ISSN:0975-9646
Venkatesh Ladi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 11 (6) , 2020, 125-129

Secured Data Transmission Using Key Exchange Mechanism for Wireless Sensor Networks

Venkatesh Ladi[1], G. BHAGYA LAKSHMI[2]
[1]II [nd] year Student, M. Tech, *Department of CSE, GITAS*
[2]Asst. Professor,*Department of CSE, GITAS*

Abstract: Many wireless sensor networks (mainly networks of mobile sensors or networks that are deployed to monitor calamity situations) are deployed in an arbitrary and unplanned fashion. For any sensor in such a network can end up being adjacent node to any other sensor node in the network. To Establish a secure communication between every pair of adjacent sensors node in such a network, each sensor node x in the network needs to store n -1 number of symmetric keys that sensor node x shares with all the other sensor nodes, where n is the number of sensor nodes in the present network. This memory storage requirement of the keying protocol is various, especially when n is large and the available storage in each sensor node is modest. Previous efforts to redesign this keying protocol and reduce the number of keys to be stored in each sensor node have produced protocols that are vulnerable to impersonation, eavesdropping, and collusion attacks. In this paper, we present a fully secure protocol mechanism where each sensor node needs to store (n+1)/2 keys, which is much less than the n-1 keys that need to be stored in each sensor in the original communication protocol. We also show that in any fully secure keying protocol, each sensor node needs to store at least (n -1)/2 keys.

Keywords: sensor, key sender, encryption, decryption, sensor networks, MWSN, plain-text, cipher-text, grid, probabilistic, keying protocol

INTRODUCTION

When sensor nodes want to communicate with each other, there has to be a secure connection that has to be established between nodes. The same is the case with the sensor nodes. Communication between sensor nodes has to be protected from external attacks after establishing a session of communication between each other.

Communication nodes (Sensors) are tiny devices with small size, less computation power, and a transmission range. Moreover, the positions of Communication nodes (Sensors) need not be constant; they may be moving or dynamically shift with respect to time. So, it is unavoidable that the data needs to be transferred and kept correct and secure. In the early days when sensors were first introduced there were many problems related to security. The data that was being communicated Either was too big in volume or the secure transmission was suffering a lack of proper care.

Therefore the concept of Cryptography was introduced. Cryptography ensured that the data that is being communicated is secure and also the data size is also small.

Sensors: Sensors are known as devices that calculate a physical quantity and convert it into an electric wave signal which can be understood and read by a receiver or a device. It is usually needed to communicate confidential data securely without being attacked by external world. Examples: Heat Sensors, SONAR (Battle fields), Security Alarm Systems.

Sensor Networks: A sensor network is designed with a communication infrastructure planned to record, monitor conditions at multiple locations. Multiple detection stations called sensor nodes exist in a sensor network, each of which is lightweight, tiny, and easy to carry devices. Every sensor node is equipped with a power source and transducer. The transducer produces electrical signals based on sensed phenomena and physical effects. The microcomputer processes and stores the sensor output. The power for each sensor device is obtained from the electric utility or from a battery.

Mobile wireless sensor networks (MWSNs):

Mobile wireless sensor networks aka MWSNs are usually defined as a wireless sensor network (WSN) in which the sensor devices (nodes) are easily movable. MWSNs are an emerging field of research in contrast to their well-established predecessor and they are smaller. Many of their applications are similar, such as surveillance or environmental monitoring.

Key distribution within Sensor Networks:

Within wireless sensor network (WSN) design Key distribution is an important issue. Due to power and memory limitations, the keys need to be well arranged to build a fully functional network.

Key distribution (distribution of keys to sensor nodes) happens before deployment. Initially, sensors detect their close by sensors in the network and transmit the information to the key sender. For secure transmission, the Key Sender then generates secret keys and sends them to respective sensor nodes.

Encryption and Decryption of data:

In cryptography, Encryption is the process of encoding messages or information in such a way that only authorized parties can read it. In an encryption scheme, the message or information, referred to as plaintext, is encrypted using an encryption algorithm, turning it into an unreadable ciphertext.

Decryption: the data which has been encrypted into a secret format is known as the process of decoding. Decryption requires a secret key or password, so only authorized users can only decrypt data. In simple meaning, it is the conversion of cipher text into plain text.

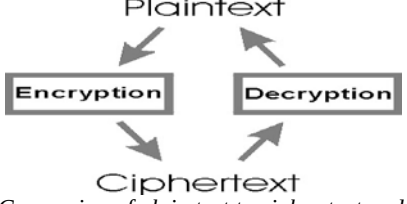

Figure 1: Conversion of plain text to cipher text and vice versa

Two main protocols were put forward in the past to minimize the number of stored keys in each sensor node in the network. We refer to these two protocols as the subjective Grid Keying Protocol and Keying Protocol. Each sensor in the network stores multiple keys that are chosen at random from a large set of keys In the Probabilistic Keying Protocol. When two side by sensor nodes need to communicate, the two sensors nodes recognize the shared keys then use a combination of their common keys as a symmetric key to encrypt and decrypt their transferred data messages. Since a particular sensor node only has a set of shared keys and doesn't have its own universal key it is prone to impersonation attack. Each sensor is allocated a unique identification number which is used to coordinate a distinct node in a two-dimensional space and each symmetric key is also assigned an identifier which is known as Grid Keying Protocol, . Then a sensor node x stores a symmetric key K if and only if the identifiers of x and K satisfy certainly given relation. When 2 adjacent sensors have to transfer the data, the two sensor nodes identify common keys and use a combination of those shared keys as a symmetric key to decrypt and encrypt data messages. The grid keying protocol has two advantages. First, this protocol can defend against impersonation - unlike the probabilistic protocol and can defend against eavesdropping - like the probabilistic protocol. Second, each sensor node in this protocol has to store only O(log n) symmetric keys, where n is the no. of sensors in the network. But the main issue with grid keying protocol is it may not protect itself from collision attack.

In our proposed paper we want to show that there could be a system with a keying protocol that reduces the number of keys maintained within the sensor to (n+1)/2 keys. The extra and very predominant feature that has been introduced is that of a Key Sender. All Sensor Networks has a corresponding Key Sender associated with it. Every sensor node has to be registered within the Key Sender. The Key Sender then distributes/shares the keys to the sensor nodes within the network. With the help of the keys distributed by the Key Sender the sensors communicate and transfers with the other sensors.

We use ix and Iy to symbolize the identifiers of sensor nodes 'x' and 'y', respectively, in this network. Each two sensors, say sensor nodes 'x' and 'y', share a symmetric key denoted K(x, y) or K(y, x). Only the two sensor nodes 'x' and 'y' know their shared key K(y, x). And if sensor nodes 'x' and 'y' ever become neighbors in the network, then they can use their shared symmetric key K(y, x) to perform two functions:

1) **Mutual Authentication**: Sensor 'x' authenticates sensor 'y', and sensor y authenticates sensor 'x'.

2) **Confidential Data Exchange**: Encrypt and then decrypt all the transferred data messages between 'x' and 'y'. In the rest of this section, we wanted to show that if the shared symmetric keys are designed to have a "special structure", then each sensor needs to store only (n+1)/2 shared symmetric keys. We need to introduce two new concepts before we present the special structure of the shared keys, : "Universal Keys" and "a circular relation, named below, over the sensor identifiers". Each sensor node 'x' in the network keeps a symmetric key, called the universal key of sensor 'x'. The universal key of sensor node 'x', denoted 'ux', is known only to sensor node 'x'. Let Ix and iy be two distinct sensor node identifiers. Identifier ix is said to be below identifier Iy if one of the below conditions holds:

1) $I_x<I_y$ and $(I_y - I_x) < n/2$

2) $I_x>I_y$ and $(I_x - I_y) > n/2$

The below relation is better explained by an example. Consider the case where n / 3. In this case, the sensor identifiers are 0, 1, 2

We have:

- Identifier 0 is below identifiers 1 and 2.
- Identifier 1 is below identifiers 2 and 0.
- Identifier 2 is below identifiers 1 and 0.

Methodology

Theorem 1: If there exists a pair of distinct but adjacent sensors 'x' and 'y' with unique identifiers 'I_x' and 'I_y' respectively then the Below condition holds true as follows :-

- 'I_x' is below 'I_y'
- 'I_y' is below 'I_x'

Theorem 2: Since there exists 'n sensors, each sensor 'x' with identifier ix has (n-1)/2 sensor identifiers I_y below it.

Theorem 3: In accordance with Theorem 1,the number of sensors with identifiers ix below the sensor 'y' with identifier I_y is (n-1)/2.

Theorem 4: If a sensor identifier ix for sensor 'I_x' is below a sensor identifier 'I_y' then the sensor 'x' needs to store the Symmetric Key '$k_{y, x'}$/ H ('I_x'|u_y) within it. Then the sensor 'y' needs to compute the Symmetric Key to verify the sensor 'x'. The Symmetric Key '$k_{(y, x)'}$ is stored only in 'x'.

Theorem 5: As discussed earlier, each sensor 'x' needs to store single Universal Key and (n-1)/2 Symmetric Keys '$k_{(y, x)}$' in order to communicate with sensor 'y' (NOTE: the sensor identifier ix should be below 'I_y').

3. A Mutual Authentication Protocol:

Each and every sensor 'x' is provided with the following information before the sensors are deployed within the network:-

1) One distinct identifier ix in the range 0-(n-1)

2) One universal key u_x

3) (n-1)/2 symmetric keys $K_{(y, x)}$ / H ('I_x'|u_y) each of which is shared between sensor 'x' and another sensor 'y' (where ix is below I_y). If the sensors 'x' and 'y' are adjacent and want to communicate with each other, then they must implement the Mutual Authentication protocol which has the following steps :-

Step 1: Sensor 'x' selects a random nonce n_x and sends a hello message that is received by sensor 'y'. x ->y: hello('I_x'|n_x)

Step 2: Sensor 'y' selects a random nonce n_y and sends a hello message that is received by sensor 'x'. x ->y: hello ('I_y'|n_y)

Step 3: Sensor 'x' determines whether ix is below iy. Then it either fetches $K_{(y, x)}$ from its memory or computes it. Finally, sensor 'x' sends a verify message to sensor 'y'.
x->y: verify ('I_x'; 'I_y'; H (I_x | I_y |n_y|$K_{(y, x)}$))
Step 4: Sensor 'y' determines whether iy is below ix. Then it either fetches $K_{y,x}$ from its memory or computes it. Finally, sensor 'y' sends a verify message to sensor 'x'. x->y: verify (I_y| I_x |H ((I_y| I_x |n_x|$K_{y,x}$))
Step 5: Sensor 'x' computes H (I_y| I_x |n_x|$K_{(y, x)}$) and compares it with the received H(I_y|I_x|n_x|$K_{(y, x)}$). If they are equal, then sensor 'x' concludes that the sensor claiming to be sensor 'y' is indeed sensor 'y'. Otherwise, no conclusion can be reached.
Step 6: Sensor 'y' computes H(I_x | I_y |n_y|$K_{(y, x)}$) and compares it with the received H(I_x | I_y |n_y|$K_{(y, x)}$). If they are equal, then 'y' concludes that the sensor claiming to be sensor 'x' is indeed sensor 'x'. Otherwise, no conclusion can be reached.

4. A Data Exchange Protocol:
Sensors 'x' and 'y' can now start exchanging data according to the following data exchange protocol:-
Step 1: Sensor 'x' combines the nonce n_y with the data to be sent, encrypts the combined data using the symmetric key $K_{(y, x)}$, and sends the result in a data message to sensor 'y'.
x ->y: data(I_x | I_y |$K_{(y, x)}$(n_y|text))
Step 2: Sensor 'y' combines the nonce n_x with the data to be sent, encrypts the combined data using the symmetric key $K_{y,x}$, and sends the result in a data message to sensor 'x'.
x ->y: data(I_y | I_x |$K_{(y, x)}$(n_x|text))

5. Optimality of Keying Protocol:
According to our keying protocol, described in Section III, each sensor in the network is required to store only (n+1)/2 keys. Thus, the total number of keys that need to be stored within the network is n(n+1)/2.
Theorem 6: There should be a minimum of n(n-1)/2 keys that are to be stored within the sensor network.
Theorem 7: According to any keying protocol (which is uniform) has to store at least (n-1)/2 keys within it to communicate with its adjacent sensors.

6. Triple Data Encryption Algorithm (TDEA):
DES (the Data Encryption Standard) is a symmetric block cipher developed by IBM. The algorithm uses a 56-bit key to encipher/decipher a 64-bit block of data. The key is always presented as a 64-bit block, every 8th bit of which is ignored. However, it is usual to set each 8th bit so that each group of 8 bits has an odd number of bits set to 1.
The algorithm is best suited to implementation in hardware, probably to discourage implementations in software, which tend to be slow by comparison. However, modern computers are so fast that satisfactory software implementations are readily available.
DES is the most widely used symmetric algorithm in the world, despite claims that the key length is too short. Ever since DES was first announced, controversy has raged about whether 56 bits is long enough to guarantee security. The key length argument goes like this. Assuming that the only feasible attack on DES is to try each key in turn until the right one is found, then 1,000,000 machines each capable of testing 1,000,000 keys per second would find (on average) one key every 12 hours. Most reasonable people might find this rather comforting and a good measure of the strength of the algorithm.
Those who consider the exhaustive key-search attack to be a real possibility (and to be fair the technology to do such a search is becoming a reality) can overcome the problem by using double or triple length keys. In fact, double length keys have been recommended for the financial industry for many years.
Use of multiple length keys leads us to the Triple-DES algorithm, in which DES is applied three times. If we consider a triple length key to consist of three 56-bit keys K1, K2, K3 then encryption is as follows:
• Encrypt with K1
• Decrypt with K2
• Encrypt with K3
Decryption is the reverse process:
• Decrypt with K3
• Encrypt with K2
• Decrypt with K1
Setting K3 equal to K1 in these processes gives us a double length key K1, K2. Setting K1, K2 and K3 all equal to K has the same effect as using a single-length (56-bit key). Thus it is possible for a system using triple-DES to be compatible with a system using single-DES.

7. Data Analysis
Key Sender: It detects the sensors present in its regionand updates their details in its table. Here we used java simulation to create an interface to key-sender. Symbolically it may look like(before sensors detection and after detection)

ID	IP Address	Universal Key

Figure 2: KeySender table before detection

ID is the unique number given to each sensor, IP address is the logical address and universal is the symmetric key used for encryption and decryption
Key Sender Table after sensor detection:

ID	IP Address	Universal Key
0	127.0.0.1	98
1	127.0.0.1	8

Figure 3: Key-sender table after clients/sensors are detected

Here we take 2 sensor nodes or clients (say receiver 0 and receiver 1) which are detected by the key sender. The key sender then calculates the universal keys(as shown in figures) and sends them to the respective clients

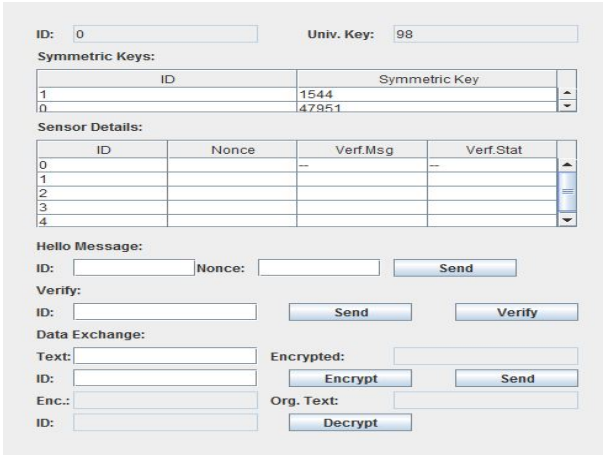

Figure 4:Receiver0

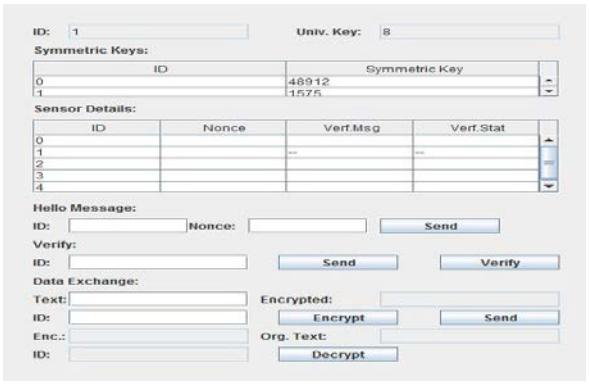

Figure 5:Receiver1

After the key sender sends the symmetric keys to both the receivers the clients now look like

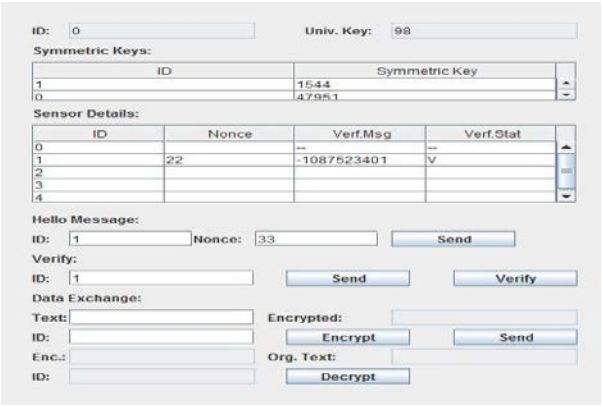

Figure 6:Reciever0 with keys updated

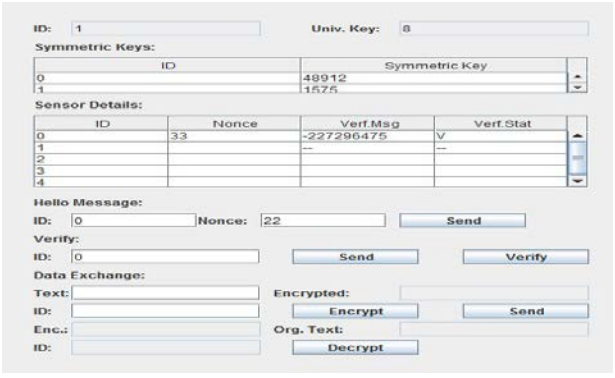

Figure 7:Receiver1 with keys updated

After the keys are updated both the nodes need to authenticate each other for that they send verification messages and confirm the authentication. After authentication is done message transfer is done using encryption and decryption.

Message transfer done by Receiver0-original message is hello, it is encrypted and sent. The encrypted message from Receiver1 is decrypted

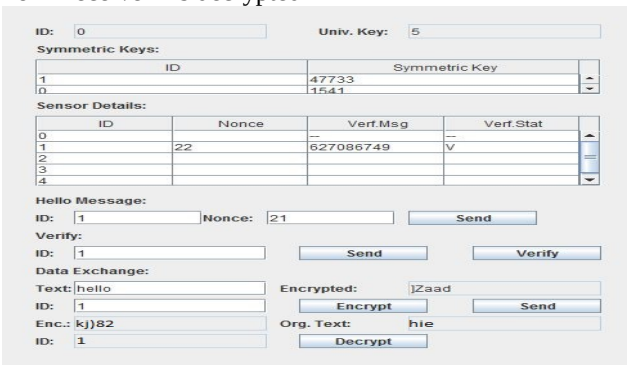

Figure 8: Receiver0 sending and receiving messages

Message transfer done by Receiver1-original message is hie, it is encrypted and sent. The encrypted message from Receiver0 is decrypted

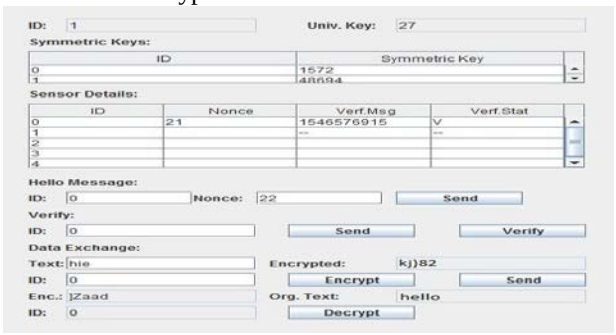

Figure 9: Receiver1 sending and receiving messages

Whenever a receiver0 wants to communicate with receiver2, it cannot communicate directly, first of all the receiver0 must communicate with receiver1 and then the receiver1 communicates that message with receiver2

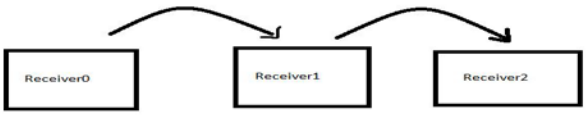

Figure 10: nodes communication

REFERENCES

1. Communication within Sensor Networks by Using Key Distributor by ch.d.naiduijcsit 2014.
2. Taehwan Choi, H. B. Acharya and Mohamed D. Gouda, " The Best Keying Protocol ", December 2011 IEEE
3. http://en.wikipedia.org/wiki/Sensor -Sensors
4. Sensor Networks by Margaret Rouse.
5. http://en.wikipedia.org/wiki/Mobilewirelesssensornetwork-MWSN's
6. "Key Distribution Mechanisms for Wireless Sensor Networks" by Seyit A., C¸Amtepe and BulentYener
7. Communication within Sensor Networks by Using Key Distributor by M.V.Kishore in International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 4906-4910

8. "Cryptography and Network Security", Fourth Edition by William Stallings
9. L. Gong and D. J. Wheeler, "A matrix key-distribution scheme," Journal of Cryptology, vol. 2, pp. 51–59, January 1990.
10. "Advanced Encryption Standard" by Douglas Selent
11. http://en.wikipedia.org/wiki/Advanced_Encryption_Standard#cite_note-fips-197-4.
12. "Comparative Study of Energy Aware QoS for Proactive and Reactive Routing Protocols for Mobile Ad-hoc Networks". International Journal of Computer Applications (0975 – 8887) Volume 31– No.5, October 2011.
13. Steganography Detection using Functional Link Artificial Neural Networks, International Journal of Computer Applications (0975 - 888), Volume 47 No.5 June 2012.
14. Secure Group Communication using Multicast Key Distribution Scheme in Ad-hoc Network, International Journal of Computer Applications. (0975 - 8887)Volume 1 – No. 25, Nov-2010.
15. A Novel Dual Phase Mechanism for Data Transmission to Provide Compression and Security by M.V.Kishore in International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 12, December 2013 ISSN: 2277 128X