Design of Secured Data Transmission in Wireless Sensor Network using Blowfish Algorithm

[1]Yashodha Bilagi, [2]Vinutha C B , [3]M Z Kurian

[1]M.Tech Digital Electronics Dept. of ECE SSIT, Tumakuru 572105, Karnataka, India
[2]Assistant Professor Dept. of ECE SSIT, Tumakuru 572105, Karnataka, India
[3]HOD Dept. of ECE SSIT, Tumakuru 572105, Karnataka, India

ABSTRACT

In Wireless Sensor Network, Information Security has been very important issue during data communication. Any loss or threat to the information can prove to be great loss to the organization[2]. Hence, securing the data is very important in wireless sensor networks. Data security helps to keep the private data to be private. The work considers the security issues to identify the problems in wireless sensor networks. In this direction, the proposed work considers a technique, known as Blowfish Algorithm, which is perfect for use in wireless sensor networks. This Blowfish algorithm uses two processes for data transformation: Encryption and Decryption. The Encryption process converts the plaintext message into cipher text and Decryption process converts the cipher text into back plaintext message.

Keywords: WSN, Blowfish algorithm, encryption, decryption.

1. INTRODUCTION

Wireless Sensor Network can be defined as a special class of ad hoc wireless network that can be used to provide a wireless communication infrastructure that allows us to sense, observe and react to the events & phenomena in the natural environment[1].

Most of the threats and attacks against security in wireless networks are almost similar to their wired counterparts while some are exacerbated with the inclusion of wireless connectivity. Wireless sensor networks are usually more vulnerable to various security threats as the unguided transmission medium is more susceptible to security attacks than those of the guided transmission medium. The wireless sensor networks could have a base station.

The architectural aspect of wireless sensor network could make the employment of a security schemes little bit easier as the base stations or the centralized entities could be used extensively in this case. Nevertheless, the important challenge is induced by the constraint of resources of the tiny sensors. In many cases, sensors are expected to be deployed randomly in the enemy territory (especially in military scenario) or over dangerous areas. Therefore, even if the base station (sink) resides in the friendly or safe area, the sensor nodes need to be protected from being compromised[6].

Blowfish Algorithm is a Feistel Network, iterating a simple encryption function for 16 times. Blowfish is a 64-bit block cipher with a variable-length key. The block size is 64 bits, and the key can be any length up to 448 bits. However, there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors[8].

2. SECURITY THREATS AND ISSUES IN WIRELESS SENSOR NETWORKS

Wireless Sensor Networks are vulnerable to security attacks due to the broadcast nature of the transmission medium. Basically attacks are broadly classified in two categories i.e. active attacks and passive attacks. This paper points out both of these attacks in details.

2.1 Passive Attacks

The monitoring and listening of the communication channel by unauthorized attackers are known as passive attack. Some of the more common attacks against sensor privacy are:

2.1.1 Monitor and Eavesdropping:

This is the most common attack to privacy. By snooping the data, the adversary could easily discover the communication contents.

2.1.2 Camouflage Adversaries:

One can insert their node or compromise the nodes to hide in the sensor network. Then these nodes can copy as a normal node to attract the packets, then misroute the data packets, conducting the privacy analysis.

2.2 Active Attacks

The unauthorized intruders monitors, listens to and modifies the data stream in the communication channel are called as active attack. The following attacks are active in nature.

2.2.1 Routing Attacks in Sensor Networks:

The attacks which act on the network layer are called routing attacks. Below are the attacks that happens while routing the messages.

2.2.1.1 Attacks on Information in transit:

In a sensor network, sensors monitor the changes of specific parameters or values and report to the sink according to the requirement. While sending the report, the

data in transit might be altered, spoofed, replayed again or vanished. As wireless communication is vulnerable to eavesdropping, any intruder can monitor the traffic flow and get into action to Interrupt, intercept, modify or fabricate packets thus, provide wrong information to the base stations or sinks.

2.2.1.2 Selective Forwarding:

A malicious node can selectively drop only certain packets. Especially effective if combined with an attack that gathers much traffic through the node. In sensor networks it is assumed that nodes faithfully forward received messages. But some compromised node might refuse to forward packets, however neighbours might start using another route.

2.2.1.3 Black hole/Sinkhole Attack:

In this attack, a malicious node acts as a black hole to attract all the traffic in the sensor network. In fact, this attack can affect even those nodes which are considerably far from the sink node. Figure 2.1 shows the conceptual view of a black hole/sinkhole attack.

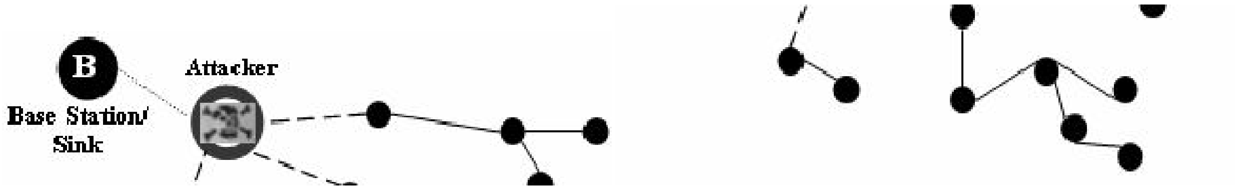
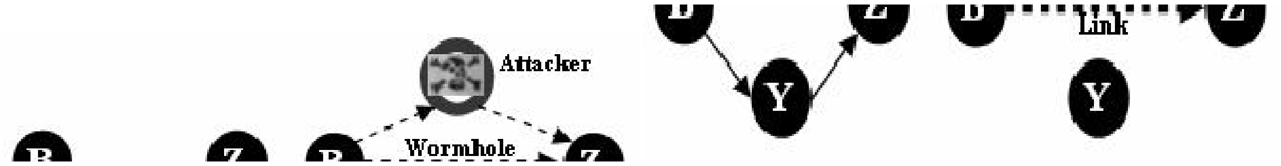


Figure 2.1 shows the conceptual view of a black hole/sinkhole attack.

2.1.1.1 Wormholes Attacks:

Wormhole attack is a critical attack in which the attacker records the packets (or bits) at one location in the network and tunnels those to another location.



(a) (b)

Figure 2.2: Wormhole Attack

Figure 2.2 (a and b) shows a situation where a wormhole attack takes place. When a node B (for example, the base station or any other sensor) broadcasts the routing request packet, the intruder receives this packet and replays it in its

neighbourhood. Each neighbouring node receiving this replayed packet will consider itself to be in the range of Node B, and will mark this node as its parent. Hence, even

if the victim nodes are multi-hop apart from node B, intruder in this case convinces them that B is only a single hop away from them, thus creates a wormhole.

2.2.1.4 HELLO flood attacks:

An attacker sends or replays a routing protocol's HELLO packets from one node to another with maximum energy. This attack uses HELLO packets as a weapon to convince the sensors in WSN.

2.2.2 Denial of Services:

Denial of Service (DoS) is produced by the unintentional failure of nodes or malicious action. In wireless sensor networks, several types of DoS attacks in different layers

might be performed.

2.2.3 Node Malfunction:

A malfunctioning node will produce inaccurate data that could expose the integrity of sensor network especially if it is a data-aggregating node such as a cluster head.

2.2.4 Node Outage:

Node outage is the situation that occurs when a node stops its function. In the case where a cluster head stops functioning, the sensor network protocols should be robust enough to mitigate the effects of node outages by providing an alternate route.

2.2.5 Physical Attacks:

Unlike many other attacks mentioned above, physical attacks will destroy the sensors permanently, hence the losses are irreversible.

2.2.6 Message Corruption:

Any modification in the content of a message by an attacker compromises its integrity.

2.2.7 False Node:

A false node involves the addition of a node by an adversary and causes the injection of malicious data. An attacker might add a node to the system that feeds false data or prevents the passage of true data. Inserting of a malicious node is one of the most dangerous attacks that can occur.

2.2.8 Node Replication Attacks:

Conceptually, a node replication attack is quite simple; an attacker seeks to add a node to an existing sensor network by copying the node ID of an existing sensor node. A node replicated in this approach can severely effect the sensor network's performance. Data Packets can be corrupted or even can be misrouted.

2.2.9 Passive Information Gathering:

An adversary with powerful resources can collect information from the sensor networks if it is not encrypted. To minimize these threats, strong encryption techniques needs to be used[7].

3. BLOWFISH ALGORITHM

Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms[8]. Blowfish became quite popular soon after its advent, just because Bruce Schneier himself is one of the most famous cryptology expert. Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. It defines 2 distinct boxes: P box and four S boxes. Taking into consideration the P box, P is a one-dimensional field with 18 32-bit values. The boxes contain variable values; those can be implemented in the code or generated during each initialization. The S boxes S1, S2, S3, and S4 each contain 256 32-bit values[3].

Basic terms used in the Blowfish algorithm:

Plain Text: The original message that we wish to communicate with the others is defined as Plain Text. The actual data that has to be send to the other is referred as Plain Text.

Cipher Text: The message which has been converted by the encryption algorithm is called cipher text. The original message is transformed into non readable message.

Encryption: A process of converting plain text into cipher text is called as Encryption. Blowfish uses the encryption algorithm and a key to send confidential data through an insecure channel.

Decryption: A reverse process of encryption is called decryption. It is a process of converting cipher text into plain text. Decryption requires decryption algorithm and a key[2].

3.1 Feistel Networks

A Feistel network is a general method of transforming any function (usually called an F function) into a permutation. It was invented by Horst Feistel and has been used in many block cipher designs. The working of a Feistal Network is given below:

- Split each block into halves
- Right half becomes new left half
- New right half is the final result when the left half is XOR'd with the result of applying *f* to the right half and the key.
- Note that previous rounds can be derived even if the function *f* is not invertible.

The figure 3.1 shows block diagram of Feistal network[8].

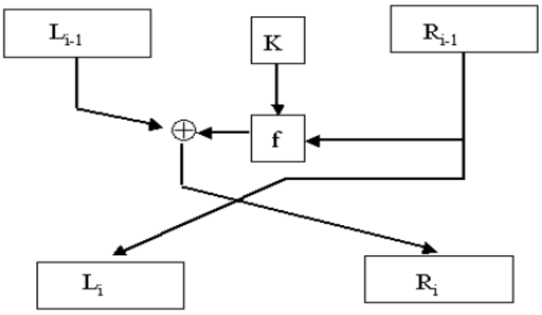


Figure 3.1: Block diagram of Feistel Network.

3.2 Description of the Blowfish Algorithm

Blowfish uses a large number of subkeys. These keys must be precomputed before any data encryption or decryption[8].

- The P-array consists of 18 32-bit subkeys:
  P1, P2,..., P18.
- There are four 32-bit S-boxes with 256 entries each:
  S1,0, S1,1,..., S1,255;
  S2,0, S2,1,..,, S2,255;
  S3,0, S3,1,..., S3,255;
  S4,0, S4,1,..,, S4,255;

The figure 3.2.1 shows Blowfish Encryption Algorithm.

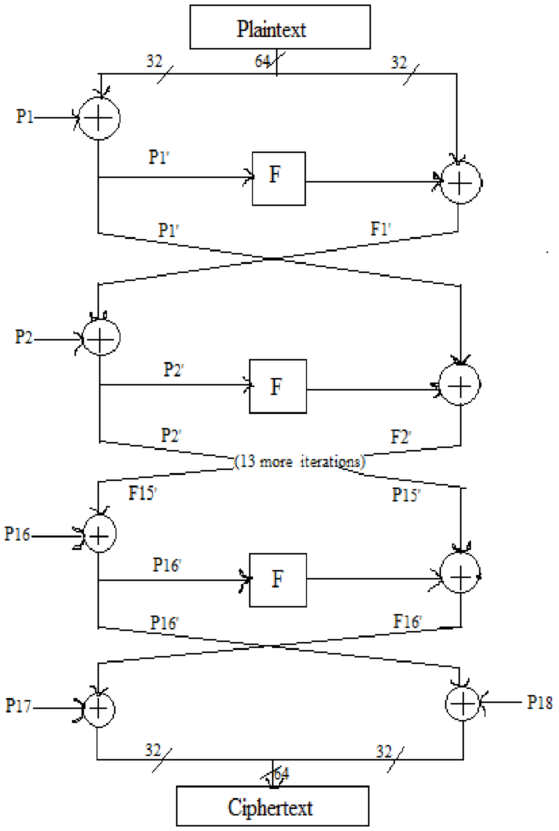


Figure 3.2.1: Blowfish Encryption Algorithm [4].

Blowfish consists of two parts: key-expansion and data encryption. During the key expansion stage, the in-putted key is converted into several sub key arrays of total 4168 bytes. There is the P array, which is eighteen 32-bit boxes, and the S-boxes, which are four 32-bit arrays with 256 entries each. After the string initialization, the first 32 bits of the key are XORed with P1 (the first 32-bit box in the P-array). The second 32 bits of the key are XORed with P2, and so on, until all 448, or fewer, key bits have been XORed Cycle through the key bits by returning to the beginning of the key, until the entire P-array has been XORed with the key. Encrypt the all zero string using the Blowfish algorithm, using the modified P-array above, to get a 64 bit block. Replace P1 with the first 32 bits of output, and P2 with the second 32 bits of output (from the 64 bit block). Use the 64 bit output as input back into the Blowfish cipher, to get a new 64 bit block. Replace the next values in the P-array with the block. Repeat for all the values in the P-array and all the S boxes in order.

Encrypt the all zero string using the Blowfish algorithm, using the modified P-array above, to get a 64 bit block. Replace P1 with the first 32 bits of output, and P2 with the second 32 bits of output (from the 64 bit block). Use the 64 bit output as input back into the Blowfish cipher, to get a new 64 bit block. Replace the next values in the P-array with the block. Repeat for all the values in the P-array and all the S boxes in order.
Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order[3].

Algorithm

```
Divide x into two 32-bit halves: xL, xR
For i = 1to 32:
xL = XL XOR Pi
xR = F(XL) XOR xR
Swap XL and xR
Swap XL and xR (Undo the last swap.)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR
```

For decryption, the same process is applied, except that the sub-keys Pi must be supplied in reverse order. The nature of the Feistal network ensures that every half is swapped for the next round (except, here, for the last two sub-keys P17 and P18)[3].
A graphical representation of F appears in Figure 3.2.2. The function divides a 32-bit input into four bytes and uses those as indices into an S-array. The lookup results are then added and XORed together to produce the output.

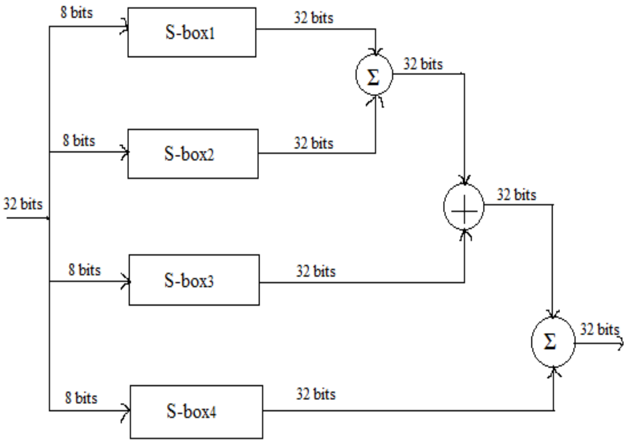


Figure 3.2.2: A graphical representation of F.

The P-array and S-array values used by Blowfish are precomputed based on the user's key. In effect, the user's key is transformed into the P-array and S-array; the key itself may be discarded after the transformation. The P array and S-array need not be recomputed (as long as the key doesn't change), but must remain secret. Let's refer to the source code for computing the P and S arrays and only briefly summarise the procedure as follows:

- P is an array of eighteen 32-bit integers.
- S is a two-dimensional array of 32-bit integer of dimension 4x256.
- Both arrays are initialised with constants, which happen to be the hexadecimal digits of π (a pretty decent random number source).
- The key is divided up into 32- bit blocks and XORed with the initial elements of the P and S arrays. The results are written back into the array.
- A message of all zeros is encrypted; the results of the encryption are written back to the P and S arrays. The P and S arrays are now ready for use[4].

4. CONCLUSION

Security in Wireless Sensor Network is vital to the acceptance and use of sensor networks. In particular, Wireless Sensor Network product in industry will not get acceptance unless there is a full proof security to the network[7]. Most of the attacks against security in wireless sensor networks are caused by the insertion of false information by the compromised nodes within the network. For defending the inclusion of false reports by compromised nodes, a means is required for detecting false reports[6]. This paper summarizes the various attacks and their classifications in wireless sensor networks.

In this paper we discussed Blowfish algorithm, which is a variable-length key block cipher. It is suitable for applications where the key do not change often, like a communications link. Blowfish is a 16 pass block encryption algorithm that is never broken. Blowfish is used frequently because:

- It is fast due to built-in instructions on the current microprocessors for basic bit shuffling operations.
- It is available in the public domains.
- Finally we can conclude that Blowfish is a better option. In future we can also implement this on image, audio & video to develop a stronger encryption algorithm with high speed and minimum energy consumption[3].

REFERENCES

[1] I. Akylidiz, W. Su, Sankarasubramaniam, and E.Cayrici, " Wireless Sensor Networks: a survey", Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. Received 12 December 2001; accepted 20 December 2001.

[2] Pratap Chnadra Mandal "Superiority of Blowfish Algorithm" Asst. Prof, Dept of Computer Application B.P.Poddar Institute of Management & technology, West Bengal, India. International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 9, September 2012 ISSN: 2277 128X.

[3] Ms NehaKhatri – Valmik, Prof. V. K Kshirsagar, "Blowfish Algorithm" Dept. of Comp. Science &Engg. Govt. College of Engg. Aurangabad, India. Dept. of Comp. Science &Engg. Govt. College of Engg. Aurangabad, India. IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. X (Mar-Apr. 2014), PP 80-83

[4] Bill Gatliff "Encrypting data with the Blowfish Algorithm" Consultant, Gdbstubs Library, eetindia.co, August 2003,EE Times-India.

[5] Kevin Allison, Keith Feldman, Ethan Mick "Blowfish".
[6] Al-Sakib Khan Pathan, Hyung-Woo Lee, Choong Seon Hong, "Security in Wireless Sensor Networks: Issues and Challenges" Department of Computer Engg.

[7] Vikash Kumar, Anshu Jain and P N Barwal, "Wireless Sensor Networks: Security Issues, Challenges and Solutions". International Journal of Information & Computation Technology. ISSN 0974-2239 Volume 4, Number 8 (2014), pp. 859-868 © International Research Publications House http://www. irphouse.com.
[8] Bruce Schneier, "Blowfish Encryption Algorithm", Pocketbrief.