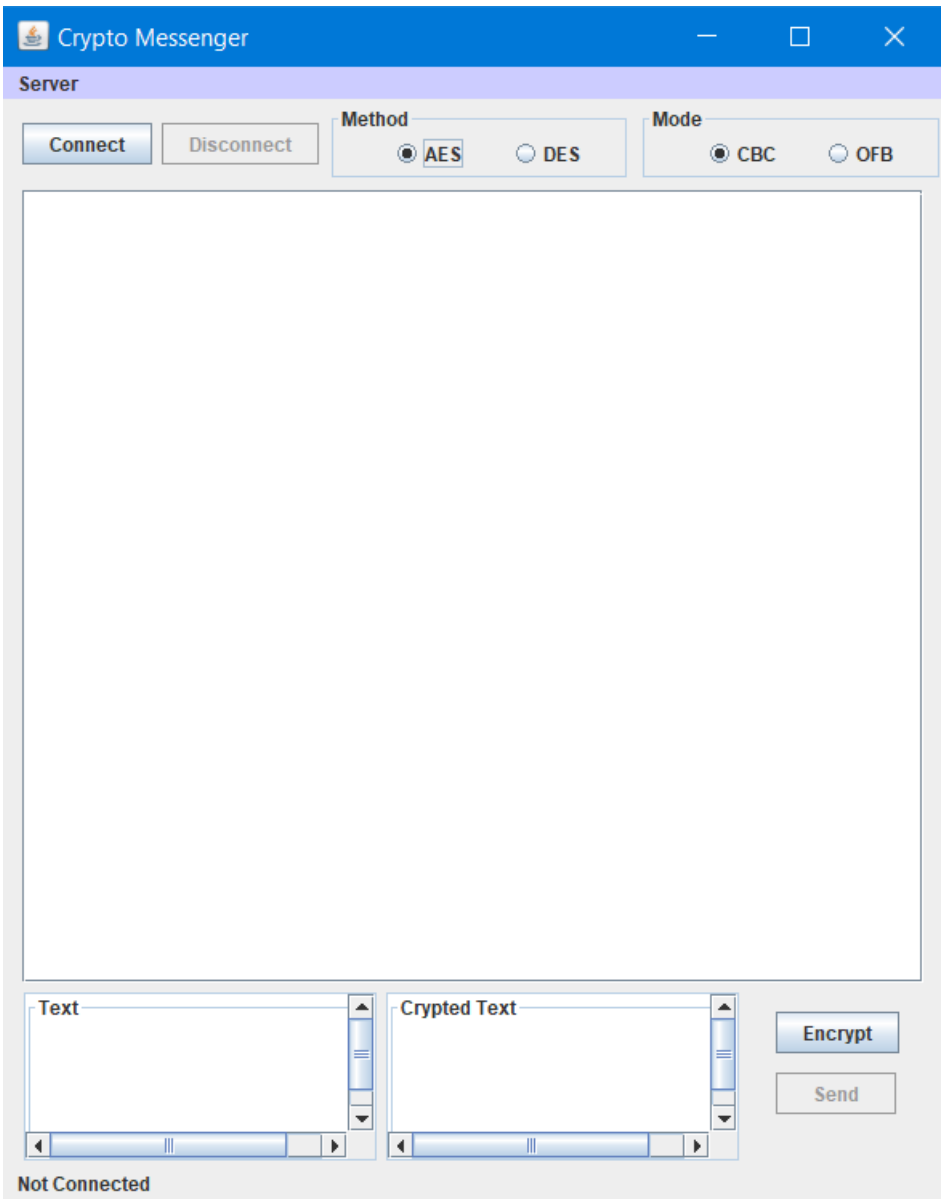
BBM 465 INFORMATION SECURITY LAB

ASSIGNMENT I

Emre DOĞAN 21426864 Akif ÇAVDAR 21280107

1.INTERFACE



There are 4 buttons :

Connect : Starts a connection between the client and the server.

Disconnect : Stops the connection between the client and the server.

Encrypt : Encrypts the text in the Text area according to method and mod parameters. Writes the cipher text to the Crypted Text area.

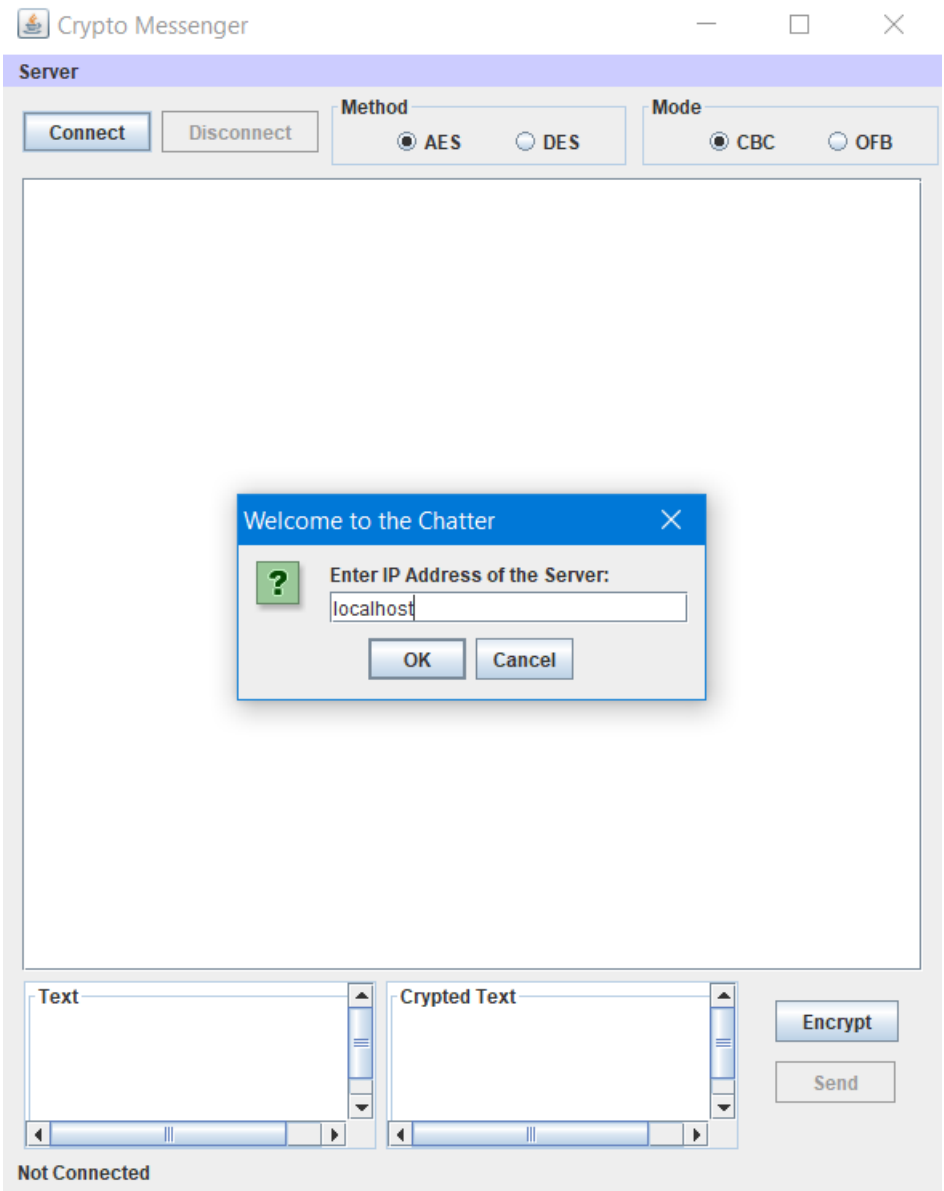
Send : Sends the cipher text to the server and clears the text areas.

There are 2 radio button sections :

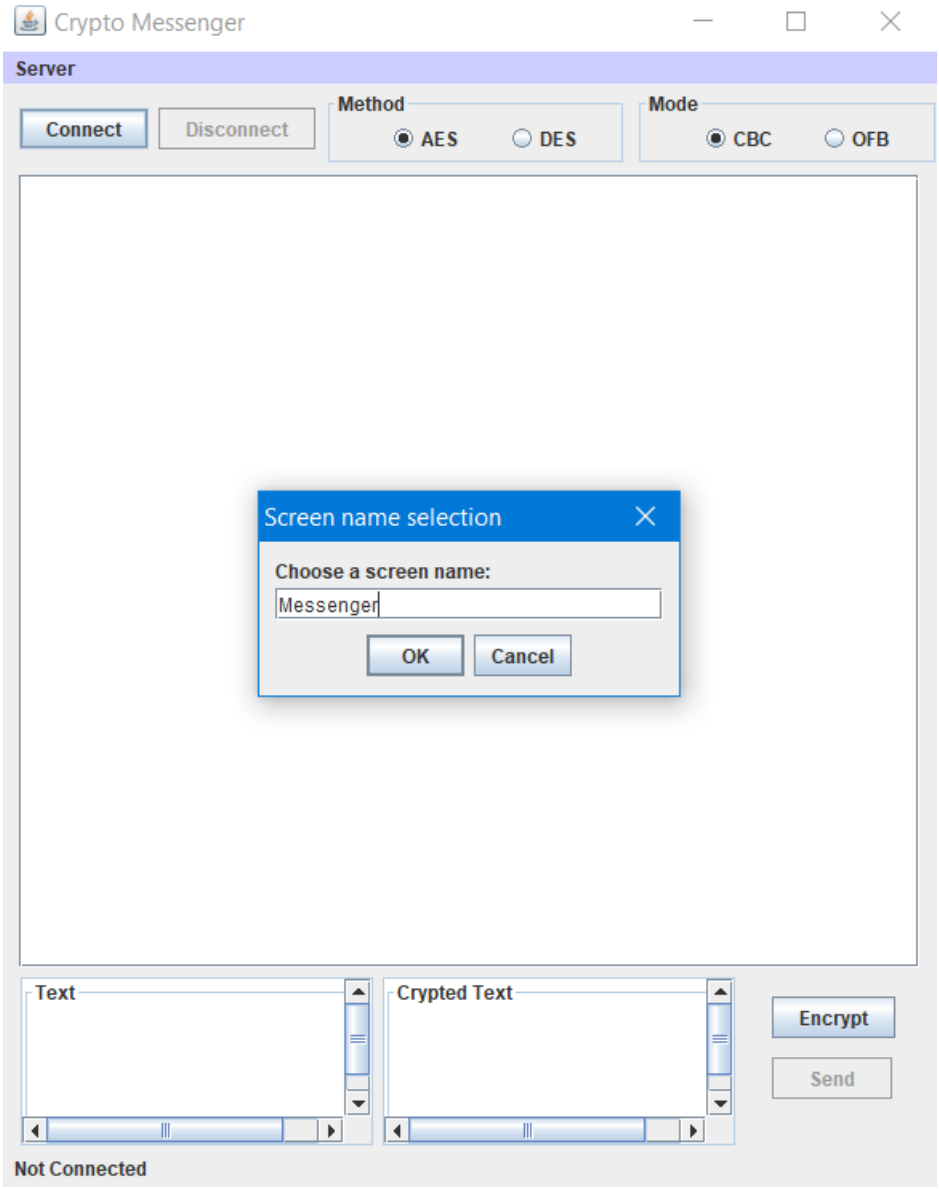
Method : Determines the method; AES or DES.

Mod : Determines the mode; CBC or OFB.

2.USAGE

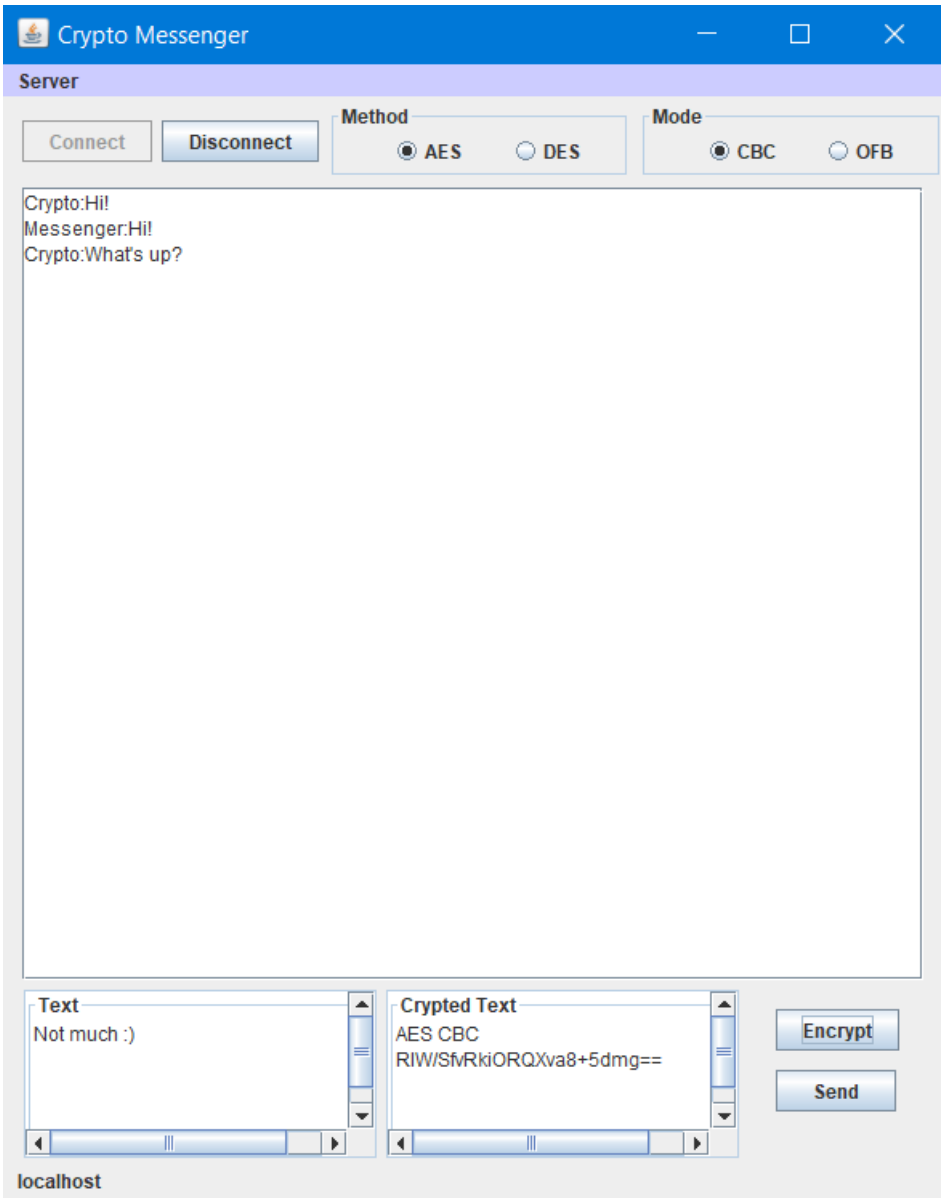


Click the connect button and enter the IP address of the Server.



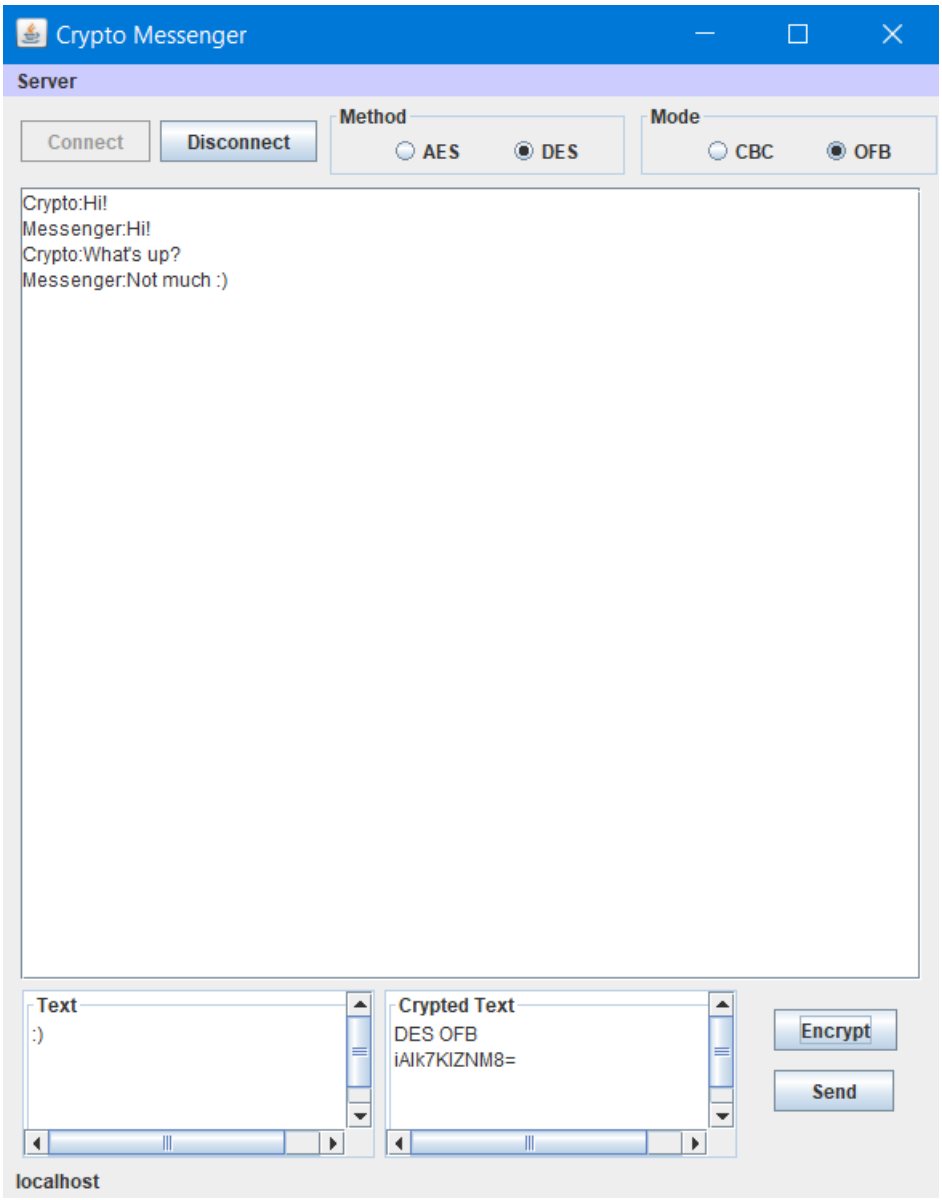
Then enter the name.

Than, if the connection is able to start, the “Not Connected” text turns into the IP address.



To send a crypted message, type the message to the Text area and click the Encrypt button.

It generates a cipher text. Than click to Send button.



To change the method and mode, click the radio buttons.

3.IMPLEMENTATION DETAILS

There are 3 classes in this project. The Server class will be run on the server side and the Client class will be run on the client side. The Gui class includes the user interface of the client process and it will be run on the client side with the Client class.

```
byte[] ivAES = { 0, 0, 0, 1, 3, 0, 8, 0, 0, 4, 0, 0, 0, 1, 0, 0 };
byte[] ivDES = { 0, 1, 0, 3, 0, 0, 6, 0};

byte[] key = ("asdasbaskasdasdaga").getBytes("UTF-8");
MessageDigest sha = MessageDigest.getInstance("SHA-1");

key = sha.digest(key);
```

```
aesKey= new SecretKeySpec(Arrays.copyOf(key,  16), "AES");
desKey= new SecretKeySpec(Arrays.copyOf(key,  8), "DES");
ivspecAES = new IvParameterSpec(ivAES);
ivspecDES = new IvParameterSpec(ivDES);
```

We use the static keys for encryption and decryption. So it's not required to send the keys to the server.

```
cipher = Cipher.getInstance(method+ "/" + mode + "/PKCS5Padding");
```

We use the PKCS5Padding in addition to the method and mode parameters to for encrypt and decrypt the texts.

```
t = new Thread(new Runnable() {

	public void run() {

		try {
			while (true) {

				String line = in.readLine();
				System.out.println(line);
				if (line.startsWith("SUBMITNAME")) {
				out.println(name);
				} else if (line.startsWith("NAMEACCEPTED")) {
				gui.jLabel2.setText("    " + serverAddress);
				} else if (line.startsWith("MESSAGE")) {
					String[] mes1 = line.split(" ");
					System.out.println(mes1[2]);
					String mes = Decrypt(mes1[2]);
					gui.jTextArea1.append(mes1[1] + mes + '\n');
				}
			}

		}catch(IOException e) {
			e.printStackTrace();
		}
	}
});

t.start();
```

We use a thread that listens to the server for incoming messages to prevent to the lock on the interface.