

**Faculty of Engineering and Technology**

**Department of Electrical and Computer Engineering**

**LINUX LABORATORY- ENCS3130   
Section#2**

**PROJECT#1 “Shell Scripting”**

**Prepared by:**

**Hazar Michael 1201838**

**Afaf Amwas 1203359**

**Instructor:**

**Dr. Aziz Qaroush**

**Instructor’s Assistant:**

**Eng. Ayham Hashish**

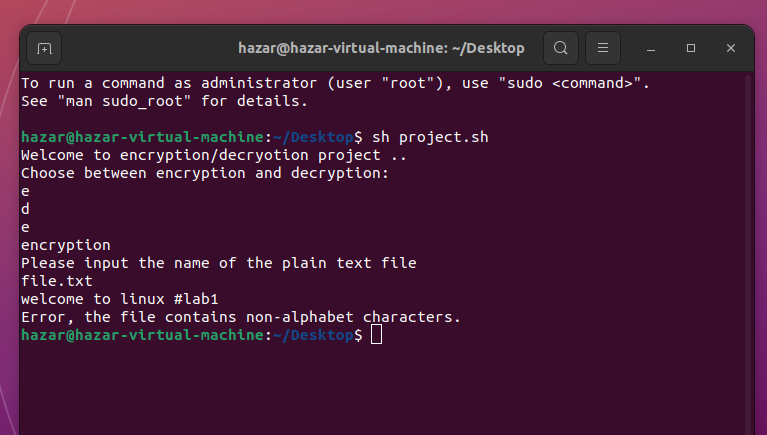
**August 2022**

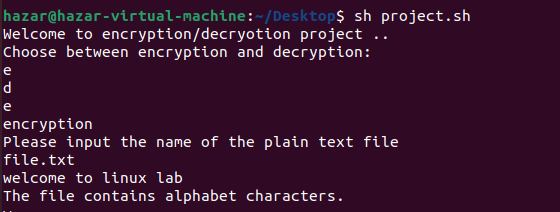
Introduction:

Encryption and Decryption are used by many programmers for security reasons. For example, if a company needs to secure their data in a file, they will encrypt and decrypt it so that only the programmer and the ones that have the key can access the file. In this Project, we have created a shell script that does encryption and decryption.

The first step was generating a key, then we represented the result of the key as an 8-bit binary number. After that, we did the XOR operation between each character’s Ascii code and the key, which will give an 8-bit binary result. Then we swapped the first 4 bits with the last 4 bits of each result. To increase the security even more, another swap operation is done on the 8-bit binary key, where the first 4 bits are swapped with the last 4 bits. The result is then added as the last character of the generated file.

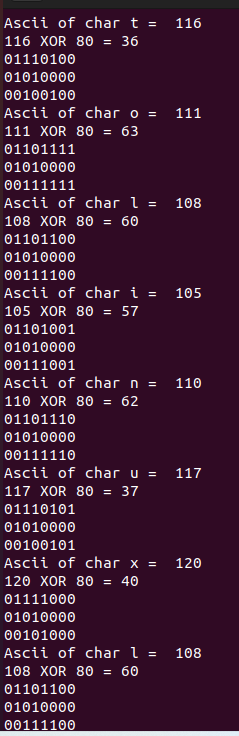
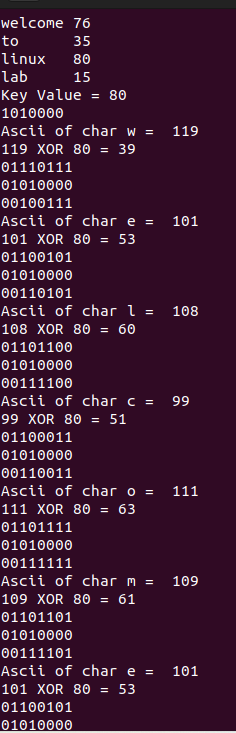
The process of decryption is similar to that of encryption but in the opposite order. The last character in the encryption file which is the key is swapped again to its original value. Then for each character in the encrypted file, the first 4 bits are swapped with the last 4 bits. Finally, an XOR operation is done with the key and each character.

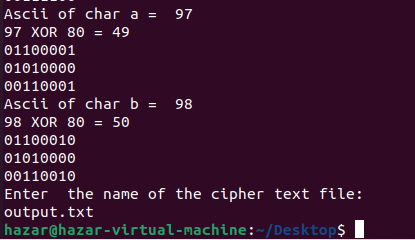
Here we check if the file contains only alphabet characters, if not the program prints an error message:



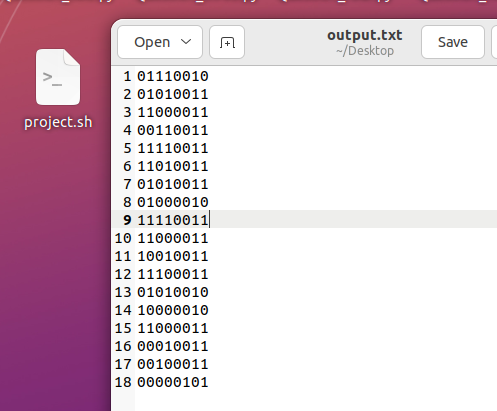
**Encryption Process:**

1. Calculating the KEY by adding the character’s index in each word, then taking mod 256 to make sure that the number will be represented in 8-bit binary number and taking the maximum as a result for the key then obtaining the Ascii value for every single character, then convert it to 8-bit binary number and XORing it with the KEY.



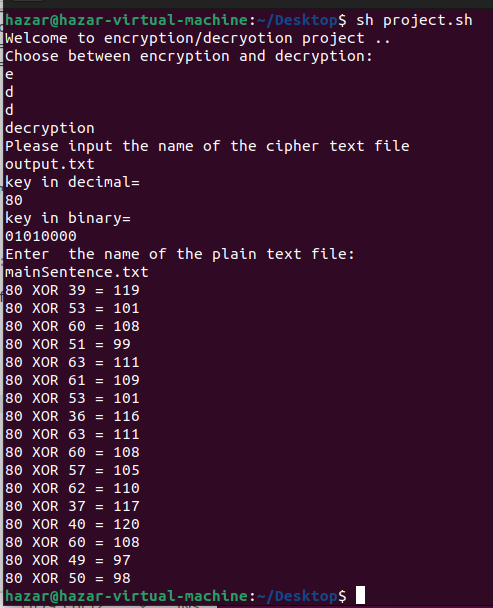


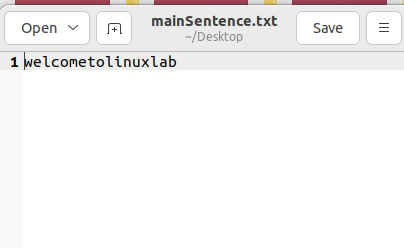
1. For each 8-bit binary result, swapping the first 4 bit with the last 4 bit. Key Swap: Swapping the first 4 bit with the last 4 bit. program will write the generated cipher text on the cipher file (“output.txt”) with the Key added as the last character.



**Decryption Process**:

1. user enters the option “d” for Decryption and is asked to enter the name of the cipher text file “output.txt”. Then it prints the value of the key, and asks the user to enter the name plain text file “mainSentence.txt”.



1. The program will write the generated plain text on the plain text file.

***The Code:***

echo "Welcome to encryption/decryotion project .. "

echo "Choose between encryption and decryption:"

echo "e"

echo "d"

read var

str="abcdefghijklmnopqrstuvwxyz"

if [ "$var" = "e" ]

then

echo "encryption"

echo "Please input the name of the plain text file"

read nameFile

value=`cat $nameFile`

value=$(echo $value | tr '[:upper:]' '[:lower:]') # transfer upper case letters to lower case letters

echo "$value"

if echo "$value" | grep '[^A-Za-z ]' $nameFile #to check the non-alphabet characters

then

echo "Error, the file contains non-alphabet characters."

else

echo "The file contains alphabet characters."

sumCharIndexWord=0 #summation of character's index

> word.txt

> sumword.txt

for i in $value # to iterate the words

do

for j in $(echo $i | sed -e 's/\(.\)/\1\n/g') #iterate the charecters for each word

do

prefix=${str%%$j\*}

index=${#prefix}

index=$((index+1))

echo $j

echo $index

sumCharIndexWord=$((index+sumCharIndexWord)) #sum of the indexes for each word

num=0

done

echo $i

echo $sumCharIndexWord

echo $i >> word.txt #print the words in a text file called word.txt

echo $sumCharIndexWord >> sumword.txt # print the summation of the character's indexes

sumCharIndexWord=0 #set to zero for the next word

done

paste word.txt sumword.txt #concatenate the two files

> wordPrint.txt

sort -nr sumword.txt | head -1 > wordPrint.txt #to obtain the max value (decs)

maxValue=`cat wordPrint.txt`

keyValue=$((maxValue % 256)) #to calculate the key value

echo "Key Value = "$keyValue

echo "obase=2; $keyValue" | bc #to print the key value as a binary number

#iterate each character in each word and XOR it with the key value, then swap the first four bits with the last four bits

for i in $value

do

for j in $(echo $i | sed -e 's/\(.\)/\1\n/g')

do

val=$(echo $j | tr -d "\n" | od -An -t dC)

res=$((keyValue ^ val))

echo "Ascii of char "$j" = "$val

echo $val" XOR "$keyValue" = "$res

echo "obase=2; $val" | bc | numfmt --format=%08f

echo "obase=2; $keyValue" | bc | numfmt --format=%08f

echo "obase=2; $res" | bc | numfmt --format=%08f >> result1.txt

echo "obase=2; $res" | bc | numfmt --format=%08f

done

done

cut -c1-4 result1.txt >> first4bits.txt

cut -c5-8 result1.txt >> last4bits.txt

echo "obase=2; $keyValue" | bc | numfmt --format=%08f >> key.txt

cut -c1-4 key.txt >> first4bits.txt

cut -c5-8 key.txt >> last4bits.txt

echo "Enter the name of the cipher text file:"

read filePrint

paste -d'\0' last4bits.txt first4bits.txt > $filePrint

fi

elif [ "$var" = "d" ]

then

echo "decryption"

echo "Please input the name of the cipher text file"

read nameFile

value=`cat $nameFile`

echo $(tail -n 1 $nameFile ) > tag.txt #to get the key

cut -c1-4 tag.txt > keyFirst.txt #to get the first four bit of the key

cut -c5-8 tag.txt > keyLast.txt #to get the last four bit of the key

paste -d'\0' keyLast.txt keyFirst.txt > keyDec.txt #to swap the bits of the key

keyValue=`cat keyDec.txt`

echo "key in decimal="

echo "ibase=2;obase=A; $keyValue" | bc

echo "key in binary="

echo $keyValue

cut -c1-4 $nameFile > decFirst.txt

cut -c5-8 $nameFile > decLast.txt

paste -d'\0' decLast.txt decFirst.txt > dec.txt

value=$(head -n -1 dec.txt) #to get the cipher text except the last line which contains the key value

echo "Enter the name of the plain text file:"

read filePrint

for i in $value

do

a=$(echo "ibase=2;obase=A; $keyValue" | bc)

b=$(echo "ibase=2;obase=A; $i" | bc)

res=$((a ^ b))

echo $a" XOR "$b" = "$res

echo $a" XOR "$b" = "$res >> outp.txt

printf \\$(printf '%03o' "$res") >> $filePrint #to convert from ascii to character

done

fi