

Secure communication Chat program using encryption

The assignment is to write an encrypted network client and server applications using encryption over regular network sockets.

We will build on assignment 0 to complete this assignment.

The client will take the following command-line arguments, a host name or IP address (both forms must be accepted), a port number, a "username" and a filename that contains the key used named "username.key".

The client initiates the connection to the IP address and port of the server. Upon connection, it will transmit a single line containing the username (not encrypted) and then all subsequent messages encrypted using DES and AES in a mode of your choice.

For more information about Cryptographic libraries in Java:

<http://java.sun.com/products/jce/index.jsp>

and

<http://www.cs.auckland.ac.nz/~pgut001/cryptlib/>

(both for java and C)

Until terminated, the client will read characters from the standard input (keyboard) and it will transmit them to the server program via the network. Anything it receives from the network connection transmitted by the server (encrypted), it will first decrypt and then display on the screen.

The server listens for and accepts a single connection. When a client connects, it uses the client's username to determine the right file containing the key to decrypt all subsequent messages received from this client. This assumes that both client and server have the same file containing the symmetric key used for encryption.

For all the messages he receives, the server replies back to the client adding "username said: " where username is the client's username and encrypting the messages using the connection key before transmitting to the client.

Extra Credit

- a) Extend the server to accept many connections (clients) with different keys. Use a single file to store all the keys in the server.

- b) Add an extra optional command line argument that will allow the transfer of a file from the client to all other clients connected to the server using encryption.
- c) Use a multi-thread client-server application to allow faster processing of requests
- d) Benchmark your implementation for various encryption algorithms (DES, AES, Blowfish, RC4, RC5, your choice)