Universidad de Carabobo
Facultad Experimental de Ciencias y Tecnología "FACyT"
Departamento de Computación
Redes de Computadoras II

Proyecto de cifrado

Algoritmo de cifrado por transposición.

"El objetivo de éste proyecto consiste en desarrollar un esquema de Criptografía que cumpla las condiciones requeridas del Proyecto y aplicarlo a un texto llano para ser evaluados por los demás integrantes del curso para intentar descifrarlo."

Autor:
Vanessa Cruz
23426481

Profesor:
Antonio Castañeda

Naguanagua, 25 de marzo del 2017

Proyecto de cifrado

Algoritmo de cifrado por transposición.

Marco teórico:

En el proyecto a continuación se utilizó el algoritmo de cifrado por transposición. El cual tomara del archivo "textollano.txt" el texto elegido para ser cifrado por medio de una clave de 10 caracteres de forma alfanumérica, la cual es "**Ñ3hOXLsFQy**", donde su equivalente en números enteros según su posición carácter por carácter en la tabla ASCII es: **[165][51][104][79][88][76][115][70][81][121]**. El algoritmo funciona de la siguiente manera: cada letra de la clave representará una columna de la matriz en donde se encuentra cargado el texto llano, y cada columna será ordenada según la letra más pequeña de la clave tomando en cuenta el número con el cual se simboliza por medio del código ascii, de ésta forma se generará un patrón el cual repetirá la clave hasta que llené el total de las columnas de la matriz, según su orden. Para mayor entendimiento vea la siguiente figura donde explica el patrón que genera el algoritmo a partir de la clave para hacer el ordenamiento de columnas de la matriz:

165	51	104	79	88	76	115	70	81	121

Clave en código ASCII:

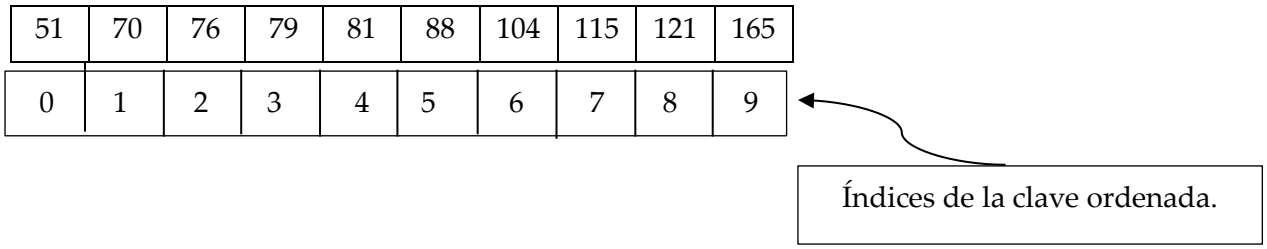
Matriz con texto llano cagado en ella:

Cifrado por transposición

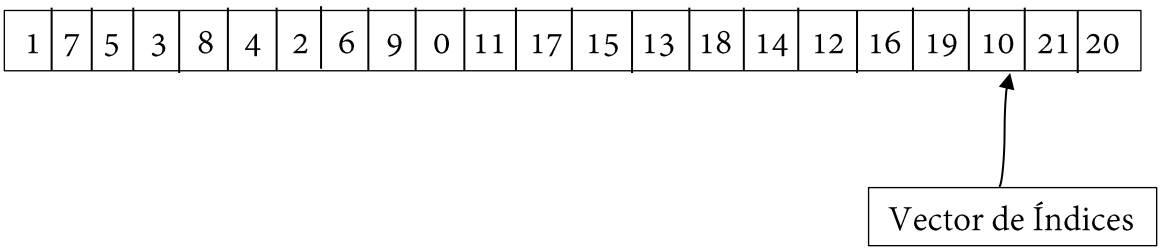
Los cifrados por transposición, en contraste, reordenan las letras, pero no las disfrazan o las sustituyen por otras. La transposición columnar consiste en confiarle el cifrado en una clave la cual será el pilar del cifrado del algoritmo. La clave del cifrado es una palabra o frase que no contiene letras repetidas. El propósito de la clave es numera las columnas, estando la columna "k" bajo la letra clave más cercana al inicio del alfabeto, y así sucesivamente hasta completar todas las columnas de la matriz donde esté cargado el texto llano listo para cifrar.

Cada cuadrito representará el índice de la columna por el cual se debe empezar a ordenar la nueva matriz cifrada, sin embargo el tamaño de la clave no es el mismo que el tamaño en total de las columnas que se generan al cargar el texto llano en la matriz es por ellos que la técnica empleada para continuar con el algoritmo de transposición es generar un patrón de ordenamiento de columnas por medio de la clave, de forma que se repetirá la clave cuando la longitud de tamaño de esta termine, y el patrón se generará hasta llenar la cantidad de columnas en la matriz.

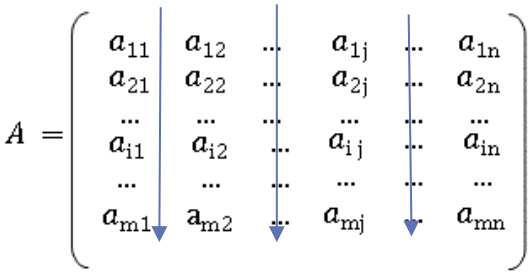
Ordenándolo de menor letra de la clave a mayor, el patrón quedaría de la siguiente forma:



Ahora bien, como nos interesa es ordenar cada columna de la matriz del texto llano de forma como el patrón de la clave nos diga, se hará lo siguiente. Se creará un vector de índices el cual nos dirá cual columna debemos tomar primero y cual debemos tomar después de la matriz, el orden de este vector de índices será según la clave y la dimensión de éste será el tamaño de las columnas que sean generadas por la matriz que contenga el texto llano. Para que esto última se cumpla y tenga sentido la clave se repetirá las veces necesarias para crear el patrón antes mencionado en la explicación anterior. Un ejemplo



De esa manera será ordenada la matriz cifrada para luego recorrerla en una sola dirección para obtener el nuevo texto cifrado. Y Así hacer la encriptación de forma que cumpla con todos los requerimientos del proyecto y a su vez generará un archivo con el texto cifrado. Cada índice representa el índice de la columna que se tomará para generar el texto cifrado.



Ejemplo de cómo recorrerá la matriz cifrada por el patrón de la clave

Resultados:

Para finalizar este fue el texto llano que se escogió de la biblia para luego ser cifrado por el algoritmo implementado para el proyecto:

```
La transfiguracion:
De cierto os digo que hay algunos de los que estan aqui, que de ningun
modo gustaran la muerte hasta que hayan visto al Hijo del Hombre viniendo
en su reino.
Y despues de seis dias, Jesus toma consigo a Pedro, a Jacobo y a su
hermano Juan, y los lleva aparte, a un monte muy alto.
Y fue transfigurado ante ellos, y su rostro resplandecio como el sol y sus
vestiduras se hicieron blancas como la luz.
Y he aqui se les aparecieron Moises y Elias hablando con El.
Entonces intervino Pedro y dijo a Jesus:
Señor bueno es quedarnos aqui Si quieres, hare aqui tres enramadas:
Una para ti, una para Moises, y otra para Elias.
Estando el aun hablando, he aqui una nube de luz los cubrio, y de la nube
salio una voz, diciendo:
Este es mi Hijo amado, en quien me he complacido; a El oid.
Y los discipulos, al oirlo, cayeron sobre sus rostros y temieron en gran
manera.
```

Dicho texto ya fue depurado para el uso del programa que ya fue implementado. De esta forma el texto cuenta con 880 caracteres incluyendo los espacios y saltos de líneas para el uso del programa y para el cumplimiento de los requerimientos del proyecto. El texto se encuentra en la página 24 del .pdf del nuevo testamento de la biblia que fue facilitada por el profesor a través del correo.

Luego tenemos el texto cifrado después de haber sido empleado el algoritmo antes explicado:

```
L ysueie.a  sorsdso.rsoybiipe
de , m ,benonqoalessrraugucin oai eu iod lbeacYisnrrg o  ndJehe f
oslhebsjo e yahzncoe. s  iaa ttvY, ultn cv YchcdeStr,s,ladi iaeos ou s
npuom yu odcannnasie,toaoenmo r  telnmeoi  P lesyieb iaeinira t u ijhdl
nac tnrs
sas na,e n
e n u  asEo l Heo rratuadhaieedevmis taerl o qst ne ui
```

```
ouefosigtHdus,aa rrmuaq dt  en r qs damllrgeaeguvroioaomtonurzaat  uu
s.ndezi lsoig d s j staoalds a io reuean lica:ols,saoq e a  eob,a
eelhleylPSor:aEhn,n q;iryecoo a es  ca,
n seo s o:rhaaanai tndsysari uau ndmJJto rl ossoiud,a aaubiE cdcr nuq
lyHrino  f s ias .eese
 lau aeu po raglered eaaue.ao smeennsa dprunroseiiaomudeqtqoe o  rtotesc
icvsesmap  ul
,a  tni sdnhluscon Ytro  l E
nasr   oue oce n:eunaaoess yuuepyc  E
dÃ±  UMibby siauntais ,uaioe  npaaoorsuMoeJqrruaeu sodpooorDh nm bndg l
laseupinorqqaisa domnEosm

 ei nmi iy nell ilalEroaanoale v e l e.

*
```

Este archivo también contiene los 880 caracteres sin contar el * que nos indica el fin de archivo tal cual como se pide en las condiciones del proyecto.

A continuación, está el código realizado para la presentación de la primera parte del proyecto:

```java
/*encriptando mensaje por medio del algoritmo de transposicion By: Vanessa Cruz*/
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

public class encriptado {
    public static void main (String args[]) throws IOException {
        BufferedReader entrada;
        String linea, texto="", clave="Ñ3hOXLsFQy", textdepurado="";
        int dim, columna, fila;
        char O[][];

        //clave en numeros ASCII: |165|51|104|79|88|76|115|70|81|121|

        entrada= new BufferedReader(new FileReader("textollano.txt"));
        FileWriter archsalida = new FileWriter("textoCifrado.txt");//archivo de salida para el cifrado...
        PrintWriter apt = new PrintWriter(archsalida);

        //Leyendo archivo del texto llano.
        while((linea = entrada.readLine())!=null) texto = texto + linea;
        entrada.close();
        //Agregan saltos de lineas para depurar el texto tal cual como sale en el archivo...
        for(int k=0; k<texto.length(); k++){
            if(texto.charAt(k)=='.'||texto.charAt(k)==':'){
                if(texto.charAt(k)=='.') textdepurado= textdepurado + ".\n";
                else textdepurado= textdepurado + ":\n";
            }else{
                textdepurado= textdepurado + texto.charAt(k);
            }
        }

        /*System.out.println("TEXTO LLANO DEPURADO: ");
        System.out.println(textdepurado);
        */

        texto=textdepurado;
```

```
37
38          texto=textdepurado;
39          //Ingresadando texto en la matriz ...!!
40          dim=texto.length();
41          columna=(dim/(clave.length()*4)); //generando un tamano para las columnas relacionado con la clave
42          fila=(dim/columna);//tamano en filas
43          O = new char[fila+1][columna+1];
44
45          /*System.out.println("\n"+dim);
46          System.out.println("\n"+columna);
47          System.out.println("\n"+fila);*/
48
49          int pos=0;
50
51          for(int i=0; i<fila; i++){
52              for(int j=0; j<columna;j++){
53                    O[i][j]=texto.charAt(pos);
54                    pos++;
55              }
56          }
57
58          //imprime matriz por pantalla
59          /*for(int i=0;i<fila;i++){
60              System.out.print("|");
61              for(int j=0;j<columna;j++){
62                  System.out.print(O[i][j]+"|");
63              }
64              System.out.print("");
65          }
66          System.out.println("");
67          */
68
```

```
65          }
66          System.out.println("");
67          */
68
69      /*System.out.println("TEXTO LLANO: ");
70      System.out.println(texto);
71      System.out.println("\n");*/
72
73          /************ALGORITMO TRANSPOSICION*************/
74          /***************GENERANDO PATRON SEGUN LA CLAVE***************/
75          //Haciendo vector de indices para la clave.
76          //SEGUN EL PATRON QUE GENERA LA CLAVE HASTA EL NUM DE COLUMNAS.
77          int movcolunm[] = new int[22];
78          int menor,k=0,p=0;
79          String claveA;
80         //se escoge a partir de la letra menor en la clave siguiendo un patron...
81         while(p<30){
82          menor=1000;
83          if(p==20) claveA="Ñ3";//rellenamos lo que falta en columnas con lo que viene del patron de la clave
84          else claveA=clave; //utilizamos una clave auxiliar siempre inicializada en la clave establecida, para generar un patron repetitivo h
85          for(int i=0; i<claveA.length(); i++){
86           for(int j=0; j<claveA.length(); j++){
87              if(menor>(int)claveA.charAt(j)&&claveA.charAt(j)!='#'){
88                  menor=(int)claveA.charAt(j);
89                  k=j+p;
90              }
91           }
92           movcolunm[i+p]=k;//vector de indices...
93           claveA=claveA.replace(claveA.charAt(k-p),'#');//marcamos la letra de menor valor para ir descartando...
94           menor=1000;//volvemos a inicializar la variable menor...
95          }
96          p=p+10;//actualizamos contador que nos dira en que patron repetitivo estaremos...
97        }
98
99          //imprime vector de indices por pantalla
100         /*for(int i=0; i<movcolunm.length; i++){
101             System.out.print(movcolunm[i]+" ");
102         }
```

```
104
105         /******GENERAMOS MATRIZ CIFRADA *******/
106         /**A PARTIR DEL VECTOR DE INDICES CON EL PATRON DE LA CLAVE********/
107         char M[][] = new char[fila+1][columna+1];
108         //Introduciendo matriz cifrada.
109         for(int i=0;i<columna;i++){
110             for(int j=0; j<fila; j++){
111                 M[j][i]=O[j][movcolunm[i]];
112             }
113         }
114
115         //imprime matriz cifrada por pantalla
116             /*for(int i=0;i<fila;i++){
117                 System.out.print("|");
118                 for(int j=0;j<columna;j++){
119                     System.out.print(M[i][j]+"|");
120                 }
121                 System.out.println("");
122             }
123             System.out.println("");
124             */
125         /**********OBTENEMOS EL TEXTO CIFRADO RECORRIENDO LA MATRIZ CIFRADA **********/
126         /******************DE ARRIBA HACIA ABAJO*************************/
127         String textoCifrado="";
128         int o, a=0;
129
130         while(textoCifrado.length()!=dim){
131             for(o=0; o<fila;o++){
132               textoCifrado=textoCifrado + M[o][a];
133             }
134             a++;
135         }
136         //Escribe texto cifrado por pantalla
137         /*System.out.println("TEXTO CIFRADO: ");
138         System.out.println(textoCifrado);
139         System.out.println("\n");*/
140
141         //Escribe texto cifrado en el archivo de salida
142         apt.println(textoCifrado);
143         apt.println("\n*");
144         apt.close();
145     }
146 }
147
```

El lenguaje de programación utilizado para este proyecto fue Java, para un mejor uso de las librerías de las cadenas y no tener problemas con el alfabeto pedido en el proyecto.

Conclusiones y recomendaciones:

Al finalizar la primera parte del proyecto, se llevó a cabo una nueva forma de implementar el algoritmo de transposición para la buena complicación que tendrá el analista de encriptamiento al momento de querer desencriptar el mensaje, (compañeros de clase). Sin embargo, debido a todas las herramientas que existen para subir el nivel de ésta complicación al momento de programar el algoritmo se sintieron muchas limitaciones las cuales pueden ser posibles debilidades a que puedan romper el cifrado creado para éste proyecto. Como recomendación se podría recrear la clave por medio de librerías de encriptado que nos facilita el lenguaje "Java" llamado keyGenerator la cual en su defecto usa el algoritmo de clave simétrica "RCA" para generar claves aleatorias, de ésta forma el patrón que generaría sería más de 16 bits y el patrón que se genera a partir de la clave empleada sería más difícil de descifrar. Así también como éste lenguaje tiene una librería para el encriptado también tiene una para el desencriptado, de la misma clave que genera la librería anterior. Para los casos que contenga la letra "ñ" sea minúscula o mayúscula se le recomienda al cifrador que guarde el archivo .txt del texto llano como formato UTF-8, para así evitar que se llene de basura la cadena dentro del programa que se va a cifrar, y se complique la salida. Otra recomendación sería permitir hacer cualquier recorrido de la matríz cifrada de forma que ya no sea un algoritmo de transposición simple sino más bien más complicaciones para que el programador no lo pueda desencriptar.

Referencias Bibliográficas:

[Redes de computadoras, 4ta Edición – Andrew S. Tanenbaum] Libro de teoría de redes de computadoras.

[https://docs.oracle.com/javase/7/docs/api/] Página oficial de Oracle de librerías para la implementación del código del proyecto.

[http://www.javahispano.org/documentacion/] Documentación de Java para desarrolladores SE.

[Redes de computadoras, 5ta Edición – Andrew S. Tanenbaum] Libro de teoría de redes de computadoras.