**National Institute of Technology,Raipur**

**Department of Computer Science & Engineering**

****

**A Term Project on Network Programming**

**GitHub Project Link: https://github.com/anku5991/NP-Project**

Submitted By:

Roll no:14115009 Name: Amitesh Gupta

Roll no:14115021 Name:Ayushi Priyadarshi

Roll no:14115088 Name:Swati Verma

Roll no:14115092 Name: Vinita Nirmal

**ABSTRACT**

Internet protocol security (IPsec) is a set of protocols that provides security for Internet Protocol. It can use cryptography to provide security. IPsec can be used for the setting up of virtual private networks (VPNs) in a secure manner.

**Secure IP** is a project which will secure the communications by authenticating and encrypting each IP packet of the communication session; includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of the cryptographic keys to be used during the session.

Secure IP uses RSA algorithm for encryption and decryption of messages which are transferred between sender and receiver. RSA is asymmetric cryptography algorithm which works on two different keys i.e. **Public key and Private key.** Public key is given to everyone and private key is kept private. For example a client sends its public key to the server and requests for some data. The server encrypts the data using client’s public key and sends the encrypted data. Client receives this data and decrypts it.

Initial authentication is done by generating nonce and by mutual exchange and verification of nonce we can actually authenticate if the other person is actually the designated receiver.

To make sure that any third person does not get access to the sent message and the receiver can receive the message without any difficulty. We have used SHA-1 cryptographic hash function.

SHA-1 produces a 160-bit hash value or message digests from the inputted data (data that requires encryption), which resembles the hash value of the MD5 algorithm. It uses 80 rounds of cryptographic operations to encrypt and secure a data object. SHA-1 is commonly used in cryptographic applications and environments where the need for data integrity is high. It is also used to index hash functions and identify data corruption and checksum errors. A detailed description of the project, its working and networking concepts will be discussed later in the project.

**LIST OF FIGURE**

**Figure1:** Flowchart.

**Figure2:** Server Side Socket Creation.

**Figure3:** Client Side Socket Creation.

**Figure4:** Encrypting Function defined in Trial Class.

**Figure5:** Decrypting Function defined in Trial Class.

**Figure6:** buildKeyPair function in Trial class Which returns a KeyPair object.

**Figure7:** Server sending Public Key.

**Figure8:** Client receiving server's Public Key.

**Figure9:** Client side NONCE Authentication.

**Figure10:** Message Digest using SHA defined in Trial class.

**Figure11:** Server Side Screen Shot.

**Figure12:** Client Side Screen Shot.

**TABLE OF CONTENTS**

* Abstract
* List of Figures
* Table of Contents
* Introduction

MOTIVATION

PROBLEMS (faced and overcame)

PROJECT DESCRIPTION

* Network Programming Side

CRYPTOGRAPHY SIDE

* Screenshots
* Conclusion

**INTRODUCTION**

**MOTIVATION**

There is a great plethora of hackers around the globe and that too is increasing at an exponential rate. A hacker can be very catastrophic especially when he can do the following:

1. If a hacker gets hold of IP address of a server, he/she could launch a Distributed Denial of Service (DDoS) attack on it, and make the service unavailable for the intended users.
2. He/she could launch Brute Force SSH attacks and attempt to gain access to the machine.
3. He/she could also scan for the services running OR any open ports on the host and would try to exploit it in some manner. (any FTP, Mail, Mysql, VNC etc)
4. If the user and the hacker share the same LAN, the hacker could target the user with Address Resolution Protocol spoofing and launch a Man in the Middle (communications attack) attack for all the internet communications.

So I pondered on making my project on network security by contributing to secure IP.

**PROBLEMS (faced and overcame)**

* As we have implemented our project in java and java is completely object oriented and contains many classes and built in functions, so it was difficult for us to understand which class contains which method and how to use them in our project.
* Secondly, we had only read about cryptography in distributed operating system and never actually implemented it, hence it was totally a new domain for us.
* Other than RSA algorithm, there were many more algorithms for encryption and decryption of messages, so it was difficult for us to choose among different algorithms.
* Generation of key pairs was quite complex.
* The greater challenge was exchanging of keys such that it cannot be tampered by the attacker and authentication of messages.

**PROJECT DESCRIPTION JAVA**

**FLOWCHART**

**Socket Creation**

**Exchanging of public keys**

**Generation and encryption of nonce(a random number) by client using server’s public key**

**Decryption of nonce by server using its private key & encryption of the same by server using clients public key**

**Decryption of nonce by client using its private key**

**Is Rec\_nonce== sent-nonce**

**Classes made**

1. Frame.java ::: A class which contains the data to be sent to via Socket.

2. Trial.java ::: Contains the functions used in common between the client and server.

3. server.java ::: Server side Code.

4. client.java ::: Client side Code.

**Socket Creation**

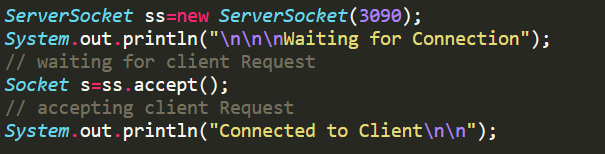
****

Fig 2:Server Side

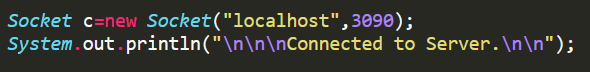


Fig 2:Client Side

1. Socket Creation using ServerSocket ss=new ServerSocket(3090) and asking it to listen to the port number 3090.

2. Client creating Socket using Socket c=new Socket("localhost",3090); and requesting to connect IP:"LOCALHOST" in the Port number 3090.

3. Socket accepting Clients request using Socket s=ss.accept();

**Encryption And Decryption**

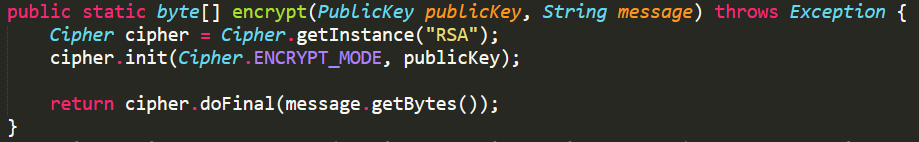


Fig 3:Encrypting Function defined in Trial Class.

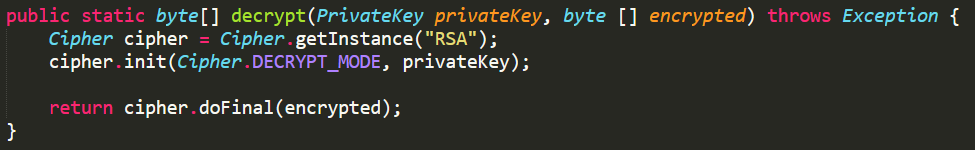
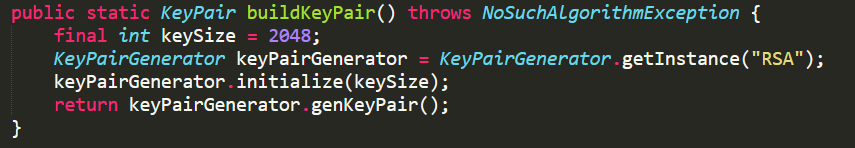


Fig 4:Decrypting Function defined in Trial Class.

**Generating KeyPair**

****Fig 6:buildKeyPair function in Trial class Which returns a KeyPair object

**KeyExchange**

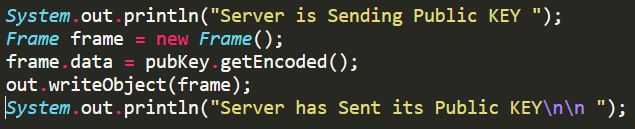
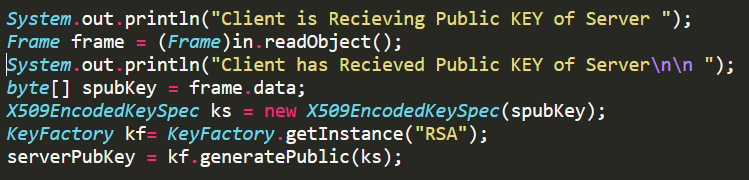
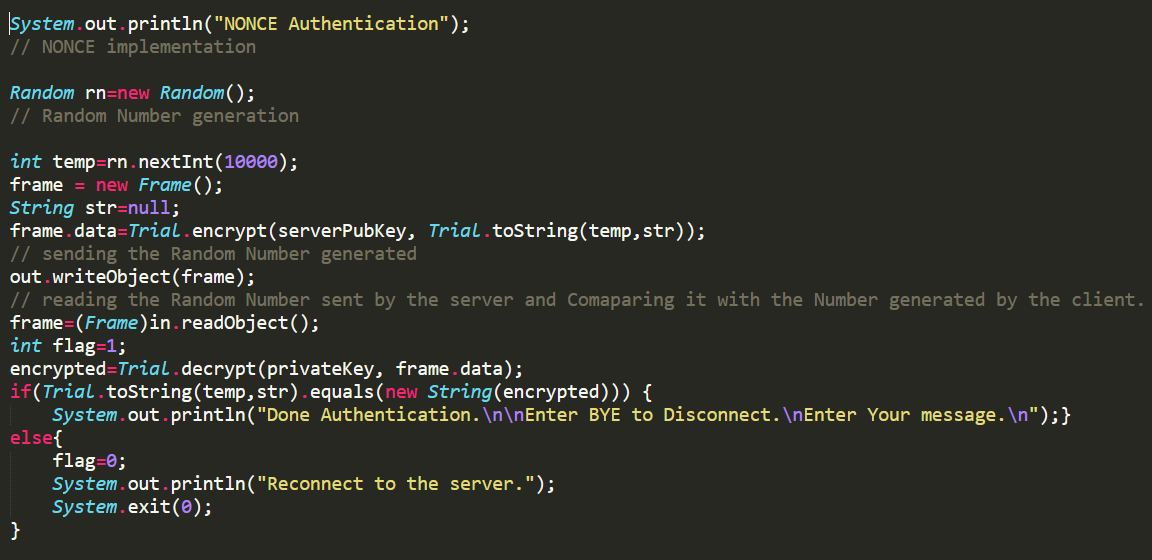
****

Fig 7:Server sending Public Key.

**** Fig 8:Client receiving server's Public Key.

**NONCE Authentication**

****Fig 9: Client side NONCE Authentication

**Verification of messages using MessageDigest**

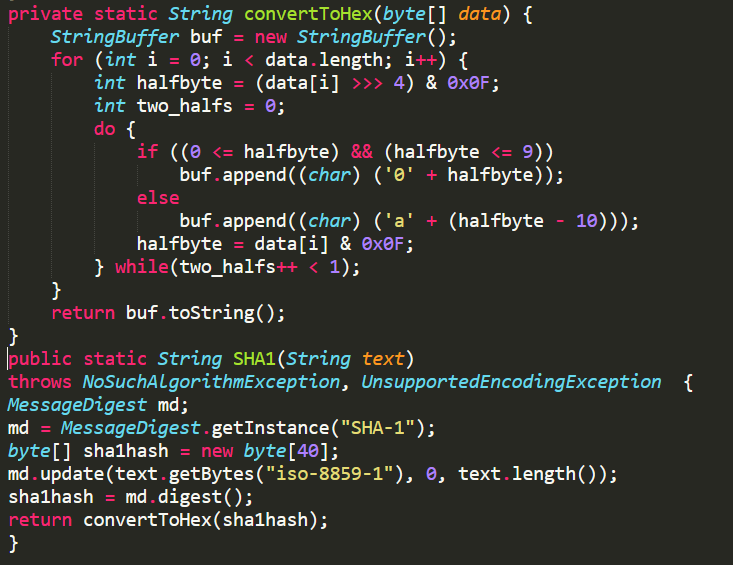
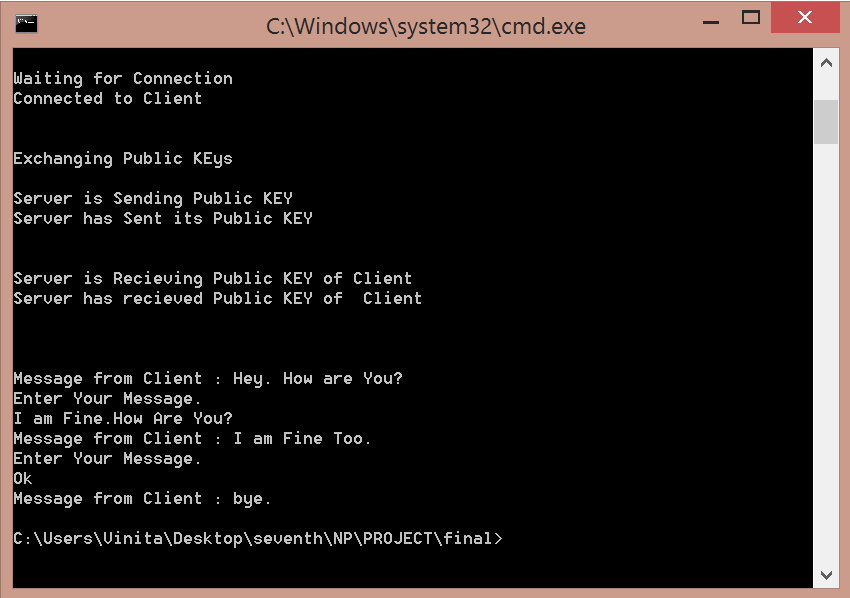
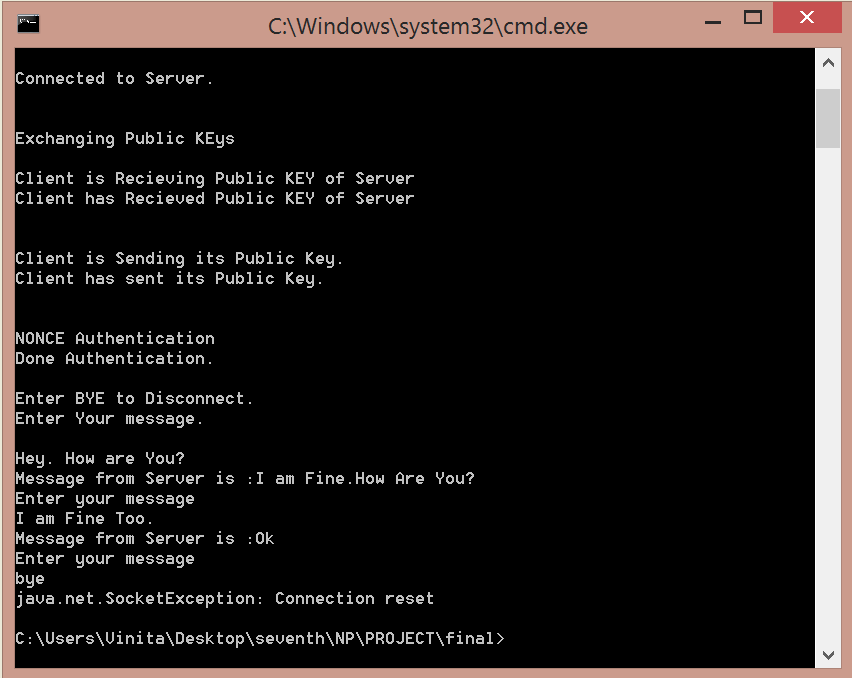
****

Fig 7: Message Digest using SHA defined in Trial class

**Server Side Screen Shot**

****Fig 8: Server Side Screen Shot

**Client Side Screen Shot**

****Fig 9: Server Side Screen Shot

**CONCLUSION**

We have successfully developed a project on Secure IP that allows two users to communicate by exchanging public keys in a secure way. Authentication and verification of messages helps to ensure privacy of the message. We have implemented the programming concepts in an effective and efficient way minimising the use of memory and space. We have made the code in a very reusable form and the code is easily understandable.

From our side we have tried our level best to explain the project in the most simple manner by adding as many screenshots as possible.

Thanking you !!