**Lab.4**

**Caesar Cipher**

# Overview

Implement a simple Caesar cipher with your lab.1 code. Section 1 goes over the Caesar Cipher algorithm, an example of the cipher, and two screen captures of the cipher being half implemented in lab.1 code. Section 2 has the questions for this lab.

# Section 1: Caesar Cipher

In a Caesar cipher, encoding is done by shifting the letter by n spaces down the alphabet. For example, “a” can be shifted to “b” by n =1. Figure 1 shows an example of shift n=13, and encodes hello using this shift. Once the message is received at the destination, the decoding is done simply by reversing the encoding spaces. In the case of figure 1, the receiver receives “uryyb”, which then receiver can decode by shifting 13 letters backwards on the alphabet for each letter.

Figure 2 shows the server receiving encoded message from the client. The server has not implemented code to decode using the key.

Figure 3 shows Wireshark capturing encoded message from the client.

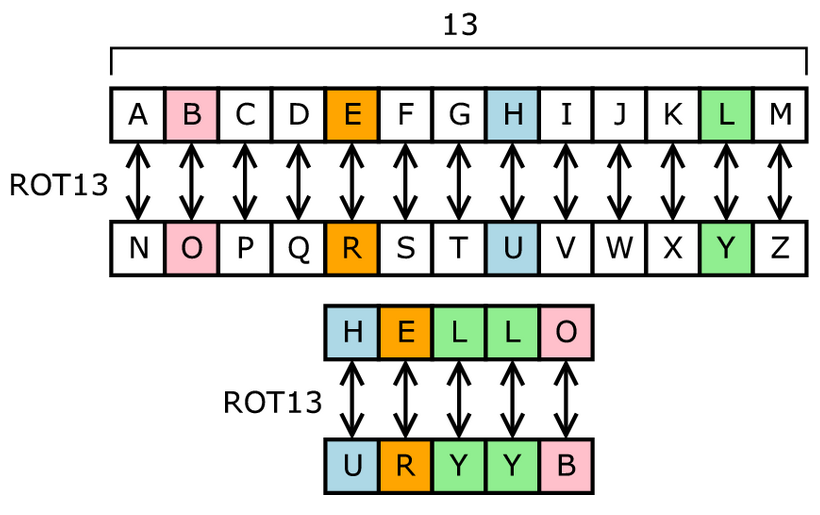
**

Figure 1 shows a Caesar Cipher that’s using a shift of 13.

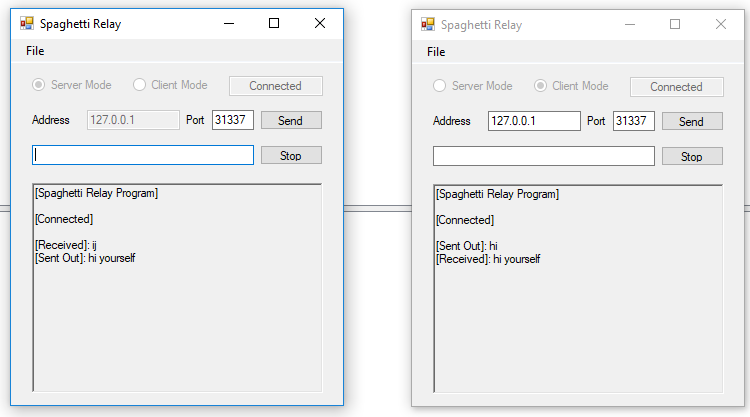


Figure 2 shows an example of an implementation in which the server is showing encoded cipher text and client is showing plain text.

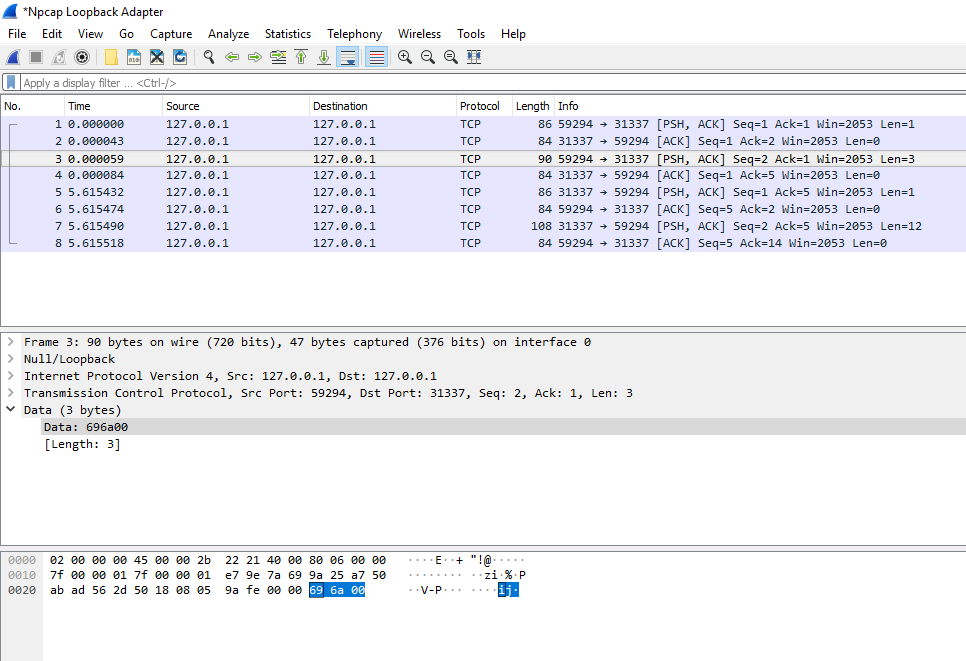


Figure 3 shows an example of Wireshark capturing a ciphered packet in which the message is shown as ij.

# Section 2: Questions

1. Implement so that **BOTH** client and server window screens receive encoded cipher text and any packet sniffed by a third party shows cipher text. What was the code added and where did you add it?

// decode

int encryptData = 13;

char cC;

int shift = 5;

int i = 0;

while (i < size)

{

cC = buffer[i];

//if (isupper(data[i]))

if (cC >= 'A' && cC <= 'Z')

{

buffer[i] = (char)((int)(buffer[i] - encryptData + 65) % 26 + 65);

}

if (cC >= 'a' && cC <= 'z')

{

buffer[i] = (char)((int)(buffer[i] - encryptData + 97) % 26 + 97);

}

i++;

}

I add this in the server and client readMessage() this allow to decode the encrypted message.

1. Implement so that **BOTH** client and server window screens receive decoded plain text and Wireshark receives cipher text. What was the code added and where did you add it? At this point, attach Client and Server files to the submission package.

// encode

int encryptData = 13;

char cC;

int shift = 5;

int i = 0;

while (i < length)

{

cC = data[i];

if (cC >= 'A' && cC <= 'Z')

{

data[i] = (char)((int)(data[i] + encryptData - 65) % 26 + 65);

}

if (cC >= 'a' && cC <= 'z')

{

data[i] = (char)((int)(data[i] + encryptData - 97) % 26 + 97);

}

i++;

}

I add this in the server and client sendMessage() this allow to encrypted the message.

1. Starting with *all* the files that are used to create Spaghetti Relay, find a place that you can add **ONE** block of code so that **BOTH** client and server users are receiving cipher text (on the GUI), but Wireshark receives plain text. What was the code added and where did you add it?

StringBuilder s = new StringBuilder();

while (i < buffer.Length)

{

if (buffer[i] == (byte)'\0')

{

break;

}

if (char.IsUpper((char)buffer[i]))

{

buffer[i] = (byte)(char)((int)(buffer[i] + encryptData - 65) % 26 + 65);

//s.Append(ch);

}

else

{

buffer[i] = (byte)(char)((int)(buffer[i] + encryptData - 97) % 26 + 97);

//s.Append(ch);

}

i++;

}

I add this in the SpaghettiForm file in the Connect() function this allow to decode the message and be able to see it in the user interface window.

# Turn In

* This assignment must be turned in using the .zip file format using <***LastName.FirstName.Lab3.zip*>**
* The zip file must include this word document, and any files asked for in the question section, including the header and source files.