Fase final: Sistema de comunicación de texto (chat), de forma segura orientado apersonas con discapacidad visual*

Oscar Jose Urizar Orozco, 201504352,[1,**] Nestor Eduardo de León Aguilón, 201906466,[1,***] and Héctor Fernando Carrera Soto, 201700923[1,****]

[1] *Facultad de Ingeniería, Departamento de Física, Universidad de San Carlos, Edificio T1, Ciudad Universitaria, Zona 12, Guatemala.*

Esta Fase contiene los avances del proyecto de Laboratorio del curso de Comunicaciones 2, el cual va dirigido a personas que carecen de la capacidad visual. En la cual se presentan los avances de Cifrado Hill para asegurar la comunicación entre dos personas, y facilitar así la detección de errores que se puedan generar al momento de transmitir los mensajes por medio del código Hamming. Un teclado braille será el dispositivo de entrada que permite representar cualquier carácter mediante la pulsación simultánea de unas pocas teclas. Se asociara cada letra del alfabeto con un número. La forma más sencilla de hacerlo es con la asociación natural ordenada, aunque podrían realizarse otras asociaciones diferentes. Para evitar posibles errores se determinara la probabilidad de error con el Código de Hamming, Este proceso es, por así decirlo, una forma de detección. En donde existen bits de redundancia". Por ultimo el funcionamiento del chat esta basado en la utilización de Sockets sin embargo para esta fase aun se encuentra en proceso de mejoras.

I. DIAGRAMA DE BLOQUES

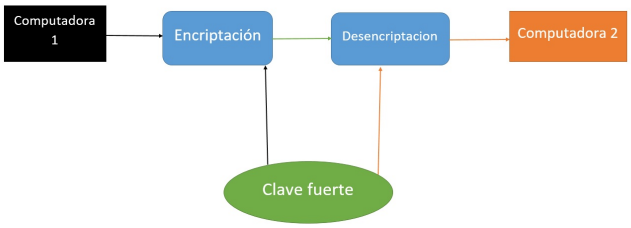

Figura 1: Diagrama de bloques básico

Computadora 1: Es la fuente del mensaje
Encriptación: es la que codifica el mensaje para que no pueda ser leído fácilmente
Clave fuerte: aquí es donde se encuentra la clave del cifrado, depende que tipo sea, será una clave diferente
Desencriptacion: aquí es donde se "traduce" el código mandando con ayuda de la clave
Computadora 2: es el receptor del mensaje original

II. MARCO TEÓRICO

A. Sistema Braile

Es un medio para la lectura y escritura empleado por personas con discapacidad visual. también llamado como cecografía y fue ideado por Luis Braille. Es un sistema constituido por celdas en las cuales se encuentran un conjunto de puntos en relieve, los cuales se encuentras distribuidos en columnas paralelas de tres o cuatro puntos. Además cada punto Braille corresponde a un número, cada una de las posiciones forman diferentes combinaciones logrando representar las letras del alfabeto, números

B. Cifrado Cesar

El Cifrado Cesar es un tipo de Cifrado de sustitución, en el que cada letra del texto simple es reemplazada por otra letra en algunas posiciones fijas de la letra actual del alfabeto.

Por ejemplo, si desplazamos cada letra en tres posiciones a la derecha, cada una de las letras de nuestro texto plano será reemplazada por una letra en tres posiciones a la derecha de la letra del texto plano.

Veamos esto en acción – encriptemos el texto "HELLO WORLD" usando un desplazamiento a la derecha de 3.

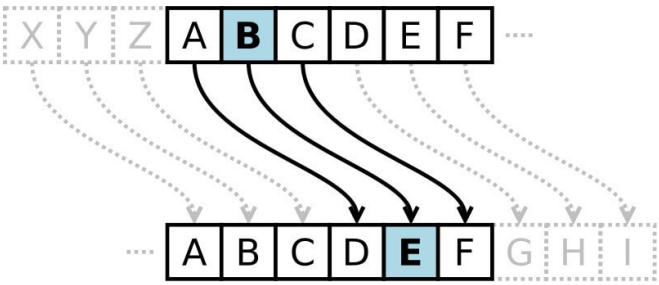

Figura 2: Explicación gráfica del cifrado cesar.
Fuente: `https://n9.cl/hjckl`.

* Laboratorios de Física
** e-mail: 2456240150101@ingenieria.usac.edu.gt
*** e-mail: 3095454391210@ingenieria.usac.edu.gt
**** e-mail: 3505043180101@ingenieria.usac.edu.gt

1. Código del cifrado Cesar

Enlace al repositrio: Cesar.py

C. Codigo Hamming

En informática, el código de Hamming es un código detector y corrector de errores que lleva el nombre de su inventor, Richard Hamming. En los datos codificados en Hamming se pueden detectar errores en un bit y corregirlos, sin embargo, no se distingue entre errores de dos bits y de un bit (para lo que se usa Hamming extendido). Esto representa una mejora respecto a los códigos con bit de paridad, que pueden detectar errores en solo un bit, pero no pueden corregirlo.

Si se añaden junto al mensaje más bits detectores-correctores de error y si esos bits se pueden ordenar de modo que diferentes bits de error producen diferentes resultados, entonces los bits erróneos podrían ser identificados. En un conjunto de siete bits, hay solo siete posibles errores de bit, por lo que con tres bits de control de error se podría especificar, además de que ocurrió un error, en qué bit fue.

Hamming estudió los esquemas de codificación existentes, incluido el de dos entre cinco, y generalizó sus conclusiones. Para empezar, desarrolló una nomenclatura para describir el sistema, incluyendo el número de los bits de datos y el de los bits detectores-correctores de error en un bloque. Por ejemplo, la paridad incluye un solo bit para cualquier palabra de datos, así que las palabras del Código ASCII que son de siete bits, Hamming las describía como un código (8.7), esto es, un total de 8 bits de los cuales 7 son datos. con base a la anterior repetición, sería un código (3.1), siguiendo la misma lógica. La relación de la información es el segundo número dividido por el primero, por nuestro ejemplo de la repetición, 1/3.

Hamming también estudió los problemas que surgían al cambiar dos o más bits a la vez y describió esto como "distancia"(ahora llamada distancia de Hamming en su honor). La paridad tiene una distancia de 2, dado que cualquier error en dos bits no será detectado. La repetición (3.1)tiene una distancia de 3, pues son necesarios el cambio simultáneo de tres bits para obtener otra palabra de código. La repetición (4.1) (cada bit se repite cuatro veces) tiene una distancia de 4, así que el cambio de dos bits en el mismo grupo quedará sin definir.

D. Cifrado Hill

En criptografía clásica, el Cifrado Hill es un cifrado de sustitución poligráfica basado en el álgebra lineal. Inventado por Lester S. Hill en 1929, fue el primer cifrado poligráfico que era práctico para operar sobre más de tres símbolos inmediatamente. El cifrado de Hill fue inventado, basándose en el álgebra lineal.

1. En que consiste el Cifrado Hill

Expliquemos en qué consiste el cifrado de Hill. En primer lugar, se asocia cada letra del alfabeto con un número. La forma más sencilla de hacerlo es con la asociación natural ordenada, aunque podrían realizarse otras asociaciones diferentes. Además, en este ejemplo solamente vamos a utilizar las 27 letras del alfabeto, pero también podrían añadirse otros símbolos usuales, como el espacio en blanco " ", el punto "." o la coma ",", la interrogación "?", las 10 cifras básicas, etcétera.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13
Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	
14	15	16	17	18	19	20	21	22	23	24	25	26	

Figura 3: Cada letra corresponde a un valor de manera ordenada, sin embargo puede cambiar el orden y agregar mas carcteres

Fuente: `https://culturacientifica.com/2017/01/11/criptografia-matrices-cifrado-hill/`.

La matriz usada para la encriptación es la llave de cifrado, y tiene que ser escogida aleatoriamente del conjunto de matrices invertibles n×n (modular 27). El cifrado puede naturalmente, ser adaptado a un alfabeto representado con cualquier orden numérico y/o cambiando el número (modular 27) siempre y cuando la matriz n×n (modular x) sea invertible. En referencia un ejemplo en lenguaje Java con alfabeto en parámetro que utiliza todo el ASCII.

Como en la correspondencia anterior, entre letras/signos y números, solamente aparecen 27 números, hay que trabajar con los números enteros "módulo 27". Es decir, se consideran los números enteros 0, 1, 2,... , 26 y el resto se identifica con estos de forma cíclica. Así, el 27 es igual a 0, el 28 a 1, el 29 a 2, etcétera, y lo mismo con los números negativos, de forma que – 1 es igual 26, – 2 es igual 25, etc. Además, se reducen las operaciones aritméticas (suma, resta, multiplicación y división) al conjunto de los números enteros módulo 27 de forma natural, es decir, al operar dos números enteros (módulo 27) el resultado se considera también módulo 27. Por ejemplo, si se realiza la multiplicación de los números 6 y 13, módulo 27, el resultado dará 24 (módulo 27), puesto que 6 * 13 = 78 y 78 = 2 * 27 + 24. O el inverso de 2, es decir, el número a tal que 2 * a es igual a 1 (módulo 27), es 14, puesto que 2 *14 = 28, que es igual a 1, módulo 27.

2. Ejemplo

"CUADERNO DE CULTURA CIENTIFICA",

cuya transcripción numérica, teniendo en cuanta la tabla de sustitución anterior, es “2, 21, 0, 3, 4, 18, 13, 15, 3, 4, 2, 21, 11, 20, 21, 18, 0, 2, 8, 4, 13, 20, 8, 5, 8, 2, 0”. Como la transformación lineal es de orden 3, vamos a agrupar los números en grupos de tres, en ternas, sobre las que luego aplicaremos la transformación lineal, (2, 21, 0), (3, 4, 18), (13, 15, 3), (4, 2, 21), (11, 20, 21), (18, 0, 2), (8, 4, 13), (20, 8, 5), (8, 2, 0).

A continuación, vamos a transformar las ternas de números anteriores, mediante la transformación lineal dada por la clave, en nuevas ternas, que serán el mensaje numérico cifrado. ¡Ojo!, que en la transformación lineal no hay que olvidar que seguimos trabajando con los números enteros módulo 27.

$$\begin{pmatrix}1&2&3\\0&4&5\\1&0&6\end{pmatrix}\cdot\begin{pmatrix}2\\21\\0\end{pmatrix}=\begin{pmatrix}44\\84\\2\end{pmatrix}\equiv\begin{pmatrix}17\\3\\2\end{pmatrix}$$
$$\begin{pmatrix}1&2&3\\0&4&5\\1&0&6\end{pmatrix}\cdot\begin{pmatrix}3\\4\\18\end{pmatrix}=\begin{pmatrix}65\\106\\111\end{pmatrix}\equiv\begin{pmatrix}11\\25\\3\end{pmatrix}$$
$$\begin{pmatrix}1&2&3\\0&4&5\\1&0&6\end{pmatrix}\cdot\begin{pmatrix}13\\15\\3\end{pmatrix}=\begin{pmatrix}52\\75\\31\end{pmatrix}\equiv\begin{pmatrix}25\\21\\4\end{pmatrix}$$

Figura 4: Procedimiento
Fuente: https://culturacientifica.com/2017/01/11/criptografia-matrices-cifrado-hill/.

Aunque la transformación lineal de la terna (2, 21, 0) es inicialmente (44, 84, 2), como estamos trabajando con enteros módulo 27, esta terna se convierte en (17, 3, 2), ya que 44 = 1 x 27 + 17 y 84 = 3 x 27 + 3. E igual para el resto.

$$\begin{pmatrix}1&2&3\\0&4&5\\1&0&6\end{pmatrix}\cdot\begin{pmatrix}4\\2\\21\end{pmatrix}=\begin{pmatrix}71\\113\\130\end{pmatrix}\equiv\begin{pmatrix}17\\5\\22\end{pmatrix}$$
$$\begin{pmatrix}1&2&3\\0&4&5\\1&0&6\end{pmatrix}\cdot\begin{pmatrix}11\\20\\21\end{pmatrix}=\begin{pmatrix}114\\185\\137\end{pmatrix}\equiv\begin{pmatrix}6\\23\\2\end{pmatrix}$$
$$\begin{pmatrix}1&2&3\\0&4&5\\1&0&6\end{pmatrix}\cdot\begin{pmatrix}18\\0\\2\end{pmatrix}=\begin{pmatrix}24\\10\\30\end{pmatrix}\equiv\begin{pmatrix}24\\10\\3\end{pmatrix}$$
$$\begin{pmatrix}1&2&3\\0&4&5\\1&0&6\end{pmatrix}\cdot\begin{pmatrix}8\\4\\13\end{pmatrix}=\begin{pmatrix}55\\81\\86\end{pmatrix}\equiv\begin{pmatrix}1\\0\\5\end{pmatrix}$$
$$\begin{pmatrix}1&2&3\\0&4&5\\1&0&6\end{pmatrix}\cdot\begin{pmatrix}20\\8\\5\end{pmatrix}=\begin{pmatrix}51\\57\\50\end{pmatrix}\equiv\begin{pmatrix}24\\3\\23\end{pmatrix}$$
$$\begin{pmatrix}1&2&3\\0&4&5\\1&0&6\end{pmatrix}\cdot\begin{pmatrix}8\\2\\0\end{pmatrix}=\begin{pmatrix}12\\8\\8\end{pmatrix}$$

Figura 5: Procedimiento
Fuente: https://culturacientifica.com/2017/01/11/criptografia-matrices-cifrado-hill/.

Por lo tanto, el mensaje numérico cifrado es “17, 3, 2, 11, 25, 3, 25, 21, 4, 17, 5, 22, 6, 23, 2, 24, 10, 3, 1, 0, 5, 24, 3, 23, 12, 8, 8”, que al transformar de nuevo los números en sus correspondientes letras, se convierte en el mensaje cifrado «QDCLYDYUEQFVGWCXKDBAFXDWMII”. Y este es el mensaje que se envía para que no sea comprendido por el “enemigo” aunque este lo intercepte en el camino. Para poder decodificar los mensajes cifrados mediante el método de Hill se necesita que la matriz de la transformación lineal utilizada, la clave, sea una matriz inversible. La matriz del ejemplo lo es, puesto que su determinante es no nulo, |A| = 22. Además, la matriz inversa de A, que es la necesaria para decodificar un mensaje cifrado, es

$$A^{-1}=\begin{pmatrix}\frac{24}{22}&\frac{-12}{22}&\frac{-2}{22}\\\frac{5}{22}&\frac{3}{22}&\frac{-5}{22}\\\frac{-4}{22}&\frac{2}{22}&\frac{4}{22}\end{pmatrix}$$

Figura 6: Procedimiento
Fuente: https://culturacientifica.com/2017/01/11/criptografia-matrices-cifrado-hill/.

Pero ojo, se está trabajando con los enteros módulo 27 y se va a transformar la matriz inversa anterior en una matriz con números enteros módulo 27. Para empezar, se

necesita el inverso del número 22. Se ve fácilmente que 22*16 = 352, que es igual a 1, módulo 27, luego 1/22 = 16. Y la matriz inversa se transforma, módulo 27.

$$A^{-1} = \begin{pmatrix} \frac{24}{22} & \frac{-12}{22} & \frac{-2}{22} \\ \frac{5}{22} & \frac{3}{22} & \frac{-5}{22} \\ \frac{-4}{22} & \frac{2}{22} & \frac{4}{22} \end{pmatrix} = \begin{pmatrix} 24\times16 & -12\times16 & -2\times16 \\ 5\times16 & 3\times16 & -5\times16 \\ -4\times16 & 2\times16 & 4\times16 \end{pmatrix}$$

$$= \begin{pmatrix} 384 & -192 & -32 \\ 80 & 48 & -80 \\ -64 & 32 & 64 \end{pmatrix} = \begin{pmatrix} 6 & 24 & 22 \\ 26 & 21 & 1 \\ 17 & 5 & 10 \end{pmatrix}$$

Figura 7: Procedimiento
Fuente: `https://culturacientifica.com/2017/01/11/criptografia-matrices-cifrado-hill/`.

E. Cifrado propio

Este proyecto de Python se centrará en el cifrado de clave simétrica. Es decir, se utilizará la misma clave para cifrar y descifrar el mensaje. En otras palabras, tanto el remitente como el receptor utilizarán la misma tabla para cifrar y descifrar el mensaje. Se trata de una forma sencilla y ligera de empezar a trabajar con cifrado. La clave que usaremos es una versión ligera de una codificación de caracteres UTF-8.

- **Detalles clave :** cubre más de 990 caracteres, incluido el alfabeto latino tanto en mayúsculas como en minúsculas, números, puntuación y los símbolos especiales más utilizados.

- **Mapeo de caracteres-bytes :** dentro de una tabla de claves, tenemos nuestro mapeo listo para usar la equivalencia entre caracteres y bytes, teniendo un número como equivalencia de letras y símbolos especiales.

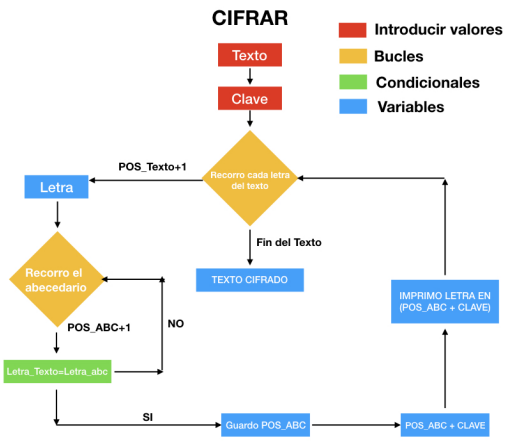

Figura 8: Explicación gráfica del cifrado propio.

III. RESULTADOS

A. Sistema de chat encriptado

Figura 9: Sistema de chat en funcionamiento
Fuente: Elaboración propia

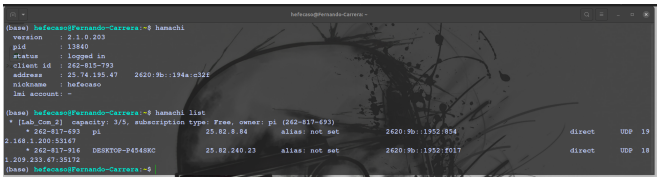

Figura 10: Lista de comunicación entre dispositivos externos a la red.
Fuente: Elaboración propia

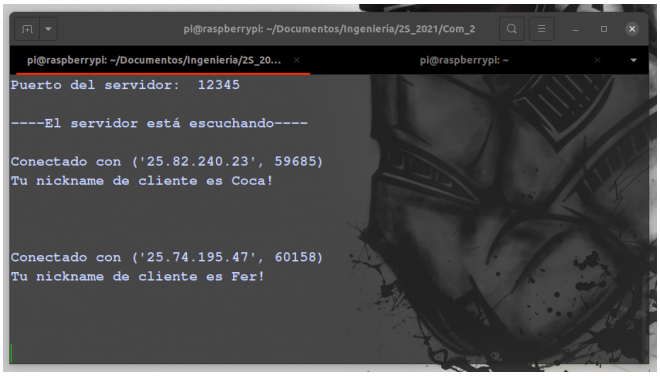

Figura 11: Shell de comunicación vía ssh al host (Raspberry pi 4).

Fuente: Elaboración propia

Figura 12: Sistema de chat encriptado, en funcionamiento.

Fuente: Elaboración propia

B. Teclado braile

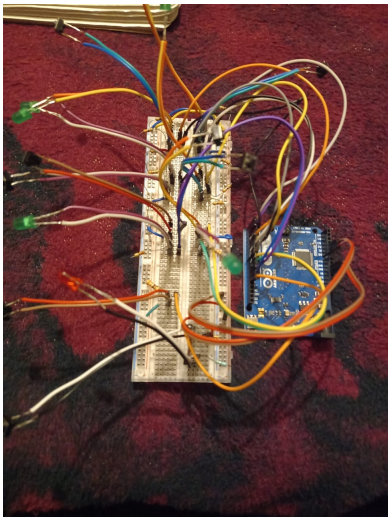

Figura 13: Fabricación de un teclado braile (Prototipo).

Fuente: Elaboración propia

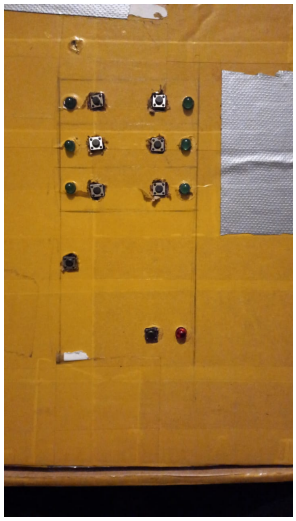

Figura 14: Fabricación de un teclado braile, resultados finales.

Fuente: Elaboración propia

C. Repositorio del código

Enlace al repositorio: https://n9.cl/bvysk

IV. DISCUSIÓN DE RESULTADOS

I. En la figura 9, pudimos observar el sistema de chat encriptado en funcionamiento óptimo, en el cual constaba de tres terminales. En la figura 11 observamos el host en funcionamiento, en el cuál se le colocó una IP pública para la raspberry, permitiendo a otras computadoras conectarse externamente de la red, usando hamachi. En la figura 12 observamos a las computadoras que entra como cliente, en el cual podemos observar a dos usuarios conectados, estos también se reflejaron en la terminal de la figura 11, la lista de computadoras conectadas se a la raspberry se mostraba en la terminal de la figura 10. Nunca se presentó problemas al realizar la comunicación entre dispositivos.

II. En la figura 13 se puede apreciar el prototipo del teclado Braille el cual fue elaborado a partir de un Arduino Leonardo, que al pulsar ciertas combinaciones permite la escritura a partir del lenguaje Braille, por lo que los resultados deseados se cumplen dado que las personas con discapacidad visual pueden hacer uso de este para poder comunicarse con otras personas. Y dado que este desarrollo su propósito sin ningún inconveniente se determino que los objetivos propuestos quedaron plenamente cumplidos.

V. CONCLUSIONES

I. Para poder generar un sistema de chat es necesario que exista un servidor, para que los clientes puedan comunicarse entre si, ademas cabe mencionar que para que este sistema funcione en cualquier sistema operativo es necesario que cuente con una dirección IP pública.

II. La Raspberry Pi es una computadora que tiene la capacidad de ser utilizada como host, y dado que esta tienen un precio reducido es ideal para realizar pruebas o pequeños proyectos.

III. Para que la comunicación existe entre los clientes es necesario utilizar un software que cumpla con las posibilidades de realizar la conexión entre los usuarios, por lo que Hamachi al ser un software simple de utilizar, es ideal para este tipo de proyectos.

IV. El cifrado permite mantener la integridad y confidencialidad del chat de los usuarios por lo que es necesario que se implemente, en todo sistema de chat para que los usuarios puedan estar seguros.

V. Dado que por motivos de ruidos, rayos cosmicos, y otros fenomenos a los que estan expuestos los canales de comunicación y los equipos, es posible que se genere un error en los sistemas de comunicación por chat por lo que es necesario implementar metodos correctores de errores, tal es el caso del codigo Hamming.

[1] CODIFICA-ME.*Código Hamming | Detectar errores por paridad* [En linea][02 de septiembre de 2021]. Disponible en:
`https://www.codifica.me/codigo-hamming-detectar-errores-por-paridad/`

[2] DA.*Sockets* [En linea][02 de septiembre de 2021]. Disponible en:
`http://www3.uji.es/~ochera/curso_2002_2003/e52/t_sockets.pdf`