Network Security and Management Assignment 2

Daniel Jenkins

February 20th, 2023

Question 1:

Encode the following plaintext messages (M) using Caesar cipher encryption.

- M = zoo, E(M) = 2rr
- M = xray, E(M) = ud1
- M = rellis, E(M) = uhoolv
- M = college station, E(M) = froohjh3vwdwlrq
- M = csci458, E(M) = fvfl78b

substitution_functions.py output

```
plaintext is zoo ciphertext is 2rr
plaintext is xray ciphertext is  ud1
plaintext is rellis ciphertext is uhoolv
plaintext is college station ciphertext is froohjh3vwdwlrq
plaintext is csci458 ciphertext is fvfl78b
```

Question 2:

Show the results of your client and server program running. For instance, your client sends the words we used before, such as "zoo", "xray", "rellis", "college station", "csci458". Then you can show what the server receives and the results of the decryption for each message.

tcpclient.py output

```
127.0.0.1
message: zoo ,sending (encrypted): 2rr
received (encrypted): 2rr , message: zoo

message: xray ,sending (encrypted):  ud1
received (encrypted):  ud1 , message: xray

message: rellis ,sending (encrypted): uhoolv
received (encrypted): uhoolv , message: rellis

message: college station ,sending (encrypted): froohjh3vwdwlrq
received (encrypted): froohjh3vwdwlrq , message: college station

message: csci458 ,sending (encrypted): fvfl78b
received (encrypted): fvfl78b , message: csci458
```

tcpserver.py output

```
waiting for a connetion on port 10000

Got a connection from ('127.0.0.1', 1323)
recieved (encrypted) 2rr , data: zoo

recieved (encrypted)  ud1 , data: xray

recieved (encrypted) uhoolv , data: rellis

recieved (encrypted) froohjh3vwdwlrq , data: college station

recieved (encrypted) fvfl78b , data: csci458

No more data from('127.0.0.1', 1323)
waiting for a connetion on port 10000
```

Question 3:

Encode the following plaintext messages using ROT13 encryption.

- M1 = zoo , ROT13(M1) = mbb
- M2 = xray , ROT13(M2) = kenl
- M3 = rellis , ROT13(M3) = eryyvf
- M4 = college station , ROT13(M4) = pbyyrtr fgngvba
- M5 = csci458 , ROT13(M5) = pfpv458

substitution_functions.py output

```
Before ROT13 :: zoo
After ROT13 :: mbb
After ROT13 again:: zoo


Before ROT13 :: xray
After ROT13 :: kenl
After ROT13 again:: xray


Before ROT13 :: rellis
After ROT13 :: eryyvf
After ROT13 again:: rellis


Before ROT13 :: college station
After ROT13 :: pbyyrtr fgngvba
After ROT13 again:: college station


Before ROT13 :: csci458
After ROT13 :: pfpv458
After ROT13 again:: csci458
```

Show the results of your client and server program running for the new function ROT13. For instance, your client sends the words we used before, such as “zoo”, “xray”, “rellis”, “college station”, “csci458”. Then you can show what the server receives and the results of the decryption for each message.

tcpclient.py output

```
127.0.0.1
message: zoo ,sending (encrypted): mbb
received (encrypted): mbb , message: zoo

message: xray ,sending (encrypted): kenl
received (encrypted): kenl , message: xray

message: rellis ,sending (encrypted): eryyvf
received (encrypted): eryyvf , message: rellis

message: college station ,sending (encrypted): pbyyrtr fgngvba
received (encrypted): pbyyrtr fgngvba , message: college station

message: csci458 ,sending (encrypted): pfpv458
received (encrypted): pfpv458 , message: csci458
```

tcpserver.py output

```
waiting for a connetion on port 10000

Got a connection from ('127.0.0.1', 30431)
recieved (encrypted) mbb , data: zoo

recieved (encrypted) kenl , data: xray

recieved (encrypted) eryyvf , data: rellis

recieved (encrypted) pbyyrtr fgngvba , data: college station

recieved (encrypted) pfpv458 , data: csci458

No more data from('127.0.0.1', 30431)
waiting for a connetion on port 10000
```

Question 4:

Encode the following plaintext messages using S-Box encryption and inverse S-Box for decryption.

M	E(M)
zoo	fuu
xray	wycn
rellis	yhppsj
college station	iuppeh jqcqsuo
csci458	ijis458

substitution_functions.py output

```
Before sBox :: zoo
```

```
After sBox :: fuu
After inv_sBox zoo


Before sBox :: xray
After sBox :: wycn
After inv_sBox xray


Before sBox :: rellis
After sBox :: yhppsj
After inv_sBox rellis


Before sBox :: college station
After sBox :: iuppheh jqcqsuo
After inv_sBox college station


Before sBox :: csci458
After sBox :: ijis458
After inv_sBox csci458
```

Question 5:

Using three rounds of encryption in your client and server code attach the results of your client and server windows showing the messages that you sent. To make sure, test them with the words we used before.

tcpclient.py output

```
127.0.0.1
message: zoo , sending (encrypted): 2hh
received (encrypted): 2hh , message: zoo

message: xray , sending (encrypted):  zr1
received (encrypted):  zr1 , message: xray

message: rellis , sending (encrypted): zkllns
received (encrypted): zkllns , message: rellis

message: college station , sending (encrypted): jhllkgk3strtnhm
received (encrypted): jhllkgk3strtnhm , message: college station

message: csci458 , sending (encrypted): jsjn78u
received (encrypted): jsjn78u , message: csci458
```

tcpserver.py output

```
waiting for a connetion on port 10000

Got a connection from ('127.0.0.1', 2093)
recieved (encrypted) 2hh , data: zoo

recieved (encrypted)  zr1 , data: xray

recieved (encrypted) zkllns , data: rellis
recieved (encrypted) jhllkgk3strtnhm , data: college station

recieved (encrypted) jsjn78u , data: csci458

No more data from('127.0.0.1', 2093)
waiting for a connetion on port 10000
```