Implementation of File Transfer between Server and Client using Python and RSA Encryption

|  |
| --- |
| Joel Jogy George  ECE Dept., Section B, MIT Manipal |

***Abstract*—This project puts forward the idea of using Telnet in which clients can connect to Python chat server and globally communicate with one another. Messages submitted to the chat server are viewed by others (in addition to management information, such as clients joining or leaving the chat server). Sockets can be thought of as endpoints in a communication channel that is bi-directional, and establishes communication between a server and one or more clients. Here, we set up a socket on each end and allow a client to interact with other clients via the server. The socket on the server side associates itself with some hardware port on the server side. Any client that has a socket associated with the same port can communicate with the server socket.**

**.*Keywords—socket; cloud computing; java system; cloud storage; network programming***

I. INTRODUCTION

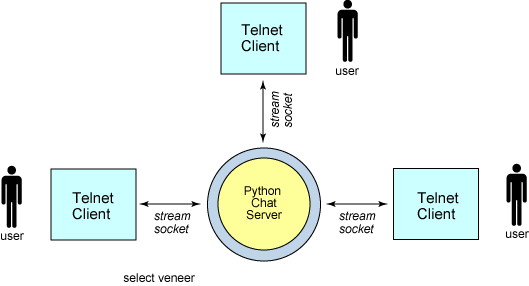
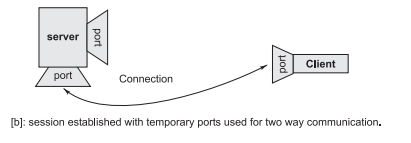
The server side script will attempt to establish a socket and bind it to an IP address and port specified by the user (windows users might have to make an exception for the specified port number in their firewall settings, or can rather use a port that is already open). The script will then stay open and receive connection requests, and will append respective socket objects to a list to keep track of active connections. Every time a user connects,  
a separate thread will be created for that user. In each thread, the server awaits a message, and sends that message to other users currently on the chat. If the server encounters an error while trying to receive a message from a particular thread, it will exit that thread. This server can be set up on a local area network by choosing any on computer to be a server node, and using that computer’s private IP address as the server IP address.  
For example, if a local area network has a set of private IP addresses assigned ranging from 192.168.1.2 to 192.168.1.100, then any computer from these 99 nodes can act as a server, and the remaining nodes may connect to the server node by using the server’s private IP address. Care must be taken to choose a port that is currently not in usage. For example, port 22 is default for ssh, and port 80 is default for HTTP protocols. So these two ports preferably, shouldn’t be used or reconfigured to make them free for usage.

Fig.1 Client-Server communication

II.CHAT SERVER

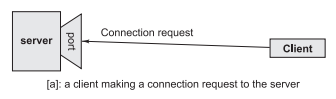
The chat server does the following things

1. Accept multiple incoming connections for client.  
2. Read incoming messages from each client and broadcast them to all other connected clients.



The server handles multiple chat clients with select based multiplexing. The select function monitors all the client sockets and the master socket for readable activity. If any of the client socket is readable then it means that one of the chat client has send a message.

III.CHAT CLIENT

The client is based on the telnet. It connects to a remote server, sends messages and receives messages.

The chat client does the following two things:

1. Listen for incoming messages from the server.  
2. Check user input. If the user types in a message then send it to the server.

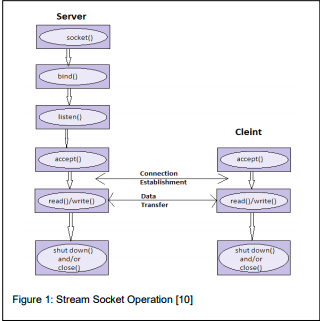


Fig.4. Stream Socket operation

IV.SOCKET

Sockets allow communication between two different processes on the same or different machines. To be more precise, it's a way to talk to other computers using standard UNIX file descriptors. In UNIX, every I/O action is done by writing or reading a file descriptor. A file descriptor is just an integer associated with an open file and it can be a network connection, a text file, a terminal, or something else.

V.RSA

The following is a description of how RSA is used:

The RSA is a block cipher whereby the plaintext and ciphertext are integers between 0 and n-1, for some n.

A typical size for n is 1024 bits.

In the RSA algorithm, one party uses a public key and the other party uses a secret key, known as the private key. Each station randomly and independently choose two large primes p and q number, and multiplies them to produce n=pq. This is the modulus used in the arithmetic calculations of the RSA algorithm (Rivest, Shamir, & Adleman, 1978).

The process of the RSA algorithm is as described below:

1. Select p and q (both should be prime numbers)
2. Calculate n=pq
3. Calculate z=(p-1)(q-1)
4. Select integer D which is relatively prime to 2. Gcd φ(n) D=1(φ9n)=z)
5. Calculate ED-1 mod(φ(n))
6. For Encryption:

C=PE mod n

1. Where P is Plaintext, C is Cipertext (encryption)
2. For Decryption:

P=CD mod n

Public key encryption algorithm uses a public key of PU=(e,n) and private key of PR=(d,n).

V.NUMERICAL STUDY ON RSA ALGORITHM

*Example: 1*

1. Select two prime numbers, p=17 and q=11
2. Calculate n=pq=17\*11=187
3. Calculate φ(n)=(p-1)(q-1)=(17-1)(11-1)

=16\*10=160

1. Select e such that e is relatively prime to φ(n)-160. So, we select e=7
2. Determine d such that ed=1 mod φ(n)

7d=1 mod 160 7\*23=1 mod 160 161=1 mod 160

(d is calculated using extended Euclid’s Algorithm) Here, Public key PU(e, n)=7, 187

Private key PR(d, n)=23, 187 Suppose, the Plaintext value (M) is 88 then,

1. For Encryption,

Ciphertext C = Me mod n

= (88)7 mod 187 = 888832 mod 187 = 11

1. For Decryption,

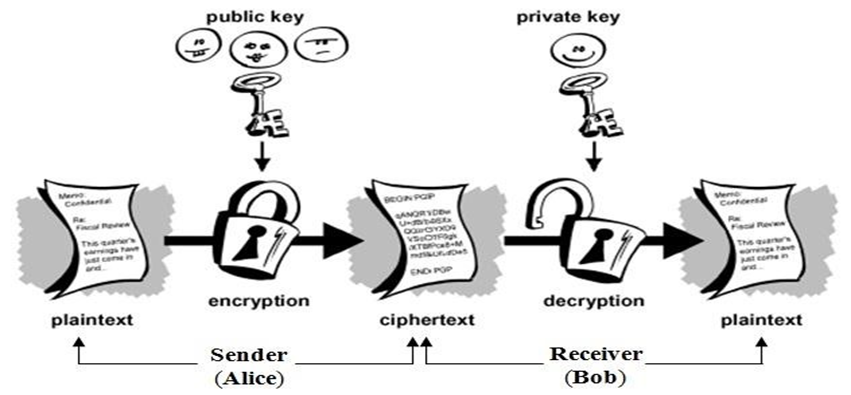
Plaintext P = Cd mod n

* 1123 mod 187
* 79720245 mod 187
* 88

VI.PUBLIC KEY CRYPTOSYSTEM

The development of public key cryptography is the greatest and perhaps the only true revolution in the entire history of cryptography [4]. With public key techniques, each user has two different keys, one made available to the public and the other kept secret. One of the keys is used to encrypt a message, and the other is used to decrypt the message. If Alice wants to send a secret message to Bob, for example, she looks up Bob's public key and uses it to encrypt the message. Because Bob's public key cannot undo the encryption process, no one who intercepts the message can read it. Only Bob, who possesses the secret key corresponding to his public key, can read the message. Alice never has to meet Bob out of the hearing of others to exchange keys or passwords; this is a substantial improvement over older encryption methods in which an exchange of private keys was necessary.聽

This system can also be used as a means for Bob to be sure a message comes from Alice. If Alice wants to sign a message, she can encrypt it with her private key. When Bob receives an encrypted message which purports to be from Alice, he can obtain Alice's public key and decrypt the message. If a readable message emerges, Bob can have confidence that the message came from Alice, because Alice's public key would only properly unlock a message which was locked with her private key (known only to Alice). Figure-1 illustrates the public-key encryption process.

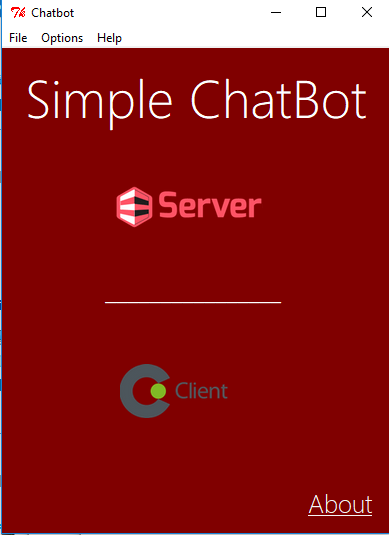


VII.SECURITY OF RSA

The security of RSA algorithm depends on the ability of the hacker to factorize numbers. New, faster and better methods for factoring numbers are constantly being devised. Obviously the longer a number is the harder is to factor, and so the better the security of RSA. As theory and computers improve, large and large keys will have to be used. The advantage in using extremely long keys is the computational overhead involved in encryption/decryption. This will only become a problem if a new factoring technique emerges that requires keys of such lengths to be used that necessary key length increases much faster than the increasing average speed of computers utilizing the RSA algorithm. RSA's future security relies solely on advances in factoring techniques.

VII.FRONT-END

Use of Tkinter module under Python libraries was used to develop the front-end of this project. Users have the option to send/receive messages from server/client and also set up RSA keys through the front-end.



VII.CONCLUSION

In this article, we have analysed on the value of usage of socket programming to be able to send and receive data among a single server and multiple clients. Also the value of RSA encryption/decryption was analyzed to encrypt data before sending it.. If the value of exponent in RSA is high, the security of RSA algorithm is also high. So, we proposed to implement the high value of exponent in RSA algorithm to produce a better security before sending packets.

# REFERENCES

1. Tanenbaum, “Computer Networks”, Second edition
2. Socket Concepts: http://publib.boulder.ibm.com/infocenter/iseries
3. H. Schildt, The Complete Reference, Seventh Edition, Chap. 27
4. Author P. Burden,” Socket Programming”
5. J. F. Kurose and K. W Ross, “Computer Networking- A Top-Down Approach featuring the Internet” , 2nd Edition (Addison Wesley World Student Edition)