Congr. Int. en Ing. Electrónica. Mem. ELECTRO, Vol. 43 pp.120 - 124, Oct 2021, Chihuahua, Chih. México
http:// electro.itchihuahua.edu.mx/revista
ISSN 1405-2172

SEGURIDAD EN PÁGINAS WEB A TRAVÉS DEL ALGORITMO DE HILL

,
Bárbara Emma Sánchez Rinza, Luis F Enríquez Cortes, Gregorio Trinidad
Benemérita Universidad Autónoma de Puebla
Facultad de ciencias de la computación
14 sur y avenida San Claudio
Tel.,52 222 2295500
brinza@hotmail.com

RESUMEN.

En el presente trabajo se utilizará el algoritmo de Hill, para poder mantener una página web segura, por métodos criptográficos. La idea es poder aplicar esta técnica de cifrado a distintas páginas web. En este caso, el algoritmo fue aplicado a una página web especializada en "intercambios de libros y otros artículos por parte de la comunidad estudiantil" para que los datos de los usuarios dentro y fuera de la comunidad universitaria sea seguros y su almacenamiento como su tráfico de la información también.

Palabras Clave:Hill, hackeada, cifrar, descifrar

ABSTRACT.

In the present work, Hill's algorithm will be used, in order to maintain a secure web page, by cryptographic methods. The idea is to be able to apply this encryption technique to different web pages. In this case, the algorithm was applied to a web page specialized in "exchanges of books and other articles by the student community" so that the data of the users inside and outside the university community is safe and its storage as their traffic of information as well.

Keywords: Hill, hacked, encrypt, decrypt

1. INTRODUCCIÓN

La seguridad en una página web inicia en instalar un certificado SSL en la web, pero ¿por qué es tan importante hacerlo? ¿Qué puede ocurrir cuando las conexiones entre el navegador y el servidor en el que se aloja la web no son seguras? La seguridad de una página web es crucial no solo para los clientes, sino también para la reputación del negocio.

Que una página web sea segura no es nada más importante, sino que es necesario, si se quieres proteger a los clientes o usuarios de posibles ataques informáticos o robos de datos personales e información de pago. Y no solo esto, existen muchas razones por las que es imprescindible que la seguridad de la página web se vuelva una de las tareas prioritarias en una empresa online.

Si la página web es víctima de un ataque informático afectará directamente a la empresa. Cuando hackean una página o inyectan código malicioso en ella afecta directamente a la experiencia de usuario de sus visitas.

Imagina que tienes una web de reserva de hoteles, un usuario aterriza en ella y encuentra contenido malicioso que no se corresponde con lo que está buscando. Automáticamente cerrará el navegador y posiblemente no volverá a visitarte. Es decir, no solo daña su experiencia de usuario, sino también la reputación online y la credibilidad de la marca de la empresa que tiene esa página web.

Y lo más importante, si Google u otro navegador, detecta que tu web ha sido hackeada o que es potencialmente peligrosa para las visitas de esa página web, la pondrá en cuarentena y la posición en los motores de búsqueda caerá drásticamente.

Es por este motivo que este proyecto se creó, ya que hoy en día las compras en línea a nivel mundial han aumentado por la pandemia, para tal proyecto se utilizo el algoritmo de Hill.

2. ALGORITMO DE HILL

Este sistema está basado en el álgebra lineal y ha sido importante en la historia de la criptografía. Fue Inventado por Lester S. Hill en 1929, y fue el primer sistema criptográfico poli alfabético que era práctico para trabajar con más de tres símbolos simultáneamente [1,2].

Este sistema es poli alfabético pues puede darse que un mismo carácter en un mensaje a enviar se cifre en dos caracteres distintos en el mensaje cifrado.

Para explicar el algoritmo solo utilizaremos 26 caracteres, aunque el trabajo se realizó para 256 caracteres.

Las letras se numeran en orden alfabético de forma tal que A=0, B=1, ... ,Z=25

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Se elige un entero d que determina bloques de d elementos que son tratados como un vector de d dimensiones.

Se elige de forma aleatoria una matriz de d × d elementos los cuales serán la clave a utilizar.
Los elementos de la matriz de d × d serán enteros entre 0 y 25, además la matriz M debe ser invertible en z_{26}^{n}
Para el cifrado, el texto es dividido en bloques de d elementos los cuales se multiplican por la matriz d × d
Todas las operaciones aritméticas se realizan en la forma módulo 26, es decir que 26=0, 27=1, 28=2 etc.
Dado un mensaje a cifrar debemos tomar bloques del mensaje de "d" caracteres y aplicar:
$M \times P_i = C$, donde C es el código cifrado para el mensaje P_i
Ejemplo:

$$A = \begin{pmatrix} 5 & 17 & 20 \\ 9 & 23 & 3 \\ 2 & 11 & 13 \end{pmatrix}$$

Si tomamos la matriz como matriz de claves.
Para cifrar el mensaje "CODIGO" debemos cifrar los seis caracteres de "CODIGO" en bloques de 3 caracteres cada uno, el primer bloque

$$P_1 = \text{"COD"} = \begin{pmatrix} 2 \\ 14 \\ 3 \end{pmatrix} \qquad P_2 = \text{"IGO"} = \begin{pmatrix} 6 \\ 8 \\ 14 \end{pmatrix}$$

$$A.P_1 = \begin{pmatrix} 5 & 17 & 20 \\ 9 & 23 & 3 \\ 2 & 11 & 13 \end{pmatrix} \begin{pmatrix} 2 \\ 14 \\ 3 \end{pmatrix} = \begin{pmatrix} 308 \\ 349 \\ 197 \end{pmatrix} = \begin{pmatrix} 22 \\ 11 \\ 15 \end{pmatrix} (\text{mod} 26)$$

El primer bloque "COD" se codificará como "WLP"

$$A.P_2 = \begin{pmatrix} 5 & 17 & 20 \\ 9 & 23 & 3 \\ 2 & 11 & 13 \end{pmatrix} \begin{pmatrix} 8 \\ 6 \\ 14 \end{pmatrix} = \begin{pmatrix} 422 \\ 252 \\ 264 \end{pmatrix} = \begin{pmatrix} 6 \\ 18 \\ 4 \end{pmatrix} (\text{mod} 26)$$

El segundo bloque "IGO" se codificará como "GSE".Luego 'CODIGO' cifrado equivale a 'WLPGSE'.
Observar que las dos "O" se codificaran de forma diferente.
Para descifrar el método es idéntico al anterior, pero usando la matriz inversa de la usada para cifrar.

2.1 Cálculo De La Matriz Inversa

Antes que nada, debemos verificar que la matriz elegida sea invertible en módulo 26. Hay una forma relativamente sencilla de averiguar esto a través del cálculo del determinante. Si el determinante de la matriz es 0 o tiene factores comunes con el módulo (en el caso de 26 los factores son 2 y 13), entonces la matriz no puede utilizarse. Al ser 2 uno de los factores de 26 muchas matrices no podrán utilizarse (no servirán todas en las que su determinante sea 0, un múltiplo de 2 o un múltiplo de 13)

Para ver si es invertible calculo el determinante de A

$$\begin{vmatrix} 5 & 17 & 20 \\ 9 & 23 & 3 \\ 2 & 11 & 13 \end{vmatrix}$$

$5\,(23 \cdot 13 - 3 \cdot 11) - 17\,(9 \cdot 13 - 3 \cdot 2) + 20\,(9 \cdot 11 - 23 \cdot 2) =$
$1215 - 1734 + 1060 = 503$
$503 = 9 \bmod 26$

La matriz A es invertible en módulo 26 ya que 26 y 9 son coprimos. Para hallar la inversa de la matriz, módulo 26, utilizamos la formula

$$A^{-1} = C^{T}\left(\det(A)\right)^{-1}$$

Donde CT es la matriz de cofactores de A transpuesta.
Hay que tener en cuenta que $\left(\det(A)\right)^{-1}$ debe realizarse en módulo 26 por lo tanto, para el ejemplo la inversa de 9 (mod 26) es 3 (mod 26) ya que
$9 \pmod{26} \cdot 3 \pmod{26} = 27 \bmod 26 = 1 \pmod{26}$

Por lo tanto 3 es la inversa multiplicativa de 9 en módulo 26
Para calcular C hay que calcular los cofactores de A

$$C_{11} = +\begin{vmatrix} 23 & 3 \\ 11 & 13 \end{vmatrix} \quad C_{12} = -\begin{vmatrix} 9 & 3 \\ 2 & 13 \end{vmatrix} \quad C_{13} = +\begin{vmatrix} 9 & 23 \\ 2 & 11 \end{vmatrix}$$

$$C_{11} = -\begin{vmatrix} 17 & 20 \\ 11 & 13 \end{vmatrix} \quad C_{12} = +\begin{vmatrix} 5 & 20 \\ 2 & 13 \end{vmatrix} \quad C_{13} = -\begin{vmatrix} 5 & 17 \\ 2 & 11 \end{vmatrix}$$

$$C_{11} = +\begin{vmatrix} 17 & 20 \\ 23 & 3 \end{vmatrix} \quad C_{12} = -\begin{vmatrix} 5 & 20 \\ 9 & 3 \end{vmatrix} \quad C_{13} = +\begin{vmatrix} 5 & 17 \\ 9 & 23 \end{vmatrix}$$

$$C = \begin{pmatrix} 266 & -111 & 53 \\ -1 & 25 & -21 \\ -409 & 165 & -38 \end{pmatrix} \quad C = \begin{pmatrix} 266 & -1 & -409 \\ -111 & 25 & 165 \\ 53 & -21 & -38 \end{pmatrix}$$

Ahora aplicamos la fórmula de la inversa

$$A^{-1} = C^{T}(\det(A))^{-1} = \begin{pmatrix} 266 & -1 & -409 \\ -111 & 25 & 165 \\ 53 & -21 & -38 \end{pmatrix} 3$$

$$A^{-1} = \begin{pmatrix} 798 & -3 & -1227 \\ -333 & 75 & 495 \\ 159 & -63 & -114 \end{pmatrix} \quad A^{-1} = \begin{pmatrix} 18 & 23 & 21 \\ 5 & 23 & 1 \\ 3 & 15 & 16 \end{pmatrix} (\text{mod} 26)$$

Esta última es la matriz que se utiliza para des cifrar

2.2 Criptoanálisis

El sistema de Hill plantea a los criptoanalistas problemas mucho mayores a los que planteaba 'el algoritmo de CESAR'. Para empezar el espacio de claves es mucho mayor, en este caso es de 4C25, es decir las permutaciones de 4 elementos tomados de entre 25 posibles, en este trabajo se utilizó 256 caracteres. Y usando una matriz más grande la cantidad de posibles claves se puede hacer tan grande como sea necesario para hacer que sea imposible un ataque por fuerza bruta [3,4].
Lo mejor que puede hacer un criptoanalista es tratar de conseguir un código para el cual se conozca una parte del mensaje. Y ver si con ambos datos es capaz de encontrar cual fue la matriz utilizada para cifrar el mensaje.

3 IMPLEMENTACION

La implementación del cifrado se realizó en la página de inicio de un sitio web de "Intercambios de artículos en la comunidad BUAP" (Figura 1), de desarrollo propio que se encuentra en un localhost

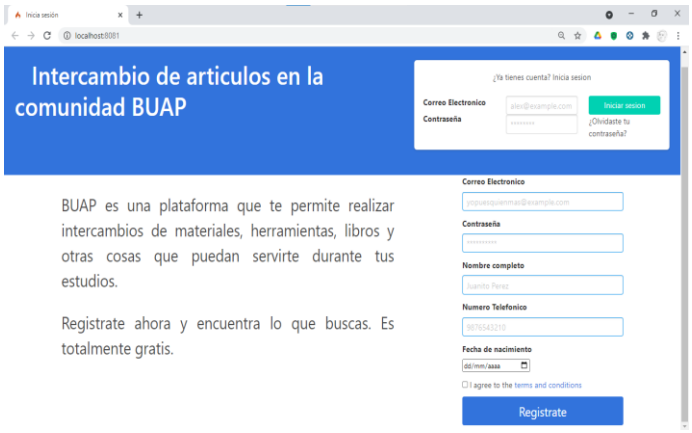

Figura. 1.TruequeoBUAP.com

El código necesario para implementar el cifrado se desarrolló en PHP. Se trata de 6 funciones; dos funciones para recorrer el alfabeto y encontrar los índices (Figura 2), dos para cifrar los mensajes dados (Figuras 3 y 4) y otras dos para descifrar [5].
La razón de tener dos funciones para que realicen el proceso de cifrado o descifrado se debe a que la función "encrypt" (Figura 3) se encarga de contar el tamaño del texto enviado y separarlo en bloques. Estos bloques pasan a la función "encryptBlock" (Figura 4) y es dentro de esta función donde se realiza todos los pasos descritos en el algoritmo de Hill [6].
El alfabeto utilizado en esta implementación fue 256 caracteres.

```php
public function indexOfChar($c){
    $allChar="ABCDEFGHIJKLMNÑOPQRSTUVWXYZ <>/¿?!,.@";
    $vari = mb_strpos($allChar, $c);
    if ($vari !== false){
        return $vari;
    }
    else{
        $vari = -1;
        return $vari;
    }
}

public function charAtIndex($pos){
    $allChar="ABCDEFGHIJKLMNÑOPQRSTUVWXYZ <>/¿?!,.@";
    return mb_substr($allChar, $pos, 1);
}
```

Figura 2.Funciones Index

```php
public function encryptBlock($plain){
    $allChar="ABCDEFGHIJKLMNÑOPQRSTUVWXYZ <>/¿?!,.@";
    $key = array(array(3,10,20), array(20,19,17), array(23, 78,17));

    $block = count($key[0]);

    $sum=0;
    $plain= mb_strtoupper($plain);
    $cipher="";

    for($i=0; $i<$block; $i++)
    {
        $a[$i][0]=$this->indexOfChar(mb_substr($plain, $i, 1));
    }


    for($i=0; $i<$block; $i++){
     for($j=0; $j<1; $j++){
        for($k=0; $k<$block; $k++){
            $sum=$sum+$key[$i][$k]*$a[$k][$j];
        }
        $cipherMatrix[$i][$j] = $sum % mb_strlen($allChar);
        $sum = 0;
     }
    }

    for($i=0; $i<$block; $i++){
     $cipher= ($cipher . $this->charAtIndex($cipherMatrix[$i][0]));
    }
    return $cipher;
}
```

Figura 3.Función de cifrado por bloque

```php
public function encrypt($plainText){
    $allChar="ABCDEFGHIJKLMNÑOPQRSTUVWXYZ <>/¿?!,.@";

    $key = array(array(3,10,20), array(20,19,17), array(23, 78,17));

    $block = count($key[0]);

    $cipherText="";
    //keyInsert(block);

    $plainText= mb_strtoupper($plainText);

    $len= mb_strlen($plainText);
      // System.out.println(plainText.substring(1,2+1));
    while($len%$block!==0){
        $plainText= ($plainText."X");
        //System.out.println(len);
        $len= mb_strlen($plainText);
    }

   for($i=0; $i<$len-1;$i=$i+$block){
    $cipherText=($cipherText . $this->encryptBlock(mb_substr($plainText, $i, $block)));
    //$cipherText=($cipherText." ");
   }
   return $cipherText;
}
```

Figura 4. Función de cifrado

Para mantener más corto el alfabeto todas las letras del mensaje a cifrar fueron convertidas a mayúsculas (Figura 5), esto no afecta gravemente al cifrado, ni al mensaje [7.8].

```php
$plain= mb_strtoupper($plain);
```

Figura 5.Función para texto a mayúscula

El texto a cifrar es enviado desde el esqueleto HTML de la página a las funciones anteriormente creadas, dando como resultado una página web muy diferente a la que teníamos a inicio, ahora está cifrada por el algoritmo de Hill.

```
<p><?php
$label7 = "Registrate ahora y encuentra lo que buscas. Es totalmente gratis";
$guarda = $hola->encrypt($label7);
echo($guarda);
//echo($hola->decrypt($guarda))?>
</p>
</div>
```

Figura 6.Texto a cifrar

Para poder descifrar la página web es necesario ingresar la matriz inversa de las claves en la función decryptBlock (Figura. 7), donde la variable "$keyin" es la matriz inversa.

```php
public function decryptBlock($cipher){
    $allChar="ABCDEFGHIJKLMNÑOPQRSTUVWXYZ <>/¿?!,.@";
    //$key = array(array(5,15,18), array(20,0,11), array(4, 26,0));
    //$keyin = array(array(23,9,21), array(11,9,2), array(22, 23,6));
    $key = array(array(3,10,20), array(20,19,17), array(23, 78,17));
    $keyin = array(array(8,32,13), array(9,4,5), array(11,8,27));
```

Figura 7. Función de descifrado

Para poder obtener la matriz inversa se utilizó un script realizado en Python (Figura 8), debido a que en ese lenguaje de programación existe una librería llamada "egcd" que agiliza el proceso de cálculo de la matriz inversa

```python
def matrix_mod_inv(matrix, modulus):
    """We find the matrix modulus inverse by
    Step 1) Find determinant
    Step 2) Find determinant value in a specific modulus (usually length of alphabet)
    Step 3) Take that det_inv times the det*inverted matrix (this will then be the adjoint) in mod 26
    """

    det = int(np.round(np.linalg.det(matrix)))  # Step 1)
    det_inv = egcd(det, modulus)[1] % modulus  # Step 2)
    matrix_modulus_inv = (
        det_inv * np.round(det * np.linalg.inv(matrix)).astype(int) % modulus
    )  # Step 3)

    return matrix_modulus_inv
```

Figura 8.Script Python

4 RESULTADOS

A continuación, se muestran los resultados obtenidos después de implementar el cifrado de Hill a nuestra página web.

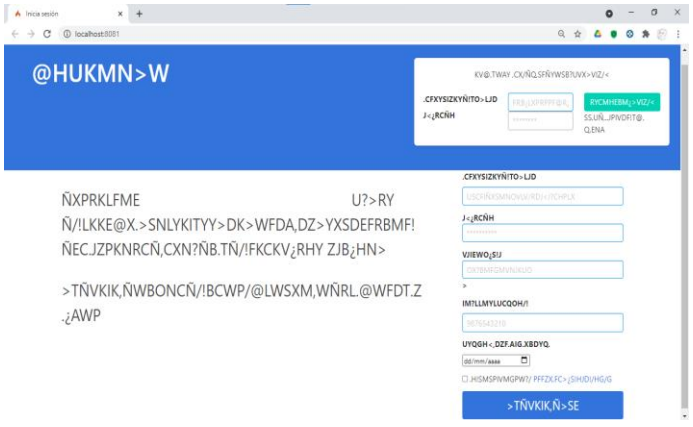

Figura 9 TruequeBUAP.com cifrada

La razón de solo cifrar los textos de la página web en lugar de todo el script HTML se debe a que el intérprete de los

navegadores web no podría descifrar el contenido y no se visualizaría nada en pantalla. Además, al descifrar pueden perderse algunas propiedades del CSS.
Si revisamos el código fuente a pesar de poder ver las etiquetas propias de HTML no podremos identificar cuál es el texto que se encuentra cifrado.

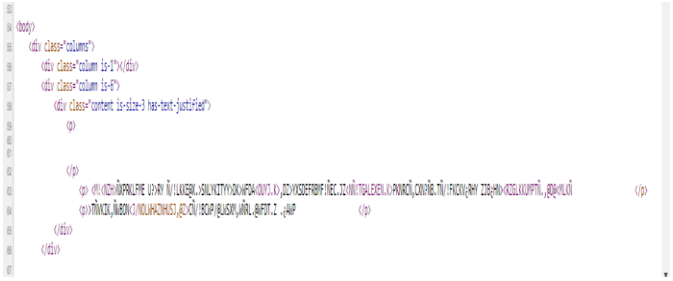

Figura 10 código fuente

Cuando se descifra la página web en un proceso similar al del cifrado, solo que llamando a la función de descifrado se tiene como resultado una página web con algunas variaciones en las palabras, añadiendo algunas "X" al final de ellas.

Figura 11 TruequeBuap.com descifrada

La razón de las "X" al final de algunas palabras como "GRATIS", "SESION" o "REGISTRATE" se debe a la naturaleza del algoritmo. El cifrado de Hill utiliza matrices cuadradas y en caso de que una palabra no cumpla con esa propiedad se le añade una letra "X" al final hasta que cumpla con la propiedad para ser multiplicada [9].

5 CONCLUSIONES

Este algoritmo de cifrado tiene la ventaja de poder cambiar el alfabeto, añadir o eliminar elementos de él, lo que hace añade más dificultad al momento de querer descifrar el mensaje. Incluso es necesario encontrar una matriz específica para cada tipo de clave.
La desventaja de esta técnica de cifrado es que se basa en matrices cuadradas. Añadir letras extras al final de las palabras para que puedan ser multiplicadas por la matriz de claves altera el significado del texto una vez que se ha descifrado, aún es posible entender el mensaje, pero es necesario realizar un ejercicio para evitar notar las letras que no deben estar ahí.
Y si se quiere aplicar este algoritmo en una página ya elaborada para la validación de contraseñas se tiene el mismo problema, dado que esas letras extra harían que no coincidan la contraseña ingresada con la que este guardada en la base de datos. La solución de este problema está en implementar este algoritmo desde la creación del sitio web o actualizando todas las contraseñas guardadas en la base de datos.

REFERENCIAS

[1] Stinson, Douglas R., Paterson, Maura B. Cryptography : theory and practice. Cuarta Edición. CRC Press, Taylor & Francis Group. 2018. ISBN: 9781138197015.
[2] Stallings, William. Cryptography and Network Security Principles and Practices. Cuarta Edición. Prentice Hall. 2005. ISBN: 9780131873162
[3] Upadhyay, Isha. JIGSAWACADEMY. Hill Cipher: A Comprehensive Guide (2021). 5 Enero 2021. https://www.jigsawacademy.com/blogs/cyber-security/hill-cipher/
[4] Rubén Daniel Varela Velasco. (10 de julio de 2006). Criptografía, una necesidad moderna. Revista Digital Universitaria, Número 7, Volumen 7, 1067-6079.
[5] Gibrán Granados Paredes. (10 de julio de 2006). Introducción a la criptografía. Revista Digital Universitaria, Número 7, Volumen 7, 1067-6079.
[6]Manuel J. Prieto. (2020). Historia de la criptografia. Madrid: La Esfera de los Libros, 978-84-9164-737-9.
[7] Gabriel Sánchez Cano. (2018). Seguridad cibernética. Hackeo ético y programación defensiva. México: Alfaomega, 978-607-538-294-4.
[8] David Moisés Terán Pérez. (2018). Administración y seguridad en redes de computadoras. México: Alfaomega, 978-607-538-132-9.
[9] Christof Paar, Jan Pelzl. (2010). Understanding Cryptography. A textbook for studenst and practtitioners. Alemania: Springer, 978-3-642-04100-6.