AN ANTI-CLONE ATTACK KEY MANAGEMENT SCHEME FOR WIRELESS SENSOR NETWORKS

Heshem A. El Zouka

Department of Computer Engineering, College of Engineering and Technology
Arab Academy for Science & Technology and Maritime Transport,
Alexandria, Egypt
helzouka@aast.edu

ABSTRACT

Wireless Sensor Networks (WSNs) are subject to various kinds of attacks such as replaying of messages, battery exhausting, and nodes compromising. While most of these attacks can be dealt with through cryptographic security protocols provided by key management schemes, there are always a few that manage to really cause problems. One such attack that is most common and significant in WSNs is cloning attack. In clone attack, the intruder tries to capture and compromise some nodes and inject them into several locations throughout the network in order to conduct other types of attacks. Moreover, if this attack is not detected early, then these replicated injected nodes will consume a large amount of the network resources. In this paper, we analyze several key management schemes that can be used for checking integrity and preventing cloning attacks. After analyzing the problems associated with these schemes, we propose a model that allows us to distinguish between legitimate nodes and cloned nodes in such sensor networks.

KEYWORDS

Node replication; Network security; Energy efficient; clone attacks ; Key management schemes.

1. INTRODUCTION

WSNs are increasingly becoming the networks of choice in many areas, including military, industrial, environmental, and medical applications. Sensor nodes are gaining interest due to their low-cost and their low power consumptions. A WSN consist of a set of sensor nodes that are distributed over a large geographic area in order to cooperatively pass the sensed data. It is expected to operate cooperatively over a long time with minimal power consumptions. Sensor nodes themselves consist of sensing, data processing, coordinating circuits and communicating components.

Therefore, the design of secure and survivable node is one of the most vital issues in designing energy-efficient protocols for wireless sensor network where the energy, memory and

Dhinaharan Nagamalai et al. (Eds) : CSE, DBDM, CCNET, AIFL, SCOM, CICS, CSIP- 2014
pp. 179–190, 2014. © CS & IT-CSCP 2014 DOI : 10.5121/csit.2014.4415

computational power of sensor nodes are limited. In this paper, some of the challenges facing the key management schemes in WSNs are discussed in attempting to evaluate them and propose a based security solution against cloning attacks, and hence securing the communication channel.

Furthermore, utilizing the existing security protocols in wireless sensor networks has led us to propose a secure framework which incorporates Kerberos authentication protocol [1] in a way that reduce the communication overhead especially over low bandwidth networks.

The rest of this paper is organized as follows: In the following sections different types of cloning attacks will be reviewed with respect to the existing pairwise key setup schemes and their vulnerabilities. Then, the case with which sensor nodes can be compromised using regular off the shelf technology and readily available free software will be demonstrated, thereby examining the vulnerability of the existing key pre-distribution schemes. Following that, additional issues associated with cloning attacks focusing on preventive techniques rather than detective approaches will be described, for example, several possible approaches are suggested to improve the effectiveness of key management in WSNs and to avoid the problem of cloning. Finally, the last section gathers everything together; the implementation discussed along with all the simulation results obtained and a comparison of the results is presented.

2. RELATED WORK

Several possible approaches are proposed in the literature to improve the security, authentication protocols, and key management schemes in WSNs. Indeed, most existing key management schemes in sensor networks are designed to establish a pairwise key among the nodes, no matter whether these nodes communicate with each other or not, and this cause the network to suffer from many attacks and vulnerabilities [2].

These vulnerabilities allow remote attackers to sniff the network, easily create clones in the compromised nodes and inject them in several locations on the network trying to launch other types of attacks. In fact, the simplicity and low-cost of these sensor nodes can make cloning attacks more likely, especially during the maintenance phase, where some of the network nodes are replaced with new ones to prolong the battery's lifetime.

Recently, several solutions have been presented to defend a WSN against these attacks. Most of these solutions have been proposed based on the use of strong cryptographic techniques and robust key management schemes that control access among sensor nodes [ ].

To control access and secure the communication channels between nodes, each of the proposed schemes try to establish a symmetric key between every pair of neighboring nodes. The use of strong symmetric cryptography system, however, requires a robust key management scheme to handle, distribute and when needed, revoke and refresh the symmetric shared keys used for securing the communications between nodes. These established keys are often used to ensure the integrity of the overall traffic exchanged between the network nodes.

However, the establishment of pairwise keys between communicating neighbor nodes is a challenging problem due to the dense deployment and randomness nature of sensor networks. Hence, in most key management schemes, the problem of joining new node and discovering its direct neighbors in order to establish a proper pairwise keys, may remain a difficult task since the

nodes are randomly scattered across large geographical area, causing non uniform distribution of the nodes. Yet, there are many other issues that affect the design of robust and secure key management schemes. For example, the design of energy efficient protocols pushed researches to develop lightweight authentication protocols that can be used to validate the legitimate nodes in WSNs [4]. Many of these proposed protocols were presented, but none of which employs asymmetric cryptography schemes due to the limited resources of the sensor nodes.

Moreover, the lack of hardware memory protection may allow the attackers to extract sensitive information from the physical memory of the nodes. Even with well hardware protection, nodes in W|SNs are prone to failure due to hardware malfunction caused by their dense deployment of sensor nodes, thereby exposing the information stored in nodes [5]. All of these vulnerabilities may allow the attacker to reproduce new clones and inject them in several locations of the network. These clones can be easily project themselves as legitimate nodes to the network and explore other types of attacks [6]. Therefore, the detection of clone attacks is another major challenge in securing wireless networks, and will be discussed further in the following sections. Analysis of current master key based schemes in WSNs

3. CLONNING ATTACK AND KEY MANAGEMENT

To minimize the impact of cloning attacks in WSNs, a variety of key management schemes have been proposed over the past few years. These schemes can be classified into three main categories: (1) Time based schemes; (2) Geographic location based schemes; and (3) Third party based schemes. These three basic schemes are analyzed for their defense against cloning attacks, where, for example, the sensor nodes are subject to physical compromise that is hard to defend against. However, in order to analyze these schemes, it is useful to consider some assumptions which permit us to generalize the protection scope against cloning attacks. First, it is assumed that all nodes' locations are fixed and there are no mobile nodes. Secondly, all sensor nodes are deployed in a two dimensional area and each node has the knowledge of its own position and its own ID. Thirdly, it is assumed that there is a time limit T_{min} to compromise the node, and the attacker can successfully compromise the node within that time limit and obtain all the stored keys. Finally, it is assumed that every node has a setup time T_{set}, where T_{set} is the maximum time a newly deployed node needed to discover its immediate neighbors in order to establish a trusted pairwise keys with them.

Meanwhile, the base station (BS) maintains the record of IDs, master key, and positions of all sensor nodes. All the data mentioned above can be acquired during either the initial deployment of the sensor nodes or during maintenance phases of WSN.

A. Time based schemes

In this key management schemes, a master key (K_m) is preloaded into each sensor node. A sensor node uses this key to set up a pairwise key with each of its neighbors. After completion the key setup phase, each node erases the key K_m from its memory. Localized Encryption and Authentication Protocol (LEAP) is one of the most popular example of this schemes [7]. In LEAP, every node is preloaded with a master key K_m (sometimes called the primary key) under the assumption that this master key will be removed when the network is deployed.

In a network of N nodes, each node is assigned with an ID from 0 to N-1, where a node with IDu and its key Km can establish a secure one way hash function Ku = *f* Km (IDu). Then, in the neighbor discovery stage, node u broadcasts a message containing its identity IDu and set a timer, which will be triggered when the elapsed time of neighbor discovery is greater than Tmin. The response message from a neighbor node v contains its identity and message authentication code (MAC) will be used later for verifying node v's identity. In general, the following example shows how the conversation is established to generate a pairwise key between any two adjacent nodes:

$u \rightarrow$ □ ; Broadcast to all neighbors (1)
$v \rightarrow u : v \mid MAC(K_v , u|v)$; Response message (2)
$K_{u,v,\_} = f(f(k_m , v) , u)$; Computed pairwise key (3)

Therefore, by exchanging ID numbers, each node can set up a shared key with its neighbor nodes. Once T_{min} is expired, every node, such as node v, will erase the master key K_m from its memory, while keeping its own individual key (K_v). However, in case of a cloning attack, a number of security breaches can be introduced in this keying scheme. Most important, if the initial master key becomes known to the attacker at any time less than T_{min}, then the attacker can easily forge any pairwise key between two adjacent nodes. In this case, the attacker will not only be able to compromise all previously established pairwise keys in the network, but will also be able to compromise all future pairwise keys. Moreover, even if the master key is not compromised, the attacker can inject any number of malicious nodes during the maintenance operation phase of the network. In case of hardware failure of node components, the node keeps the initial master key in memory without erasing it and hence the key will be captured easily. The chance of hardware failure is more likely to increase if a deployment method uses an airplane to deploy sensor nodes.

To overcome these vulnerabilities in the basic LEAP scheme, S. Zhu et al. further proposed the extended scheme to LEAP, which was named as LEAP++ [8]. In this scheme, authors assume that the attacker is capable of recovering K_m before T_{est}. They propose a solution to this problem by having time slots for the distributed keys. Therefore, every master key is only valid for a certain time slot T, and every new joining node in the network is preloaded with a master key and a set of individual keys for all other time periods t , where $t > T$. In this scenario, if the master key is compromised, the attacker can only know the pairwise keys setup within that time T, and the pairwise keys setup in other time periods are still secure.

However, this solution introduces other potential problems, which make LEAP++ less attractive in terms of timing, control, and process. For example, one key question is how to calculate the length of time slot. If the length of time slot is too long and there are many nodes required to set up keys during this time, the approach is not relatively new compared to the LEAP protocol. On the other hand, by reducing the length of time slot, then the number of compromised pairwise keys will be also reduced. Clearly that shorter time duration will also increase the difficulty of management and deployment.

Another problem with this approach is that it does not offer support for backward authentication. So, encrypted data recorded earlier can be easily decrypted including key exchanging data between neighboring nodes. Therefore, the vulnerability of cloning attacks remains high due to the lack of backward authentication between nodes. Additionally, the attacker can add malicious nodes to the network if he in possession of the initial master key [9]. The open broadcast nature of radio communications also makes it possible for any faulty node to be impersonated without knowing it, and hence revealing the stored keys [10].

B. Geographic Location Based Schemes

In localization based schemes, each sensor node knows the coordinates of its location using either global positioning system (GPS) or any other localized methods.

For example, in case of deterministic deployment, the position of the node is calculated according to its relative distance to neighbors, and in which any pair of nodes comes under transmission range of the WSNs are considered neighbor nodes.

Generally, all localization schemes are based on Eschenauer and Gligor's random key pre-distribution (RKP) [11]. RKP scheme is a probabilistic key management scheme where each node is preloaded with a number of keys that are randomly selected from a large key pool. Neighboring nodes use these preloaded keys to set up their pairwise keys. All communication will then use this pairwise keys to authenticate and verify the integrity of the exchanged messages. In addition, based on the location of nodes, the confidentiality is maintained by assigning an index for each key, and the index of keys is exchanged between nodes and their neighbors to determine their shared pairwise keys. Therefore, information about the position of the node can be used to ensure confidentiality between neighbor nodes and hence preventing cloning attacks.

However, compromising one node will reveal its keys and any established pairwise keys, although the attacker cannot inject malicious nodes elsewhere into the network. This is because of the location of the nodes which were deployed on predefined regions of the network. Another problem with the location based schemes is that they consume more memory than other key management schemes of WSNs since each node needs to store the coordinates of its neighbors, and the relative amount of memory in WSN is very limited. However, in such schemes, the energy consumption will be balanced among all the sensor nodes and hence the network lifetime can last longer.

C. Third parity based schemes

These types of schemes depend on a trusted third party (e.g. the base station) or a server that acts as a key distribution center (KDC) where a pairwise key is generated upon request of any two sensor nodes in the SN wishing to communicate. The KDC normally sends this key in encrypted form to the communicating nodes. An example of this scheme is Kerberos, which was built on the Needham- Schroeder protocol. Kerberos was originally designed to enable two parties to exchange secret information across an otherwise open network [12,13,14].

In this key management scheme, each sensor node of the network shares a different secret key with the KDC, which enables the nodes to verify the received message originated from the base station. The Kerberos server itself provides a centralized server whose function is to validate sensor nodes by providing them with ticket to grant request to the base station. Actually, both uthentication server and a ticket granting server, the main two components of Kerberos, work together as a trusted third party (TTP), and the authentication server knows all the nodes' passwords and stores them in a centralized place.

Actually, both authentication server and a ticket granting server, the main two components of Kerberos, work together as a trusted third party (TTP), and the authentication server knows all the nodes' passwords and stores them in a centralized place.

On the other hand, the purpose of the ticket granting server is to certify to the server/Base station in the network and to ensure that a node is really what it claims to be. In this way, both the authentication and authorization servers are used to authenticate node to each other in WSN.

Figure 1 describes how the node and the base station are jointly configured to verify each other's identity via the Kerberos server. In this flat connection protocol, the Kerberos key exchange mechanism specifies three exchanges: the Kerberos authentication exchange, the key granting service exchange and base station to node service exchange. In this way the connection is established between the nodes and the servers to enable them to exchange the keys and certificates. However, the deficiency with these protocols is that they use what is known as "hierarchical authentication protocols" where each sensor node in network has only one authentication provider, which is Kerberos in this case. When the network density is high, all the sensor nodes have to wait for a long time to be authenticated and establish a semi SSL connection with the base station. From energy consumption perspective, most amount of energy is consumed in such authentication and authorization process. In order to avoid energy consumption and unnecessary traffic, which in turn may increase the average delay and cause a cloning attack, an alternative practical approach that uses the envelope model is presented and described in section 3 of this paper, but with some changes.

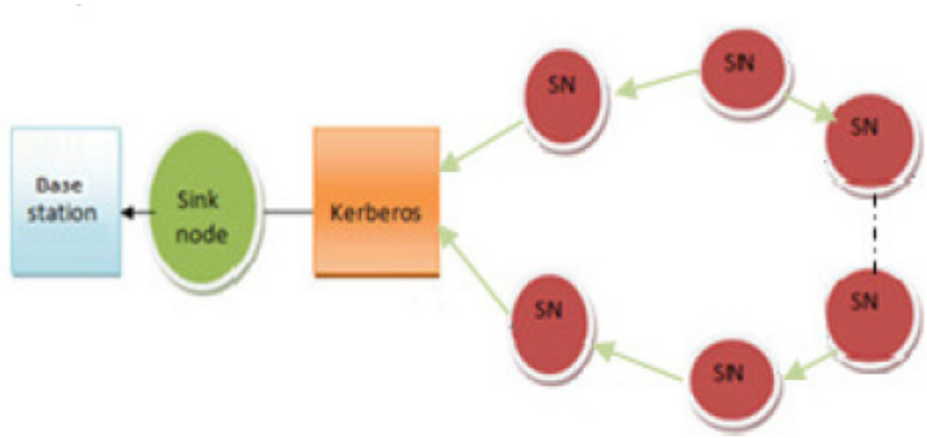

Figure 1. Flat Connection Model

In this model, the network is divided into clusters and a set of Kerberos controllers as shown in Figure 2. Each controller works as an authentication authority and a key management for one cluster in the control group of the WSN. On the other hand, all the nodes inside each cluster will communicate with the CH node using AES encryption Algorithm.

The CHs themselves will authenticate and communicate securely with each other using Kerberos. The effectiveness of this model is that it distributes the keys among the upper hierarchy of CHs using Kerberos authentication, and strong symmetric cryptosystem among cluster nodes, making it impossible for cloning attacks to take place. Even if the attacker succeeded to compromise one cluster, the other clusters are still protected.

The proposed Hierarchical model uses multiple Kerberos controller as apposed to the Flat model.

Clearly, because of the constraints imposed on WSNs, such as energy limitation, the cost of having many Kerberos controllers tend to be quite complex and usually defy analytical methods that have been proved to be fairly effective for Flat connection model.

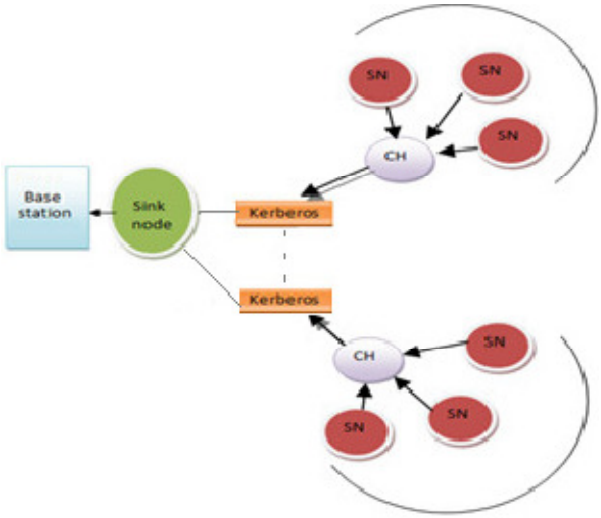

Figure 2. Hierarchical Connection Model

Another advantage of Hierarchical model over Flat is the minmized overhead as there is no common master key shared between the nodes across the network to help each node realize its closest neighbor. Also the Flat model creates a single point of failure acting as a bottleneck in the whole network. In addition, the model uses AES-128 encryption in the communication between nodes of the same cluster which offered faster computation, thus minimizing energy dissipation in these cluster nodes.

However, because of many constraints imposed in modeling Hierarchical networks, such as the dependency measures of multiple Kerberos controllers, modeling of such networks tend to be quite complex. Furthermore, few controllers have come into existence, for there are still many research experiments that need to be considered.

4. DISCUSSION AND SIMULATION RESULTS

In this paper the cloning attack problem and its impact with respect to three categories of key management schemes were presented. In time based scheme, the master key in basic LEAP protocol is used to calculate all of its neighbor pairwise keys. We noticed that the node can be compromised by reproducing clones which will allow the intruder to infiltrate the sensor network, and then other types of attacks can be conducted. Therefore, the first type of key management scheme exploits seriously degrades the resilience of such schemes.

To overcome the vulnerabilities in basic LEAP protocol, we showed how LEAP++ used a time slots for the distribution of the pairwise keys. In this protocol, every master key is valid for certain time slot T, and every new joining node is preloaded with a master key. However, we found that LEAP++ did not offer advantages compared to LEAP in terms of timing, control, and process. Besides, it is not easy to calculate the length of time slot. The analysis also showed that

the vulnerability in time based schemes remains high due to the lack of backward authentication between nodes, which make these schemes vulnerable to the cloning attacks.

We analyzed the localization based schemes, and found several constraints and limitations which can limit the use of such schemes. We defined the problem of localization systems as estimating the position or coordinated of sensor nodes. In localization schemes, nodes can be equipped with a GPS system, but this is a costly solution in terms of memory and power consumption. We also found that most of the deterministic deployment algorithms were not aware of range measurement inaccuracy or had not considered the scaling problems in designing their localization algorithms. However, one of the benefits of using localization ased schemes is their ability to store all the information needed to determine the position of the nodes which can assist in strengthening the process of key establishment and hence, in preventing cloning attack.

Then, we examined the schemes which involve the base station in the process of key management. We presented the strengths and weaknesses and what are the possible attacks to these management schemes in general. In these schemes, the base station plays a central role in generating the pairwise keys and authenticating the nodes. Two authentication schemes were discussed, one is Flat connection model and the other is Hierarchical connection model.

In Flat model, the connection is established between nodes and servers in a manner that is secure and efficient in terms of authenticity. However, the performance of these schemes degrades significantly when the number of sensor nodes increases. Clearly, a network that has only one authentication provider will cause considerable routing overheads and longer authentication time.

In Hierarchical model, the cluster heads are selected according to their battery life time and in a way similar to [15, 16]. In this scheme sensor nodes play the roles of cluster heads periodically. Whenever a cluster head is elected in a cluster, the CH broadcasts a message to other member in the cluster that it becomes a cluster head.

We evaluated the performance of Hierarchical compared with Flat structure in detail including energy consumption and battery life time. We used OMNET [17] as a simulator to analyze the performance of Flat and Hierarchical.

The basic assumptions used in performance analysis assumes that different energy consumption values would be generated according the key management process performed by nodes and servers, making a distinction between the distance among sensor nodes and the authentication servers. The network size was simulated as a square area of 100 x 100 m3, and the performance of algorithms was analyzed with respect to the lifetime of the network.

Table 1 Network variables

Items	Value
Sensing area (m^2)	100 x 100 m
Number of nodes	100, 500
Initial energy (J)	10 J
Tx energy	50 nJ/bit/m^3
Rx energy	50 nJ/bit/m^3
Packet size (bytes)	32 bytes

On the other hand, the amount of consumed energy was measured by considering the energy consumption required for the replacement of cluster heads and the broadcasting messages between all nodes and their servers. The model is implemented based on the assumptions listed in table 1. As shown in the table, 100 sensor nodes were randomly deployed over an area of 100x100 m3 to be used in the simulation, and then we increased the number of sensor nodes to be 500 distributed over the same area.

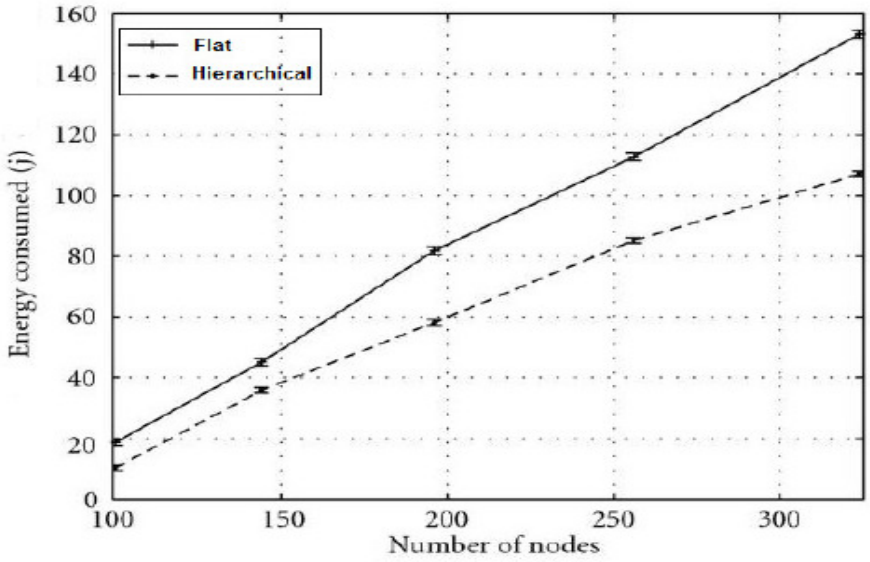

Figure 3. Energy Consumption

All nodes are assumed to have fixed locations and no mobility feature. All nodes are homogeneous and have the same initial energy of 10 J. The energy required by the radio to run the transmitter or receiver circuitry = 50 nJ/bit/m3. For modeling the Kerberos authentication server, we applied a four byte SHA-1 algorithm such that an intruder has to generate 231 packets on average and the sensor nodes would be dead. The compressed data packet size in bytes = 16. We plotted the average of 100 simulate experiments, and the compare results are shown in Figure 3 and Figure 4.

As illustrated in Figure 3, we can observe that Hierarchical is more energy efficient than Flat. Based on these results, we noticed that more than 75% of the sensor nodes in the Hierarchical model preserved their energy as the energy is consumed mostly around the cluster heads. On the

other hand, the Flat model introduces more energy consumption due to the longer paths to Kerberos and consequently higher end-to-end packet transmission time. Therefore, based on these results, we conclude that Hierarchical model is better than Flat in terms of balancing the energy consumption in wireless sensor networks.

As illustrated in Figure 4, the average throughput measured over the Hierarchical model tends to be higherthan the Flat model due the aggregation of all packets at the CHs. Clearly, the flat model offers a higher end-to-end delay as the data travels a long distance before it reaches the BS/Kerberos controllers.

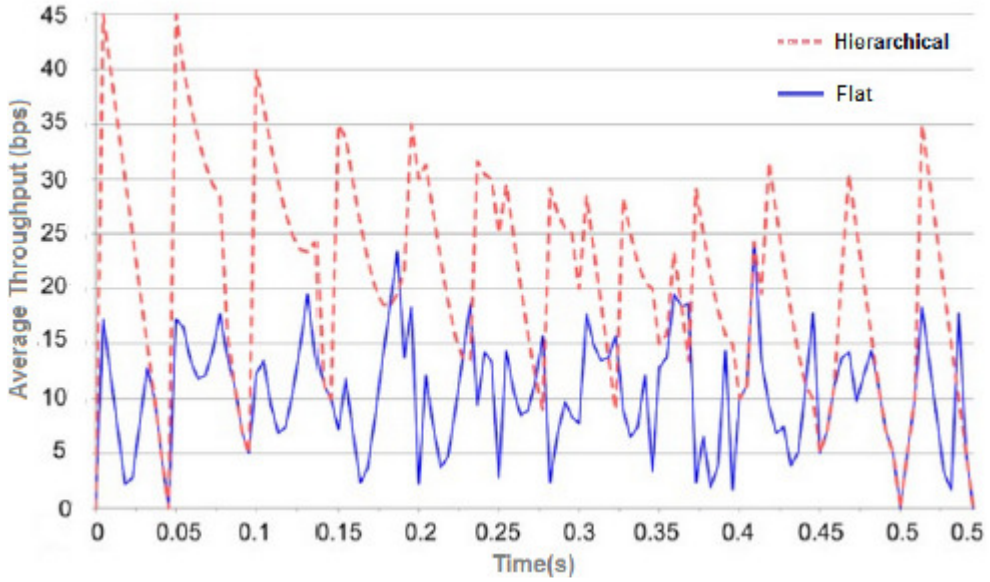

Figure 4. Communication Overhead

On the other hand, the Hierarchical model offers a higher throughput, faster key management scheme, and lower authentication delay than the Flat model. Therefore, we can finally conclude that the Hierarchical model has achieved better simulation results than the traditional Flat model in terms of energy, throughput performance, and network life time.

5. CONCLUSION AND FUTURE WORK

In this paper, the challenges and the approaches for the security and routing protocols of WSNs were surveyed. Then, a framework that secures the communications between the wireless nodes was proposed. In the first experiment, a Hierarchical model that uses Kerberos controller along with a cluster head in a hybrid manner to preserve the energy and increase the life time of WSN was implemented. In the second experiment, the process of employing the base station to enhance the authentication protocol of the sensing nodes was examined. To improve the performance of the Flat model, the proposed Hierarchical architecture is implemented using two security layers, one for establishing authenticity and one for generic trust that authenticates the distributed Cluster Heads. The existing key management schemes were surveyed, and based on their response, a

Hierarchical model that uses multiple Kerberos controllers to improve the effectiveness of key management in WSNs was proposed.

The analysis showed that the proposed Hierarchical model provides a significant increase in the life of the entire network as more than 75% of the nodes reserved their energy while the consumption is limited to the CHs. As for evaluating the effectiveness of employing a strong authentication technique, the analysis showed that the distributed Kerberos controllers experienced fewer losses by sending fewer instructions per packet and the resulting compressed data rate was improved.

In the future, the scale of the network will be increased and more than one base station will be examined, also we plan to make our protocols aware of data freshness by adding time stamp to the authenticated packet. Additionally, we plan to study the performance of our model on different motes and build a comparison over different architectures.

REFERENCES

[1] J. G. Steiner, C. Neuman, and J. I. Schiller, ``Kerberos, an Authentication Service for Open Network Systems'," USENIX Association Conferences Proceedings, February 1988, pp. 191-202.

[2] D. Manivannan and P. Neelamegam, "WSN: Key Issues in Key Management Scheme – A review," Research Journal of Applied Science, Engineering and Technology, vol. 4, 2012, pp. 3188-3200.

[3] S. Othman, A. Trad, and H youssef, "Performance Evaluation of Encryption Algorithm for Wireless Sensor Networks," International Conference on Information Technology and e-Service (ICITeS), March 2012, pp. 23-35.

[4] O. D. Mohatar, A. F. Sabater, and J. M. Sierra, "A lightweight Authentication Scheme for Wireless Sensor Networks," Ad Hoc Networks, vol. 9, no. 5, 2010, pp. 727-735.

[5] C. Sreedhar, S. Vema, and P. Kasiviswanath, "A Survey on Security issues in Wireless ad hoc Routing Protocols, International Journal 2(2), 2010, pp. 242-232.

[6] A. Pandey and R. Tripathi, "A Survey on Wireless Sensor Networks Security," International Journal of Computer Applications, vol.3, no.2, June 2010, pp. 8887 – 8975.

[7] S. Zhu, S. Setia, and S. Jajodia, " LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," ," in Proceedings of CCS'03, The 10th ACM Conference on Computer and Communications Security, Washington D.C, USA, October 2003, pp. 27-31.

[8] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient Security Mechanisms for Large Scale Distributed Sensor Networks," ACM Transactions on Sensor Networks vol. 2, 2006, pp. 500 – 528

[9] J. Jang, T. Kwon, and J. Song, "A time-based key management protocol for wireless sensor networks," in Proceedings of ISPEC07, Information Security Practice and Experience, 2007, pp. 314–328.

[10] B. Tian, S. Han, L. Liu, S. Khadem, and S. Parvin, "Towards Enhanced Key Management in Multi-phase ZigBee Network Architecture," , in Proceedings of Computer Communication, vol.35, no.5, pp. 579-588.

[11] Laurent Eschenauer and Virgil D. Gligor, "A Key Management Scheme for Distributed Sensor Networks," in Proceedings of the 9th ACM Conference on Computer and Communication security, November 2002, pp. 41-47.

[12] R. Needham and M. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," Communications of the ACM, vol. 21, no. 12, December 1978, pp.993-999.

[13] B. Clifford and Theodore Ts'o, " Kerberos: An Authentication Service for Computer Networks," From IEEE Communications Magazine, vol. 32, no. 9, September 1994, pp. 33-38

[14] C. Chang, D. J. Nagel, and S. Muftic, "Assessment of Energy Consumption in Wireless Sensor Networks: A Case Study for Security Algorithms," In 4th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (IEEE MASS 2007), Pisa, Italy, October 2007, pp. 1-6.

[15] S. Mostafa, H. El Zouka, and M. Abouelnasr, "Hybrid Encryption Secure Routing Protocols for Wireless Sensor Networks," Proceeding of the ISCA, First International Conference on Sensor Networks and Applications (SNA), San Francisco, November 2009, pp. 109-114

[16] H. El Zouka, "Challenges in Securing Wireless Sensor Networks," in Proceedings of SENSORCOMM 2013, The Seventh International Conference on Sensor Technologies and Applications, Barcelona, Spain, August 2013, pp. 145-150.

[17] The OMNeT++ Simulator. http://www.omnetpp.org [Retrieved on January, 2014].