SOOAWSN: A complete framework for Wireless Sensor Networks

Graduate Student Paper

Mohammed A. Abuhelaleh and Khaled M. Elleithy
School of Engineering
University Of Bridgeport, Bridgeport, CT
{mabuhela, elleithy} @bridgeport.edu

Abstract

Secure Communication is a mandatory requirement in many network applications. Wireless sensor networks (WSN) require more attention regarding securing the communication between its nodes. In these kinds of networks, communications may occur between sensors themselves or between sensors and the node that is responsible of collecting information from sensors which called a Base Station (BS). Limited resources of such kind of networks require special techniques to be applied on those kinds of networks that differ from those applied to other kinds of networks. Limited resources include power, low speed processors, and small memories. In this research, we present a complete framework that covers all the aspects of WSN communications (i.e. Routing Protocol, Security Key Management, and Intrusion Detection). The framework is called Secure Object Oriented Architecture for Wireless Sensor Networks (SOOAWSN). In this paper, we present each module as an object that contains some methods that cover all the aspects of that part. Then, these objects are applied together as a complete solution or as methods selected from each object to create a partial solution that satisfy specific application’s requirements.

1. Introduction

Many current networks applications have special needs in order to be deployed. Some of these applications have some constraints that make it hard to be deployed using the traditional kinds of networks such as, the space required, the way of distributing nodes, the nature of the place where the application need to be created, and the cost. Some examples are: traffic surveillance, building monitoring, whether stations, and battlefields. In these kinds of applications, wireless sensor network is the best choice, where nodes can be distributed in a way that satisfies the application’s needs. On the other hand, these kinds of networks have many properties that may affect their function. Limited resources such as: limited power source, small memories, and low speed processors make it hard to deal with such kind of networks. Researchers have been working on two major challenges occur from dealing with such kind of networks, and these are: Network performance, and network security. Unfortunately, none of them did present a complete solution that addresses both aspects. Some researchers discussed security related issues, and others discussed performance related issues.

In this paper we present a unique and a complete framework that should cover the whole network lifecycle from the time it is deployed. We present a Secure Object Oriented Architecture for Wireless Sensor Network communication (SOOAWSN) framework. We divide the Wireless Sensor Network solution into objects that contain methods. Each method represents a part of the wireless sensor networks communication aspects. The idea of dividing the main solution into parts is to break a large and complex problem into smaller and clearer problems. Another goal is to achieve flexibility in the final solution in order to be applicable to many situations as parts or as a complete solution. The final goal is to have the solution easier to be implemented and applied to real-world situations. In this work, we will provide and test each part independently then we will combine them together in a complete framework. Some methods of the objects will be optional to be used depending on the application’s requirements and some will be mandatory. This will lead to more flexibility and more benefits that may achieved from ignoring some parts in some applications.

In this research we focus on a specific kind of network distribution, which is clustering hierarchy distribution. This kind of distribution has been demonstrated by many researches to be the best choice in many wireless sensor
networks applications [1, 2, 3, 4, 5, 6, and 7]. For our solution to be more powerful, we will focus on Dynamic Clustering Hierarchy. Dynamic Clustering Hierarchy has also been shown by many researches to have better security, better performance, and more flexibility than other solutions [1, 2, and 7].

Our paper organized as the following: In section one; we discuss some related work of our solution. In section three, we present the three objects that will form the framework. The three objects are Routing Protocol, Key Management, and Intrusion Detection and Recovery. In section four we present an analysis and results of some experiments we applied on our solution. In section five, we conclude our paper.

2. Related Work

Our proposal presents a complete solution that contains issues related to WSN communications. In order to give a clear idea of each component of SOOAWSN, we review in this section some of existing research related to each part of our proposal. This section is divided into: Routing Protocols, Key Management Solutions, and Intrusion detection solutions.

2.1 Routing Protocols

For any kind of networks, the way that nodes communicate with each other is considered as the most important issue. Communication consists of transmitting data from party A to party B which may involves some other nodes as connectors between those parties. The main target is to get these data arrived to the destination accurately with minimum cost. For WSN, cost includes the power consumption, computational needed and the space required. Many researches proposed variety of solutions to accomplish a reliable communications. In this paper we consider some of these proposals which mainly relates to a specific kind of WSN architecture, which is Cluster Hierarchy architecture.

Heinzelman et al. [1] presented a technique called Low-energy adaptive clustering hierarchy (LEACH). LEACH is a Cluster based routing protocol which minimizes the energy consumption of WSN. The idea is to have the sensors elect among themselves CHs in one complete cycle of communication. Each cycle contains two phases, Setup phase and Steady-State phase. The election depends on the desired percentage of CHs and the time passed for the sensor since it has been CH. LEACH does not follow hop-by-hop routing, and it is non-energy aware algorithm.

Santos et al. [8] proposed Minimum Energy Communication Network (MECN). MECN utilizes low power geographic positioning system (GPS) which establishes and maintains a minimum energy network for wireless networks. The basic idea of MECN is to locate the sub-network with the smallest number of sensors that requires the minimum transmission power between any two particular sensors.

2.2 Key Management Solutions

Although many of routing protocols provide some ways to secure communications, there is still a need to provide better security to such protocols. The architecture of the security model of WSN consists of different parts. First part is the data encryption technique. Second part is the creation of the security keys which to be used with such encryption techniques. The third part is the way to distribute these keys upon the nodes in the network. In addition to these parts, there is an important part that should be covered, which is the Key Refreshing part.

Du and et al. [9] presented Pair-wise Key Pre-distribution for WSN. The idea here is to provide each sensor with at least one pair of keys prior to network deployment. These keys then can be used for sensor-sensor communication. The main target was to reduce the benefits of limited attacks by the intruders. Advanced version introduced later invests the knowledge of sensors locations prior to network deployment. This provides the sensors expectations of the nearest neighbors, even they are randomly deployed.

Eschenauer and Gligor [10] proposed a probabilistic key pre-distribution scheme. In this scheme, each sensor is provided with a subset of keys picked randomly from a large key pool prior to network deployment. Two sensors have to find one common key within their subsets and use that as their shared key in order to agree for sensor-sensor communication.

2.3 Intrusion detection solutions

Currently, there are little studies in the area of intrusion detection in WSN. In this section we present some existing works on Intrusion detection in WSN and Ad-Hoc Networks.
Silva and et al. [11] proposed Decentralized intrusion detection in wireless sensor networks. In their proposal, they suggest that nodes to be responsible of monitoring other nodes behavior. Each such node listens to traffic in its radio range to detect any abnormal behavior. These messages are provided to an Intrusion detection system. The details of how this system works are not discussed in [11].

Onat et al. [12] propose similar technique to which that has been proposed by Silva and others [11]. The authors do not include details of how the real Intrusion-detection system works. In both previous works, there is no cooperation between monitoring nodes. Instead the detection is executed locally in each node.

3. SOOAWSN framework

Our framework is mainly divided into three objects. These objects should represent and cover the complete lifecycle of network communication. We call this framework a Secure Object Oriented Architecture for Wireless Sensor Network (SOOAWSN). The objects are: Routing Algorithm, Key Management, and Intrusion detection (see Fig.1). Each object contains different parts, which we call methods, which represent this object. In this section we discuss, in more details, these objects and the methods represent these objects to give a complete picture of our framework.

3.1 Routing Algorithm

The operation of our proposed algorithm occurs in rounds. Rounds are classified into two kinds; the full transmission round and the half transmission round. At each round, new clusters are to be formed with new CHs. At the beginning of each round, sensors elect themselves to become CHs for current round. In order to determine the eligibility of a sensor to be a CH, each sensor (S) generates a random number between 0 and 1. Then this number is compared to a sensor variable threshold value T(S). If the value of the threshold is greater than the random number, the sensor is eligible to become a CH for the current round (R).

The Threshold value is calculated as follow:

First it calculates the threshold for one round [1]:

\[
T(S) = \begin{cases} 
\max(P/\left(1-P \mod 1/P\right), \text{Rem/Mx}, T_{\text{min}}) & \text{if } S \in G \\
0 & \text{otherwise} \end{cases}
\] ... (1)

Second, it calculates the threshold for two rounds

\[
T(2) = \begin{cases} 
\max(\text{Rem}/\text{Mx}, T_{\text{min}}) & \text{if } S \in G \\
0 & \text{Otherwise} \end{cases}
\] ... (2)

If formula (1) is approved, then it is ready to become a CH for one round. In this case, the sensor will check the result of formula (2) to see if it is able to become a CH for two rounds. Where P is the desired percentage of CHs.
Tmin is a minimum threshold (to avoid the possibility of remaining energy shortage), G is the set of sensors that have not been CHs in 1/P rounds, Mx is the maximum energy that the sensor could have, and Rem is the sensor remaining energy.

Each elected CH starts listening to the network; where other sensors start broadcasting their messages to their neighbors (using Carrier sense multiple access protocol for transmission to avoid collisions). When ready sensors (CHs) receive the messages, they temporarily save each report with the node ID in their memories. Then, each CH sends requests with confirmation to those sensors indicating that it is ready to become their CH for the current round. When formula (2) is applied, it also indicates that the CH is also ready to become these sensors CH for the next round.

Sensors then, reply to CHs with confirmation messages. When a CH receives the confirmations, it combines all the reports that it has in one compressed report and forwards it to the BS. The report message contains: CH id, BS id and the aggregation report.

In the next round, sensors check first if they are group members of a CH with an ability to handle two rounds. If they are, then they use this CH for the current round (half transmission round is applied. Sensors who receive the acknowledgment, remove CH information from their memories. These actions appear in one round which is limited by duration of time. If the time passed and sensors don not receive acknowledgement from other corresponding sensors it will ignore the previous communications with those sensors and proceed with the current data they carry. This limitation prevents the appearance of the deadlock. In the case that the sensor does not have a CH from the previous round, it will repeat the first scenario for full round transmission.

3.2 Key Management Solution

Key Management is one of the core requirements in any security application, especially in wireless communications. It is the way of creating and distributing keys among nodes in a network in order to be used for data encryption during nodes communication. In this part, we are proposing three methods that we plan to apply to the solution to achieve different levels of security.

3.2.1 Key Pre-distribution (KP) Method

The idea of this method is to create a pool of keys at the BS that has specific number of keys generated randomly using a pseudorandom generator function. At the same time, the BS 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 sensor-sensor communication possible (these keys called sharing keys). Meanwhile, the BS provides each sensor with at least one unique key (named Master Key) which is to be used to communicate between each sensor and the BS.

When sensors need to communicate with each other, they encrypt their messages with the keys that are shared between the two communication parties (if any). To do hand-checking between the two communicating parties they exchange the information of which keys they have by providing their keys ids. If at least one shared key exists between these parties, then the communication can be established between those parties. The communication between the sensors and the base station can be encrypted using the Master Key for each sensor.

This algorithm is proposed in [13] and it is expected to provide: authenticity, confidentiality, and keys 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.

3.2.2 Public and Private Keys Method

In this method, each sensor uses two keys to communicate with each others; 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 uses these keys for different purposes (i.e. authentication, digital signature, and keys exchange).

3.2.3 Multi-generations Keys Method
This method relates to the first method in our key management solution. The idea is to reuse the keys that produced from the key pool in Key-Predistribution technique to support key refreshes and to support the expansion of the current sensor network.

For Key-Predistribution method, BS creates a key pool that contains numbers of keys with their IDs. The keys are randomly distributed upon the sensors in order to have some sharing keys between sensors to use during their communications. This technique does not support the ability to refresh the keys implicitly in the future. In order to change a key, the BS needs to securely communicate with the related sensors and inform them with the new keys. Furthermore, this technique does not efficiently support the expandability of the network by adding or replacing sensors.

Our 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 BS randomly generates number of keys and assigns key ID to each key. The key has to start with a specific tag that represents the first generation of key (i.e. 001 for example). The BS then randomly distributes groups of these keys on the sensors prior to network deployment. The BS station also distributes a formula of one way function to all sensors. In addition to that, the BS 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 BS 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). BS then distributes the new group of keys on its new sensors. Moreover, BS may broadcasts some updated keys IDs 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 generation tag to see if there are any IDs in common. In the case that the receiver shares some keys with the sender, it then checks the flag of these IDs to see if it is from the same generation. Then the Key-Predistribution method can be applied without modifications. If the key is from a 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 Fig2).

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.

3.3 Intrusion Detection Solution

Network monitor is a mandatory requirement in any WSN application to guarantee network stability. The main target for network monitoring is to detect any misbehavior of the network communications. Usually this misbehavior occurs from intruders in the network that may affect the network work or affect the privacy of this network. In this section, we discuss how to detect such kind of intrusion and how to recover from it. In order to do that, we propose two methods of intrusion detection techniques that can be used during wireless sensor networks communications. These methods perfectly fit all kinds of wireless sensor networks that follow the clustering hierarchy.

The first method adopts public and private keys to authenticate the sensor. We called this method a Public-Private method. The algorithm works as follows (Fig3):

1) Sensor A send a message to sensor B. Part of this message (i.e. signature) is encrypted using A’s private key. The signature part consists of A’s ID and a nonce.
2) Sensor B decrypt the signature part using A’s public key. It then compares this part with the external part that consists of A’s ID and the nonce. If they are not matched, then A’s is considered as a suspected node.
3) If sensor B detects any suspected node, it informs the BS of that node. In order not to consume extra energy, sensor B sends this information as part of its regular report.
4) The BS compares the suspected sensor ID with the IDs in a table contains all sensors IDs. If the ID does not exist in its table then go to step6.
   If the ID exists then the BS stores the sensor information in a table called Suspected-Nodes table.
5) If suspected sensor ID’s is found in the suspected table then go to step6.

6) The BS broadcasts a warning signal to all sensors to ignore future communication with that sensor and terminate or renew all the keys that are shared with that sensor.

The Public-Private method results in few data overhead that is produced from the addition part added to the original message, which is the digital signature. However, we try to reduce the attributes needed to build this signature using only the sensor ID and the nonce. This will decrease the data overhead required to build such signature compared to traditional Public key authentication that is used in the traditional networks.

The second method called a Routing Attack method. It provides the ability to detect any attack that may affect the information forwarded to the BS. The algorithm works as follows:

1) Sensor A includes its previous activity in the report forwarded to the BS. This activity contains the ID of the CH who was responsible of forwarding the previous message from that sensor. The serial number of the message is also included in the activity part of the report (Fig4. a.).

2) The BS stores all activities in the network, and each time it receives new information, it compares it with the information it has regarding the activities. Any missing or mismatching information will indicate a problem that may involve two parties, the sensor itself and its previous CH (Fig4. b.).

3) If the BS finds a frequent information mismatching related to the same CH or the same sensor, it will decide the compromised node or the intruder.

4) The BS broadcasts a warning signal to all sensors to ignore future communication with that sensor and terminate or renew all the keys that are shared with that sensor.

In Routing Attack method, the number of sensors compares to the number of CHs in the network will determine the efficiency of its function. Increasing the number of sensors for a constant number of CHs will result that each CH has higher number of sensors in its group. If the CH has been compromised or in case that it is an intruder, then the number of reports that are going to be sent to the BS in the next round regarding this intrusion will be higher. This helps the BS to make a quick decision regarding this attack. On the other hand, this will lead to more damage to the network in that specific cycle where the attack occurs. This concluded from the high number of the sensors connected with that CH. It is the responsibility of the BS to choose a typical percentage of desired CHs which is to be changed during the network lifecycle depends on the number of sensors in the network.

Fig 5 shows the relation between the number of reports sent to the BS and the number of CH in a network with N nodes. It shows different values ranging from 100 sensors network size to a 1000 sensors network size with different percentages of the desired CHs ranging from 0.01 to 0.1. It shows that, for a specific network size, increasing of the desired percentage of CHs will decrease the number of the sensors involved in the attack which will also decrease the number of the reports that sent to the BS in the next round.
4. Results and Analysis

We applied parts of the solution in some experiments and we compared them with some existing solutions using MATLAB. In this section we discuss, in details, one of these experiments. In this numerical experiment, we first describe the chosen parameters for each protocol. Then we apply the same scenario on each protocol. The experiment is applied on LEACH, TCCA, and our routing algorithm. LEACH and TCCA considered of the most powerful Cluster Hierarchy techniques proposed until now [1, 2]. We applied them to three different network sizes (100, 1000, and 10000 sensors).

We analyzed the results collected from applying this experiment on each of the protocols using the same initial values and following the same scenario. We started with comparing the results based on energy saving results from each protocol. Then we analyzed them based on data overload produced by each protocol. Finally, we compared the results based on the number of the dead sensors produced at the end of the experiment for each protocol.

4.1 Energy saving

Our experiment shows that the variation of energy consumption is very small when the network size is small (i.e. 100 sensors), but it varies more if we increase the network size. Figure13 shows that, for network size of 10,000 sensors, total energy consumption is minimum in our algorithm with almost 0.1X1012 nj, then TCCA comes with energy consumption of almost 1.2X1012 nj at second place, and LEACH comes last with 3.3X1012 nj (see Fig6. a.). The variation comes from the nature of how our algorithm works. Using TTL, in addition to continuous checking of residual energy of each sensor, gives our algorithm and TCCA more energy balancing for a large network size. Double round technique provides our algorithm with more energy saving.

4.2 Data Overload

Our experiment shows that, for a large network size (i.e. 10000 sensors), the total data overhead is minimized using our algorithm. Figure14 shows that, data overhead reaches almost 0.1X10^{12} bits in our algorithm, where in TCCA it reaches 2.2X10^{12} bits and in LEACH it reaches 9.3X10^{12} (see Fig6. B.). This shows that LEACH produces...
more data overhead, almost nine times more than the data overhead produced by our algorithm. Moreover, TCCA produces more data overhead, almost twice, than data overload produced by our algorithm.

4.3 Performance

We analyze the performance of the WSN based on the expected dead sensors produced in each solution after the same number of rounds. According to energy saving analysis, we can figure out that the number of dead sensors that may appear in LEACH will be much higher than the number of dead sensors in our algorithm, where the number of dead sensors depends on the energy consumption by the network.

The results show that our algorithm and TCCA remains completely alive (i.e. no dead sensors) after 1000 rounds. On the other hand, LEACH results in a large number of dead sensors in the case of 10000 sensors.

5. Conclusions

We proposed a new framework that provides WSN with different levels of security with less energy and better performance. Our results show the efficiency of our framework when we compare it to other existing solutions. The flexibility of our framework, to be applied as parts or as one part, gives it more advantages when we compare it to other solutions.

6. References


