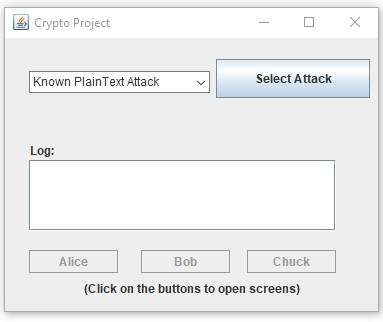
Author: Himesh Buch

4/25/2020

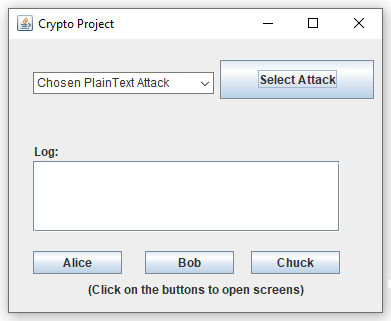
This document explains how to run and the program:

1. Run “ServerThread.java” to start the socket server which is used for sending and receiving messages
2. Once the server is running in the back, run “attack\_mode\_chooser.java” to run the program. Here is what to expect,

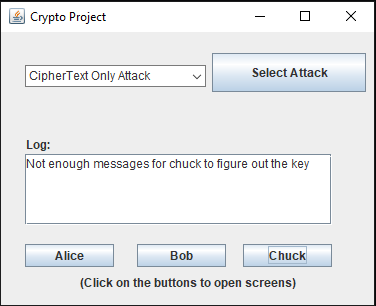
Choose the attack mode from here



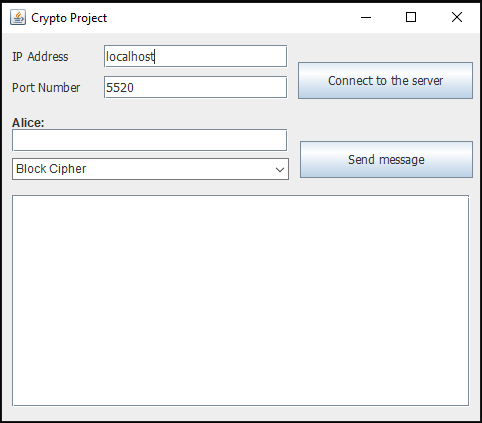
As you can see, you have a dropdown menu with what attacks you can choose. There is also a text area which will log some information there. Clicking on the button below will open up required user interface for either Alice, Bob, or Chuck. At the moment they are grayed out ,but once you click on the “Select Attack” button all the three buttons will be enabled. Let us select an attack,



Here we selected “Chosen PlainText Attack”, and as expected, all the buttons are enabled. If you click on the button “Chuck” now, it is of no use and you will see an error message. No conversation has happened between Alice and Bob, and thus, the error message will appear. Also, please make sure to send at least three messages before trying to decrypt them.



1. Clicking on “Alice” will open this interface for Alice to choose a cipher and send messages to Bob.

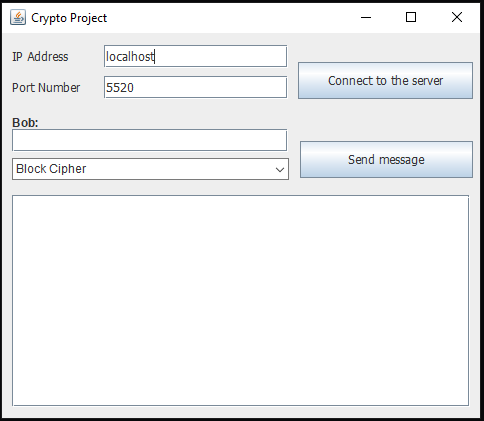


All the sent messages will appear here

Click on this button to connect to the server

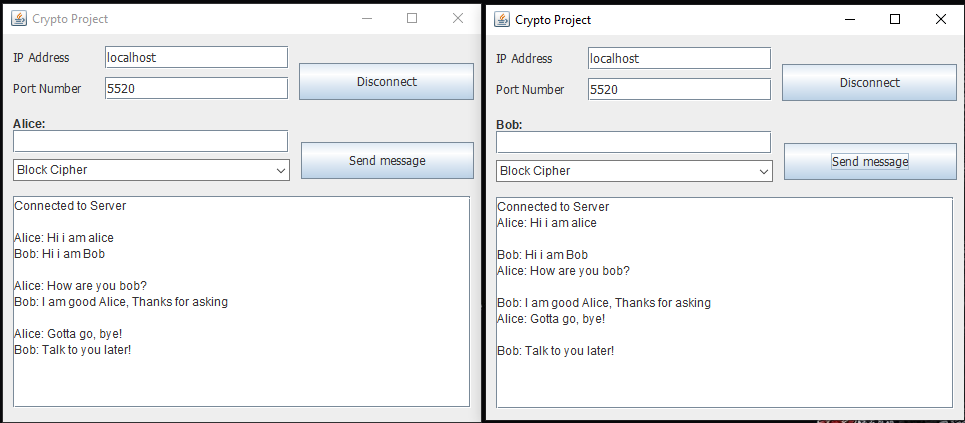
This dropdown shows what ciphers they can use

The first step is to click on “Connect to Server” button to connect to the server. If you do not press the button and start sending the messages, you will get an error “You are not connected”, written in the text area. Once connected to the server, select the cipher of your wish and start sending the messages. Note that the same rules apply to Bob. Here is what Bob’s user interface looks like. It is very identical with Alice’s UI



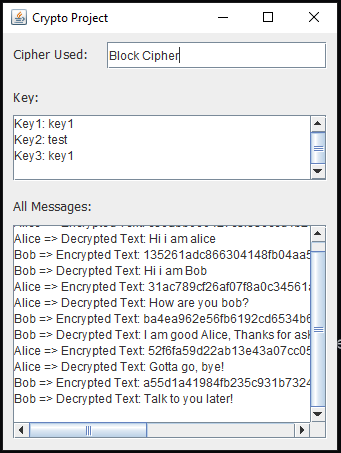
One thing to keep in mind is that, you are required to select the same cipher for both the user. You cannot select, say, Block Cipher for Alice and Stream Cipher for Bob. That will end with failure to run and the results will not be as expected as well.

1. Here is what sending messages look like. This is a side by side view, for both the users, and as you can see, this is an example conversation that they had,



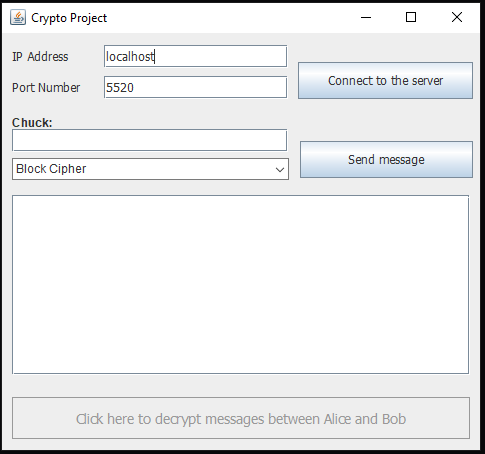
1. Depending on the attack mode, Bob’s UI will look something like this

* For Known PlainText Attack:

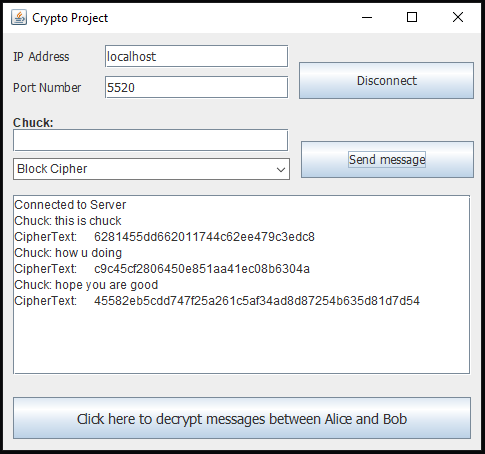


As you can see here, chuck has decrypted all the messages and figured out the key. It also shows all the encrypted and decrypted messages.

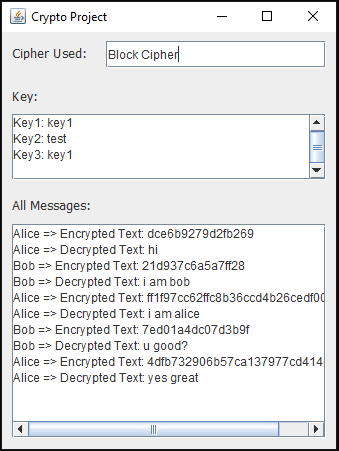
* For Chosen PlainText Attack/Chosen CipherText Attack/CipherText only Attack:



As seen in this UI, chuck has to query some messages to the oracle in order to figure out the key. The “Click here to decrypt messages between Alice and Bob” button is grayed out right now because chuck has not queried any messages to the server and does not know what the keys are. Once the server/program thinks that enough messages are queried, the button will automatically be enabled and messages then, will be able to be decrypted. Again, note that the cipher Chuck chooses is also the same as Alice and Bob (one of the requirements of the project was that the cipher used to encrypt the messages will be known to the adversary Chuck). Query at least, 3 messages to the server to efficiently decrypt the messages and figure out the key.



Once the button is enabled, meaning the decryption is done, clicking on it will show you something like this, which contains all decrypted messages between Alice and Bob, the cipher that was used, and the keys



1. This entire example was done using Block Cipher and Chosen PlainText Attack mode. Feel free to try using various cipher and attack modes
2. Please make sure you follow all the required constraints. Here is a quick review,

* Run Server.java
* Run attack\_mode\_chooser.java
* Select an attack
* Click on Alice and then click on Chuck
* Make sure they are both connected to the server
* Start sending messages
* Send enough messages and click on Chuck to decrypt the messages and see the keys