International Journal of Computer Applications (0975 – 8887)
Volume 39– No.16, February 2012

Design and Develop ECC for Wireless Sensor Network

Bhisham Sharma
Research Scholar
PEC University, Chandigarh

Yogesh kumar
Thapar University, Patiala

Vandana Ladha
Thapar University, Patiala

ABSTRACT
Wireless Sensor Networks consist of sensor nodes and few powerful control mobile laptops performing activities like routing, data aggregation etc over wireless media. These kinds of networks are getting popular these days because of small size, ease of handle and installation. Because of this property they are used in environment like military, hospitals, weather forecasting etc. for processing critical information. However such kind of environment is more prone to "Man In the Middle Attack", where attacker can easily perform malicious activities without interrupting network operation, which can further propagate to other nodes that can alter say routing information and even can degrade the network performance and stability. In this paper we have designed ECC algorithm and implemented it over a simulated network created with the help of Java Sun Spot kit, consisting of two sensor devices and a base station. Here by going through the work done by different researchers where they have compared both RSA and ECC algorithm with the help of automated tool over different factors, it can be concluded that ECC is the most optimal and efficient algorithm for wireless communication. Using Java SunSpot kit a wireless sensor network is formulated and devices are made to communicate with Elliptical Diff-Hellman Algorithm implemented over it, the packet are then captured and verified using the automated tool.

Keywords:
Wireless Sensor Networks, PKI, Elliptical Curve Cryptography, SunSpot devices.

1. INTRODUCTION
By the immense effort of researchers in wireless communication, Micro Electro Mechanical System(MEMS) have opened a route to modern civilization which has been densely populated with the low-power, cost-effective and automated devices knows as sensors. These sensors devices are capable of storing and processing real time data which is helpful in preparation and prevention during the phases of pre-event, responses during the event and post recovery along with the analysis of the event [2]. When networked, sensor networks can not only provide data collection but can also be used for performing and controlling multitude task. Because of it sensor networks are used in various applications like monitoring temperature, humidity, pressure, soil, vehicle movement, lightening conditions etc. [2]

1.1 Wireless Sensor Network
Wireless Sensor or Wisenet is the network formed by sensor devices that are capable of communicating with each other over wireless media. After the portable devices like PDA, mobiles etc. these devices are emerging at a high speed. In spite of their small size and memory, this sensor device act as a powerful CPU which can be easily portable, installed and handled also known as "Sensor Motes". Sensor motes consist of microcontroller, transceiver with antenna for receiving and transmitting data, memory having Operating System installed over it. Various companies came into play for developing sensor device like MICA, INTEL and the latest is Sun Microsystems supporting different operating system like TinyOs, JavaSqwak depending upon their performance.

1.2 Architecture of WSN
A general architecture of WSN consists of sensor nodes communicating over wireless media and a base station. Base station collect the information and broadcast it further to the Gateway which then send server and display it over the screen of the web client requesting the particular information. The entire scenario is shown in the Figure 1 given below, where the wireless media is shown with the dashed lines. In spite of the local network one can also have basestation attached to a single machine running a host application for displaying results of the data so collected.

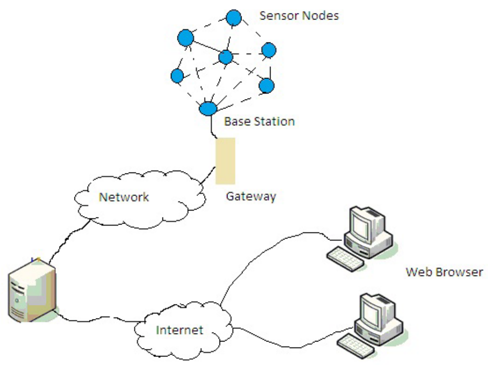

Figure 1: Architecture of WSN network.

1.3 Application Area of WSN
Wireless Sensor networks are becoming popular because of their small and ease of installation. Thereby used in very sensitive environment, some of them are discussed below [1]:

1.3.1 *Civil Engineering:* Structure Monitoring is the latest application of sensor networks where they are attached to heavy structures like bridges, building and then analyzed for the strain they gain while passing of the train and their tolerance power.

1.3.2 *Industry Automation:* It is not always possible to keep on track of maintainable of heavy and large machines and it is not always mandatory also. So in industries these machines are embedded with sensors that keep on diagnosing machines and apprise whenever the maintainable is required.

1.3.3 Military environment: In military environment sensors are used to transmit sensitive information, detecting presence of movements of enemy units on land/sea, identifying chemical and biological threats, targeting systems and controlling command within the army units.

1.3.4 Public safety: For security purpose WSN is playing an effective role, as it usage in airports help in determining presences of weapons, specialized luggage tags, physical location of disaster.

1.4 Security in WSN

Wireless sensor networks are becoming more popular in most critical environment as in military, hospitals etc where they have to perform mission-critical tasks. Security is of main concern in such sensor networks because of their resource constraints and of the nature of communication they do i.e. wireless. So implementing security in such network is more important than that of wired networks. Without taking security factor into consideration an attacker can easily analyze the packet and breach out the important information being transmitted between the nodes, known as eavesdropping or can easily inject their own packet. Researchers found that in sensor networks, security should be implemented during design time for ensuring secrecy of sensitive information, privacy of people and safe operation over sensor networks.

1.4.1 Security requirement of WSN

Major security requirement of WSN have been listed below.

1.4.1.1 Data Confidentiality: Because of the inherited vulnerability of WSN, data confidentiality should be made the major ingredient of Security policy created for such networks where sensor nodes are capable of sending the data securely to the neighbor nodes, especially in military environment. Apart from this other kind of sensitive data like public and private key should be made secure from traffic analysis.

1.4.1.2 Data Integrity: Adding confidentiality doesn't mean that entire security is achieved. An attacker after sniffing data can alter the alter it and again can inject within the network that after reaching the node can initiate some malicious activity which can give wrong results or even can crash the entire network. So Data Integrity is yet another requirement of such networks
.

1.4.1.3 Data Freshness: Sensors should made sure that data send over the network should be fresh i.e. no old messages should be replayed over network. This is basically used in case of shared keys that keep on changing and if this requirement is not considered the attacker once sniffing the key would replay with it again and again. For this counter must be used that can determine freshness of data.

1.4.1.4 Authentication: An attacker in spite of modifying the data packet can inject stream of packet by itself so the receiver must ensure that is originated from the intended source. Also this feature is necessary for performing various administrative task required for managing sensor networks.

1.5 Defensive measure

Eavesdropping is the major threats within WSN where attacker can sniff the packet and even changes it. Effective measure should be taken to save the data from the malicious attack that affect the secrecy of data. One of the solution given by researchers is in the modulating the data in a unreadable human form that ensures secrecy and authorization of data. In the following section it has been discussed that how it can be achieved:

1.5.1 Cryptography

Cryptography is a science and art for encrypting sensitive information in a unreadable human format while communicating over unsecured media so that it is transmitted and processed by intended receiver. It basically involves two core mechanisms Encryption and Decryption. Figure 2 gives a brief description of entire cryptographic mechanism. Initially the messages is encoded into human unreadable format known as Encryption from the sender

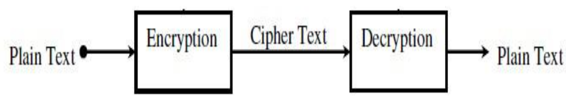

Figure 2: Cryptographic Method

side and then released over the network, on receiving the message is decoded at the receiver side using the same algorithm known as Decryption. Here the algorithm used for encryption and decryption is known as cryptophytes and the encoded message is cipher text and decoded message is known as decipher text. The algorithm or key used in cryptosystem are very complex as they consist of mathematical formulae and concept. This reveals the fact that the strength of any ciphered message depends upon the difficulty level of understanding the algorithm. Cracking such algorithm can be possible but it consumes a large amount of resources in terms of time, power and money. Basic purpose of doing cryptography is to achieve three things.

1.5.1.1 Confidentiality: It means the secrecy of message should be maintained.

1.5.1.2 Authentication: Only the intended receiver should receive the message.

1.5.1.3 Non Repudiation: It refers to the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated. The basic cryptographic algorithm is divided into two kind i.e. Symmetric and Asymmetric Cryptographic algorithm. They can be discussed briefly as in following section:

1.5.2 Symmetric Algorithm

As the name specifies in Symmetric Algorithm same key is given to both sender and the receiver and because they are kept private so it is also knows as private key algorithm. The entire mechanism has been shown in the Figure 3: Here the sender encrypts the message using the secret key and the receiver decrypts it using the same key. There is Number of Algorithms that follows symmetric key algorithm as basic principle like DES, RSA etc and many more are there.

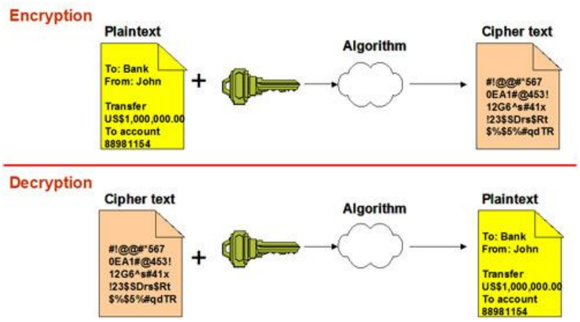

Figure 3: Symmetric Key Mechanism

Scalability, Key distribution and security are the major issue in symmetric key mechanism.

1.5.3 Asymmetric Algorithm

Asymmetric key cryptography is so called as here both sender and receiver are allocated different set of keys i.e. public key and private key. Here public key is known to all the users over the network want to communicate with the owner and the private key is only known to the owner. Here the algorithm generates a set of pair for every system, but they are not mathematically related i.e. if any intruder gets hold of any of the key that cannot be obtained. However the messages encrypted with a private key can only be decrypted using the corresponding public key. A general scenario of public key cryptography has been shown in the Figure.4: Here the sender will encrypt the message with receivers public key in secure message format this can as only be decrypted using receivers private key so ensures the confidentiality and authenticity. Main advantage of Asymmetric algorithm is Highly Scalable, Proper key distribution, proper security.

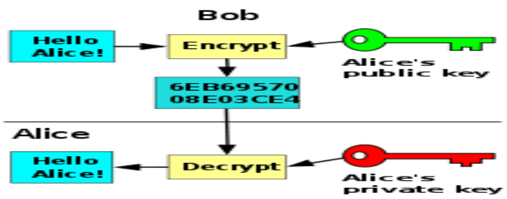

Figure 4: Asymmetric Key Mechanism

1.5.4 Elliptical Curve Cryptography

Elliptical Curve Cryptography or ECC is found to be the best solution until now as RSA was subsided because it doesn't fully satisfies the resource constraints. According to ECC based on Deffi-Hellman Algorithm is depicted in Figure 5 where a point G is selected from Elliptical curve E. For Alice(A) and Bob(B) communicating, A generates

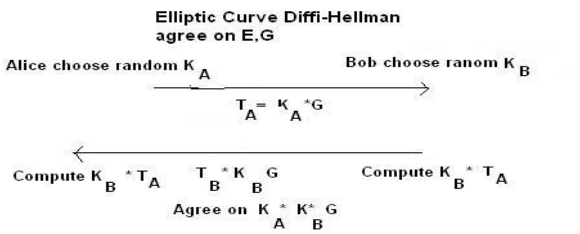

Figure 5: Elliptical Curve Cryptography Mechanism

private key K_a and the public key is generated as T_A as $T_a = K_a * G$ and B generates the K_b as private key and public key as $T_b = K_b * G$. Here they generates their shared key as $K_a * T_b = K_a * K_b * G$ and Bob computes the shared key as $K_b * T_a = K_b * K_a * G$. Because $K_a * T_B = K_b * T_b$, now Alice and bob now shares a secret key. This paper is an attempt to implement ECC on WSN and showcase its usefulness over other techniques.

1.5.5 Comparison of RSA and ECC

A public key cryptosystem is considered to be more secure as here public key is used for communication per discussion in above section of literature survey. There are number of algorithm that has been discussed in this concern like RSA and ECC. One that is adopted in WSN scenario depends upon the consumption of resources like memory, battery, power resources etc. Apart from mathematical part let us discuss how ECC is better than the RSA.

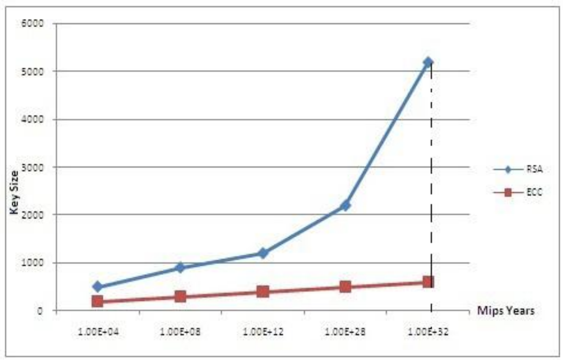

Figure 6: Comparison of Security level ECC vs RSA[3]

RSA algorithm is based upon factorizing method where main principle is to choose big number as a key i.e. hard to factorize whereas ECC, as already mentioned above, based upon elliptic curve discrete logarithmic problem of finite field, that take millions of years to break it. Following graph Figure 6 shows a comparison with the key size generated with two algorithms and the time to break each of them. With the above graph we conclude that ECC with small size provide equal security level and comparatively take hard to break which is a positive side of ECC algorithm when implemented in WSN scenario. The aspects where we can compare the two algorithms is in the terms of time and power consumption for performing verification, signature generation and key exchanged operation. This can be shown by the help of following two tables: Here the results are shown corresponding to two key sizes RSA-1024, 2048 and ECC-160,224. From the above results we conclude that ECC is an optimum algorithm that we can implement in WSN scenario. As represent in the table

Table 1. Time Consumption vs. Algorithm and Key Size [3]

Algo rithm	Key Size(Bit)	Key Exchange Client(s)	Key Exchange Server(s)	Signature Sign(s)	Signature Verify(s)
RSA	1024 2048	1.12 4.14	22.03 166.85	22.03 166.85	0.86 3.89
Algo rithm	Key Size(Bit)	Key Exchange Client(s)	Server(s)	Signature Sign(s)	Verify(s)
ECC	160 224	1.62 4.38	1.62 4.38	1.65 4.46	3.27 8.84

Table 2. The estimated Power Consumption in WSN [3]

Algo rithm	Key Size(Bit)	Key Exchange Client(s)	Server(s)	Signature Sign(s)	Verfiy(s)
RSA	1024 2048	39.96 136.62	726.99 5506.05	726.99 5506.05	28.38 128.37
ECC	160 224	53.46 144.54	53.46 144.54	54.45 147.18	107.91 291.72

2. INTRODUCTION TO JAVA SUNSPOT PLATFORM

Java SunSpot kit is basically used here for simulating a wireless sensor network. The kit has two sensors and a basestation as shown in Figureure

2.1 SunSpot Motes

Sun Spot (Sun Programmable Object Technology is a wireless sensor network (WSN) mote developed by Sun Microsystems. The device is built upon the IEEE 802.15.4 standard the IEEE 802.14 standard. Unlike other available mote systems, the Sun Spot is built on Squawk Java Virtual Machine.[4] Figure 7 demonstrates the external Figure 8(a) and internal view Figure 8(b) of SunSpot devices showing the major parts of SunSpot kit .

2.1.1 Internal Structure of SunSpots

As has been shown in Figure 4.1(a), SunSpot device major hardware part has been discussed below:

2.1.1.1Sun roof: This is the shielding part of main circuit of the device which can be opened and close by pressing a cork on its body [4].

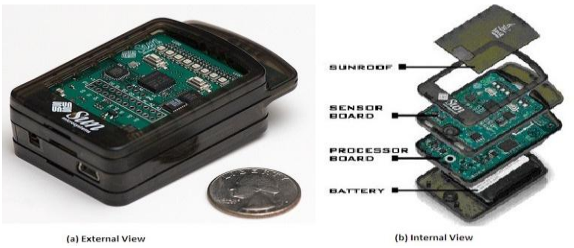

Figure 8 External and Internal view of SunSpot devices

2.1.1.2 Processor: Processor of SunSpot consists of 180 MHz 32 bit ARM920T core with 512 RAM. Along with IEEE 802.15.4 radio with integrated antenna for over the air communication and an USB interface which can be connected to external devices [4].

2.1.1.3 Sensor Board: The sensor board is the major part of the device that comprise of Light sensors with 8 tri-color LED's. Analog Inputs, 2 momentary switches and 5 general purpose I/O pins and 4 high current devices [4].

2.1.1.4 Battery: Sun Spots devices are provided with 3.7v rechargeable mAh lithium-ion battery. This can be charged by connecting it with external devices [4].

2.2 Softwares for Spot Manager

Installing Spot Manager is not an easy task as it requires number of software to be installed already on your computer. But the best part is that the setup asks and provides the required one if any of them is not installed. Here it is assumed that the installation procedure is supported by internet connection. Following is the list of the pre-requisite softwares:

2.2.1 Sun Development Kit: SDK or Sun Development kit consist of all the packages and classes requires for running and deploying application on SunSpot.

2.2.2 Java Net beans: Java Neatens provide GUI for developing SUNSpot application come along with the SDK SUNSpot modules.

2.2.3 Ant server: Apache Ant server provides various xml files required for deploying, accessing info, running application etc.

3. IMPLEMENTATION

The entire implementation is conducted by following the steps given below. The major softwares used here is Apache ant, SDK and NetBeans:

3.1 Accessing Spot Info

Connect the SunSpot device to the system for accessing its information this can be done by using ant info command in the root directory where SDK is installed as shown in Figure 9(a) From the Figure 9(b) it can be analyzed that two SunSpot devices are there having the IEEE address as 0014.4F01.0000.181D and 0014.4F01.0000.0E14. These addresses are basically used for accessing the spot remotely. And as here cryptographic mechanism is of major concern, so it can be identified that no keys are previously installed on device.

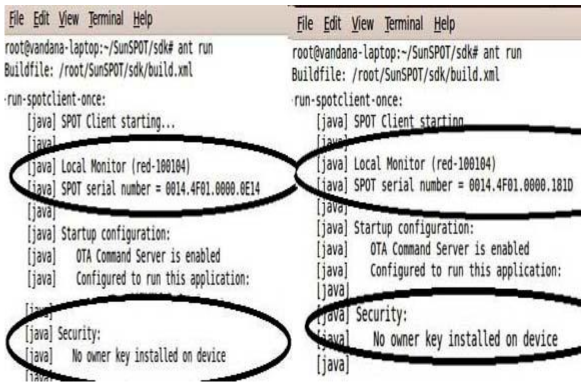

Figure 9(a, b) Initial information of a SunSpot devices.

3.2 Deploying BounceDemo Application

As the process of deploying any application on a Spot device is same let us consider it on the device having address 0014.4F01.0000.0E14.

3.2.1For running any application on SunSpot device proper library suite is to provide that assist devices to perform their function. This can be done using ant flashlibrary command. The outcome of this command when successfully build is shown in Figure 10.

3.2.2Here BounceDemo-onSpot application is deployed over the device using ant-deploy as shown in Figure 11 This application basically bounces the light ball over two devices making them communicate with each other.

3.2.3Finally the application can be made to run on the devices using ant runcommand and the outcome is shown in Figure

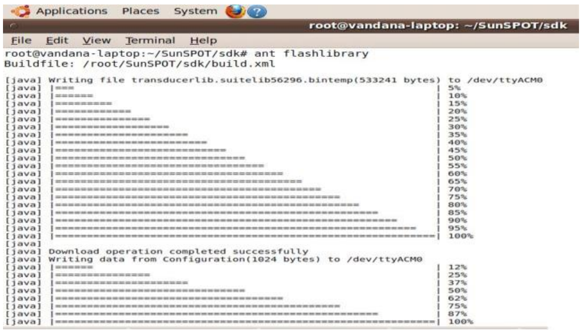

Figure 10 Output of ant ashlibrary ash the library command.

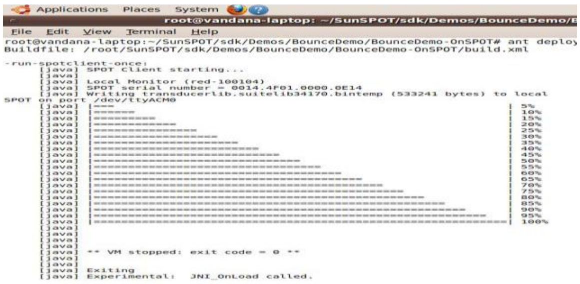

Figure 11 Output of ant deploy command on SunSpot.

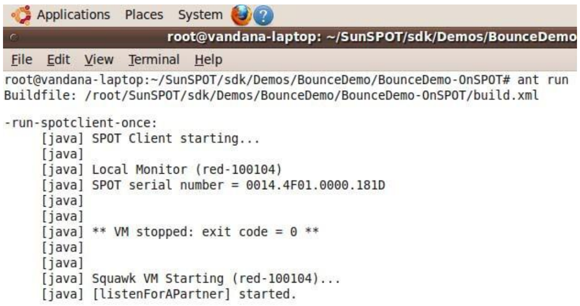

Figure 12 Output of ant run command.

3.3 Running Base Station for Capturing Packets

Now for running Packet Sniffer and Sniffer client on BaseStation. It is required to disable Over-the-Air (OTA) and mesh routing over it. This is performed in the following way:

3.3.1 **Accessing BaseStation Info**: BaseStation information can be accessed similarly as that of SunSpot using ant info.The initial info is as shown in Figure 13.

3.3.2 **Disabling OTA and Mesh routing**: Now for disabling routing and OTA we have to issue ant disableota as shown in Figure 14.

3.3.3 **Starting as a BaseStation** : A BaseStation can also be used as a Spot device. However to specifically start it as a basestation, ant startbasestation command required to be issued. This can be done in the following way:

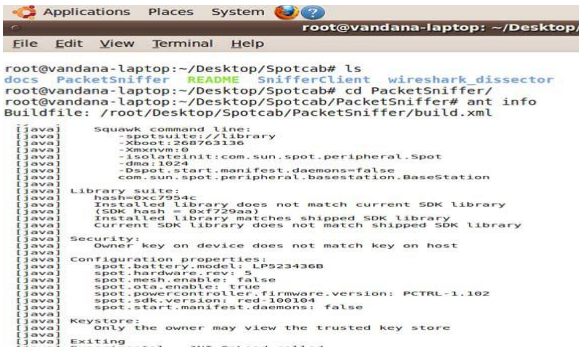

Figure 13 Output of ant info command for basestation

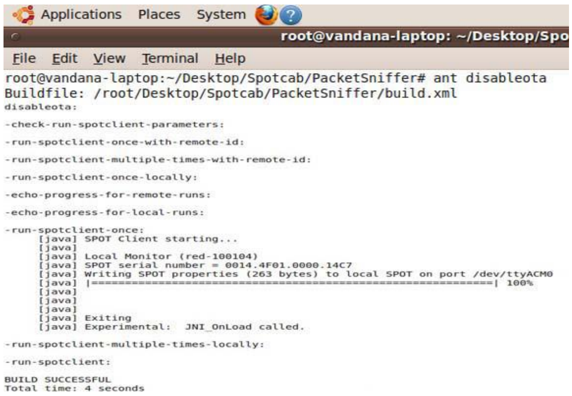

Figure 14 Output of ant disableota command.

4. SNIFFING PACKETS USING BASESTATION

After getting basestation prepared, Packet Sniffer application is to be deployed and then running the sniffer client in the following way:

4.1 Browsing and Analyzing Sniffed Packets

After running the sniffer client, the entire captured packet is stored in "sniffer.Dump" file in the same folder of application. This file can be opened using wireshark as shown in the following Figure 15

4.2 Analyzing Encrypted Communication

Until now the spot devices are made to communicate and data is captured which is quite unsecured. Now data will be crypted using ECC algorithm and then analyzed. The steps are been taken from readm.txt file in SSL folder given on the webpage [5]. The entire encryption steps are described in the following section:

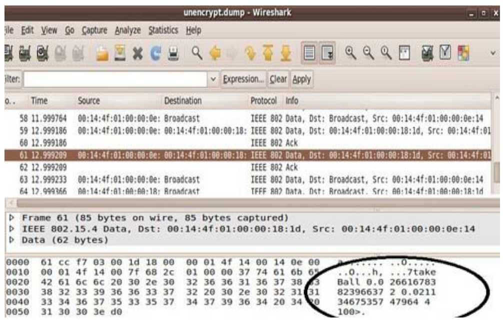

Figure 15 Wireshark output showing unencrypted data

4.3 Deleting Existing Public key:

Before generating any key or secure communication, the device has to bring in the keyless state. This can be done using ant deletepublickey command as shown in Figure 16. the keystore became empty after executing the command.

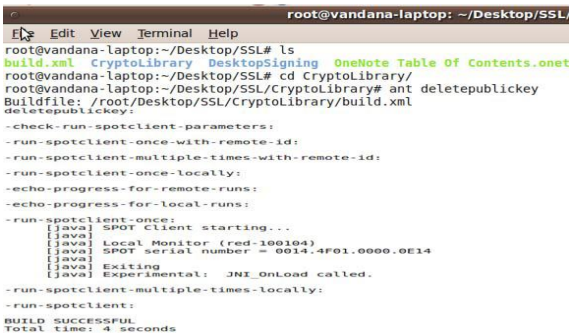

Figure 16 Output of ant publickey command for basestation

4.4 Editing sun property file and creating suitable library:

Now to create a new library certain changes has to be made in .sun:properties file stored in user account when the SDK is installed properly. This entire changes has been given in the Figure 17 and Figure 18 it can be seen that a new library "cryptolib" is to be created which contain three jar file SSL:jar, cryptocommon:jar and a host jar file SpotCryptoClient:xml:

Figure 17 Similarly the creation of SSL:jar and host jar can be shown in Figure 19 and 20

Finally a new library is created by issuing ant library command as in Figure 21 And then flashing this library on the Sun Spot as in Figure 22 With the creation of new library the ant server will support new list of commands for manipulating like listtrustedkeys, addtrustedkey, deletetrustedkeys and others. However here for making the two device trust each other, a key would be added in both the device as shown in Figure 23 Now to list out new set of keys can be listed using ant listtrustedkeys as shown in Figure 24

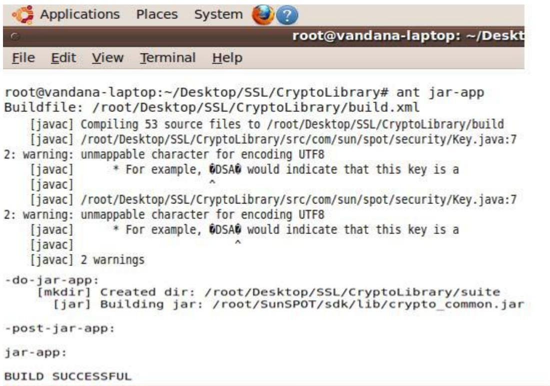

Figure 18 Output of ant jar-app command.

Figure 19 Output of ant jar apps command.

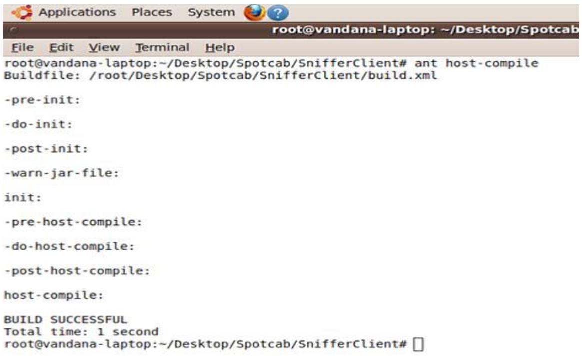

Figure 20 Output of ant info command for basestation.

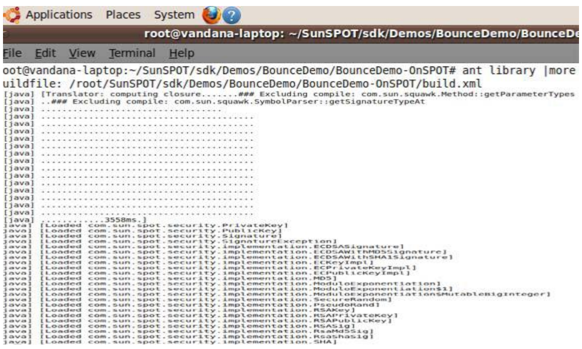

Figure 21 Output of ant library command.

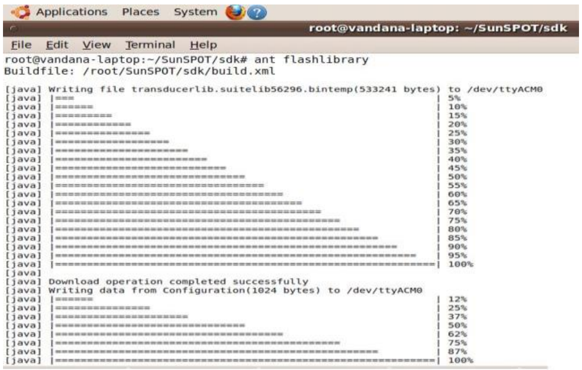

Figure 22 Output of ant reset command.

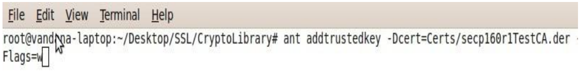

Figure 23 Output of ant addtrusted command.

4.5 Deploying application and creating keys on device : Here again BounceDemo-onSPOT application is deployed as described in previous sections. However to support the crypting algorithm “radiostream" is changed to “sradiostream" which are analogue of “http" to \https".

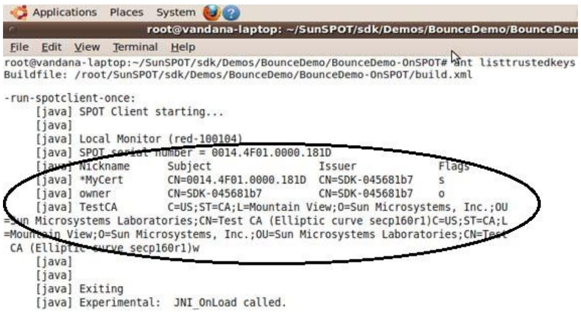

Figure 24 Output of ant listtrusted command

4.6 Analyzing Encrypted packets

Now Sniffer Client on the Base Station can be defined similarly in the same way as described in the previous section now when we open the encrypted packets in wireshark as shown in Figure 25.

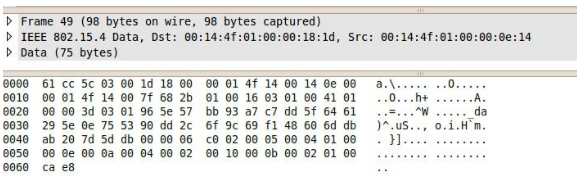

Figure 25 Output of wireshark showing encrypted data.

The Figure 25 shows any data exchanged between two spot is encrypted and cannot be deciphered easily. As in previous section we can easily see clear text flow.

5. CONCLUSION AND FUTURE SCOPE

This thesis investigated the mathematical foundation of Diffe-Hellman key exchange protocol and the elliptic curve cryptography for the purpose of understanding the practical problems of implementing the theoretical concepts on wireless sensor networks. The main results are as follows:

- Designed a technique for establishing secure communication between nodes in wireless sensor networks. The protocol is not vulnerable to man-in-middle attacks problem
- Implemented the technique over Java Sun Spots for its analyzing the cryptographic behavior. Here one spot sent the light information to other. It appears as ball bouncing between the SunSpots. The packet captured first, in human readable form and, then in cryptographic form.

In future this research work can be extended as:

- Design and implement a set of attacks against ECDH protocol.
- Testing the key generation process between multiple Sun Spots nodes and test it in a multiple using complex environment of WSN networks

6. REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E.Cayirci "Wireless sensor networks: a survey" Published by Elsevier Science B.V., 2002.

[2] Al-Sakib Khan Patan, Choong Seon Hoong, Hyung-Woo Lee, "Smartening the Enviornment using Wireless Sensor Networks in Developing Country", CACT,2006.

[3] F.Amin, A.H. Jahangir and H.Rasifard "Analysis of Public Key Cryptography", World Academy of Science Engineering and Technology, 2008.

[4] Sun Labs, “SunSpot Owner’s Manual", copyright Sun Microsystems

[5] Sun Labs, http://www.sunspotworld.com/GettingStarted /Linux.html", copyright Sun Microsystems.

[6] F.L. Lewis, Associate Director of Research and Head, Advanced Control and Sensor MEMS Group "Wireless Sensor Networks", Smart Environments: Technologies, Protocols, and Applications, Network 2004.

[7] Hemanta Kumar Kalita, Avijit Kar “WIRELESS SENSOR NETWORK SECURITY ANALYSIS",International Journal of Next-Generation NetworksIJNGN),Vol.1, No.1, December 2009.

[8] Anoop MS, "Elliptic Curve Cryptography-An Implementation", www.security.ittoolbox.com/research.

[9] J.Katz and Y.Kindell, "Introduction to Modern Cryptography", Champan and Hall/CRC 2008

[10] Ian Curry "An Introduction to Cryptography and Digital Signatures", Copyright-Entrust, version 2.0, March 2002.

[11] Dr.Rahul Banerjee, Introducton to Symmetric Key Cryptography, BITS Pilani presentation.

[12] B.A.Frouoazan, "Cryptography and Network Security", International Edition McGrraw Hill, 2008