Simulation of Wireless Sensor Network Security Model Using NS2

Nayana Hegde
Department of Electronics and Communication Engineering
Reva ITM , Bangalore, Karnataka, India

Dr.Sunilkumar S.Manvi
Department of Electronics and Communication Engineering
Reva ITM , Bangalore, Karnataka, India

Abstract- Wireless Sensor Networks (WSNs) are used for wide range of applications, such as national security, military surveillance and medical care. All these applications require a certain level of reliability and security during data transmission. Securing data transmission in WSN is a must. Since sensors have a limited power, memory and computational resources, any security mechanism for sensor network must be energy efficient and use less memory. The main purpose of this paper is to present how to use network simulator (NS2) for designing a secure network and use cryptographic algorithm for securing information. Data payload is encrypted for assuring the confidentiality and hash functions are used for assuring data integrity. Results from different encryption algorithms are compared. Since NS2 does not support any security features, this paper describes how to add a new protocol for NS-2.

Keywords – Wireless Sensor Networks (WSNs), Security, Encryption, Block Cipher, NS2, TCL, C++

I. INTRODUCTION

Wireless Sensor Networks (WSN) are usually characterized by unattended operational environments, ad-hoc style wireless communications and resource-constrained sensor nodes in terms of power, memory and computational capabilities and communication bandwidth [3]. A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions such a temperature, sound, vibration, pressure and humidity. The WSN as shown in figure 1 is built of nodes, each sensor network nodes has radio transceiver, a microcontroller and a battery [10]. The main characteristics of a WSN include power consumption constrains for nodes, node failures, mobility of nodes, communication failures, heterogeneity of nodes, and ability to withstand harsh environmental conditions and ease of use. Sensor nodes can be imagined as small computers. They usually consist of a processing unit with limited computational power and limited memory, a communication device and a power source [10].

To achieve the security requirements, several researchers have focused on evaluating cryptographic algorithms in WSNs and proposing energy efficient ciphers [1]. Examples of network simulation software are ns2/ns3, OPNET, NetSim. Network simulators are particularly used to design various kinds of networks, simulate and then analyze the effect of various parameters on the network performance [19]. In this paper we have simulated a security model for WSN using NS2.The model will comprise of simulating a wireless network of 50 nodes. It uses new agent called security packet. Data is sent encrypted from sender node to receiver node.

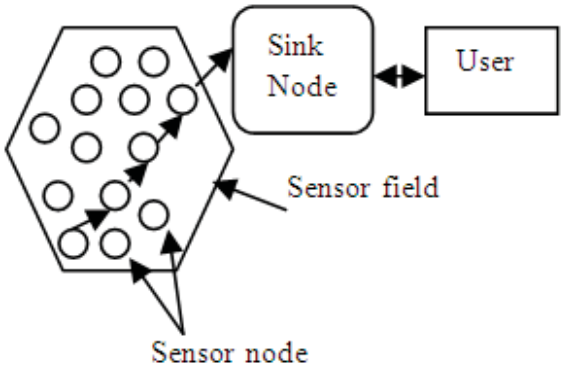

Figure 1. Basic Structure of WSN

A. Security Requirements

In WSNs, four major security requirements are integrity, confidentiality, authentication and freshness[1].Encryption is used to ensure confidentiality and message authentication code(MAC),functioning as a secure checksum, provides the data integrity and authentication in the network[2].The data encryption algorithm used in WSNs are generally divided into three major categories: Symmetric-key algorithms, asymmetric key algorithms, and hash algorithms. Asymmetric algorithms are more energy consuming. Hash functions, on the other hand, are typically used for verifying the integrity of the exchanged messages and may increase the transmission cost[4].

The paper is organized as follows: section I has Introduction, Section II Methodology. In section III we give the Design and implementation. Section IV gives Results and discussion and Section V Conclusion and future work.

II. METHODOLOGY

Network Simulator (Version 2), widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors. [6]. Figure 2 shows the basic architecture of NS2. NS2 provides users with executable command ns which take on input argument, the name of a Tcl simulation scripting file. Users are feeding the name of a Tcl simulation script (which sets up a simulation) as an input argument of an NS2 executable command ns [6].

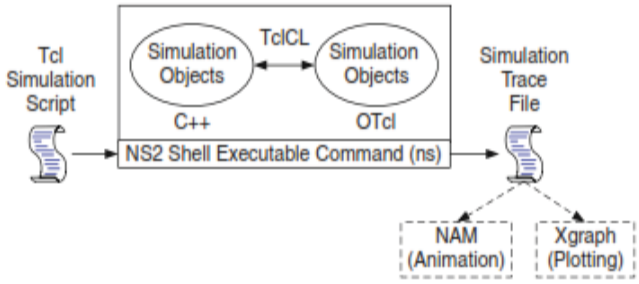

Figure 2. Basic Architecture of NS-2

A. Features of NS2

NS2 is an event driven, object oriented simulator, written in C++, with an OTcl interpreter as a frontend. NS2 supports a class hierarchy in C++ (compiled hierarchy), and a similar class hierarchy within the OTcl interpreter (interpreted hierarchy).From the user's perspective, there is a one-to-one Correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. Users create new simulator objects through the interpreter; these objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy.

- NS is an object oriented discrete event simulator
  - Simulator maintains list of events and executes one event after another.
  - Single thread of control: no locking or race conditions.
- Back end is C++ event scheduler
  - Protocols mostly.
- Source code
  - Most of NS2 code is in C++.
- Scripting language
  - It uses TCL as its scripting language.
- Protocols implemented in NS2
  - Transport layer, traffic agent, TCP.
  - Interface queue, drop tail queue.
- Scalability
  - Per-packet processing must be fast;
  - Separating control and packet handling
- Extensibility:
  - Must be easy to add new objects.
  - Object trees to understand hierarchy:
    - In C++;
    - In tcl.

III. DESIGN AND IMPLEMENTATION

Implementation of security on NS-2 is necessary in network simulation. However, currently, NS-2 does not support these features. In this paper we aim to solve this issue. The information security using cryptographic algorithms which comprises (symmetric and asymmetric) and hash function is to encrypt and send data securely between two nodes. The system must encrypt the data or systematically scramble information so that it cannot be read without knowing the coding key. This operation determines to a certain level the strength of the security system; the harder it is to break the encrypted message the more secure the system is to be. In this paper we have used CESAR cipher, XOR cipher and XTEA algorithm for encryption/decryption of messages. Results are compared. The proposed scheme is tested using ordinarily image processing. From the simulation of the experiment results, we can draw to the conclusion that this method is robust to many kinds of watermark images.

A. Approach

NS-2 is open source network simulation application.NS-2 currently supports IP protocol suite and various standard routing protocols for wire and wireless network. To add security functions. We have to modify/build new packet formats. NS-2 is open source network simulation application.NS-2 currently supports IP protocol suite and various standard routing protocols for wire and wireless network. To add security functions. We have to modify/build new packet formats. Our approach is to build a new protocol at network layer – IP layer. We also define new packet format to represent new protocols. The new protocol is represented by a class derived from built-in class in NS-2. Within new derived class we will add encryption and decryption for data field in the data packet. We will also implement message digest generation function to ensure the integrity of data packet during transmission. We consider our data as plain text. The cryptography algorithms used are CESAR cipher, XOR cipher and XTEA algorithm. The programming language is C++. Procedure is defined as follows:

- Define a new Packet format
- Derive new class from Agent class for processing this new packet format.
- Processing includes:

-Encrypting function.
-Decrypting function.

- Message digests generation function.

All NS-2 packet types are handled as if they are simple byte arrays. Carriers of this big packet structure are instances of class Packet. Adding new packet type intoNS2 can be done with the sample code given below:

```
#include "packet.h"
struct hdr_myHeader {
        // your data fields
        int myData;
        int &getMyData() { return myData; }
        // necessary for access
        static int offset_;
        inline static int& offset() { return offset_; }
        inline static hdr_myHeader* access(const Packet* p) {
                return (hdr_myHeader*) p->access(offset_);
        }
};
```

Figure 3. Sample Code for adding a new Packet.

The field offset_ points to the position in the NS2 byte array (merged packet), where the header is stored.

Static binding of packet is present from file MyPacket.cc as below:

```
static class MyHeaderClass : public PacketHeaderClass {
public:
        MyHeaderClass() :
PacketHeaderClass("PacketHeader/MyHeader",sizeof(hdr_myHeader)) {
                bind_offset(&hdr_myHeader::offset_);
        }
} classMyHeader;
int hdr_myHeader::offset_;
```

Figure 4. Sample code for binding the packet type

B. Steps Involved in Adding a new Protocol in NS2

- Define a new Packet class Security_Packet
- Add new packet entry into Packet.h file
- Add new packet type in ns-default.tcl file.
- Make an entry into ns-Packet.tcl file
- Make an entry into Makefile.info file[8].

C. Design of Wireless Sensor Network

Effect of number of active nodes and inter-node distance plays important role on the performance of Wireless Sensor Networks. Wireless Sensor Network Simulation is based on the scenario of simulation. For WSN we have used AODV routing protocol and 802.15.4 as MAC. Sensor nodes are placed randomly. Generating randomness in simulation is necessary because in real time scenario most of the systems are random in nature. The random quantities in a network could be delays, inter-event times, packet lengths, and distance between two sensor nodes or random mobility of nodes.

TABLE 1. PARAMETER VALUES FOR WSNSIMULATION

MAC protocol	IEEE 802.15.4
Routing Protocol	AODV
Agent Type	Security
Terrain Size	1000x1000
Number Of Nodes	50
Node Mobility	None
Packet Size	32 Bytes
Node Placement	Random
Number of Sources	Node 10 and Node 12
Number of Sink nodes	20 and 22

D. Encryption/Decryption functions

Cryptography is the science of using mathematics to encrypt and decrypt data. A cryptographic algorithm or cipher is a mathematical function used in encryption and decryption process. Symmetric ciphers use single key for both encryption and decryption. Asymmetric or public key encryption uses two keys one for encryption and another key for decryption. There are two types of symmetric key ciphers: Block ciphers and Stream ciphers. Block cipher is a symmetric key modern cipher that encrypts an n bit block of plaintext or decrypts an n bit block of ciphertext. Examples are AES, RC5, SkipJack. Here we have used CEAR cipher,XOR and XTEA for encryption. Figure 5. Shows the basic structure of XTEA algorithm.

In cryptography, Corrected Block TEA (often referred to as XXTEA) is a block cipher designed to correct weaknesses in the original Block TEA.XXTEA is vulnerable to a chosen-plaintext attack. XXTEA is a consistent incomplete source-heavy heterogeneous UFN (unbalanced Feistel network) block cipher. XXTEA operates on variable-length blocks that are some arbitrary multiple of 32 bits in size (minimum 64 bits). The number of full cycles depends on the block size, but there are at least six (rising to 32 for small block sizes). The original Block TEA applies the XTEA round functions to each word in the block and combines it additively with its leftmost neighbor. Slow diffusion rate of the decryption process was immediately exploited to break the cipher. Corrected Block TEA uses a more involved round function which makes use of both immediate neighbors in processing each word in the block.

E. Hash Functions

Hash functions are like checksum. Using a simple hashing algorithm we get hashed value from a string of plain text. The hash value will be attached to packet header for data integrity checking. At the other end of communication, after decryption, the decrypted text will be hashed again to get new hashed value. This new hashed value will be compared to the value attached within packet header. If they are equal, the data integrity is ensured and decrypted text is accepted; otherwise the packet is discarded. In either case, an acknowledge packet will be sent back to sender to inform of the status of the packet [7].

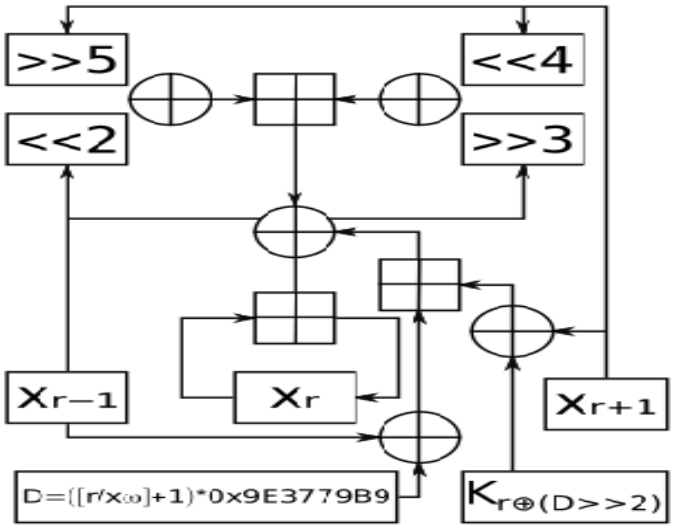

Figure 5. Block Diagram of XTEA

IV.RESULTS AND DISCUSSION

This section presents the output of the simulation. The wireless sensor network security performance depends mainly on the cryptography algorithms. We have used Caesar cipher, XOR and XTEA algorithm to transfer the secure data packets between the nodes. NAM is tcl/tk based animation tool that provides animation or Nodes, Links, Queues, Packets and Agents. The figure 6 shows the NAM output of the simulation where nodes are and rand omly placed with random topology. Data packets are sent from node 10 to node 20 and from node 12 to node 22.

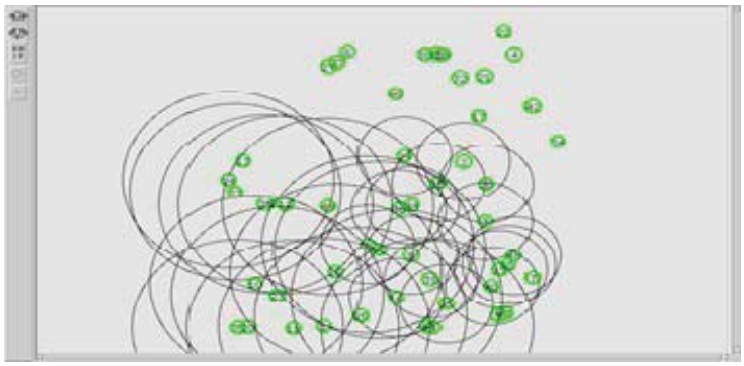

Figure 6. NAM output for Wireless Sensor Simulation

Time taken for sending the data packets from sender node to receiver node each algorithm is drawn as bar chart in figure 7. Trip time for sending data packets is measured msec.

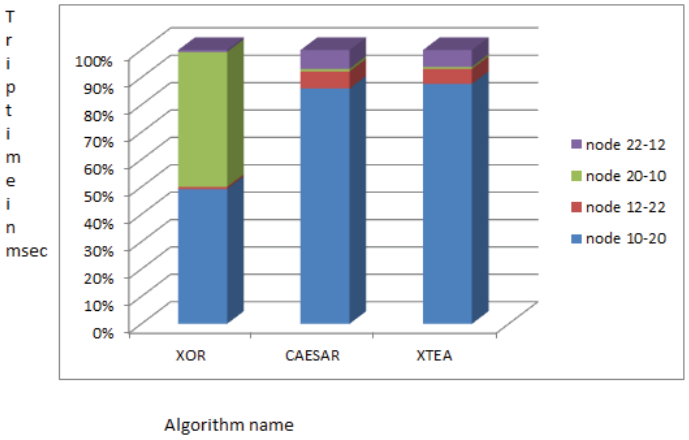

Figure 8. Comparison of memory usage for algorithms

Memory required for each algorithm is drawn as a bar chart in figure 8.Here Caesar cipher and XOR cipher uses same memory and XTEA algorithm uses more memory.

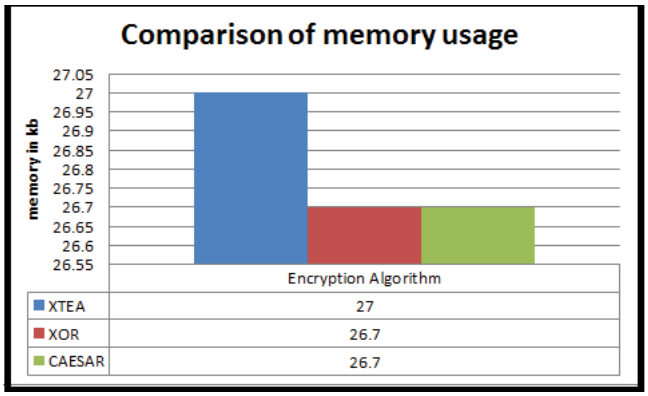

Figure 8. Comparison of memory usage for algorithms

TCL output shows the number of nodes deployed and message sent from sender node to the receiver node with encrypted message as well as decrypted message. Hash function is used for confirming the data integrity.

V. CONCLUSION AND FUTUREWORK

Wireless sensor network are more prone to attack, compared to other computer network due to their adhoc deployment and wireless nature of communication. For secure communication in WSNs, efficient cryptographic algorithm suitable for WSN environment is required. It is ideal to choose the most efficient cryptographic algorithm in all aspects; operation speed, storage and power consumption. However, since each cryptographic algorithm applied in WSNs has distinguished advantages, it is important to choose cryptographic algorithm suitable for each environment WSNs are exploited or the application being deployed. Among the XOR, CAESAR cipher and XTEA algorithms, XTEA is more secure. Its memory consumption is similar to other two ciphers. XTEA has small memory requirements which make it suitable for use in resource constraint environment such as embedded systems. The length of the cipher text is equal to the length of plaintext which means that XTEA does not require using operation modes. Future work is to include implement BlowFish, SkipJack algorithms and compare their performance.

REFERENCES

[1] Xueying Zhang,Howard M Heys, and Cheng Li “Energy Efficiency of Symmetric key Cryptographic Algorithms in Wireless Sensor Networks”, 25th Biennial Symposium on Communications IEEE, pp168-172,2010

[2] Ch.P.Antonopoulos,Ch.Petropoulos,K.Antonopoulos,V.triantafyllou,n.S.V oros,”The Effect of Symmetric Block Ciphers on WSN Performance and Behavior”,Fifth International Workshop on Selected topics in Mobile and Wireless Computing,IEEE,pp-799-806,2012

[3] Ruiping Ma, Liudong xing, Howard E.Michel,Vinod M. Vokkarane,”Linear Crypatanalysis of A Survivable Data Transmission Mechanism for Sensor Networks”,pp-562-567,IEEE 2012

[4] Soufiene Ben Othman,Abdelbasset Trad,Habib Youssef,”Performance Evaluation Of Encryption Algorithm for ge WSNS”, Internatonal d of l Security and Its Applications(IJNSA), vol.2, no.3,pp.52-68, July 2010

[5] Shaik Sahil Babu,Karuppiah,S.Rajaram,”Energy Efficient Algorithm for Wireless Network”,International journal of Engineering Research and Technology,vol 1,may 2012.

[6] Introduction to Network Simulator NS2.pdf

[7] Sirwan A.Mohammed sattar B.Sadkhan,,”Design and Simulation of Network Using NS-2” IJECIERD 2011

[8] Sam Tran,Tuan Anh Nguyen, "System Design Specification for Encryption/Decryption function for NS2”, Houston - April 25, 2004

[9] Nour El Din M. Khalifa, Mohamed H. Taha, Hesham N. Elmahdy and Imane A. Saroit,"A Secure Energy Mechanism for WSN and Its in NS-2" CiiT International Journal of Wireless Communication, Vol 4, No. 16, December 2013.

[10] Neha Singh, Prof. Rajeshwar Lal Dua, Vinita Mathur, ,"Network Simulator NS2-2.35 " IJARCSSE Volume 2, Issue 5, May 2012