



UNIVERSITY WITH A PURPOSE



# Project Title

**SECURE FILE STORAGE IN CLOUD ENVIRONMENT USING CRYPTOGRAPHY TECHNIQUES**

Project Guide

**Ms. Avita Katal**  
Assistant Professor,  
School of Computer Science



# Team Members & Role

Mr. Ekanshu Dargan	:	Literature Review, Documentation, Implementation and Coding
Mr. Babanjot Singh	:	Literature Review, Documentation, Designing and Coding
Mr. Manik Khurana	:	Literature Review, Documentation, Coding, Integration and Testing

# Introduction

Cryptography's aim is to construct schemes or protocols that can still accomplish certain tasks even in the presence of an adversary.

- Safe communication between Alice and Bob.
- Providing privacy and authenticity remains a central goal of Cryptography Techniques.
- Implemented Client Server Communication (using Socket programming) and RSA and AES.

# Problem Statement

Security is compromised in the course of Data Communication between the client and the server

- All time network connectivity on cloud has increased the chances of ever getting hacked.
- User data can not be compromised.
- Consequences are devastating.
- Encryption becomes a crucial link to overcome this security issue.

We will be comparing the two famous algorithms of Cryptography: RSA and AES and highlighting the advantages and disadvantages of both.

# Motivation

- Inspired by the stale state of security, the developers have been trying and failing to securely encrypt information for ages.
- Most of their attempts were not grounded regarding the codes and the data they shared.
- In other words, the simple intuitive methods of encrypting information or data provides no guarantee of keeping it safe.
- That's why there was a need to introduce of the various algorithms of Encryption and Decryption.

# Objectives

Transfer of data between a cloud server and a client without any risk.

- Create a simple, effective and secure network.
- Prevention of loss caused by any unauthorized access.
- Maintain authenticity, confidentiality
- Compare different encryption algorithms and their working.

# Software/ Hardware Requirement

## Hardware:

- Intel(R) Core(TM) i3-3200 CPU @ 1.5 GHz
- 2 GB RAM
- System type is 32/64-bit Operating System

## Software:

- Linux based Operating System – Ubuntu
- GCC compiler



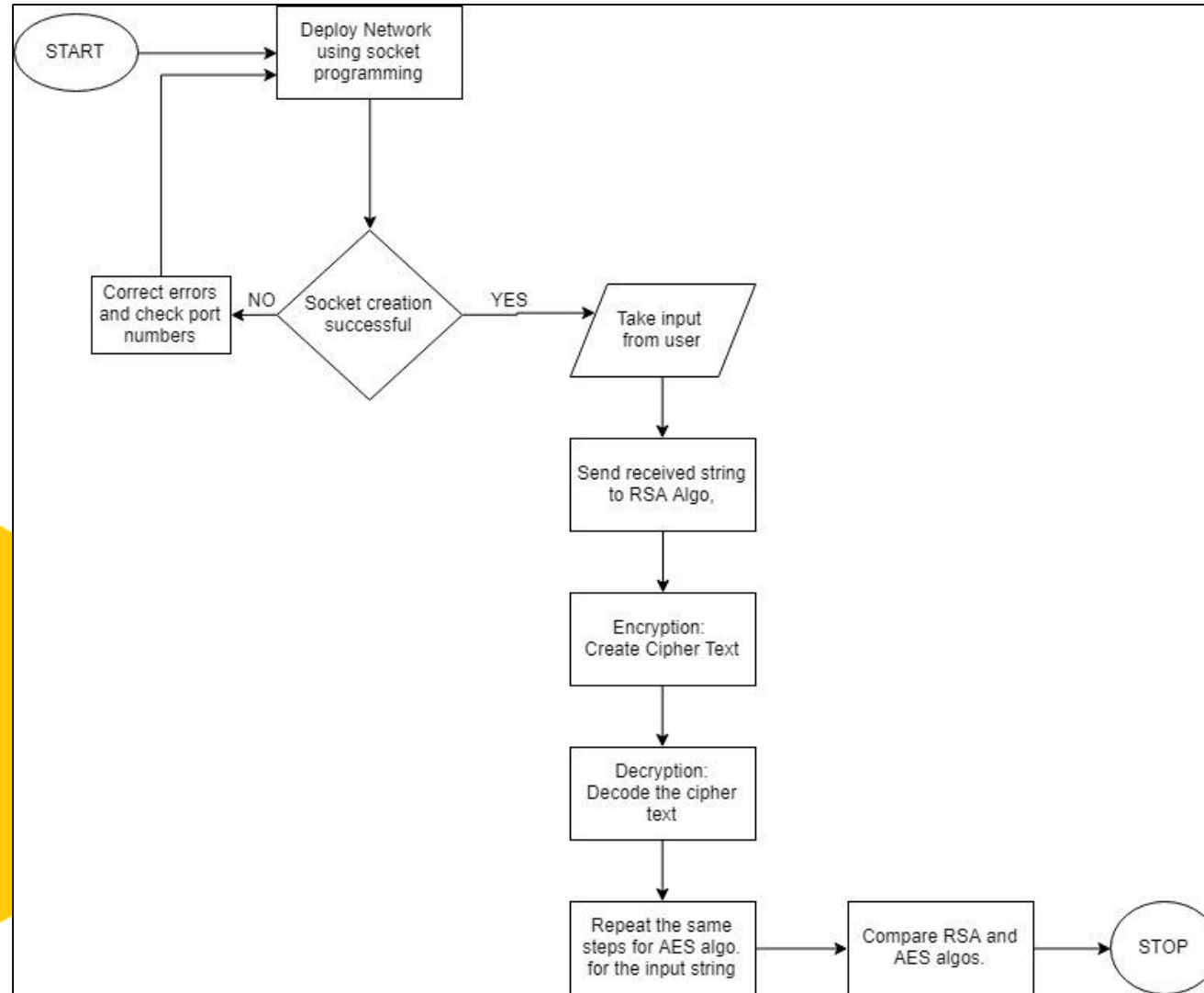
# Literature Review

- **Enabling cryptography to enable users to communicate securely over an insecure channel in a way that guarantees their transmissions' privacy and authenticity[1].**  
By- JEAN SÉBASTIEN, *University of Luxembourg*.
- **Comprehensive study and working of Socket Programming[2].**  
By- Socket Programming Tutorials(<http://www.binarytides.com/socket-programming-c-linux-tutorial>)
- **Implementing the Asymmetric RSA algorithm with two different keys[3].**  
By- Hellman, M. and J. Diffie, *IEEE Transactions on Information theory*
- **Implementing the Symmetric Advanced Encryption Standard (AES) with a key length of 128 bits using Verilog hardware description language (HDL)[7].**  
By- Abhinandan Aggarwal, Gagandeep singh, Prof. (Dr.) Neelam Sharma, *Maharaj Agarsen Institute Of India*

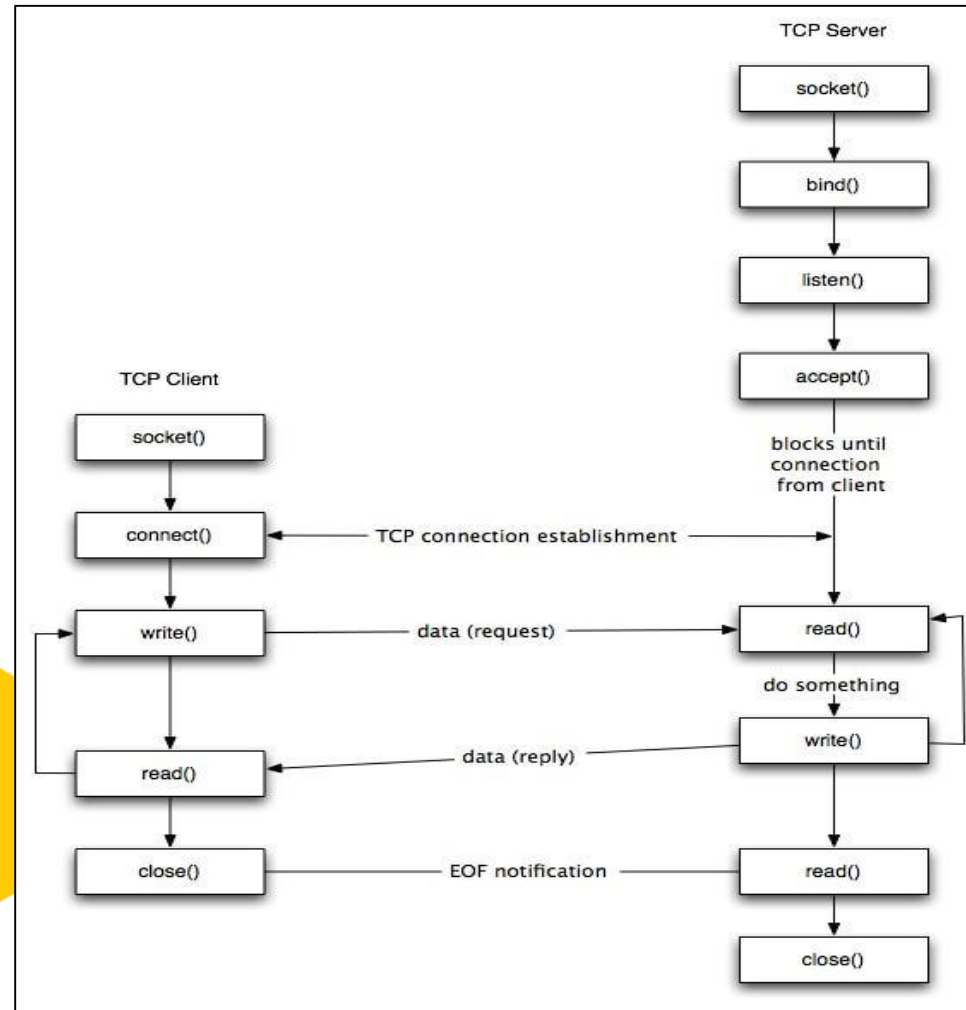
# Methodology

- **Module 1:** Deploying the network using socket programming.
- **Module 2:** Data Encryption Using RSA Algorithm.
- **Module 3:** Encryption using AES Algorithm.
- **Module 4:** Comparative study of both algorithms.

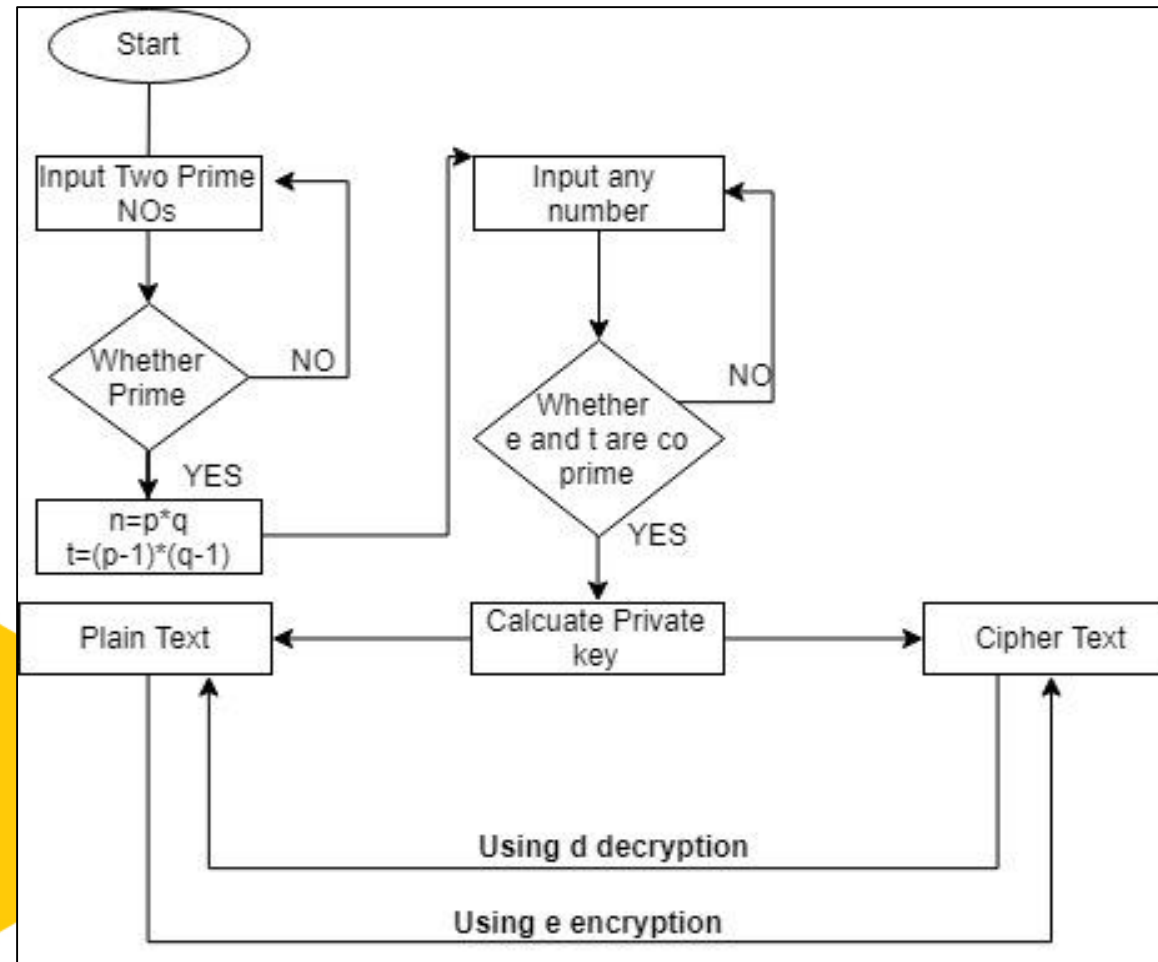
# Flowchart of the System



# Flowchart of Socket Programming Module

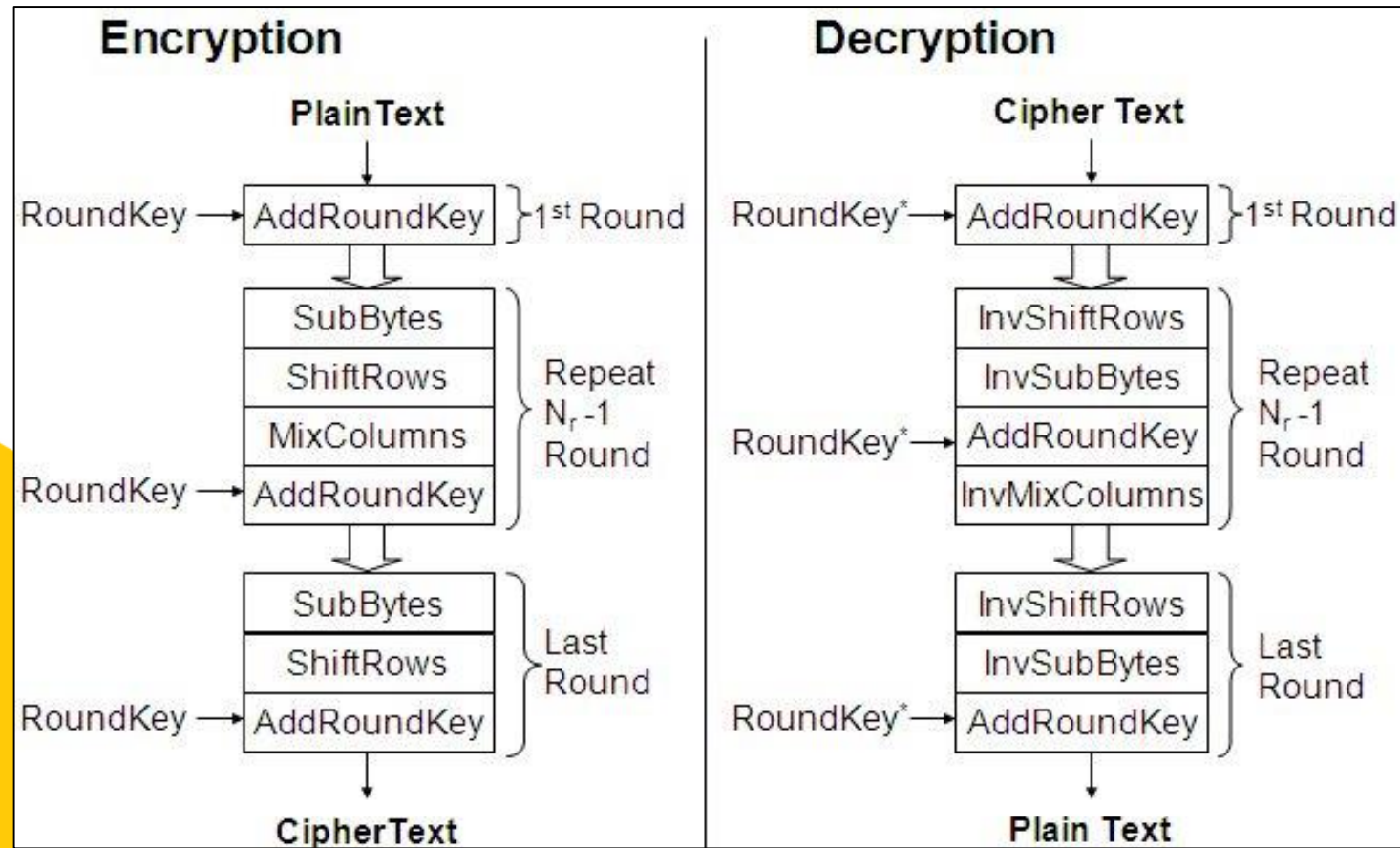


# Flowchart of RSA Algorithm





# Flowchart of AES Algorithm



# Algorithm for the System

Step 1: Start

Step 2: Deploy the network using Socket Programming

Step 3: If not successful, then check errors and correct the port number associated.

Step 4: If successful, create Socket(), Accept() data, Connect() to server, and Close() the socket after the completion.

Step 5: Take input from the user for both RSA and AES Algorithm

Step 6: Send the received input form the user

Step 7: Send input to the RSA Module Code in first iteration and to AES Module in the second Iteration

Step 8: Perform the Encryption in the RSA format, for the first iteration and in the AES format for the second iteration

Step 9: Show the Cipher Text to user

Step 10: Decrypt the Cipher Text to get the plain text

Step 11: Repeat Steps 6, 7, 8, 9 and 10 for the AES Module (the second Iteration)

Step 12: Compare the working of two Algorithms

Step 13: Show results to the user

Step 14: Display "Thank You"

Step 15: End

# Pseudo Code for RSA

1. Generate (Find) two Prime Numbers (P and Q)
2. Calculate  $N = P * Q$ .
3. Calculate  $M = \Phi(N) = (P-1)*(Q-1)$ .
4. Select any integer e, the rules to select E are:
  - a. E is positive integer
  - b.  $0 < E < M$
  - c.  $\text{GCD}(M, E) = 1$
5. Calculate D a use Extended Euclid Theorem
$$ed = 1 \bmod (p - 1)(q - 1)$$
6. At Encryption:
$$C = P^e \bmod n$$
7. At decryption:
$$C * d \bmod n$$

# Algorithm for AES

## Encryption:

- Derive the set of round keys from the cipher key.
- Initialize the state array with the block data (plaintext).
- Add the initial round key to the starting state array.
- Perform nine rounds of state manipulation.
- Perform the tenth and final round of state manipulation.
- Copy the final state array out as the encrypted data (ciphertext).

## Decryption:

The order of operation in decryption is:

1. Perform initial decryption round:
  - XorRoundKey
  - InvShiftRows
  - InvSubBytes
2. Perform nine full decryption rounds:
  - XorRoundKey
  - InvMixColumns
  - InvShiftRows
  - InvSubBytes
3. Perform final XorRoundKey

The same round keys are used in the same order.

# Algorithm for Socket Programming

## Steps to create a client using TCP/IP

- Create a socket using the `socket()` function in c.
- Initialize the socket address structure as per the server and connect the socket to the address of the server using the `connect()`;
- Receive and send the data using the `recv()` and `send()` functions.
- Close the connection by calling the `close()` function.

## Steps to create a server using TCP/IP

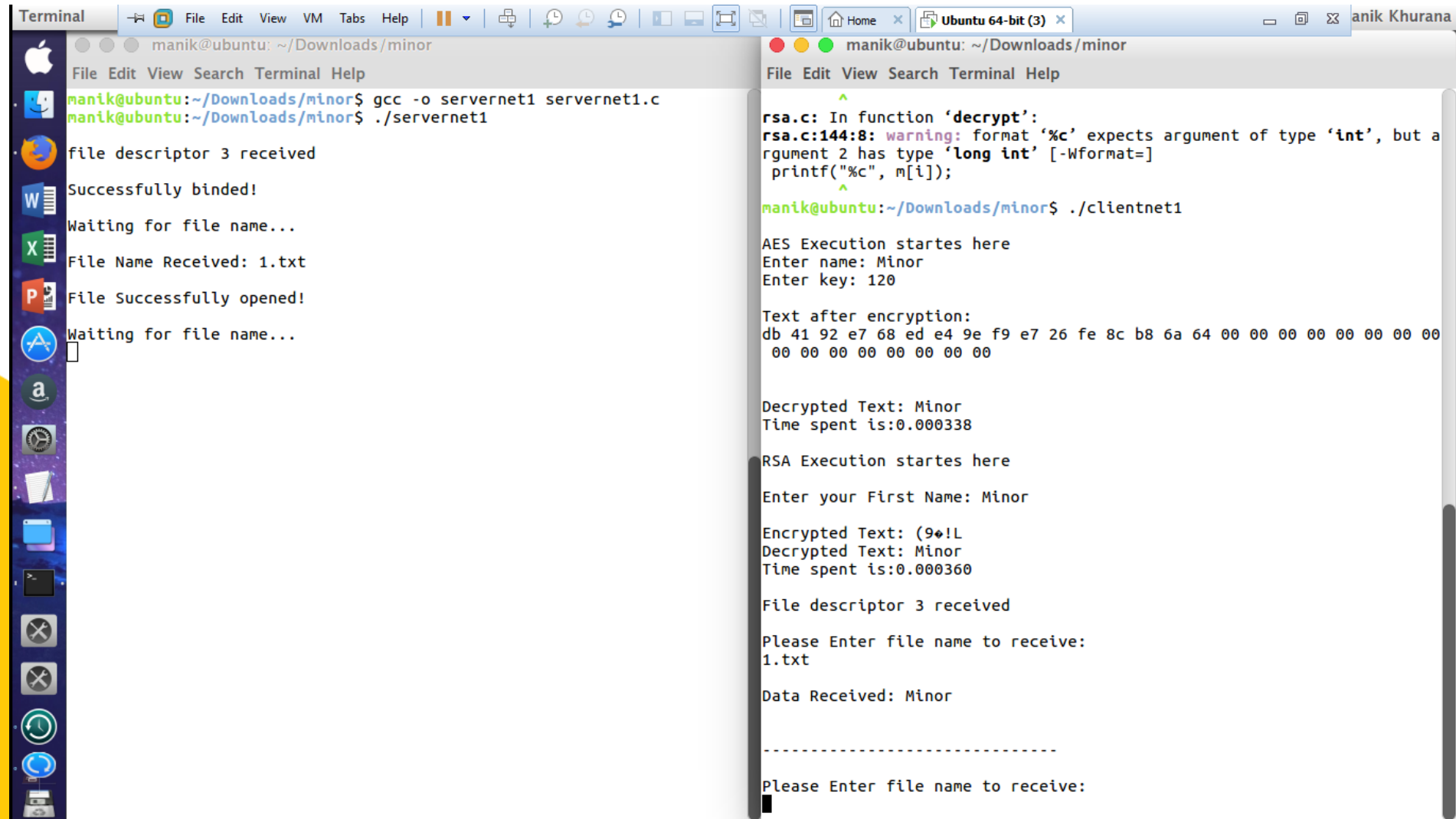
- Create a socket using the `socket()` function in c.
- Initialize the socket address structure and bind the socket to an address using the `bind()` function.
- Listen for connections with the `listen()` function.
- Accept a connection with the `accept()` function system call. This call typically blocks until a client connects to the server.
- Receive and send data by using the `recv()` and `send()` function in c.
- Close the connection by using the `close()` function.



# Final result of Comparison

FEATURE	RSA	AES
DEVELOPED	1977	2000
KEY LENGTH	MORE THAN 1024 bits	128,192,256 bits
CIPHER TYPE	ASYMMETRIC BLOCK CIPHER	SYMMETRIC BLOCK CIPHER
SECURITY	LESS SECURE	MUCH SECURE
ENCRYPTION/DECRYPTION SPEED	SLOWER	FASTER
HARDWARE AND SOFTWARE IMPLEMENTATION	NOT EFFICIENT	EFFICIENT IMPLEMENTATION

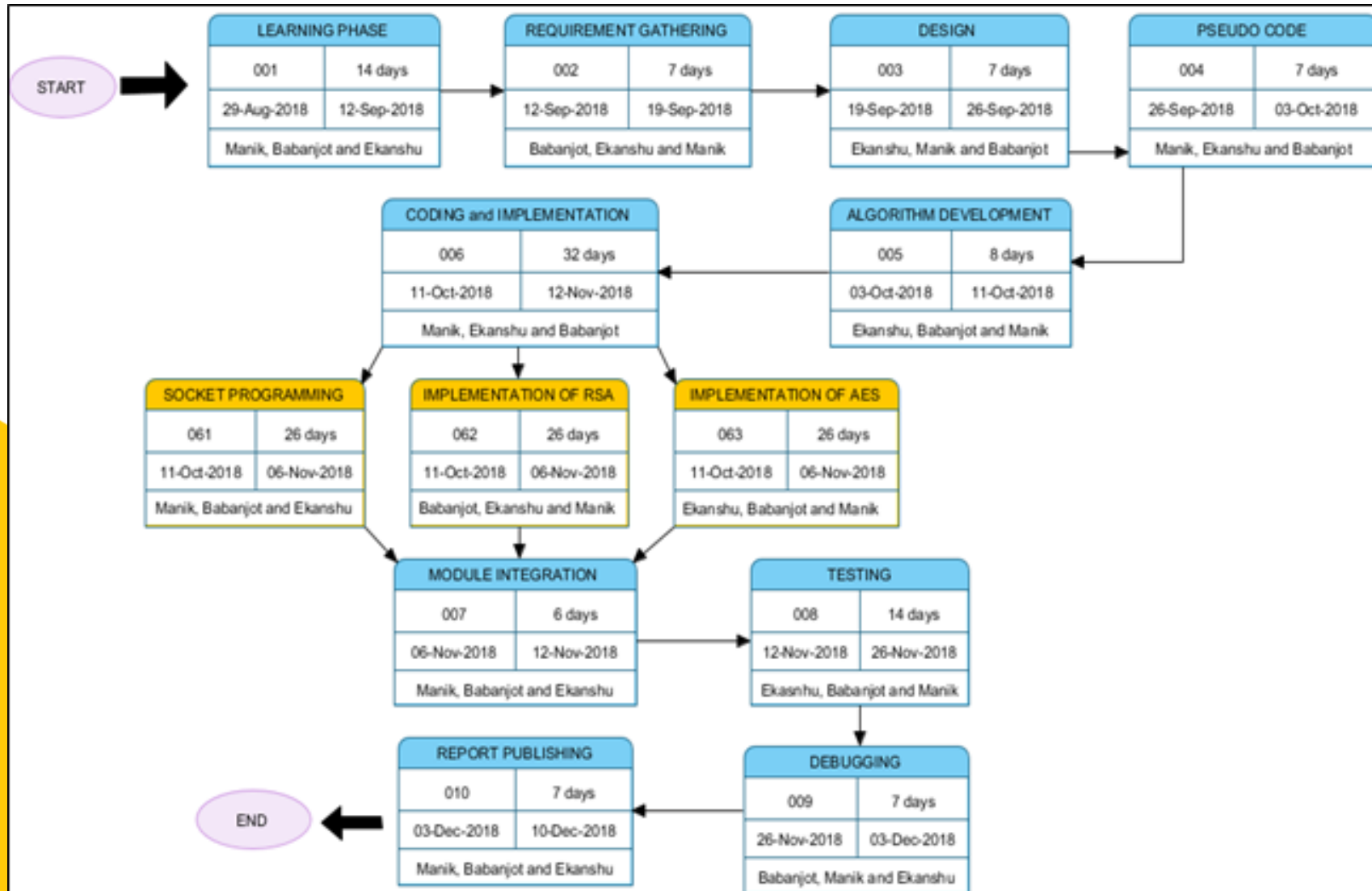
# Final result of Comparison: Screenshot



```
Terminal
manik@ubuntu: ~/Downloads/minor
File Edit View Search Terminal Help
manik@ubuntu:~/Downloads/minor$ gcc -o servernet1 servernet1.c
manik@ubuntu:~/Downloads/minor$ ./servernet1
file descriptor 3 received
Successfully binded!
Waiting for file name...
File Name Received: 1.txt
File Successfully opened!
Waiting for file name...
manik@ubuntu:~/Downloads/minor$

rsa.c: In function 'decrypt':
rsa.c:144:8: warning: format '%c' expects argument of type 'int', but a
rgument 2 has type 'long int' [-Wformat=]
printf("%c", m[i]);
manik@ubuntu:~/Downloads/minor$ ./clientnet1
AES Execution startes here
Enter name: Minor
Enter key: 120
Text after encryption:
db 41 92 e7 68 ed e4 9e f9 e7 26 fe 8c b8 6a 64 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
Decrypted Text: Minor
Time spent is:0.000338
RSA Execution startes here
Enter your First Name: Minor
Encrypted Text: (9*!L
Decrypted Text: Minor
Time spent is:0.000360
File descriptor 3 received
Please Enter file name to receive:
1.txt
Data Received: Minor
-----
Please Enter file name to receive:
```

# PERT Chart



# References

- [1] JEAN SÉBASTIEN CORON “Crypto Corner”, *University of Luxembourg*, 2006.
  - [2]. Socket Programming Tutorials (<http://www.binarytides.com/socket-programming-c-linux-tutorial>)
  - [3] Hellman, M. and J. Diffie, 1976. New Directions in Cryptography. *IEEE Transactions on Information theory*, vol. IT-22, pp:644-654.
  - [4]. Jamgekar, R. S., & Joshi, G. S. (2013), File Encryption and Decryption Using Secure RSA, *International Journal of Emerging Science and Engineering (IJESE)*, 1(4), 11–14.
  - [5] Fei Shao, Zinan Chang, (2010) *Second International Conference on Communication Software and Networks*.
  - [6]. M.Pitchaiah, Philemon Daniel, Praveen, Implementation of Advanced Encryption Standard Algorithm, *International Journal of Scientific & Engineering Research Volume 3, Issue 3, March -2012 1 ISSN 2229-5518*
  - [7] Abhinandan Aggarwal<sup>1</sup> , Gagandeep singh<sup>2</sup> , Prof. (Dr.) Neelam Sharma<sup>3</sup> *Department of electronics and communication, Maharaja Agrasen Institute of Technology, New delhi ,India.*
- Links:
- [a].[http://www.academia.edu/97169/Study\\_of\\_Data\\_Transmission\\_Using\\_Sockets](http://www.academia.edu/97169/Study_of_Data_Transmission_Using_Sockets).
  - [b]. <https://www.geeksforgeeks.com/socket-programming-in-c-using-tcpip/>
  - [c]. [https://www.tutorialspoint.com/cryptography/public\\_key\\_encryption.htm](https://www.tutorialspoint.com/cryptography/public_key_encryption.htm)
  - [d]. [https://www.tutorialspoint.com/cryptography/advanced\\_encryption\\_standard.htm](https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm)

THANK YOU

---



UNIVERSITY WITH A PURPOSE