Cifrado de Hill Implementado en Java

Andrés Cruz Chipol
Facultad De Ciencias de la Computación - Criptografía
Benemérita Universidad Autónoma de Puebla
Veracruz, México
andres.cruzc@alumno.buap.mx

***Abstract*—Implementación del algoritmo de cifrado de Hill en java con Netbeans 8.0.**

Keywords: cripto, cifrado, hill, descifrado

I. INTRODUCCIÓN

Dentro de la criptografía clásica podemos encontrar el cifrado de Hill, un cifrado de sustitución poligráfica, es decir, un mismo signo, en este caso una misma letra, puede ser representado en un mismo mensaje con más de un carácter.

Así, en el ejemplo que vamos a analizar a continuación, la letra A del mensaje original aparece representada en el mensaje codificado de tres formas distintas, como C, K e I. está basado en algebra lineal.

Este algoritmo fue inventado por Lester S. Hill en 1929, fue el primer cifrado de su tipo que era practico para operar sobre más de tres símbolos inmediatamente.

II. LESTER S. HILL

Fue un Matemático estadounidense y docente que se intereso en las aplicaciones matemáticas de las comunicaciones recibido en Columbia College y un Doctorado en la Universidad de Yale.

Su más notable contribución fue el cifrado de Hill. También desarrollo métodos para detectar errores en números de código telegrafiado. Escribió 2 libros.

III. CIFRADO DE HILL

El cifrado consiste en asociar cada letra del alfabeto con un número. La forma mas sencilla de hacerlo es con una asociación ordenada con los números naturales, aunque se podrían realizar otro tipo de asociaciones.

Se puede utilizar el alfabeto de 27 letras, incluso también añadir otros símbolos como el espacio en blanco o el punto, la coma.

Dado el ejemplo. Trabajaremos con números enteros modulo 27, se consideran los números enteros del 0 al 26, el resto se obtiene de forma cíclica, así el 27 será el 0.

El cifrado de Hill utiliza una matriz cuadrada de números como clave, la cual determinara la transformación lineal.

Cuando tengamos nuestras matrices cuadradas podemos entonces a operar con ellas:

Para cifrar un mensaje donde K es nuestra matriz llave, M nuestra matriz del mensaje llano, y A nuestro numero del alfabeto. Usaremos la siguiente formula:

$$\boldsymbol{C = (K * M)\ mod\ |A|}$$

M = Mensaje original, ***K*** = Clave, ***C*** = Mensaje cifrado, ***A*** = *Alfabeto*

Para descifrar un mensaje donde K es nuestra lave, C nuestro mensaje cifrado y A el alfabeto Usaremos la siguiente formula:

$$\boldsymbol{M = \left(K^{-1} * C\right)\ mod\ |A|}$$

M = Mensaje original, ***K*** = Clave, ***C*** = Mensaje cifrado, ***A*** = *Alfabeto*

Con unos pasos antes de obtener nuestro mensaje original, tenemos que obtener nuestra matriz de la llave inversa de la siguiente fomula:

$$K^{-1} = [T_{ADJ(K)} * [(|A| + 1)/\det(K)]] mod\ |A|$$

Consideraciones:

1.- Nuestra Matriz
estas cuentas pueden realizar en un par de segundos, dado a la potencia de computación algunos métodos de cifrado clásico ya no son un estándar o incluso son simplemente nulos a la hora de cifrar el texto.

IV. DESAROLLO

Para poder desarrollar el algoritmo utilizamos los siguientes métodos en Java:

Nuestras funciones principales:

```
public static String encriptrar(HillMessage msj)
```

La cual recibe un objeto de tipo HillMessage la cual contiene la llave y el texto. Regresa el mensaje cifrado.

```
public static String desencriptrar(String llave, String mensaje)
```

Desencriptar solo recibe la llave y el mensaje, devolviendo el texto desencriptado.

Para poder realizar este algoritmo se necesitaron los siguientes métodos programados en Java:

Inversa: Regresa la inversa de la matriz como se proporcionó anteriormente.

getMessage: Convierte el mensaje original en una matriz para poder operar en ella

getKey: Convierte la llave en una matriz cuadrada

V. CODIGO

```
public static int[][] inversa(int[][] matriz){
    int [][] resultado = new int[2][2];
    resultado[0][0] = matriz[1][1];
    resultado[0][1] = matriz[0][1] * -1;
    resultado[1][0] = matriz[1][0] * -1;
    resultado[1][1] = matriz[0][0];
    int determinante = (matriz[0][0]*matriz[1][1]) - (matriz[0][1]*matriz[1][0]);
    int escalar = 256/determinante;
    //System.out.println("Determinante:"+determinante);

    for (int i = 0; i < resultado.length; i++) {
        for (int j = 0; j < resultado.length; j++) {
            resultado[i][j] = resultado[i][j]*escalar;
        }
    }
    for (int i = 0; i < resultado.length; i++) {
        for (int j = 0; j < resultado.length; j++) {
          //  System.out.print(resultado[i][j] + " ");
        }
       // System.out.println("");
    }
    for (int i = 0; i < resultado.length; i++) {
        for (int j = 0; j < resultado.length; j++) {
            int mod = resultado[i][j];
            if(mod >= 0){
                resultado[i][j] = resultado[i][j]%255;
            }else{
                int div = resultado[i][j]/255;
                int mul = (div*-1) * 255;
                resultado[i][j] =( resultado[i][j] + mul ) + 255;
            }
        }
    }
    for (int i = 0; i < resultado.length; i++) {
        for (int j = 0; j < resultado.length; j++) {
            //System.out.print(resultado[i][j] + " ");
        }
       // System.out.println("");
    }
    return resultado;
}
```

```
public static int [][] getMessage(String mensaje,int n) {
    //CREACION DE LA MATRIZ M
    int n2 = (int)Math.ceil((float)mensaje.length()/n);
    //System.out.println("n2 = "+n2);
    int lenMensaje = mensaje.length();
    int contadorRelleno = 0 ;
    int matrizMensaje[][] = new int [n][n2];
    //RELLENAR LA MATRIZ CON ESPACIOS
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n2; j++)
            matrizMensaje[i][j] = (int)' ';
    //RELLENAR LA MATRIZ CON EL MENSAJE
    for (int j = 0; j < n2; j++) {
        for (int i = 0; i < n; i++) {
            if(contadorRelleno < lenMensaje){
                matrizMensaje[i][j] = mensaje.charAt(contadorRelleno);
                contadorRelleno++;
            }
        }
    }
    //Visualizar la matriz
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n2; j++) {
            //System.out.print(matrizMensaje[i][j]+" ");
        }
        //System.out.println("");
    }
    //System.out.println("");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n2; j++) {
            //System.out.print((char)matrizMensaje[i][j]+" ");
        }
        //System.out.println("");
    }
    return matrizMensaje;
}
```

```
public static int[][] getKey(String llave){
    int n =(int) Math.sqrt(llave.length());
    int llaveMatriz[][] = new int[n][n];
    int contador = 0;
    //Convertir Matriz
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            llaveMatriz[i][j] = llave.charAt(contador);
            contador++;
        }

    }
    //Visualizar matriz
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            //System.out.print(llaveMatriz[i][j] +" ");
        }
        //System.out.println("");
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
           // System.out.print((char)llaveMatriz[i][j] +" ");
        }
        //System.out.println("");
    }
    return llaveMatriz;
}
```

```
class HillMessage{
    String textoClaro;
    int longitudN;
    String llave;
    public HillMessage(String m1,int m2,String m3){
        textoClaro = m1;
        longitudN = m2;
        llave = m3;
    }
}
```

```
public static String encriptrar(HillMessage msj){
   // System.out.println("\nLLave K:");
    int keyNxN[][] = getKey(msj.llave);
   // System.out.println("\nMatriz M:");
    int matrizM[][] = getMessage(msj.textoClaro,msj.longitudN);
    int n = msj.longitudN;
    int n3 = (int) Math.sqrt(msj.llave.length());
    int n2 = (int)Math.ceil((float)msj.textoClaro.length()/n);
    int matrizC[][] = new int [n][(int)Math.ceil((float)msj.textoClaro.length()/n)];
    String mensaje = "";
    int contadorRelleno = 0 ;
    int lenMensaje = msj.textoClaro.length();

    int matrizCifrada[] = new int [n*n2];
    int suma = 0;
    int contadork = 0;
    for(int i = 0; i < n2; i++){
        for(contadork = 0; contadork < n; contadork++){
            suma = 0;
            for(int j=0; j < n; j++){
                //System.out.print(matrizM[j][i] + " ->");
                //System.out.println(keyNxN[contadork][j]+ "*"+matrizM[j][i] + "="+
                suma += keyNxN[contadork][j] * matrizM[j][i] ;
            }
            //System.out.println("Suma:" + suma + " Modulo-255= " + (suma % 255) + "
            mensaje =  mensaje + (char)(suma % 255);
            //System.out.println(mensaje);
        }

        suma = 0;
    }
    return mensaje;
}
```

```
public static String desencriptrar(String llave, String mensaje){
    //Matriz = la llave inversa por C  modulo |A|
    //System.out.println("");
    int [][] matrizMensaje = getMessage(mensaje,2);
    //System.out.println("");
    int [][] key = getKey(llave);
    int [][] keyInversa = inversa(key);
    String texto = "";
    for (int i = 0; i < mensaje.length()/2; i++) {
        texto = texto + ((char)((keyInversa[0][0] * matrizMensaje[0][i] + keyInversa[0][1] * matrizMensaje[1][i])%255));
        texto = texto + ((char)((keyInversa[1][0] * matrizMensaje[0][i] + keyInversa[1][1] * matrizMensaje[1][i])%255));
    }
    return texto;
}
```

VI. PRUEBA

Llave: bfjn

Mensaje: Hola mis amigos como estan?

```
Matriz M:
72 108 32 105 32 109 103 115 99 109 32 115 97 63
111 97 109 115 97 105 111 32 111 111 101 116 110 32

H l   i   m g s c m   s a ?
o a m s a i o   o o e t n
```

```
LLave K:
98 102
106 110
b f
j n
```

Matriz Resultante:

```
18 78 229 90 25 227 251 254 114 74 178 152 71 3
207 188 82 65 37 154 178 155 9 49 222 215 197 253

↕ N å Z ↓ ã û þ r J ² ? G
Ï ¼ R A % ? ² ?         1 Þ × Å ý
```

```
Mensaje Cifrado:
↕ÏN¼åRZA↓%ã?û²þ?r        J1²Þ?×GÅ ý
```

Descifrado:

```
Determinante:-32
```

```
-880 816
848 -784
```

Matriz inversa Resultante:

```
140 51
83 236
```

```
Texto Descifrado: Hola mis amigos como estan?
PS C:\Users\andy_\Desktop\Hill>
```

VII. CONCLUSIÓN

El cifrado de Hill es una propuesta interesante para la criptografía clásica, si bien la solución de las operaciones para descifrar y cifrar el mensaje con un alfabeto reducido como el alfabeto ingles o español no obtendremos complicaciones para realizar nuestras operaciones adecuadamente.

Sin embargo en este caso utilizamos los 256 caracteres de la computadora, lo cual ocasionaba algunos errores a la hora de calcular la determinante o la inversa de la llave. Por lo que se tuvo que hacer algunas modificaciones para poder obtener nuestro resultado.

Por último podemos observar cómo podemos hacer uso de las matemáticas, en este caso el algebra lineal para poder obtener algoritmo criptográfico.

VIII. REFERENCIAS

[1] Lester S. Hill - Wikipedia

[2] Cifrado Hill - Wikipedia, la enciclopedia libre

[3] Criptografía con matrices, el cifrado de Hill — Cuaderno de Cultura Científica (culturacientifica.com)

[4] Criptosistema Hill | Textos Científicos (textoscientificos.com)

[5] Cifrado Hill - es.LinkFang.org

[6] Seguridad Informática I - Cifrado de Hill - YouTube

[7] Cifrado de Hill - YouTube

[8] Cifrado Hill – Numerentur.org