Instituto Politécnico Nacional

Escuela Superior de Cómputo

Cryptography

Reporte

Laboratorio 1

Cifrado por sustitución

Santiago Mancera Arturo Samuel

Caballero Huesca Carlos Eduardo

Grupo: 3CV1

Profesora: Díaz Santiago Sandra

27 de agosto del 2016

1. Teoría

Ataque de fuerza bruta al cifrado Afín

EL ataque por fuerza bruta al cifrado afín consiste en probar todas las posibles claves conformadas por *a* y *b* sobre un alfabeto n*.*

$$\text{Cifrado: } E(x)=(ax+b)\bmod m$$
$$\text{Descifrado: } d(x)=a^{-1}(x-b)\bmod m$$

En lo que respecta al coeficiente *b*, éste puede ser cualquier valor comprendido entre *0* y *n-1*, puesto que su función es simplemente de realizar un desplazamiento al carácter por cifrar.
Por otro lado, debido a que la llave debe permitir el descifrado, es necesario que *a mod n* tenga un inverso multiplicativo, lo cual puede comprobarse si obtenemos *1* al calcular el máximo común divisor entre *a* y *n*.

$$mcd(a,n)=1$$

Así pues, mientras que para *b* tenemos *n* posibilidades, para *a* es necesario encontrar todos números entre *0* y *n-1* que tengan inverso multiplicativo.

En el caso de la presente práctica, se trabajó sobre un alfabeto de 26 caracteres. Por lo tanto, los posibles valores para *a* son:

$$a=1,3,5,7,9,11,15,17,19,21,23,25$$

Por lo tanto, el cifrado afín puede romperse comparando todas las 312 posibles combinaciones entre *a* y *b.*

a = 1, b = 1
a = 1, b = 2
...
a = 25, b = 25

Llaves de matrices 2x2 y 3x3 para el cifrado Hill

El cifrado Hill, requiere para su funcionamiento matrices invertibles que seŕan usadas como llaves para el cifrado. Sin embargo, las llaves deben cumplir con ciertas condiciones:

- Debe ser una matriz cuadrada.
- Debe ser invertible.

Para comprobar que dicha matriz sea invertible, es necesario obtener su determinante, el cual debe ser distinto de cero. Una vez obtenido, calculamos su módulo n y comprobamos que el valor resultante sea coprimo con *n.*

Es decir: $mcd(det, n)=1$

Una vez probado que la matriz llave tiene inversa, se utiliza algún método para obtenerla. La matríz resultante será útil en el proceso de descifrado.

$$K^{-1}=(Det(K))^{-1}[A]$$

Se debe tener en cuenta que una vez obtenido el determinante módulo n, su inverso será el inverso multiplicativo módulo n.

2. Ejercicios de programación

1. Diseña un programa en C/C++ para descubrir el texto en claro de cada texto cifrado. Todos ellos fueron cifrados con el método afín, con diferentes valores para *a* y *b*.

```
55  void descifra(char *textC){
56      int a[]={1,3,5,7,9,11,15,17,19,21,23,25};
57      int b=0,i=0,j=0;
58      int caracter1,caracter2;
59
60      for(b=0;b<26;b++){
61          for(i=0;i<12;i++){
62              printf("a=%d  b=%d\n",a[i],b);
63              for(j=0;textC[j]!='\0';j++){
64                  caracter1=(a[i]*(textC[j]-65))-b;
65                  if(caracter1<0){
66                      caracter1=26-caracter1;
67                  }
68                  caracter2=(caracter1%26);
69                  printf("%c",caracter2+65);
70              }
71              printf("\n\n");
72          }
73      }
74      return;
75  }
```

La función descifra define los valores posibles que puede tener el coeficiente *a* conforme a la teoría revisada.

Mientras que los posibles valores de *b* se definen mediante un ciclo que va desde 0 a 25.

El segundo ciclo recorre los valores de *a* almacenados en el vector 'a'.

El tercer ciclo descifra cada carácter del mensaje cifrado.

Se calcula: $caracter\,1=a^{-1}(x-b)$

Si el resultado es negativo se realiza 26 – caracter1 y se calcula su módulo 26.

Los textos decifrados son:

c1-171.txt
a=15, b=9
IFYOUCANTEXPLAINITSIMPLYYOUDONOTUNDEJSTANDITWELLENOUGH

c2-171.txt
a=19, b=3
IFNOBODYHATESYOUYOUREDOINGSOMETHINGWRONG

c3-171.txt
a=25, b=22
IFYOULOOKFORTHWLIGHTYOUCANOFTWNFINDITBUTIFYOULOOKFORTHWDARKT
HATISALLYOUWILLWVWRSWWV

2. Diseña un programa que genere llaves válidas (2x2) para el cifrado Hill. El programa debe generar la matriz de manera aleatoria y verificar las propiedades para obtener la llave válida. La matríz y su inversa deben guardarse en un archivo cuyo nombre será ingresado como parámetro por el usuario.

```
                //Crear Matriz 2X2
int MatrizK[2][2]={rand()%26,rand()%26,rand()%26,rand()%26};
                //Imprimir Matriz
```

```
16                  //Calcular el determinante
17          determinante = (MatrizK[0][0]*MatrizK[1][1]) - (MatrizK[1][0]*MatrizK[0][1]);
18              //Determinante mod 26 para determinantes negativos
19          while(determinante<0){
20          determinante = determinante+MOD;
21          }
22          //Determinante mod 26 para determinantes positivos
23          if(determinante>=MOD){
24              determinante = determinante%MOD;
25          }
26          // Determinar si tiene inversa
27          if(determinante!=0&&determinante%2!=0&&determinante%13!=0){
28              tieneinversa = mcd(determinante,MOD);
29              //Calcular Inverso Multiplicativo Modular
30              if(tieneinversa==1){
31              inversoMulti=inversoMultiplicativoModular(determinante, MOD);
32              printf("Esta Llave si se puede usar\n");
33              printf("El determinante de la matriz es %d mod 26\n",determinante);
34              printf("Esta matriz si tiene inversa\n");
35              printf("El inverso multiplicativo de esta matriz es %d mod 26\n",inversoMulti);
36                              }
37          }
38          else{
39          printf("La matriz llave no se puede usar\n");
40          printf("El determinante de la matriz es %d mod 26\n",determinante);
            }
Line 2, Column 1
```

```c
int inversoMultiplicativoModular(int a, int modulo){
    int residuo,x,y,cociente;
    int coeficienteA = 1;
    int coeficienteB = -1*(modulo/a);
    int corrida = 0;
    int corrida2 = 1;

    residuo = modulo%a;
    x = a;
    y = residuo;
    while( residuo != 0 ){
        cociente = x / y;
        residuo = x % y;
        coeficienteA *= -1*cociente;
        coeficienteB *= -1*cociente;
        coeficienteA += corrida;
        coeficienteB += corrida2;
        corrida = -1*( coeficienteA-corrida )/cociente;
        corrida2 = -1*( coeficienteB-corrida2 )/cociente;
        x = y;
        y = residuo;
    }
    if( x == 1 ){
        if( corrida2 >= 0 )
        {return corrida2;}
        else
        {return corrida2 + MOD;}
    }
    else
        { return -1; }

}
```

3. Repita el proceso anterior para una matriz 3x3.

matrizK3x3.c

```c
    int determinante,tieneinversa,inversoMulti;
    srand(time(NULL));
                    //Crear Matriz 3X3
    int MatrizK[3][3]={rand()%26,rand()%26,rand()%26,rand()%26,rand()%26,rand()%26,rand()%26,rand()%26,rand()%26};
                        //Imprimir Matriz
    int i,j;
    for (i=0; i<3; i++){
        for (j=0; j<3; j++){
            printf(" %d ",MatrizK[i][j]);
        }
            printf("\n");
    }
    //Calcular el determinante
    determinante = (MatrizK[0][0]*MatrizK[1][1]*MatrizK[2][2]+MatrizK[1][0]*MatrizK[2][1]*MatrizK[0][2]+MatrizK[2][0]*MatrizK[0][1]*MatrizK[
                   -(MatrizK[0][2]*MatrizK[1][1]*MatrizK[2][0]+MatrizK[1][2]*MatrizK[2][1]*MatrizK[0][0]+MatrizK[2][2]*MatrizK[0][1]*Matriz
    //Determinante mod 26 para determinantes negativos
        while(determinante<0){
        determinante = determinante+26;
        }
        //Determinante mod 26 para determinantes positivos
        if(determinante>=26){
            determinante = determinante%26;
        }
        // Determinar si tiene inversa
        if(determinante!=0&&determinante%2!=0&&determinante%13!=0){
            tieneinversa = mcd(determinante,26);
            //Calcular Inverso Multiplicativo Modular
            if(tieneinversa==1){
            inversoMulti=inversoMultiplicativoModular(determinante, MOD);
            printf("Esta Llave si se puede usar\n");
            printf("El determinante de la matriz es %d mod 26\n",determinante);
            printf("Esta matriz si tiene inversa\n");
            printf("El inverso multiplicativo de esta matriz es %d mod 26\n",inversoMulti);
                            }
        }else{
        printf("La matriz llave no se puede usar\n");
        printf("El determinante de la matriz es %d mod 26\n",determinante);
        }
    return 0;
```

3. Bibliografía

Rodríguez, Francisto. *Códigos Y Criptografía*. 1st ed. Ciudad de México: N.p., 2016. Web. 27 Aug. 2016.