CRYPTOGRAPHY GROUP ASSIGNMENT

****

GROUP MEMBERS

1 AMEN BEFEKADU 117/11

2 MINTESNOT BELETE 138/11

3 HAVANA DANIEL 133/11

4 AHMED IBRAHIM 113/11

SUBMITTED TO : SOLOMON

**DEFENCE UNIVERSITY COLLEGE**

**OF ENGINEERING**

**How do we Run the program?**

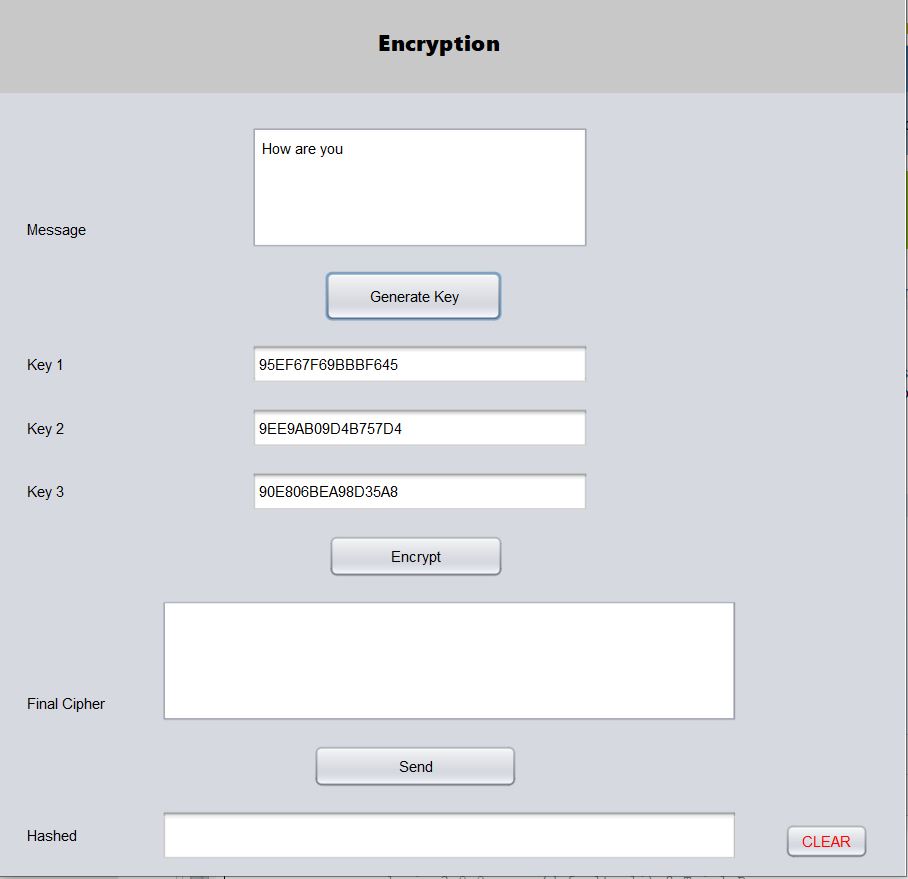
First we run the encrypt.java class then we run the decrypt.java class the next step after running the two classes is entering the message that we want to send and press the “Generate Key” button. Here the entered message will go to in the String message from the message area like (message=msg\_area.getText();). The key is generated from keyLookupTable.java class there are a list of keys there and it will randomly select 3 keys per “Generate Key” button pressed.

So we enter the message “How are you”

And the keys key1 95EF67F69BBBF645

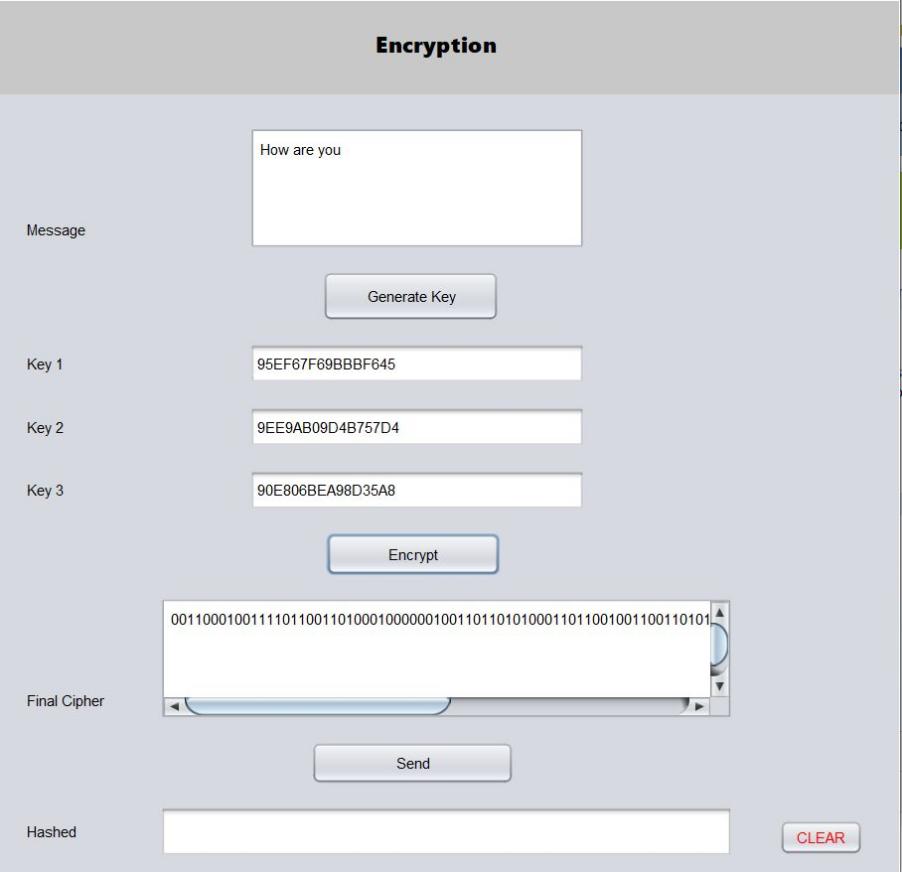
Key2 9EE9AB09D4B757D4

Key3 90E806BEA98D35A8



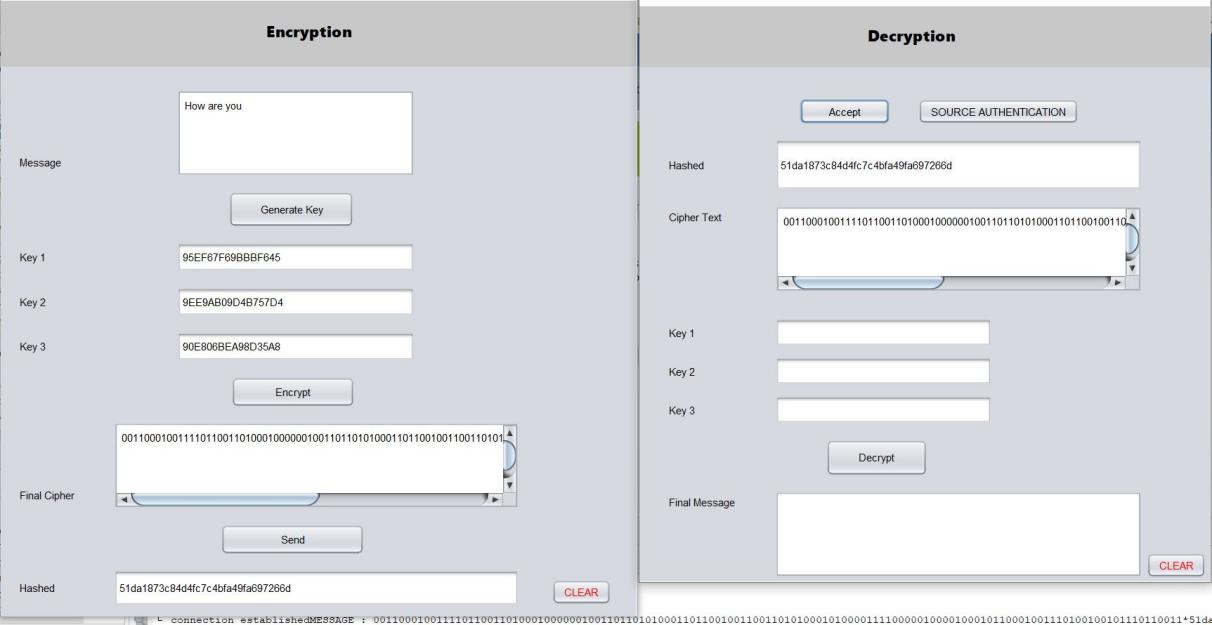
After we generate the key the next step is the encryption process as we know in triple des Encryption works 1st encrypt the message by using the 1st key then it will decrypt using the 2nd key lastly it will encrypt it using the 3rd key so the 1st encryption process works in a step that by converting the message to binary using the class “converttobinary.java” then the converted message’s number of block will be calculated using the class “no\_of\_blockcalculator.java” and the process of key generation, the selection of the key and converting the generated key to binary is done in the class “roundkeygeneration.java” then the message will initial permuted using the class “plaintxtintialpermutation.java” the division of two 32 bits is done in the class “plaintxt32division.java” After initial permutation the 16 rounds of encryption will begin in the rounds there are 4 main things PERMUTATION EXPANISTION, xor, SENDING TO 8 SBOXES , SEND TO PERMUTATION 32 and XOR LEFT AND PASSED RIGHT THROUGH FUNCTION then after we the encrypted text in binary form then it will change it to text by using the class “binarrytostrin.java class”

Then after finishing all the 3 processes it will get the final cipher text



Then we get Final Cipher 00110001001111011001101000100000010011011010100011011001001100110101000101000011110000010000100010110001001110100100101110110011

Then we use the concept of hashing for Authentication so when we press the “Accept” button in the “decrypt.java” class it will wait for the “encrypt.java” class to send the ciphered message for it so when we press the “send” button in the “encrypt.java” class it will accept the hashed message and the the ciphered message



So the as we can see the hashed message and the ciphered message are the same as that we can found in the encryption class

After that we have to use the same keys as we use in the encryption process so we copied the keys from the encrypt.java class in real life we can use kdc for distribution of keys

Here we get all things that we want to decrypt the message so we pressed the “Decrypt” button

The decryption process works as 1st it will decrypt the ciphered message by using the 3rd key then it will encrypt it using the 2nd key finally it will decrypt it using the 1st key the encryption and the decryption process is the same as we found the encrypt.java class but the difference is the using of the keys

Finally we get the message “How are you” that sent from the encryption class.

