INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

CRIPTOGRAFÍA

Sesión de Laboratorio 1

Cifrados de sustitución

Grupo: 3CV2

Alumnos:

Martínez Galindo Angélica

Montaño Castañeda Daniel

Maestra:

Días Santiago Sandra

Contenido

Objetivo

En esta sesión vamos a trabajar con códigos de sustitución, en particular, con el cifrado afín. Además, vamos a hacer un ataque de texto cifrado, sólo para el cifrado afín.

Teoría

1. Breve descripción de los cifrados por sustitución

Es aquel cifrado que sustituye cada letra o grupo de letras por otra letra o grupo de letras distinta/s para cifrar texto en claro. Los primeros y antiguos métodos de cifrado se basaban en este principio, aunque en aquella época no eran muy robustos ni difíciles de descifrar, pero les resultaban muy útiles.

2. Definición de ataque solo a texto cifrado

Este tipo de ataque se da cuando el atacante tiene acceso solo al texto cifrado, por lo que solo es posible aplicar un análisis estadístico al mismo, en el caso de que se conozca el algoritmo de cifrado se podría también realizar un ataque por fuerza bruta.

3. Recursos del adversario cuando se realiza un ataque solo de texto cifrado

El adversario tiene sólo copias del texto cifrado, en estos casos el adversario podría realizar análisis estadísticos, en el caso de que este tenga como recurso el conocimiento de saber qué tipo de cifrado se está usando, esto facilitaría mucho su labor, ya que solo le quedaría encontrar la llave, además de esto puede contar con recursos como computadoras lo demasiado potentes como para poder descubrir el tipo de cifrado usado mediante análisis. Todos estos recursos se deben tomar en cuenta a la hora de elegir un tipo de cifrado para saber cuál es más conveniente usar dependiendo de qué tan sensible sea la información con la que se trata.

Desarrollo

Ejercicios de programación

Programa 1

1. Diseño de un programa en C / C ++ para cifrar y descifrar utilizando el cifrado afín. Considera el según las exigencias.

a) Los valores de a y b deben ser elegidos por el usuario. Su programa debe comprobar que a es un valor válido.

b) Su programa debe funcionar sólo con el alfabeto Inglés, que tiene 26 símbolos.

c) Suponga que su texto plano solamente tiene símbolos del alfabeto Inglés.

d) No cifre los espacios en blanco.

e) El programa debe trabajar con archivos de texto de cualquier tamaño.

Código 1

```
#include <stdio.h>
#include <stdlib.h>

int inverso(int x,int m);
char encriptar(char aux,int m,int a,int b);
char desencriptar(char aux,int m,int a,int b);
int gcd(int x,int y);

int main()
{
    FILE *entrada;//archivo de entrada
    FILE *salida;//archivo de salida
    char caracter,caracter_c;//auxiliar para caracteres
    int a,b,m,opc;//parametros de cifrado y opcion de cifrar o descifrar
    m=26;

    printf("Quieres cifrar(1) o descifrar(*): ");
    scanf("%d",&opc);
    printf("\nIntroduce a:");
    scanf("%d",&a);
    printf("\nIntroduce b:");
```

```
    scanf("%d",&b);

    if(gcd(a,m)!=1){ //si no son coprimos
        printf("\nValor de 'a' no valido.");
        exit(0);
    }

    entrada = fopen("entrada.txt","r");
    salida = fopen("salida.txt","w");

    if (entrada == NULL){
        printf("\nError de apertura del archivo. \n\n");
        }
        else{
            while ((caracter=fgetc(entrada))!=EOF)
            {
                if(opc==1){
                    caracter_c=encriptar(caracter,m,a,b);
                    printf("%c", fputc(caracter_c, salida));
                }
                else{
                    caracter_c=desencriptar(caracter,m,a,b);
                    printf("%c", fputc(caracter_c, salida));
                }
            }
        }
        fclose(entrada);
        fclose(salida);
    return 0;
}



int inverso(int x,int m)
{
    int i;
    for(i=1; i<m; i++)
    {
        if( (x*i)%m ==1)
            return i;
    }
}

char encriptar(char aux,int m,int a,int b){//funcion que encripta un
caracter dados los parametros de cifrado
    char letra_c;
        if(aux<65||aux>90){
            letra_c=aux;
        }
        else{
            letra_c=((a*(aux-65)+b)%m)+65;
        }
        return letra_c;
}

char desencriptar(char aux,int m,int a,int b){//funcion que desencripta
un caracter dados los parametros de cifrado
```

```
    char letra_c;
        if(aux<65||aux>90){//Si se recibe un caracter que no pertenezca
al abecedario que no lo modifique
            letra_c=aux;
        }
        else{
            letra_c = (inverso(a,m)*((aux-65)-b))%m;
            if(letra_c<0){
                letra_c=letra_c+26+65;
            }
            else{
                letra_c=letra_c+65;
            }
        }
        return letra_c;
}


int gcd(int x,int y) //calculamos el maximo comun divisor de dos numeros
{
    int c;
    while(x!=0){
        c=x;
        x=y%x;
        y=c;
    }
    return y;
}
```

Pruebas

```
C:\Users\Angelica\Desktop\CRIPTO>gcc practica1.c -o practica1

C:\Users\Angelica\Desktop\CRIPTO>practica1
```

Podemos ver la forma de compilar y ejecutar nuestro primer programa, descrito anteriormente. A continuación se mostrará una prueba del programa con un ejercicio visto en clase. Hay que recordar que este programa lee el mensaje a cifrar de un archivo, llamado entrada.txt y el mensaje encriptado se mostrará en un archivo, llamado salida.txt. Sin embargo en este caso lo mandaremos a imprimir para poder visualizar el resultado sin tener que estar abriendo los demás archivos.

M = HELLO

K = (3,2)

Ek(x) = ax + b mod 26

H	E	L	L	O
7	4	11	11	14

Ek(H) = Ek(7) = 3*7+2 mod 26 = 23

Ek(E) = Ek(4) = 3*4+2 mod 26 = 14

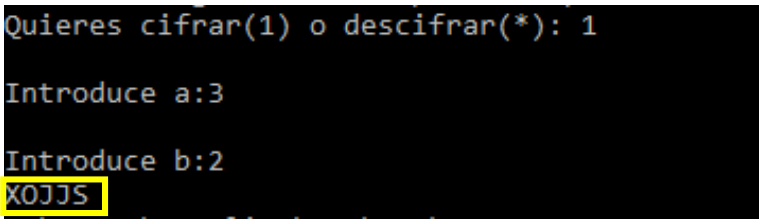
Ek(L) = Ek(11) = 3*11+2 mod 26 = 9

Ek(L) = Ek(11) = 3*11+2 mod 26 = 9

Ek(O) = Ek(14) = 3*14+2 mod 26 = 18

X	O	J	J	S
23	14	9	9	18

E = XOJJS

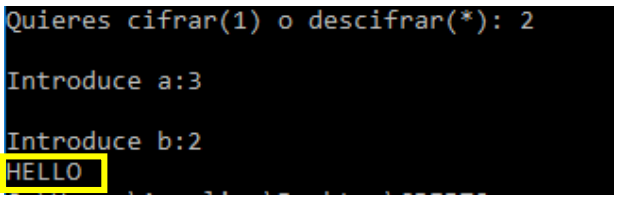


Podemos visualizar como es que el programa funciona correctamente. Para descifrar es el mismo procedimiento pero es necesario utilizar una ecuación diferente.

M = XOJJS

K = (3,2)

Dk(x) = a^{-1}(y-b) mod 26



Programa 2

2. Ya sabemos que si realizamos un ataque de fuerza bruta para cambiar de cifrado, sólo tenemos que probar 26 teclas. Una mejor manera de cifrar mediante la sustitución se permutar el alfabeto, y luego sustituir cada carácter en el texto en claro por el correspondiente

Código 2

```
#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;

bool isInAlphabet(char c, char alphabet[2][26],int tam);
int row(char c, char alphabet[2][26], int col);
string encrypt(string s, char alphabet[2][26]);
string decrypt(string s, char alphabet[2][26]);
bool isInKeyWord(char c, string key_word);
bool isValidKey(string key);

int main()
{
    string s,s1,key_word,ciphertext;
    char alphabet_c [2][26];
    int a,b,m=26,i;


    cout << "Introduce la cadena a cifrar: ";
    getline(cin, s);
    cout << "Introduce la llave: ";
    cin >> key_word;

    for(int i=0;i<s.size();i++) //pasamos a mayusculas nuestro texto a
cifrar
        s[i]=toupper(s[i]);

    for(int i=0;i<key_word.size();i++) //pasamos a mayusculas nuestra key
word
            key_word[i]=toupper(key_word[i]);
```

```
    if(isValidKey(key_word) == false) //validamos nuestra key word

    {
        cout << "Llave no debe contener caracteres repetidos" << endl;
        exit(0);
    }


    for(int alph=0; alph< m; alph++)
        alphabet_c[0][alph] = 65+alph;      //llenamos nuestra primera
columna con el alfabeto

    for( i=0; i<key_word.size() ; i++)      //agregamos en la segunda
columna la key word
        alphabet_c[1][i] = key_word[i];

    int tam= i;
    for(int sc=0; sc<m; sc++)   //generamos el alfabeto para nuestro
cifrado
    {
        if(!isInAlphabet(sc+65,alphabet_c,tam))//si el caracter no se
encuentra dentro de nuestro arreglo lo agregamos
        {
            alphabet_c[1][tam] = (65+sc);
            tam++;
        }
    }

    cout << "Texto plano: " << s << endl;
    ciphertext = encrypt(s,alphabet_c);
    cout << "Texto cifrado: " << ciphertext << endl;
    cout << "Texto descifrado: " << decrypt(ciphertext,alphabet_c) <<
endl;


   return 0;
}

bool isInAlphabet(char c, char alphabet[2][26],int tam) //checamos si el
caracter "c" se encuentra en el arreglo de nuestro alfabeto
{
    for(int i=0; i<tam; i++)
    {
        if(c == alphabet[1][i])
            return true;
    }

    return false;
}

int row(char c, char alphabet[2][26], int col) //devuelve el renglon en
donde se encuentra el caractet "c"
{
    for(int i=0; i<26; i++)
    {
        if(c == alphabet[col][i])
```

```
            return i;
        }
    }

    string encrypt(string s, char alphabet[2][26])
    {
        string s_enc;
        for(int i=0; i<s.size(); i++)
        {
            if(s.at(i) != ' ' )
                s_enc.push_back(alphabet[1][row(s[i],alphabet,0)]);
            else
                s_enc.push_back(' ');
        }
        return s_enc;
    }

    string decrypt(string s, char alphabet[2][26])
    {
        string s_dec;
        for(int i=0; i<s.size(); i++)
        {
            if(s.at(i) != ' ' )
                s_dec.push_back(alphabet[0][row(s[i],alphabet,1)]);
            else
                s_dec.push_back(' ');
        }
        return s_dec;
    }

    bool isInKeyWord(char c, string key_word) //checamos si el caracter "c"
    se encuentra en el arreglo de nuestro alfabeto
    {
        for(int i=0; i<key_word.size(); i++)
        {
            if(c == key_word[i])
                return true;
        }

        return false;
    }

    bool isValidKey(string key)
    {
        for(int i=0; i<key.size(); i++){
            if(isInKeyWord(key[i],key))
                return false;
        }
        return true;

    }
```

Pruebas

Para compilar y ejecutar:

```
C:\Users\Angelica\Desktop\CRIPTO>g++ practica1_2.cpp -o practica1_2

C:\Users\Angelica\Desktop\CRIPTO>practica1_2
```

Siguiendo el ejemplo de la práctica insertamos nuestra llave como se muestra a continuación.

K = CRYPTO

M = We will meet at midnight

A	B	C	D	E	F	G	H	I	J	K	L	M
C	**R**	**Y**	**P**	**T**	**O**	A	B	D	E	F	G	H
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
I	J	K	L	M	N	Q	S	U	V	W	X	Z

```
Introduce la cadena a cifrar: We will meet at midnight
Introduce la llave: CRYPTO
Texto plano: WE WILL MEET AT MIDNIGHT
Texto cifrado: VT VDGG HTTQ CQ HDPIDABQ
Texto descifrado: WE WILL MEET AT MIDNIGHT
```

K = DANIEL

M = ABCDEFGHIJKLMNOPQRSTUVWXYZ

A	B	C	D	E	F	G	H	I	J	K	L	M
D	**A**	**N**	**I**	**E**	**L**	B	C	F	G	H	J	K
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
M	O	P	Q	R	S	T	U	V	W	X	Y	Z

```
C:\Users\Angelica\Desktop\CRIPTO>practica1_2
Introduce la cadena a cifrar: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Introduce la llave: DANIEL
Texto plano: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Texto cifrado: DANIELBCFGHJKMOPQRSTUVWXYZ
Texto descifrado: ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Validación de llave

```
Introduce la cadena a cifrar: ANGELICA
Introduce la llave: ALA
Llave no debe contener caracteres repetidos
```

Programa 3

3. Uso de su programa para el punto 1, el diseño de un programa para aplicar ataque de fuerza bruta a los textos cifrados abajo. Descubre el texto en claro y la clave utilizada para los valores es decir, a y b.

a) QNMCNURAPIJDPCIRYFJQCNCHACREPJQLKRQWRPMIIRWWRPTNCQ
RAKPMMTNAANMNAQXCRADNIRJKNTRCMPHANCINIMPTNA
b) CXDKRPQVUKXDGRURPDAKVSZNUVCUMVZHMUPTVZNKXDUX
c) ZSTNBPUHDNNWPQFNRHQOZCDCHMHUWNZQSNUSUHH

Código 3

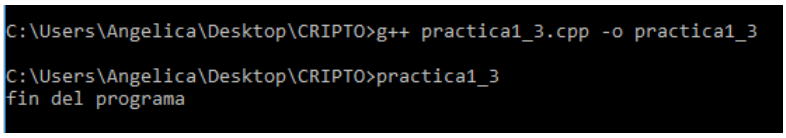
```
#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;


int gcd(int x,int y);
int inverso(int x,int m);
char desencriptar(char aux,int m,int a,int b);
void generadorLlaves(ifstream& fe,int m);

int main()
{
    string cadena,cadena_cifrada;
    char c;
    int a,b,m=26;

    ifstream fe("c2_a.txt");
    generadorLlaves(fe,m);

    cout << "fin del programa" << endl;


   return 0;
}

int gcd(int x,int y) //funcion para calcular el maximo comun divisor
{
```

```
    int c;
    while(x!=0)
    {
        c=x;
        x=y%x;
        y=c;
    }
    return y;
}

int inverso(int x,int m)//funcion para calcular el inverso de un numero
{
    for(int i=0; i<m; i++)
    {
        if( (x*i)%m == 1)
            return i;
    }
}

char desencriptar(char aux,int m,int a,int b){//funcion que desencripta
un caracter dados los parametros de cifrado
    char letra_c;
        if(aux<65||aux>90){//Si se recibe un caracter que no pertenezca
al abecedario que no lo modifique
            letra_c=aux;
        }
        else{
            letra_c = (inverso(a,m)*((aux-65)-b))%m;
            if(letra_c<0){
                letra_c=letra_c+26+65;
            }
            else{
                letra_c=letra_c+65;
            }
        }
        return letra_c;
}

void generadorLlaves(ifstream& fe,int m)
{
    string cadena_cifrada,cadena_descifrada;

    ofstream fs("salida.txt");

    while(!fe.eof())
       getline(fe,cadena_cifrada);

    for(int a=0; a<m; a++)
    {
        if(gcd(a,m)==1)
        {
            for(int b=0; b<m; b++)
            {
                fs << a << " " << b << " ";
                for(int i=0; i<cadena_cifrada.size(); i++)
                    fs << desencriptar(cadena_cifrada[i],m,a,b);
```

```
                fs << endl;
            }
        }

    }

    fs.close();

}
```

Pruebas

Para compilar y ejecutar:



En este ejercicio se nos ha proporcionado 3 archivos diferentes los cuales contienen los textos cifrados que se muestran en la descripción del ejercicio 3, para esto fue necesario utilizar el código del programa 1 pero con una función nueva la cual se encargará de la generación de llaves. Sabemos que solamente hay 12 números coprimos de 26 (número de letras que tiene el alfabeto inglés), por lo tanto se generarán 12*26 = 312 diferentes llaves con sus respectivos textos descifrados. Sin embargo, se nos solicitó encontrar el texto que tenga más sentido a continuación se mostrarán los resultados obtenidos para cada uno de los tres textos encriptados.

- Para el inciso a

  No se encontró algún texto con sentido.

- Para el inciso b

```
5 1  VUQHYIDEJHUQBYJYIQFHETKSJEVJXEKWXJIOEKSHUQJU
5 2  AZVMDNIJOMZVGDODNVKMJYPXOJAOCJPBCONTJPXMZVOZ
5 3  FEARISNOTREALITISAPRODUCTOFTHOUGHTSYOUCREATE
5 4  KJFWNXSTYWJFQNYNXFUWTIZHYTKYMTZLMYXDTZHWJFYJ
5 5  POKBSCXYDBOKVSDSCKZBYNEMDYPDRYEQRDCIYEMBOKDO
```

  El texto encontrado fue: "*FEAR IS NOT REATL IT IS A PRODUCT OF THOUGHTS YOU CREATE*"

  Llave: a=5, b=3

- Para el inciso c

```
11 13 URKAGMDQSAAPMFEAYQFTUZSZQHQDPAUFRADRDQQ
11 14 BYRHNTKXZHHWTMLHFXMABGZGXOXKWHBMYHKYKXX
11 15 IFYOUAREGOODATSOMETHINGNEVERDOITFORFREE
11 16 PMFVBHYLNVVKHAZVTLAOPUNULCLYKVPAMVYMYLL
11 17 WTMCIOFSUCCROHGCASHVWBUBSJSFRCWHTCFTFSS
```

El texto encontrado fue: "*IF YOU ARE GOOD AT SOMETHING NEVER DO IT FOR FREE*"

Llave: a=11, b=15

Referencias

- http://gaussianos.com/critpografia-cifrado-por-sustitucion/
- http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/1-panorama-general/14-ataques/142-ataques-a-los-metodos-de-cifrado