ENCRYPTO CHAT

Authors: Eadan Plotnizky, Naama Scandarion, Carolina Campos

INTRODUCTION

ENCrypto Chat is a secure communication program designed for two users to exchange messages on an encrypted server. The program offers a variety of encryption schemes, including DES, Triple DES, RSA, and El Gamal, which users can choose from based on their preferences and needs.

- Libraries: infint, winsock
- Code Editor: VScode
- Version Control: Git/GitHub
- Programming Language: C++

PROBLEM STATEMENT & CONTRIBUTION

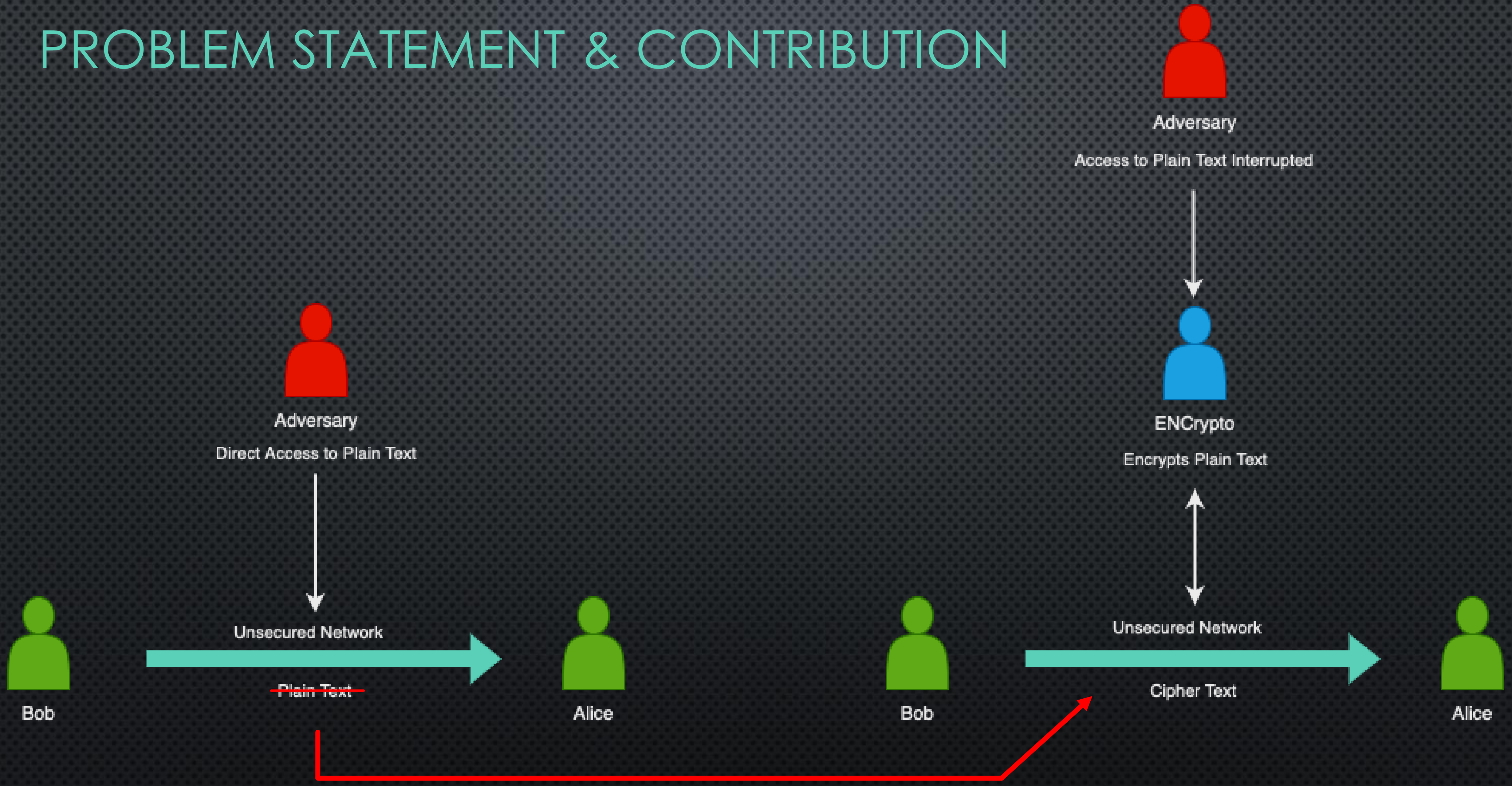
Problem

Unsecured messaging in socket communication is a significant security concern that could lead to sensitive information being intercepted and accessed by unauthorized parties. To prevent this, we created ENCrypto chat

Contribution

Solved by creating a cryptographic structure that will process communication between client and server to achieve a secured communication-ENCrypto

PROBLEM STATEMENT & CONTRIBUTION



Preliminary Material – Input/Output Files

Must Include Both

```
#include <iostream>
#include <fstream>
```

WRITING INTO A FILE

```
ofstream MyFile("filename.txt");

// Write to the file
MyFile << ”Hello";

// Close the file
MyFile.close();
```

READING OUT OF A FILE

```
ifstream MyFile("filename.txt");

while (getline (MyFile, myText)) {
// Output the text from the file
cout << myText;
}

// Close the file
MyFile.close()
```

PRELIMINARY MATERIAL

- Triple DES (3DES) is a more secure version of DES, using three different keys and performing three rounds of encryption to provide a higher level of security.

$$Cipher\ Text = E_{K3}(D_{K2}(E_{K1}(plaintext)))$$

$$Plain\ Text = D_{K1}(E_{K2}(D_{K3}(cipher\ text)))$$

- RSA is a widely-used asymmetric key encryption algorithm that uses a public key for encryption and a private key for decryption. RSA is considered secure and is commonly used in many modern communication systems for key exchange.

Two large prime numbers: p, q
Public Key: [n, e]
Private Key: d (mod phi multiplicative inverse of e)
m: original message translated to ASCII codes
R: random padding number
M (padded message using PCSK1):
0x00 | | 0x02 | | R | | 0x00 | | m
Cipher Text: M^E mod n
Plain Text: M^D mod n

PRELIMINARY MATERIAL

- DES (Data Encryption Standard) is a widely-used symmetric key encryption algorithm that utilizes a 56-bit key. While it has been historically popular, DES is no longer considered secure due to its small key size.

- El Gamal is another asymmetric key encryption algorithm that uses a similar approach to RSA but with a different mathematical foundation. El Gamal is often used in digital signature schemes and has also been used in some communication systems.

CLIENT/SERVER MODEL (IN A NUTSHELL)

THE CLIENT-SERVER MODEL DISTINGUISHES BETWEEN APPLICATIONS AS WELL AS DEVICES. NETWORK CLIENTS MAKE REQUESTS TO A SERVER BY SENDING MESSAGES, AND SERVERS RESPOND TO THEIR CLIENTS BY ACTING ON EACH REQUEST AND RETURNING RESULTS.

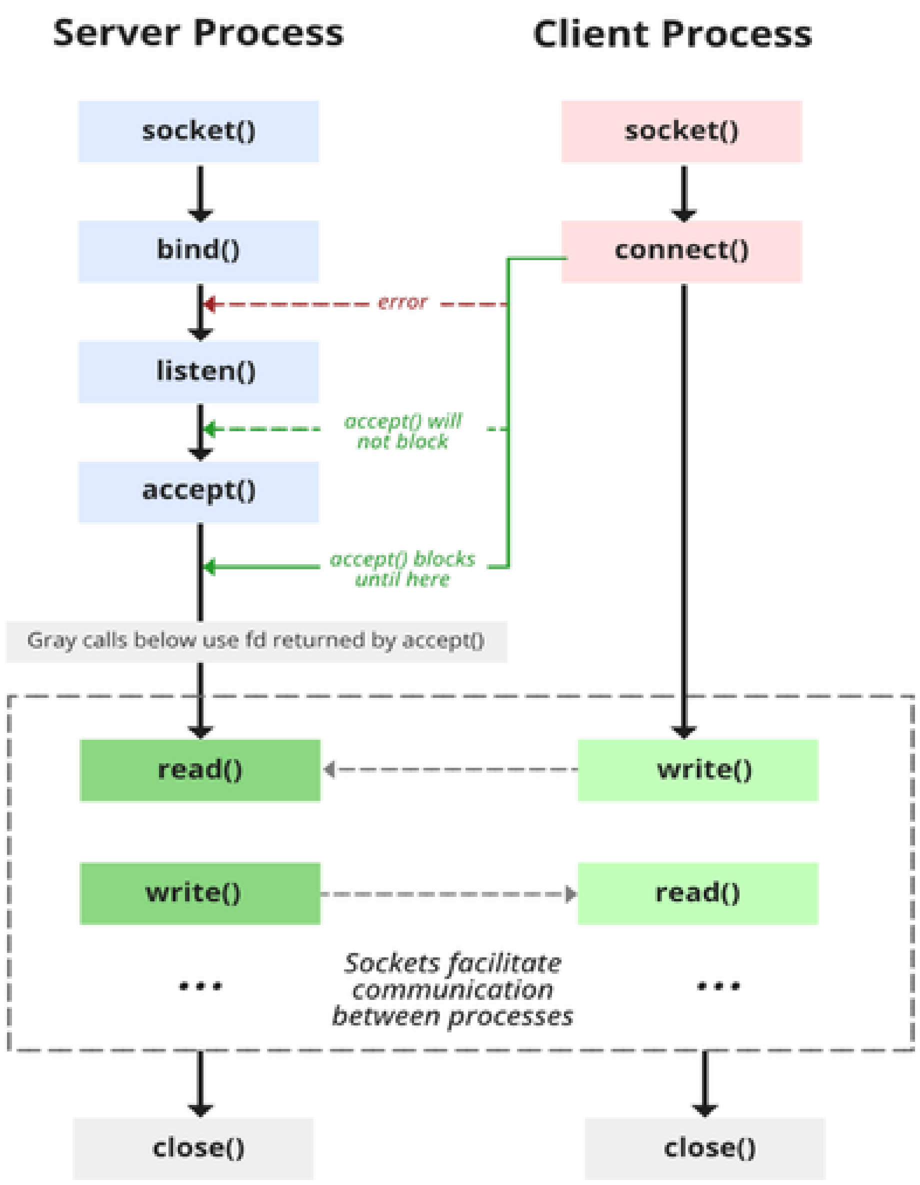
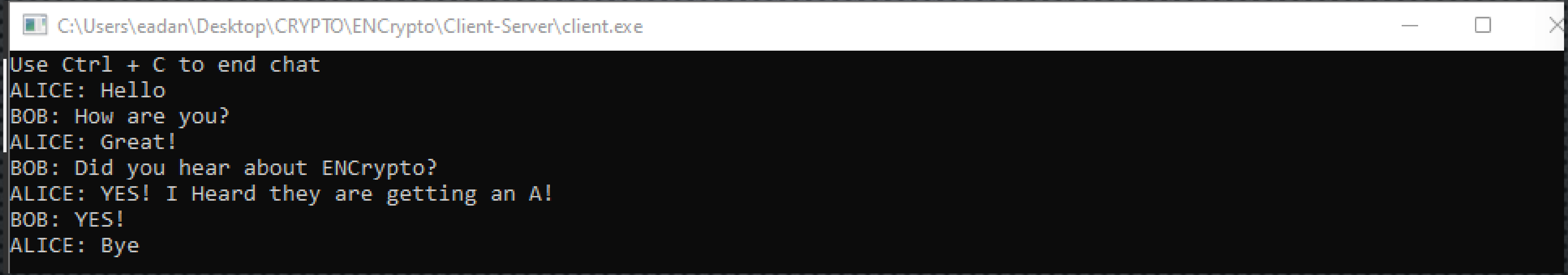


Diagram : https://www.geeksforgeeks.org/socket-programming-cc/

ENCRYPTO - FRONTEND

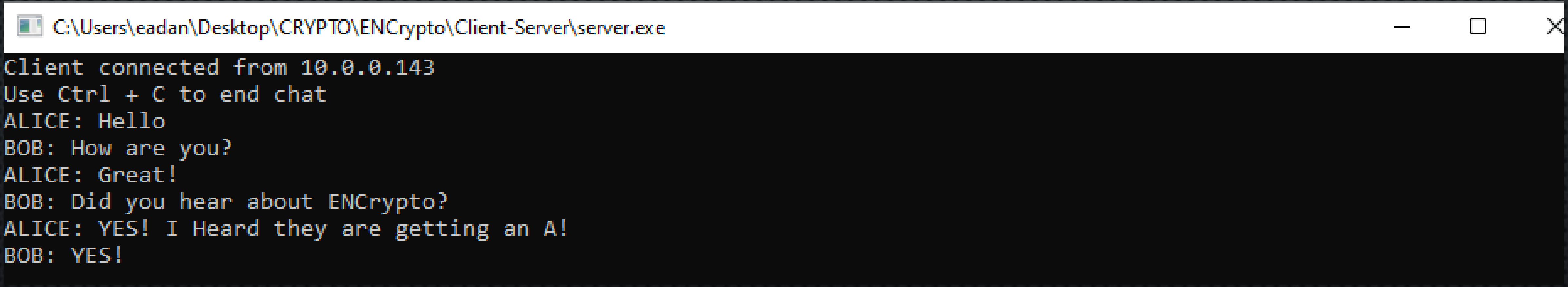
Client Side



C:\Users\eadan\Desktop\CRYPTO\ENCrypto\Client-Server\client.exe

```
Use Ctrl + C to end chat
ALICE: Hello
BOB: How are you?
ALICE: Great!
BOB: Did you hear about ENCrypto?
ALICE: YES! I Heard they are getting an A!
BOB: YES!
ALICE: Bye
```

Server Side

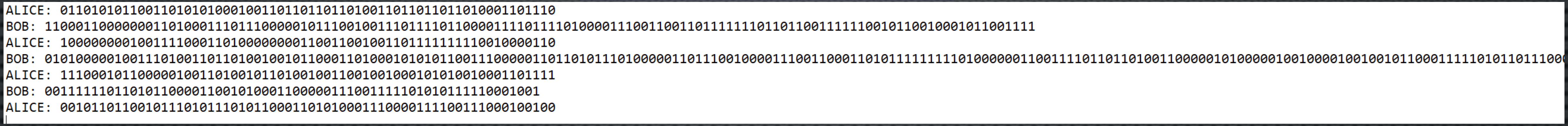


C:\Users\eadan\Desktop\CRYPTO\ENCrypto\Client-Server\server.exe

```
Client connected from 10.0.0.143
Use Ctrl + C to end chat
ALICE: Hello
BOB: How are you?
ALICE: Great!
BOB: Did you hear about ENCrypto?
ALICE: YES! I Heard they are getting an A!
BOB: YES!
```

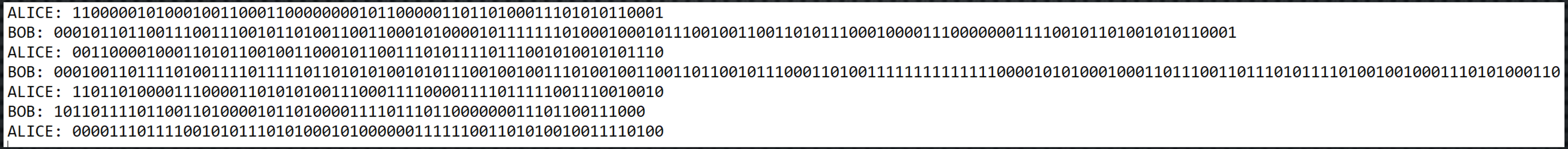
ENCRYPTO – BACKEND (LOG.TXT)

DES BACKEND



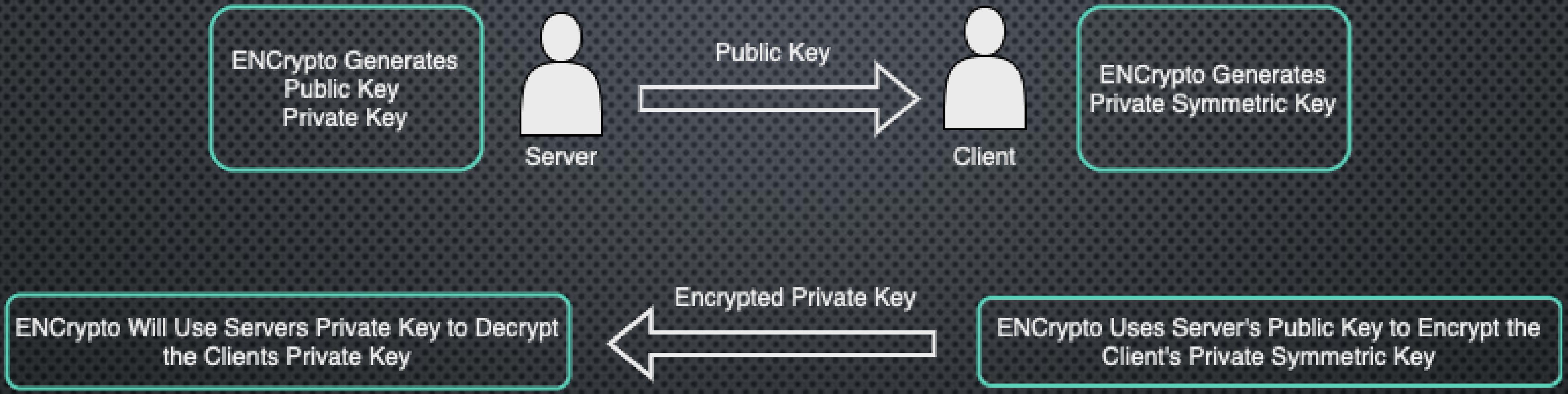
```
ALICE: 0110101011001101010100010011011011011010011011011011010001101110
BOB: 11000110000000110100011101110000010111001001110111101100001111011110100001110011001101111111011011001111110010110010001011001111
ALICE: 1000000001001111000110100000000110011001001101111111110010000110
BOB: 0101000000100111010011011010010010110001101000101010110011100000110110101110100000110111001000011100110001101011111111101000000110011110110110100110000010100000100100001001001011000111110101101110000
ALICE: 1110001011000001001101001011010010011001001000101010010001101111
BOB: 0011111101101011000011001010001100000111001111101010111110001001
ALICE: 0010110110010111010111010110001101010001110000111100111000100100
```

3DES BACKEND



```
ALICE: 1100000101000100110001100000000101100000110110100011101010110001
BOB: 000101101100111001110010110100110011000101000010111111101000100010111001001100110101110001000011100000001111001011010010101100 01
ALICE: 0011000010001101011001001100010110011101011110111001010010101110
BOB: 00010011011110100111101111110110101010010101110010010011101001001100110110010111000110100111111111111111000010101000100011011100110111010111101001001000111010100011 0
ALICE: 1101101000011100001101010100111000111100001111011111001110010010
BOB: 1011011110110011010000101101000011110111011000000011101100111000
ALICE: 0000111011110010101110101000101000000111111001101010010011110100
```

ENCRYPTO



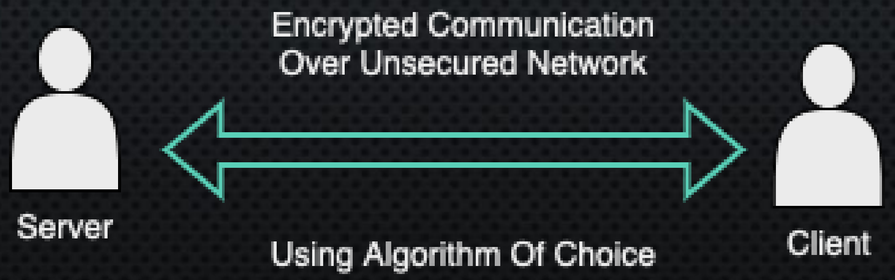
ENCrypto Generates
Public Key
Private Key

ENCrypto Generates
Private Symmetric Key

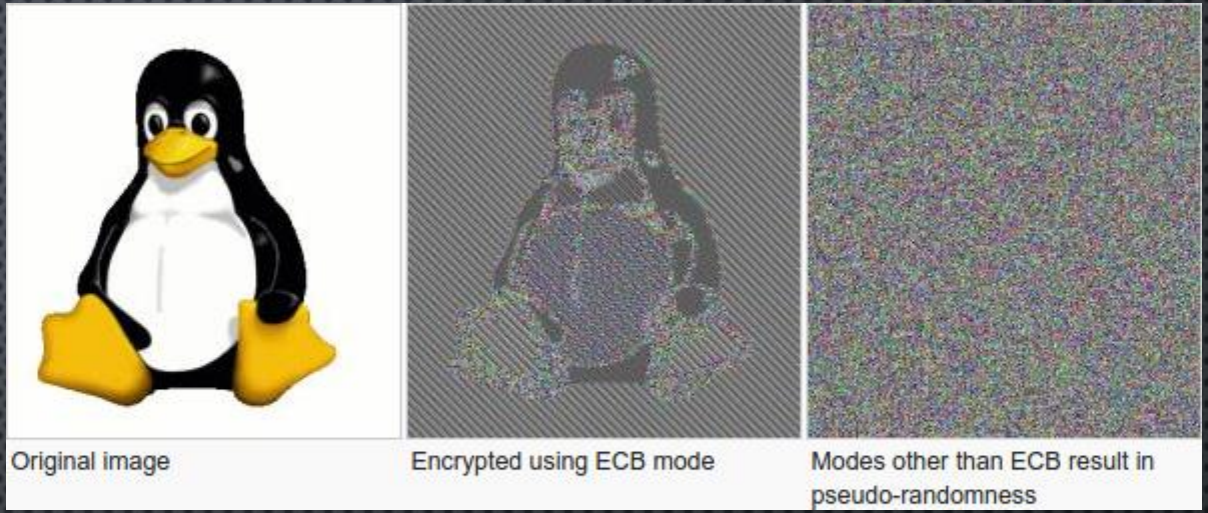
ENCrypto Will Use Servers Private Key to Decrypt the Clients Private Key

ENCrypto Uses Server's Public Key to Encrypt the Client's Private Symmetric Key

Result



CONCLUDING REMARKS



Original image

Encrypted using ECB mode

Modes other than ECB result in pseudo-randomness

- Our project is awesome
- Future modifications:
  - GUI
  - Port Forwarding – use app on many networks at once
  - Add more cryptographic schemes
  - Add more mode of operations options (current mode ECB)
  - Maybe: OAEP Padding Scheme (RSA)