|  |  |  |
| --- | --- | --- |
| **ipn** | **INSTITUTO POLITÉCNICO NACIONAL**  **ESCUELA SUPERIOR DE CÓMPUTO** |  |

**CRYPTOGRAPHY**

**“Title”**

Affine Cipher

**By:**

**Allan Ulises Zepeda Ibarra**

Professor:

M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

February2014

**Index**

Contenido

[Problem: 3](#_Toc349729798)

[Hypothesis: 3](#_Toc349729799)

[Software (libraries, packages, tools): 3](#_Toc349729800)

[Procedure: 3](#_Toc349729801)

[Results (Data): 3](#_Toc349729802)

[Conclusions: 3](#_Toc349729803)

[Code 4](#_Toc349729804)

# Problem:

# There is a message in a file with extension. Txt and wish that this can not be read by anyone who does not have the decryption key to view, using only encrypted using affine and lowercase letters of the alphabet.

# Hypothesis:

# Using an encryption method in which we have affine "C = ap + b" equation where we as a multiplicative facto (a = alpha) for the plaintext (p) and a factor addition (b = beta) for encryption and using only lowercase alphabet we have to use a ring Z = 26 and alpha that can be used are a = {1,3,5,7,9,11,15,17,19,21,23,25} the value of beta is not of much relevance, since taking this can encrypt, but an application of this type does not work if we can not decipher what by then encrypt with the formula p =-a (C + (-b)) in which ay have the inverse of the inverse of the ciphertext by where-b = 26-by-alpha may be a = {1,9,21,15,3,19,7,23,11,5,17,25 } corresponds directly with a.

# Software (libraries, packages, tools):

# In C language functions for handling archives were used asking the name of the file you want to encrypt and decipher archive you want, within the program are given the option to encrypt or deciphered plus you have to validate that a is correct that if it becomes encryption factor of 26 would not be correct and could not be deciphered correctly performed.

# Procedure:

# The first step is to ask the name of the file to encrypt or decipher any of the two cases then you must enter an alpha and a beta and this data is read the file character by character and character by character is encrypted and written or decrypts in another file like printing on screen to know the result.

# Results (Data):

# 

# 

# Conclusions:

# In this practice the part that complicates as taking correctly the value of the character and apply the transactions so that encryption and deciphered are correct taking into account this aspect applying modulo us constantly is helpful to have manageable values​​.

# Code

#include<stdio.h>

char evaluar(char let, int alp, int b);

char evaluarin(char let, int alp, int b);

FILE \*ou;

FILE \*aut;

int main(void)

{

int cont;

char a,aux,c,con;

char tex[50];

int al[12]={1,3,5,7,9,11,15,17,19,21,23,25};

int al1[12]={1,9,21,15,3,19,7,23,11,5,17,25};

int i=0,alp=0,b=0,ax=0,j=0;

FILE \*in;

//remove("dtext.txt");

//remove("ctext.txt");

printf("Select an option\n1-..Encrypt text\n2-..Deciphering text\n3-..Exit\n");

scanf("%d",&cont);

switch(cont)

{

case 1:

printf("Enter the name of the file to encrypt\n");

scanf("%s",tex);

in=fopen(tex,"r");

ou=fopen("ciphertext.txt","r+");

printf("Enter an alpha of sig. list ");

for(i=0;i<12;i++)

{

printf("%d ",al[i]);

}

printf("\n");

scanf("%d",&alp);

for(i=0;i<12;i++)

{

if(al[i]==alp)

{

ax=1;

j=i;

}

}

while(ax!=1)

{

printf("alpha entered is invalid enter another\n");

scanf("%d",&alp);

for(i=0;i<12;i++)

{

if(al[i]==alp)

{

ax=1;

j=i;

}

}

}

printf("enter a beta\n");

scanf("%d",&b);

while(!feof(in))

{

a=fgetc(in);

printf("%c",a);

//printf("\n%d\n",a);

aux=evaluar(a,al[j],b);

//printf("%d\n",aux);

//printf("%c\n",aux);

}

rewind(ou);

printf("\n");

while(!feof(ou))

{

c=fgetc(ou);

printf("%c",c);

}

break;

case 2:

printf("Enter the name of the file to decode\n");

scanf("%s",tex);

ou=fopen(tex,"r+");

aut=fopen("decrypted.txt","w");

printf("Enter an alpha of sig. list ");

for(i=0;i<12;i++)

{

printf("%d ",al[i]);

}

printf("\n");

scanf("%d",&alp);

for(i=0;i<12;i++)

{

if(al[i]==alp)

{

ax=1;

j=i;

}

}

while(ax!=1)

{

printf("alpha entered is invalid enter another\n");

scanf("%d",&alp);

for(i=0;i<12;i++)

{

if(al[i]==alp)

{

ax=1;

j=i;

}

}

}

printf("enter a beta\n");

scanf("%d",&b);

rewind(ou);

while(!feof(ou))

{

c=fgetc(ou);

aux=evaluarin(c,al1[j],b);

}

break;

default:

return;

}

fclose(in);

fclose(ou);

}

char evaluar(char let, int alp, int b)

{

int ax=0;

char axx;

if(let<97||let>122)

{

//printf("no es");

return 0;

}

else

{

ax=let-97;

ax=ax\*alp;

ax=ax%26;

ax=ax+b;

ax=ax%26;

axx=ax+65;

fputc(axx,ou);

return axx;

}

}

char evaluarin(char let, int alp, int b)

{

int ax=0;

char axx;

if(let<65||let>90)

{

return 0;

}

else

{

ax=let-65;

ax=ax+(26-b);

ax=ax%26;

ax=ax\*alp;

ax=ax%26;

axx=ax+97;

printf("%c",axx);

fputc(axx,aut);

return axx;

}

}