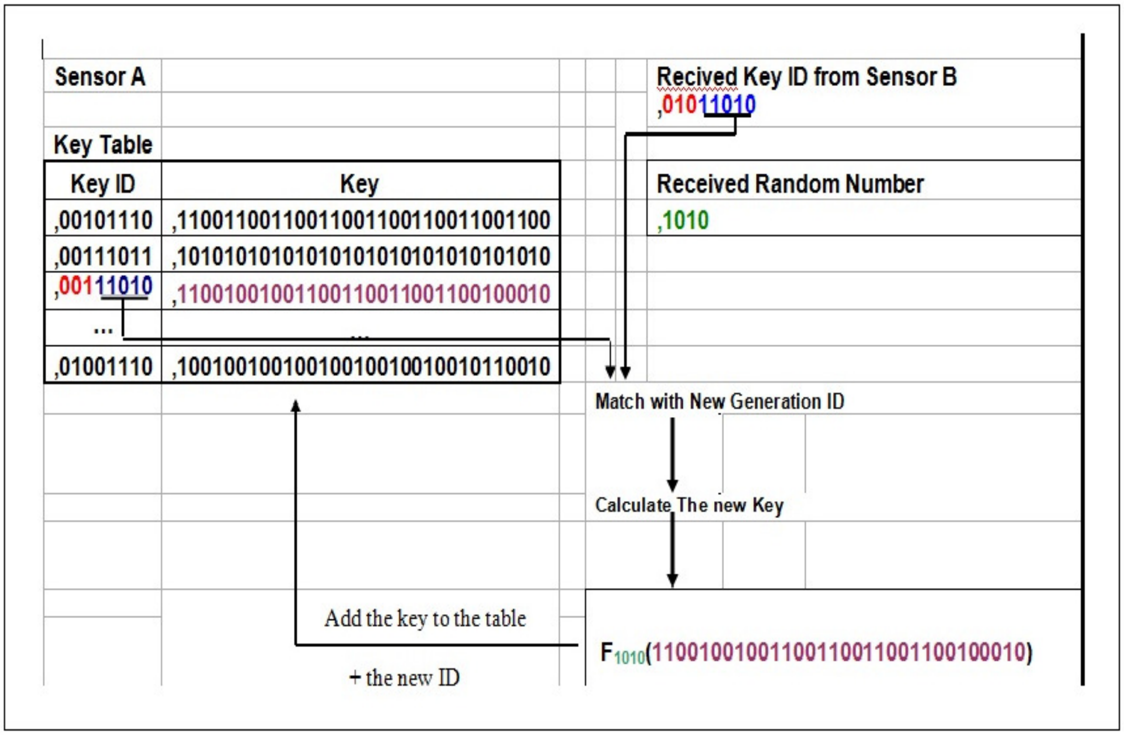
This method relates to the first method in our solution. The idea is to reuse the keys that produced from the key pool in KD technique to support key refreshes and to support the expansion of the current sensor network.

For KD method, B.S. creates a key pool that contains numbers of keys with their IDs. The keys distributed upon the sensors randomly in order to have some sharing keys between sensors to use during their communications. This technique without any extension does not support the ability to refresh the keys implicitly in the future. In order to change a key, the B.S. needs to securely communicate with the related sensors and inform them with the new key/s. Furthermore, this technique does not support efficiently expandability of the network by adding or replacing sensors.

The new method suggests having the key pool refreshed occasionally without the need to announce the existing sensors with these updates. The method works as follows: The B.S. randomly generates a number of keys and assigns key ID to each key. The key has to start with a specific flag that represents the first generation of key (i.e. 001 for example). The B.S. then randomly distributes groups of these keys on the sensors, like in KD method, prior to network deployment. The B.S. station also distributes a formula of one way function to all sensors. In addition to that, the B.S. station distributes some random numbers with unique IDs for each number (the formula and the numbers are the same for all sensors). After a period of time, the B.S. may refresh its keys by calculating a new value of each key using its related old key, and one of the numbers that are previously distributed to the sensors. The new ID for the key will be the second generation flag plus the old key (i.e. 0010… for example). B.S. then distributes the new group of keys on its new sensors. Moreover, B.S. may broadcasts some updated keys to the sensors that have been compromised by intruders (using the secret key, or public key of the specific sensor).

The sensors can be communicated with each other as follows: the sensor, which needs to send an advertising message to its neighbors, includes the IDs of the keys it has in its message with the

IDs of the numbers used to create these keys. The receiver then checks the keys IDs without the flags to see if there are any IDs in common. In the case that the receiver shares some key/s with the sender, it then checks the flag of these IDs to see if it is from the same generation. Then the KD method can be applied without modifications. If the key from different generation then, it calculates the new key from the old key and the value that is included in the message, using the one way function stored in the sensor. This new key then, can be used during the receiver communications with the sender and any other sensor which has the new or the old key with that specific key ID see Figure(3).



Key ID	Key
,00101110	,11001100110011001100110011001100
,00111011	,10101010101010101010101010101010
,00111010	,11001001001100110011001100100010
...	...
,01001110	,10010010010010010010010010110010

Figure3. Updating the key value with the new one received from sensor B.

The key has a lifetime period; this will insure that the sensor memory will always have a space for new keys. This method provides sensor networks with the ability to refresh the keys and expand the network without limitations.

5 SECURITY ANALYSIS

In this section we analyze our solution and we compare it to some of existing solutions. Applying LEACH protocol without any addition provides us with some level of security that has been discussed in [10]. This level of security gives WSN the ability to defeat several kinds of insider attacks. At the same time, the architecture of the network that LEACH applies makes it vulnerable to some attacks like spoofing, jamming, replay, etc. In addition, it is vulnerable to some stage attacks like selective forwarding (this kind of attack result from the intruder claims to be a CH). KD provides integrity, authenticity freshness and confidentiality to node-to-node communications [11].

Using of Public and Private Keys method increase the level of authentication and integrity that have been covered by KD. In addition, it gives the nodes the ability to have a backup plan, with low cost, that can be used to transfer the new keys to uncompromised nodes in case of any attack.

The technique of multi-generation method provides WSN with the ability to support the expansion and the freshness of the network effectively during network life. At the same time it provides an additional low-cost technique to isolate the compromised node from the network activities.

5.1 Security in Action

To declare the security covered by our solution, we discuss the real work of our solution on LEACH as an example of dynamic clustering hierarchy protocol.

Prior to network deployment, B.S. distributes the keys that will be used for data encryption during nodes communication. It first generates large number of keys with their IDs as a key pool. It then assigns each sensor with a group of keys, with their IDs, picked randomly from the key pool without replacement. In addition, it provides each sensor with at least one unique key shared with the B.S. which can be used during sensor-B.S. communications. B.S. also distributes some one-way hash functions with unique ID/s represents each function. In addition, it distributes some pairs of values and their IDs to be used with these formulas in order to generate new keys from the existing ones.

At the beginning of the network life, each sensor generates at least one pair of keys that has two keys relates to each other. The sensor then keeps one of them secretly as a private key, and broadcasts the other key to its neighbors as a public key. Each sensor then checks its availability to become a CH for the current round according to some formulas provided by [10]. The actual round then begins with each CH broadcasts their advertising messages announcing their ability to become CHs for current round. Each message includes the ID of the CH, the IDs of the keys that CH has, and the encryption of both CH ID and Keys IDs. The advertising message is encrypted using a MAC which is generated using the private key of CH. When the Other sensors receive the advertising messages, they check first if they have keys in common from keys IDs provided. Then they check the signature of CH by decrypt the message using the corresponding CH public key. If the decryption match with the plain text provided, then this ensure that CH with its ID is the one who claim to be. Sensors then reply to CH with join-request messages to join the CH cluster. The message includes CH ID, sensor ID and the encryption of the message using the MAC generated from the key shared with CH. In addition, part of the encrypted message is also encrypted using the MAC generated from the corresponding sensor private key. CH receives the messages and confirms that each message received is from the one who claims to be. This can be applied by decrypting the special part of the message using the corresponding sensor public key. CH then broadcasts the replies to all accepted sensors including CH ID and the time schedule for each sensor. This message is encrypted using the MAC generated from each key shared with corresponding sensors. For all messages transferred between the CH and other sensors, a nonce is included to ensure the freshness of the data.

The previous steps represent the setup phase of LEACH. In this phase, all communications are secured using the MAC that is either generated using the private and public keys, or by using the shared keys that previously picked from the key pool. This provides data integrity and confidentiality, where nonce provides freshness. In addition, the use of public and private keys (digital signature) provides authenticity for sensor-CH communications.

The steady state phase starts with all sensors send their data to their CH. Each message includes the ID of the sensor, the ID of the CH, and the encryption of sensor ID, Ch ID, the new value of the nonce (nonce+1), and the data itself. Message encrypted using the MAC which is produced using the sharing key between each sensor and the CH. CH merges all the messages received from other trusted sensors in one message and then sends it to the B.S. The message includes CH ID, B.S. ID, and the encryption of CH ID, B.S. ID and the merged reports. All encrypted using the MAC produced from the unique key shared with the B.S.

The final step supports node-B.S. secure communication by providing integrity and confidentiality of the messages between CH and B.S.

For key freshness and for the network to have new sensors involved, the B.S. regenerates the keys in the key pools using the previous keys and their IDs. It takes each key and it updates the first part of the key to represent the current generation. Then it creates a new key by applying the one way hash function on the old key using one of the values distributed previously on the sensors. The new sensors assigned group of keys randomly from the new key pool and then it applies the same steps like previously discussed with minor modifications. Sensor, who receives a request for communication, checks the IDs provided by the other sensor. It checks the second part of the ID to see if it mach with the one it has. Then it checks the first part to see if it is from the same generation. If it is from new generation, then the sensor calculates the new key using the one way function (i.e. the ID of the function used provided by the sender). The sender also includes the ID of the value used in key calculation. The receiver then stores the key in its keys table. Moreover, this key can be used in the future communication like before. This technique provides the network with key freshness that reduces the ability for attacker to trace the old keys for long period of time. In addition, this technique can be used to safely update any compromised key.

6 Conclusions

We discussed in this paper three methods represent a complete key management solution that can be applied to LEACH, or any similar protocol. Our solution adopted the pair-wise key pre-distribution to provide WSN with different level of security. The use of public and private keys provides WSN with higher security level and provides the sensors an alternative way to exchange new keys. Finally, the technique we applied on renewing the key pool provides WSN with an ability to support multi-generations of sensors. We shows our solution in action by applying it on LEACH.

References

1- D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In Mobile Computing and Networking, pages 263.270, Seattle, WA USA, 1999.

2- I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “A survey on sensor networks”, IEEE Commun. Mag. vol.40 , No.8, 2002 ,pp.102–114.

3- Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In Mobile Computing and Networking, pages 263–270, Seattle, WA USA, 1999.

4- G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. Commun. ACM, 43(5):51–58, 2000.

5- W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In IEEE Hawaii Int. Conf. on System Sciences, pages 4–7, january 2000.

6- A. Manjeshwar and D. Agrawal. Teen: A routing protocol for enhanced effeciency in wireless sensor networks. In 1st International Workshop on Parallel and Distributed Computing Issues in

Wireless Networks and Mobile Computing, 2001.

7- O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," ieee infocom, march 2004. In IEEE INFOCOM, pages 629–640, March 2004.

8- Qing Fang, Feng Zhao, and Leonidas Guibas. Lightweight sensing and communication protocols for target enumeration and aggregation. In 4th ACM international symposium on Mobile ad hoc networking & computing (MOBICHOC´03), pages 165– 76. ACM Press, 2003.

9- A. Iwata, C. Chiang, G. Pei, M. Gerla, and T. Chen. Scalable routing strategies for ad-hoc wireless networks. IEEE Journal on Selected Areas in Communications, 17(8):1369–1379, Aug. 1999. Special Issue on Ad-Hoc Networks.

10- W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In IEEE Hawaii Int. Conf. on System Sciences, pages 4–7, January 2000.

11- Leonardo B. Oliveira, Hao C. Wong, M. Bern, Ricardo Dahab, A. A. F. Loureiro. Sec-LEACH - A Random Key Distribution Solution for Securing Clustered Sensor Networks. Fifth IEEE International Symposium on Network Computing and Applications (NCA'06)

12- C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. Elsevier.s Ad- Hoc Networks Journal, Special Issue on Sensor Network Applications and Protocols, 1(2.3):293.315, 2003. Also appeared in 1st IEEE International Workshop on Sensor Network Protocols and Applications.

13- Lu, K. ; Qian, Y. ; Hu, J., "A framework for distributed key management schemes in heterogeneous wireless sensor networks", Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International

14- Tzu-Hsuan Shan ; Chuan-Ming Liu, "Enhancing the Key Pre-distribution Scheme on Wireless Sensor Networks", Asia-Pacific Services Computing Conference, 2008. APSCC '08. IEEE.

15- Sharifi, M. ; Ardakani, S.P. ; Kashi, S.S., SKEW: An efficient Self Key Establishment protocol for Wireless sensor networks", Collaborative Technologies and Systems, 2009. CTS '09.

16- Junqi Zhang ; Varadharajan, V., "A New Security Scheme for Wireless Sensor Networks", Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE ",

17- Min-Woo Park ; Jong-Myoung Kim ; Young-Ju Han ; Tai-Myoung Chung, "A Misused Key Detection Mechanism for Hierarchical Routings in Wireless Sensor Network", Networked Computing and Advanced Information Management, 2008. NCM '08.

18- Ergun, M. ; Levi, A. ; Savas, E., "A resilient key predistribution scheme for multiphase wireless sensor networks", omputer and Information Sciences, 2009. ISCIS 2009.

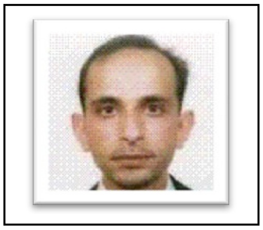
19- Kun Zhang ; Cong Wang ; Cuirong Wang, "A Secure Routing Protocol for Cluster-Based Wireless Sensor Networks Using Group Key Management", Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08.

20- A. C. Ferreira, M. A. Vilac¸a, L. B. Oliveira, E. Habib, H. C.Wong, and A. A. Loureiro. On the security of cluster-based communication protocols for wireless sensor networks. In 4th IEEE International Conference on Networking (ICN'05), volume 3420 of Lecture Notes in Computer Science, pages 449–458, Reunion Island, April 2005. Springer.

21- A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. Wireless Networks, 8(5):521–534, September 2002.

Authors

Dr. Elleithy is the Associate Dean for Graduate Studies in the School of Engineering at the University of Bridgeport. He has research interests are in the areas of network security, mobile communications, and formal approaches for design and verification. He has published more than one hundred twenty research papers in international journals and conferences in his areas of expertise. He is the co-chair of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE). CISSE is the first Engineering/Computing and Systems Research E-Conference in the world to be completely conducted online in real-time via the internet and was successfully running for four years. He is the editor or co-editor of 10 books published by Springer for advances on Innovations and Advanced Techniques in Systems, Computing Sciences and Software. He received the B.Sc. degree in computer science and automatic control from Alexandria University in 1983, the MS Degree in computer networks from the same university in 1986, and the MS and Ph.D. degrees in computer science from The Center for Advanced Computer Studies in the University of Louisiana at Lafayette in 1988 and 1990, respectively.



Mohammed Abuhelaleh is a full-time Ph.D. student of Computer Science and Engineering at the University of Bridgeport. He worked as a lecturer for Alhusein Bin Talal Universtity; He thought some computer science courses, in addition to college courses, like Data Structure, C++,and Computer Skills for three years. He has master degree Computer Science from University of Bridgeport, and graduated with a GPA of 3.48. Mohammed now is in fifth year of PHD program. Mohammed worked as Graduate Assistant Networks courses, like DCC, Network security, and Network administration under Prof. Elleithy.