IMPLEMENTATION OF A SECURITY PROTOCOL FOR BLUETOOTH AND WI-FI

U. Pavan Kumar

Dept. of Telecommunication Systems Engineering, AITTM, Amity University, Noida, India.

Pavan21cool@gmail.com

ABSTRACT

This paper is mainly based on providing security to the wireless networks through which devices like Bluetooth gets connected. The Wi-Fi connections are also prone to various attacks these days. The protocols that are required to provide security to wireless networks can be implemented by creating a wireless scenario using the software Network Simulator. This paper illustrates a scenario to check the security protocol. As NS2 mainly has the implementation of routing protocols, a new protocol should be designed especially for security purpose. This is done by following many tutorials to get a minimum basic knowledge of NS2, C/C++ coding. The security feature followed in the paper is encryption/decryption of the data that is being exchanged. Data should be ensured as and then there will be a perfect implementation of the protocol. So, the paper throughout concentrates on adding a new security protocol to NS2 and implementation of that protocol by providing a wireless scenario.

KEYWORDS

Wireless Networks, NS2[1], Security Protocol, Bluetooth, Wi-Fi, Caesar Cipher[2], Encryption/Decryption[2], Hash Function [2].

1. INTRODUCTION

Bluetooth technology is the open standard for wireless connections with the backers mostly from the industries of mobile phones and computers. Bluetooth, however, is a radio-frequency (RF) technology utilizing the unlicensed 2.4GHz Industrial-Scientific-Medical (ISM) band.

The applications which were targeted include PC and peripheral networking, hidden computing, and data synchronization such as for address books and calendars. Other applications might be comprised of home networking and home appliances of the future such as smart appliances, heating systems, and entertainment devices. Ericsson was the initial developer. Later, Bluetooth Special Interest Group (SIG) formalized it as an industrial standard. SIG, now, expanded to a group of 1800 members though it was initially formed by Ericsson, Intel, Toshiba, Nokia and IBM.

The important consideration for any network is its security. If a network gets an unauthorized access then it can lead to cases like disclosure of sensitive information, modification of data, denial of service, illicit use of resources etc. Such unauthorized access grants the access to

DOI : 10.5121/ijnsa.2013.5307

information which can reveal the usernames and passwords which can be later misused for further attacks.

Not just wired LANs but WLANs are also subjected to all the security issues. In addition to these, WLANs also have many more vulnerabilities that are associated with wireless connectivity and prone to attacks. The nature of the wireless communicating medium makes it practically impossible to confine the radio signals to an area that is controlled. These radiated signals are subject to and effected by clandestine interception and exploitation.

If we take the case of a wired LAN, the attacker must physically connect some wires to the network to gain access to the resources and such protection is not available for a WLAN. This is because of remote access of any wireless networks. This implies that anyone using compatible wireless equipment can potentially access the LAN.

To mitigate or to control these security concerns, encryption/decryption is used in an attempt to make the signal unusable by unauthorized parties if any interception takes place. Any user wishes to have a easy to use policy for a system and this became the primary concern for most commercial products. Due to this, all the current 802.11 WLAN devices are bound by this concern. This led to a default option of turning off the security features temporarily and this creates a way for the attacker.

2. NEED FOR WIRELESS SECURITY

Wireless security has the top priority in prevention of unauthorized access or damage to computers mainly using wireless networks. The most common systems of wireless security currently working are WEP (Wired Equivalent Privacy) and WPA (Wi-Fi Protected Access). WEP is one among the least secure forms of security.

FBI has cracked a network which was secured using WEP in a span of 3 minutes. WEP is a very old IEEE 802.11 standard available from 1999 which was outdated in 2003 by the entry of WPA.WPA has overcome the drawbacks of WEP and has become an alternative to it quickly. WPA2 is the present standard and is supportable only through upgraded firmware. Encryption is the special feature of WPA2.

Many laptops have wireless cards pre-installed in them. Wireless networking is prone or vulnerable to some security issues. Hackers have found wireless networks relatively easy to break into, and even use wireless technology to hack into the wired networks also. Due to this, it has become necessary for the enterprises to build a system that protects the network from unauthorized access and such a system is WIPS (Wireless Intrusion Prevention System) or WIDS (Wireless Intrusion Detection System).

The risks to users of the wireless devices have increased as the services have become much popular. There were relatively few dangers when the introduction of wireless technology took place. During the early stages of wireless, hackers have no way to get adapted to the technology as the work place is mostly a wired environment. But currently, the ignorance and carelessness of the users and the corporate brought many risks in security.

Hacking methods have become much more sophisticated and innovative with wireless. These days hacking is an interesting concept for many as Linux and windows have brought some easy while using methods in their systems without any charge.

2.1. Background

Anyone within the range of a geographical network of an open and unencrypted wireless network can sniff or capture and record the traffic through which one can gain unauthorized access to internal network resources as well as to the internet, and then can use the information and resources to perform disruptive or illegal acts. These unauthorized accesses are the current main issues for home networks and as well as for enterprises.

If router security is not active or if the owner simply deactivates it for his convenience, it creates a free hotspot. Since most of the 21st-century laptops have built in wireless networking, a third part adapter installation is not at all required. There is a default option through which we can enable the built-in wireless networking without giving any indication to the owner. This makes broadcasting, much easier, which connects all the nearby computers.

Internet Connection Sharing is a tool that makes an easy way to the users to setup a WLAN. This feature is available in all the latest operating systems. If the user does not know about the setup then it will become a way for the attacker to steal all the information through the base station which makes the WLAN. It may even be without the knowledge of the intruders if their computer automatically selects a nearby unsecured wireless network to be used as an access point.

All the user related features are made easy by the vendors and this brought the threat situation in to the wireless environment.

2.2. The Threat Situation

Wireless security is one of the aspects of computer security. Rogue access points are the primary cause for making the organizations prone to the attacks.

Vulnerabilities are common for any wireless network. For suppose, if a person working in an organization has entered in to the network through an unsecured port then he has given the start to crash the entire network. This also gives a key hole to drag all the sensitive and confidential information present in the network.

Vulnerabilities can be eradicated through the counter measures which are effective. One of them is to deactivate the unused ports within the network and later for using that port we can get some authorization. Such countermeasures must be effectively implemented and any ignorance damages the entire counter measure.

So, by undergoing the threats of wireless technology, we can understand the critical condition of needing security to wireless networks. This need was in many minds which led to build security features and new protocols for wireless networks like Bluetooth, Wi-Fi, etc.

The protocol in this paper is also about the security that a protocol can give to a wireless device if used as an application or as a feature if added to the protocol architecture.

3. IMPLEMENTATION

3.1. Introduction to NS2

Every person interested in networking has to study the behavior of the network. The network might be wired or wireless. This is the main job of a network simulator. The simulation carried out here is a method which creates a hypothetical network environment and makes it functional to study the network behavior. NS2 is among such simulation tools but it is not just another simulator like others. It is used all around the world and is available at no cost. Although in initial stages it make us feel like a hard bone but a little hard work, interest and guidance can make anyone, a master of NS2 in a little span.

3.2. Installation of NS2

Linux/Unix based systems are the primary operating systems in which NS2 can be installed. It is also available on Windows based systems which is used with the help of Cygwin. Installation on Linux/Unix is almost same and no extra tool is required to run NS2, but for a Windows based system, Cygwin software is needed in addition to a huge number of files. So if anyone has both options to work on a Linux based or a Windows based system, it is advisable to go for Linux based system.

NS2 comes with a set of protocols which are large in number and are used in different kind of networks as well as different scenarios in real life. The current version used is version 2.3.5 which comes with a total of 72 various protocols. Installation errors are frequent with many users and are often solved by many developers from time to time. As and when the software is developed, it gets validated for making it use for many interested ones who wanted to work with NS2.

3.3. NS2's Strength

NS2 lets researchers to simulate wired as well as wireless networks. Simulation can be done through static as well as dynamic nodes in this simulator. It also lets users to monitor behavior of a network. It gives the user an environment to create a new protocol and use it to the existing codes. NS2 supports TCL and Object Oriented TCL for the simulation purpose. If implementation is required then C++ code can be used along with TCL/OTCL.

3.4. Problem with NS2

Network Simulator-2 is an open source and very popular network simulation tool that provides support for IP protocols suite and many standard routing protocols for wired and wireless networks. Implementation of security on NS-2 is a necessary task in network simulation. NS2 has no way to support security features unless some protocols are added. So, a new protocol is designed in this project to give security features to NS2.

Here, a new protocol must be added and it should be provided with some functions related to security functions like providing encryption and decryption. Such a work is mentioned in the following sections of this paper.

4. ADDITION OF A NEW PROTOCOL TO NS2

4.1. Approach to NS2

NS-2 is one of the open source systems, which is created, executed and developed using C++ and Tool Control Language (TCL). Any researcher hungry to add new features and components to NS2 can easily add them.

The current latest version of the software is 2.35. A huge number of protocols are added in this version. MAC layer protocols and application protocols are the special features of this version. Routing protocols are also separately available as there are the cases of wired and wireless networks. Any new protocol can also be added by any one and can adopt it into NS2.

Our current problem is security and so we focus on adding some security features in the form of protocols to the software. In this project, we are adding a protocol to the Internet Protocol layer. Data integrity, encryption and decryption are implemented on the packet by this protocol. We consider our data as plain text.

The cryptography algorithm is CESAR cipher. The programming language is C++.
Environment development requirements:

- Personal Computer Windows XP professional and later.
- Cygwin UNIX Simulation
- NS-2 version 2.35
- C/C++ editor.

4.2. Algorithm To Add A Protocol to NS2:

For any procedure to be implemented, we need a set of steps. These steps define the algorithm for the procedure. Similarly, for the protocol to be added we require a systematic set of steps. This algorithm is the best way to add a protocol to NS2. All the steps must be performed in an orderly manner and not doing so will bring errors in the procedure. The following are the steps followed in protocol addition to NS2.

a) Create a new header file SP.h for the protocol.
b) Create SP.cc file which contains the required code to execute the protocol.
c) Add the protocol ID to the packet.h file of NS2.
d) Edit enum packet_t() and p_info() files of packet.h.
e) Add the default value of the protocol to ns-default.tcl file.
f) Add an entry for the new protocol packets in the file ns-packet.tcl.
g) Add the file SP.o to the list of object files for NS in makefile.in file.
h) Recompile NS2 software

5. LOGICAL DESIGN OF ENCRYPTION/DECRYPTION SYSTEM

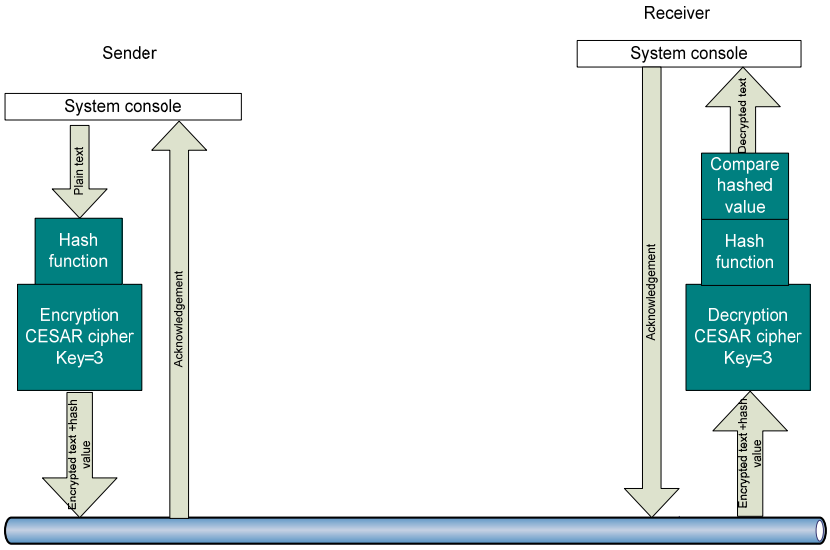

Figure 1: Logical design of the encryption/decryption system

6. PROGRAM CODE

SP.h:

```
#ifndef ns_SP_h

#define ns_SP_h

#include "agent.h"

#include "tclcl.h"

#include "packet.h"

#include "address.h"

#include "ip.h"

struct hdr_SP {

char ret;

double send_time;

double rcv_time;

int seq;

char data[128];
```

```
unsigned int hashvalue;
// Header access methods
static int offset_;
inline static int& offset() { return offset_; }
inline static hdr_SP* access(const Packet* p) {
return (hdr_SP*) p->access(offset_);
}
};
class SPAgent : public Agent {
public:
SPAgent();
int seq;
int oneway;
virtual int command(int argc, const char*const* argv);
virtual void recv(Packet*, Handler*);
void encryption(char* out);
void decryption(char* out);
unsigned int hashing (char value[], unsigned int len);
};
#endif // ns_SP_h
```

SP.cc:

```
#include "SP.h"
#include "string.h"
int hdr_SP::offset_;
static class SPHeaderClass : public PacketHeaderClass {
public:
SPHeaderClass() :
```

```
PacketHeaderClass("PacketHeader/SP",sizeof(hdr_SP)) {
bind_offset(&hdr_SP::offset_);
}
} class_SPhdr;
static class SPClass : public TclClass {
public:
SPClass() : TclClass("Agent/SP") {}
TclObject* create(int, const char*const*) {
return (new SPAgent());
}
} class_SP;
SPAgent::SPAgent() : Agent(PT_SP), seq(0),
oneway(0)
{
bind("packetSize_", &size_);
}int SPAgent::command(int argc, const char*const* argv)
{
if (argc ==3)
{
if (strcmp(argv[1], "send") == 0)
{
Packet* pkt = allocpkt();
hdr_SP* hdr = hdr_SP::access(pkt);
hdr->ret = 0;
hdr->seq = seq++;
hdr->send_time = Scheduler::instance().clock();
strcpy(hdr->data, argv[2]);
```

```
hdr->hashvalue = hashing(hdr->data,(unsigned int)strlen(hdr->data));
printf("Sent message %s with hashing %d\n",hdr->data,hdr->hashvalue);
// ---------- encrypt the data ----------
encryption(hdr->data);
// Send the packet
send(pkt, 0);
return (TCL_OK);
}else if (strcmp(argv[1], "start-WL-brdcast") == 0)
{
Packet* pkt = allocpkt();
hdr_ip* iph = HDR_IP(pkt);
hdr_SP* ph = hdr_SP::access(pkt);
strcpy(ph->data, "test");
iph->daddr() = IP_BROADCAST;
iph->dport() = iph->sport();
ph->ret = 0;
send(pkt, (Handler*) 0);
return (TCL_OK);
} else if (strcmp(argv[1], "oneway") == 0) {
oneway=1;
return (TCL_OK);
}
}
return (Agent::command(argc, argv));
}
```

```
// -- CESAR encryption function ----------
void SPAgent::encryption(char out[])
{
int key =3;
int i=0;
for (i=0;i<strlen(out);i++)
{
out[i]=(out[i]+key)%128;
}
}
void SPAgent::decryption(char out[]){
int key =3;
int i=0;
for (i=0;i<strlen(out);i++){
out[i]=(out[i]-key)%128;
}
}
unsigned int SPAgent::hashing(char value[], unsigned int len){
char *word = value;
unsigned int ret = 0;
unsigned int i;
for(i=0; i < len; i++){
int mod = i % 32;
ret ^=(unsigned int) (word[i]) << mod;
ret ^=(unsigned int) (word[i]) >> (32 - mod);
}
return ret;
```

```
}
void SPAgent::recv(Packet* pkt, Handler*){
hdr_ip* hdrip = hdr_ip::access(pkt);
hdr_SP* hdr = hdr_SP::access(pkt);
if ((u_int32_t)hdrip->daddr() == IP_BROADCAST){
if (hdr->ret == 0){
printf("Recv BRDCAST SP REQ : at %d.%d from %d.%d\n", here_.addr_, here_.port_, hdrip->saddr(), hdrip->sport());
Packet::free(pkt);
Packet* pktret = allocpkt();
hdr_SP* hdrret = hdr_SP::access(pktret);
hdr_cmn* ch = HDR_CMN(pktret);
hdr_ip* ipret = hdr_ip::access(pktret);
hdrret->ret = 1;
ipret->daddr() = IP_BROADCAST;
ipret->dport() = ipret->sport();
send(pktret, 0);
}else
{
printf("Recv BRDCAST SP REPLY : at %d.%d from %d.%d\n", here_.addr_,
here_.port_, hdrip->saddr(), hdrip->sport());
Packet::free(pkt);
}
return;
}
if (hdr->ret == 0){
double stime = hdr->send_time;
```

```
char original_data[128];
char encrypted_data[128];
strcpy(encrypted_data,hdr->data);
strcpy(original_data,hdr->data);
int rcv_seq = hdr->seq;
char out[105];
unsigned int newhash;
char authenticate_result[50];
decryption(original_data);
newhash=hashing(original_data,strlen(original_data));
if(newhash==hdr->hashvalue){
printf("data intergity ensured\n");
strcpy(authenticate_result,"Message_Accepted");
}else
{
printf("data modified %d\n",newhash);
strcpy(authenticate_result,"MESSAGE_ERRROR-Integrity violation");
}
sprintf(out, "%s recv %d %3.1f %s %s %d", name(), hdrip->src_.addr_ >>
Address::instance().NodeShift_[1],(Scheduler::instance().clock()-hdr-
>send_time)*1000,encrypted_data,
original_data,hdr->hashvalue);
Tcl& tcl = Tcl::instance();
tcl.eval(out);
// Discard the packet
Packet::free(pkt);
// Create a new packet
```

```
Packet* pktret = allocpkt();
// Access the header for the new packet:
hdr_SP* hdrret = hdr_SP::access(pktret);
hdrret->ret = 1;
hdrret->send_time = stime;
hdrret->rcv_time = Scheduler::instance().clock();
hdrret->seq = rcv_seq;
strcpy(hdrret->data, authenticate_result);
send(pktret, 0);
}
else
{
char out[105];
sprintf(out, "%s recv %d %3.1f %s _ %d", name(), hdrip->src_.addr_ >>
Address::instance().NodeShift_[1],
(Scheduler::instance().clock()-hdr->send_time) * 1000, hdr->data, hdr->hashvalue);
Tcl& tcl = Tcl::instance();
tcl.eval(out);
// Discard the packet
Packet::free(pkt);
}
}
```

The tcl file can be created as per our own imagination of the wireless scenario using NS2. But the coding must be done including the SP protocol agent in it.

Tcl file is very important because it is the only way to bring our idea of network environment to reality. To study a network, we first need to create a scenario that defines the network and this is fulfilled through the tcl script.

7. DEMONSTRATION & RESULTS

For the demonstration to be carried out, we need a scenario consisting of four wireless nodes. Node 0, 1 will send message to node 2 and 3 respectively and from node 2 and 3 back to node 0 and 1.An acknowledgement will be assumed by each sender node from the receiver node. This scenario is created using the TCL language. We use a script file to arbitrarily send the message.
The following figures are the proofs to the demonstration. The first figure shows the message transfer from node 0 to node 2. The second figure shows the message transfer from node 1 to node 3. The third figure consists of the generated output. It shows the message transmitted and hash value being used in the transfer. Data integrity is also assured in the third figure.

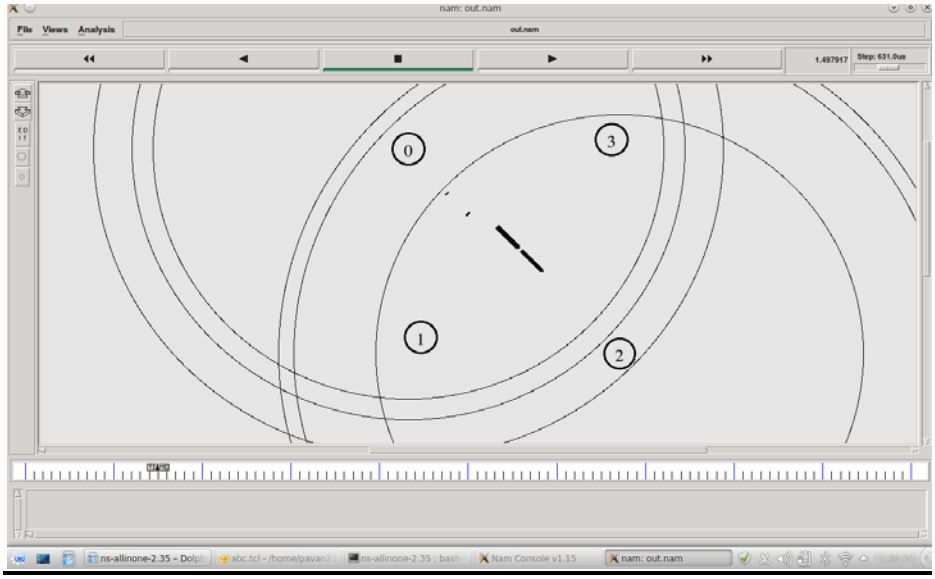

Figure 2: Message being sent from node 0 to node 2

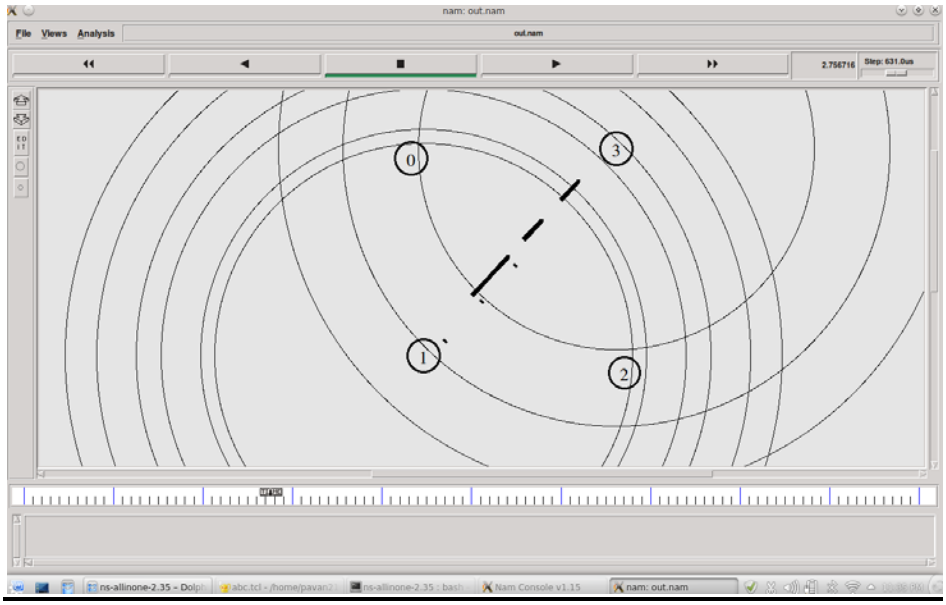

Figure 3: Message transfer from node 1 to node 3

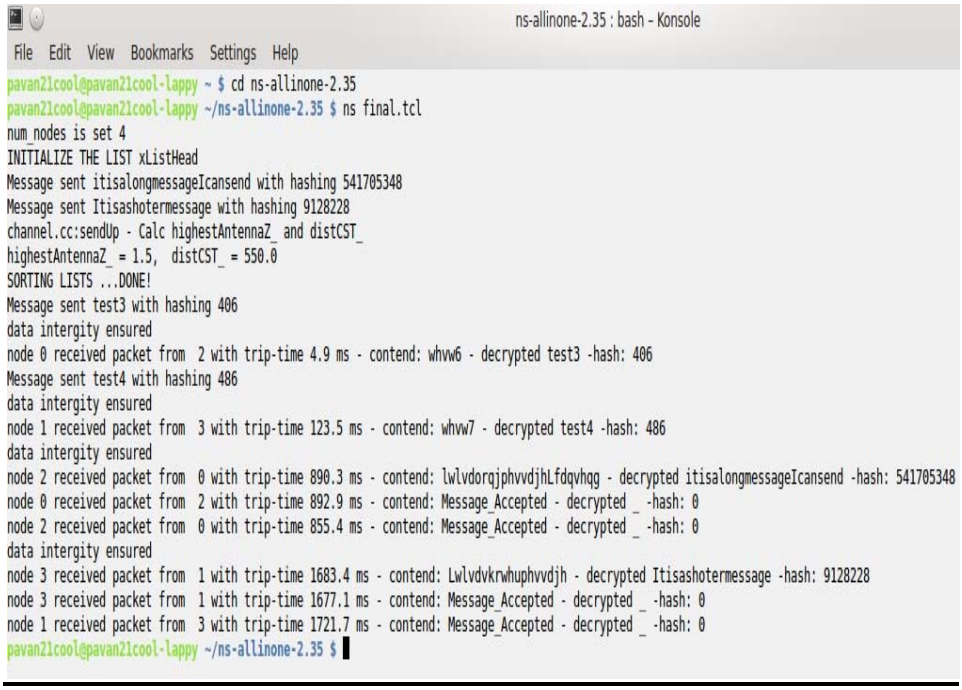

Figure 4: Data Encrypted, Decrypted and Ensured

8. CONCLUSIONS

Security in wireless networks can be assured if security protocols are being constructed and implemented on the network. NS2 is providing an environment to check such protocols and there are other software too for such protocol checks. SP protocol provides the security by encrypting the message and prevents the attackers from being attacked. The default key used for CEASAR cipher here is 3 and it can be increased to provide some complexity for the hackers. Not just CEASER cipher, the algorithms can be many to provide encryption like DES, 3DES, EAS, Blowfish etc.

Bluetooth devices can be secured using this protocol, This is because it's the data that's being encrypted here but not the pairing key(which is already provided by Bluetooth). Bluetooth implementation can also be checked by replacing the nodes here with the devices. Though there are many protocols or algorithms are made, they might get hacked and the data can be captured. So, maintaining and altering the codes and increasing the complexity in the encryption provide better security.

Finally, providing security is not just based on the protocols being used but the user must also be cautious from getting attacked. This is because as there are many ways to protect your network same as there are ways to break the security. This is possible only if the user gives some lead to the hacker by getting attracted to his tricks. So, security protocols are safe, strong and secure until and unless the algorithms being used in them are cracked.

ACKNOWLEDGEMENTS

It is my utmost duty and desire to express acknowledgement to the various torch bearers, who have rendered valuable guidance during the preparation of my paper.

Firstly, I would like to thank **Prof. R. K. Kapur,** Dy. Director & Head, AITTM, AUUP for his inspiring support to our academic efforts.

It is a great pleasure for me to acknowledge my profound sense of gratitude to my guide **Mrs. Neha Arora,** Assistant Professor, AITTM for her valuable and inspiring guidance, comments, suggestions and encouragement throughout this paper.

I am grateful to my beloved **Lt. Gen. P D Bhargava,** Group Dy. Vice Chancellor, AUUP & Advisor, AITTM for providing me a platform in the form of this great institution to showcase my talent.

Finally, I sincerely thank all those who have helped me in the endeavor of completing this paper. I should also like to thank entire fellow colleagues who really helped me directly or indirectly, whenever needed.

And my deepest thanks to my parents who, have constantly encouraged fulfilling a part of my goal completely and successfully.

REFERENCES

[1] http://sce.uhcl.edu/transa/Sourcecode/NS-2_Security_Node_Document.doc
[2] http://www.isi.edu/nsnam/ns/tutorial/index.html
[3] http://wn.com/Edimax_ADSL_Modem_Router__Setup_WPA_Wireless_Security
[4] http://wn.com/Edimax_Broadband_router__Setup_WPA_Wireless_Security
[5] http://en.wikipedia.org/wiki/Wireless_security
[6] Behrouz A. Forouzan & Debdeep Mukhopadhyay (2008) "Cryptography and Network Security", Second Edition, Tata McGraw Hill Education Private Limited.

Authors

I'm Udaragudi Pavan Kumar, born on 28th of April,1991.I'm currently persuing M.Tech in Telecommunication Systems Eng. Of batch 2012-2014. I'm interested in networking which made me to do this paper.