**MINOR PROJECT 1**

**“SECURE FILE TRANSFER”**

**by**

**NAME Roll Number**

**Ashish Pandey R100217014**

**Mridul Thapa R100217039**

**Nikhil Mishra R100217043**

**Surabhi Kumari R100217101**

**Under the guidance of**

**Mr. Pushpendra Kumar Rajput**

Assistant Professor

Department of Cybernetics



**School of Computer Science**

**University of Petroleum and Energy Studies**

**Dehradun-248007**

**2017-21**

**APPROVED BY**

**Mr. Pushpendra Kumar Rajput Dr Monit Kapoor**

**Project Mentor Department Head**

**ABSTRACT**

Secure Data transfer based upon well-designed Data encryption and authorization systems helps defining the methods used to protect passwords and other credentials from being stolen. Transferring and storing passwords in plain text form leaves them at risk of exposure to attackers, eavesdroppers and spyware. In order to avoid such exposure, powerful encryption/validation systems use various mechanisms to minimize the possibility that unencrypted credentials which are exposed, as well as be sure that any private data that does get transmitted and stored will be of minimal use to an attacker. In this project we propose a method to protect data transferring by encryption techniques: symmetric AES algorithm used to encrypt files, asymmetric RSA used to encrypt files or data to ensure a secure transmitting between server-client or client-client from verifying in-between client and server and make it hard to attack by common attacked methods.

**Keywords: Connection, Encryption, Decryption. Communication, Socket**

**TABLE OF CONTENT**

**Contents**

1. **Introduction 4**
2. **System Analysis 5**

2.1 Hardware Interface

2.2 Software Interface

1. **Design 6**

3.1 Flowchart

1. **Problem Statement 7**
2. **Literature Review 7**
3. **Objective 8**

6.1 Objectives Achieved

1. **Algorithms 9**
2. **Output Screen 11**

7.1 Before Encryption

7.2 After Encryption

7.3 After Decryption

1. **Limitations and Future Enhancements 13**
2. **Conclusions 14**

**References**

1. **INTRODUCTION**

Instant messaging has been a boon to the computer industry and is now thoroughly used in contrast to email or other archaic form of technologies. There has been a rapid increase of these kind of services in the market, but the loopholes are the same. While wirelessly transmitting a message it is prone to the security gaps. For message encryption there has been 2 kinds of encryption being used: Asymmetric and Symmetric Encryption. Asymmetric encryption provides the public key and private key, the RSA Algorithm is a type of it. RSA has been the more secure algorithm for encryption, but is also not very good at handling longer strings.

In order to send or receive a message, the server socket is attached to a port, from where it transmits. The Linux based application extends the usage of Key based encryption to share and receive files that are not prone to unauthorized access.

The application helps a user to connect to someone and send & receive encrypted file to and fro. This application helps to convert a document containing plain text to a file with encrypted text which can only be accessed by the recipient who has the symmetric key to the file.. This guarantees the privacy of the data that is being shared. This GUI application helps in encrypting and decrypting a file and is not digitally signed. The advantage of this application is sending and receiving of file with encryption can be done without any active internet connection, as we are using simple messenger, and taking down the need of a centralized server.

1. **SYSTEM REQUIREMENTS**

**Hardware Requirements**

**Minimum requirements will be as follows:**

1. 256 MB RAM required
2. Processor with speed of 500 MHZ
3. Internet or LAN Connection

**Software Requirements**

1. Dev C/C++
2. Notepad Editor
3. VMware
4. Linux based OS

**3. DESIGN**

**3.1**

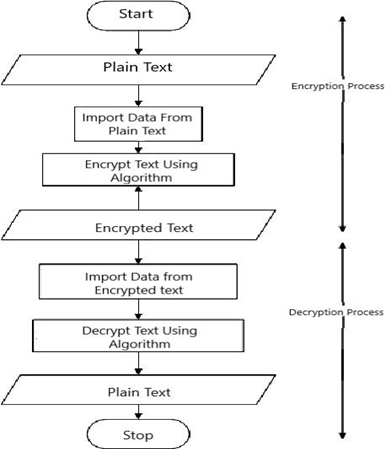
****

Figure1: Flowchart

**4. Problem Statement**

The client-server communication model is used in a wide variety of software applications. Where normally the server side is sufficiently protected and sealed from public access, but client applications running on devices like notebooks and desktops are considered insecure and exposed to security threats. The main weakness of client-server application is that there is no security provided to data which is transferred between clients. Any unauthorized client can hack the client account and can change the data. This is the main objective of this project (To secure the Data Transfer between Client and Server), using the key based encryption techniques.

**5. Literature Review**

Socket [1] is a standard connection protocol that supports data communication over the network between connected terminals. The standard connection supports the data transmission both by the TCP and UDP protocols between the terminals. TCP is a transport layer protocol used by applications that require guaranteed delivery of data. Basically, it is a connection-oriented protocol. To communicate over TCP, one must first have to establish a connection between pair of sockets, where one socket is client and the other belongs to server. After the connection is established between them then they can communicate with each other.

A client [2] is a system that accesses or desires for a service made accessible by a server. A server is a system (hardware or software) program running to provide the service requests of other system programs. Port is a software mechanism that allows the centralized connected Servers to listen for requests made by clients. Port is actually purposed as a gateway to listen for the requested parameters by the server terminals or other machines. It is a software address on a system that is on the network. Entire request response proceeding among this Application is carries through machine ports. Secure Sockets Layer protocol is used for encryption of data for secure data transmission. IP is the reasonable network address of the device on a network. It is notational called dotted-decimal.

It is more convenient to use peer-to-peer application because it operates directly to the clients. The communication to the servers consumes more time if compared to the peer-to-peer. That is why peer-to-peer is more preferred to be used for Data Transfer systems as main technique and also as an alternative method to use especially for the purpose of file transferring feature.

Advanced Encryption Standard (AES) is one of the most frequently used and most secure encryption algorithms available today. It is publicly accessible, and it is the cipher which the NSA uses for securing documents with the classification "top secret". RSA is one of the most successful, asymmetric encryption systems today. Originally discovered in 1973 by the British intelligence agency GCHQ, it received the classification “top secret”. We have to thank the cryptologists Rivest, Shamir and Adleman for its civil rediscovery in 1977. They stumbled across it during an attempt to solve another cryptographic problem.

**6. Objectives**

To develop a channel to transfer files securely, using key based encryption methods and comparing the different methods on Linux based OS. The sub-objectives incidental to the main objective are:

* Use of Asymmetric and Symmetric encryption methods to secure both ends of the communication channel
* To provide a secure channel for faster transmission of data
* To build a dependable and encrypted file transfer system between client and server
* To compare the asymmetric and symmetric encryption algorithms.

**6.1 Objectives Achieved**

* Implemented AES Algorithm in securing file transfer between client-server on Linux based OS.
* To provide a secure channel for faster transmission of data

**Algorithm**

**CLIENT**

**STEP1-**

*// Function designed for chat between client and server.*

void func (char\* b) {}

**STEP2-**

*Get the IP ADDRESS, PORT NO and print it using it as Hostname of the client.*

sockfd = socket(AF\_INET, SOCK\_DGRAM,

IP\_PROTOCOL);

*Print the Client is using Hostname and PORT NO as IP\_ADDRESS and PORT\_NO.*

**STEP3-**

When Counter == 0

*Print the name of the file*

*Send it to sockfd, net\_buf, NET\_BUF\_SIZE, sendrecvflag, tructsockaddr\*) & addr\_con, addrlen.*

**STEP4-**

*Counter++ and Return 0*

**SERVER**

**STEP1-**

Function to clear Buffer

void clearBuf(char \*b)

Function to encrypt

char chipher(char ch)

Function to send file

sendFile(FILE \*fp,char \*buf, ints)

add the socket

sockfd = socket(AF\_INET, SOCK\_DGRAM, IP\_PROTOCOL);

**STEP2-**

*Bind it*

*as if bind sockfd,struct sockaddr and sizeof addre\_con == 0 then print nextline otherwise print* ***ERROR***

*Print the server running* ***PORT NO***

**STEP 3-**

It receives the file name using func

recvfrom(sockfd, net\_buf, NET\_BUF\_SIZE, sendrecvflag,

(struct sockaddr \*)&addr\_con, &addrlen);

*and open the file using*

fp = fopen(net\_buf, "r");

*//source*

**STEP 4-**

if fp == null *then fail to open else it gets copied and again it is sent to*

sockfd, net\_buf, NET\_BUF\_SIZE,sendrecvflag,

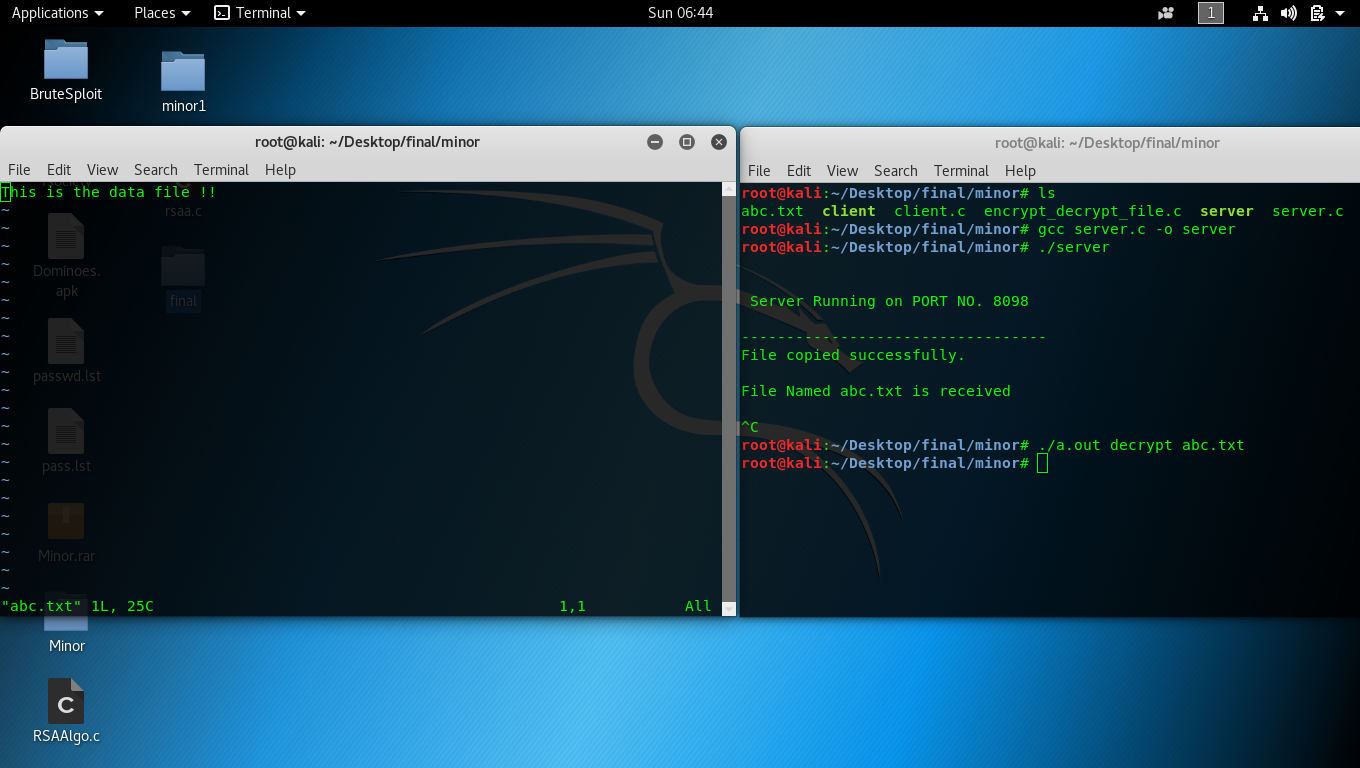
(struct sockaddr \*)&addr\_con, addrlen

**STEP5-**

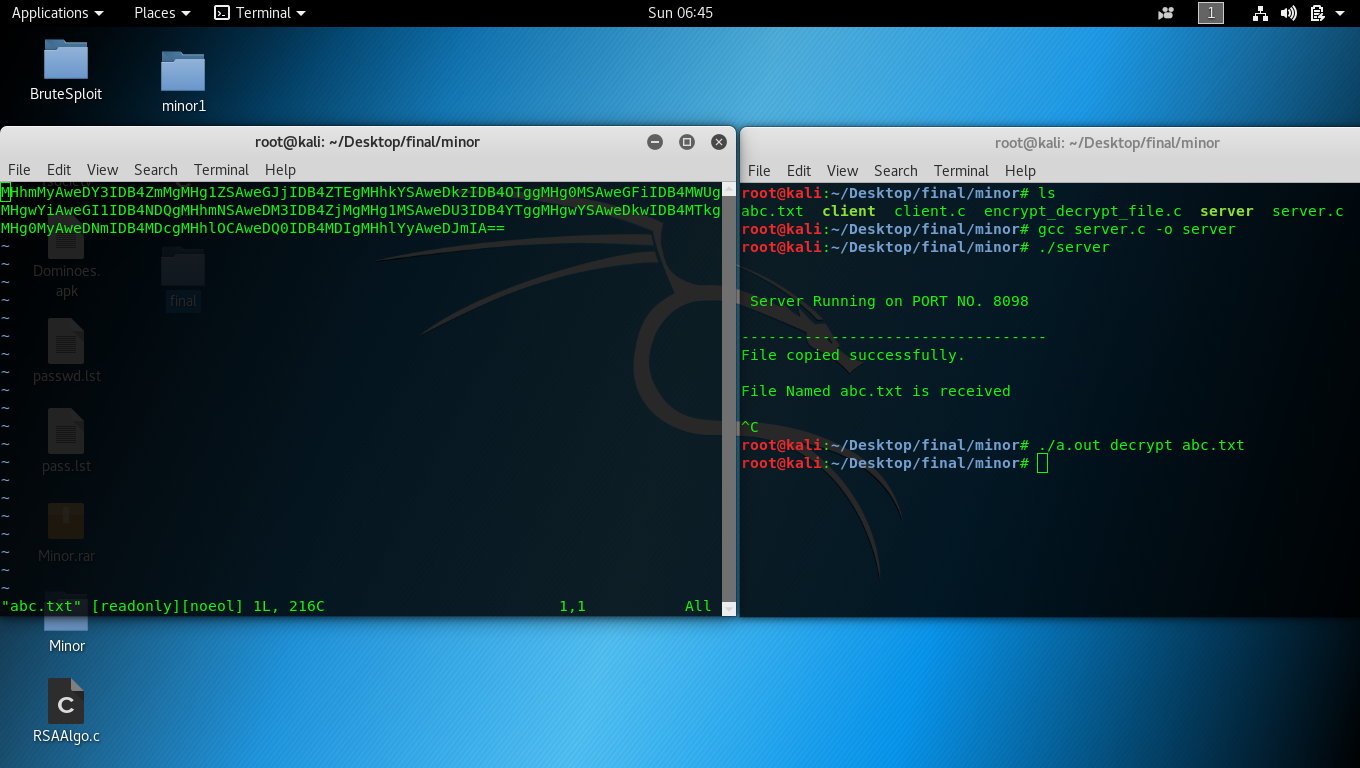
*if fp!=null then it gets close using fclose()*

**8. Output Screen**

**8.1 Before Encryption**

****

**8.2 After Encryption**

****

**8.3 After Decryption**

****

**9. Limitations and Future Enhancements**

**9.1 Limitations**

Tackling of the Error propagation of Advanced Encryption Standard (AES) is a great challenge. In literature, several studies have been made on this issue and several techniques are suggested to tackle the effect. Error propagation effect in case of selective AES and its comparison with normal AES has also been studied. Four algorithms, viz. SBM 1.1, SBM 1.2, SBM 1.3 and SBM 1.4, have been proposed for preventing error propagation effect of AES.

Apart from the error propagation effect, AES may suffer from different attacks made over the channel. Larger key lengths translate into an exponential increase in the complexity of an exhaustive search. Side-channel attacks, however, use a divide-and-conquer approach [104, 105] and hence it is generally assumed that increasing the key length cannot be used as mitigation.

Yet, the internal round structure of AES-128 and its key-scheduling seem to hinder a direct extension of the existing attacks thus challenge the proposition above. Indeed, two consecutives round keys are required to infer the secret key and the Mix Columns operation, not present in the last round, apparently increases the key search complexity from 28 to 232. Additionally, it is unclear what the impact of the different round structures is on the number of required measurements.

**9.2** **Future Enhancements**

In AES algorithm, the number of rounds involved in the encryption and decryption depends on the length of the key and the number of block columns.

This part of the application worked on the application of AES algorithm for the purpose of encryption. The path further in the project will see the implementation of RSA Algorithm for the encryption. After the successful deployment of RSA Algorithm, the project will see the comparison of both the key-based algorithm.

**10. Conclusion**

Applied the AES (256 bits) algorithm successfully.

**References**

1. Limi Kalita,”Socket Programming”,*(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4802-4807*.
2. Pranab Bandhu Nath & Md.Mofiz Uddin,”TCP-IP Model in Data Communication and Networking”,*American Journal of Engineering Research (AJER) e-ISSN: 2320-0847 p-ISSN : 2320-0936.*
3. Rushabh Balpande, Chetan Dusane, Khushboo Kashyap & Nandkumar Patil,”Client Server Based Secure Chat Application Using Peer-To-Peer Network Architecture” *International Journal of Emerging Trend in Engineering and Basic Sciences (IJEEBS) ISSN (Online) 2349-6967*
4. By Dr. Prerna Mahajan & Abhishek Sachdeva,”A Study of Encryption Algorithms AES, DES and RSA for Security”, *Global Journal of Computer Science and Technology Network, Web & Security Volume 13 Issue 15 Version 1.0 Year 2013 Online ISSN: 0975-4172 & Print ISSN: 0975-4350*