Implementación del cifrado Cesar en Matlab

García García Jonathan Eduardo
Instituto Politécnico Nacional
Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacán
Maestría en Ingeniería en Seguridad y Tecnologías de la información Ciudad de México, México
jgarciag1404@alumno.ipn.mx

I. Introducción

La función esencial de la criptografía es mantener la privacidad de la comunicación, de forma que el mensaje sea inteligible tan solo para sus destinatarios. Esta necesidad de ocultar cierta información ha estado siempre presente en la vida del ser humano, datando el primer sistema criptográfico del que se tiene constancia del siglo V a.C.

La palabra "criptografía" proviene etimológicamente del griego Kriptos (ocultar), Graphos (escritura), "ocultar la escritura". En un sentido más amplio significa aplicar alguna técnica para hacer ininteligible un mensaje. La criptografía es una herramienta muy útil cuando se desea tener seguridad informática; puede ser también entendida como un medio para garantizar las propiedades de confidencialidad, integridad y disponibilidad de los recursos de un sistema.

La criptologia consiste en permitir el intercambio de información a través de un medio de comunicación inseguro, de forma que, si la información es interceptada por un intruso, sea imposible su descifrado

Esta ciencia está dividida en dos grandes ramas:

- La criptografía: ocupada del cifrado de mensajes en clave y del diseño de criptosistemas
- El criptoanálisis: que trata de descifrar los mensajes en clave, rompiendo así el criptosistema.

II. Cifrado Cesar

El cifrado de César es un cifrado de textos planos muy antiguo, se cree que Julio César lo utilizó para dirigir mensajes confidenciales a sus generales en campañas militares en el primer siglo antes de Cristo. Se basa en un cifrado mono alfabético. En criptografía, un cifrado César se clasifica como un cifrado por sustitución en el que el alfabeto en el texto plano se desplaza por un número fijo en el alfabeto

A. Ventajas

- Uno de los métodos más fáciles de usar en criptografía y puede proporcionar una seguridad mínima a la información.
- Uno de los mejores métodos para usar si el sistema no puede usar ninguna técnica de codificación complicada

B. Desventajas

- Uso de estructura simple
- Solo puede proporcionar seguridad mínima a la información
- La frecuencia del patrón de letras proporciona una gran pista para descifrar el mensaje completo.

III. Análisis de frecuencia

En que consiste el análisis de frecuencia, en cualquier idioma tenemos unas letras más comunes que otras.

1) Se analiza la frecuencia de cada letra del idioma español (inglés o portugués) y a cada letra se le da un valor determinado en función de su frecuencia de uso.
2) Se analizan diferentes textos en español (inglés o portugués) para validar los valores anteriores
3) Una vez obtenida esta tabla se lee el texto cifrado
4) Se reconoce el texto cifrado y se hace un análisis de la frecuencia de aparición de los diferentes símbolos. Se crea una nueva tabla con estos resultados.
5) Los valores de ambas tablas se comparan luego por su frecuencia y los símbolos del texto cifrado se sustituyen por las letras del alfabeto correspondientes.

IV. Desarrollo

Para este trabajo, se requirió programar una aplicación que pueda mostrar el resultado de un texto plano tras ser cifrado con el algoritmo de Cesar.

Implementación función para cifrado y descifrado de una cadena de texto

```
%Funcion para cifrar texto
%obj = instancia de esta clase
%plainText = texto a cifrar
%offsetKey = llave de desplazamiento para el cifrado
%filterSpecialChars (true/false) = filtrar caracteres especiales del
%texto y utilizar alfabeto reducido (false) o todo el conjunto de
%caracteres ASCII (true)
function cipherText = cipher(obj,plainText,offsetKey)
    cipherText=obj.normalizedCipher(plainText,offsetKey);
end

%Funcion para descifrar texto
%obj = instancia de esta clase
%cipherText = texto a descifrar
%offsetKey = llave de desplazamiento para el cifrado
%filterSpecialChars (true/false) = filtrar caracteres especiales del
%texto y utilizar alfabeto reducido (false) o todo el conjunto de
%caracteres ASCII (true)
function plainText=decipher(obj,cipherText,offsetKey)
    %la llave de cifrado debe ser negativa para el descifrado
    invertedOffset=-1*offsetKey;
    %si se utilizara el alfabeto reducido llamar a normalizedCipher
    plainText=obj.normalizedCipher(cipherText,invertedOffset);
end

%Funcion para cifrar el texto utilizando el alfabeto reducido
%reducido especificado mas arriba
%obj = Instancia de la clase
%text = Texto a cifrar
%offset = llave para el cifrado
function rText=normalizedCipher(obj,text,offset)
    rText=""; %Variable de resultado inicializada en cadena vacia
    ASCII_LENGHT=size(obj.LanguageDefinition.Alphabet,1); % Tamano del alfabeto reducido
    NEW_LINE=obj.LanguageDefinition.Alphabet(mod(13+offset,ASCII_LENGHT)); %Caracter de salto de ...
        linea cifrado
    length=size(text,1);  %Longuitud del texto a cifrar
    for i=1:length % por cada fila del texto
 row=text{i}; %recuperar la fila de texto en la posicion i
 row=convertStringsToChars(row); %convierte la cadena de texto a un arreglo de caracteres iterable
 row_lenght=size(row,2); %longitud de caracteres
 for j=1:row_lenght %por cada caracter
     letter=row(j); %obtener el caracter en la posicion j
     if(offset>0) %si se esta cifrando el offset es positivo
  letter=obj.LanguageDefinition.normalizeChar(letter); % normalizar el texto (puede que no este ...
      normalizado)
  if(letter<0)
      continue;
  end
     end
     %encontrar la letra que corresponde en el alfabeto
     %reducido y obtener la posicion en base cero
     letter=find(ismember(obj.LanguageDefinition.Alphabet,letter),1)-1;
     letter=letter+offset; %agregar el desplazamiento de la llave de cifrado
     letter=mod(letter,ASCII_LENGHT); %modulo del valor de la letra y el tamano del diccionario
     letter=obj.LanguageDefinition.Alphabet(letter+1); %obtener la letra en la nueva posicion del ...
         alfabeto cifrado base 1
     rText=append(rText,letter); %agregar la letra al resultado del texto cifrado
 end
 if((offset>0) && (i<length)) %si se esta cifrando y aun no estamos en la utlima fila agregar un ...
     salto de linea
  rText=append(rText,NEW_LINE);
 end
    end
end
```

Para el uso de este cifrado se implemento una interfaz grafica que cuenta con las siguientes funcionalidades:

1) Selección de archivo de texto
2) Selección de idioma
3) Selección de llave de cifrado
4) Campo de texto para ingresar texto a tratar
5) Campo de texto donde se muestra el resultado del texto
6) Opciones para cifrar,Limpiar y descifrar el texto

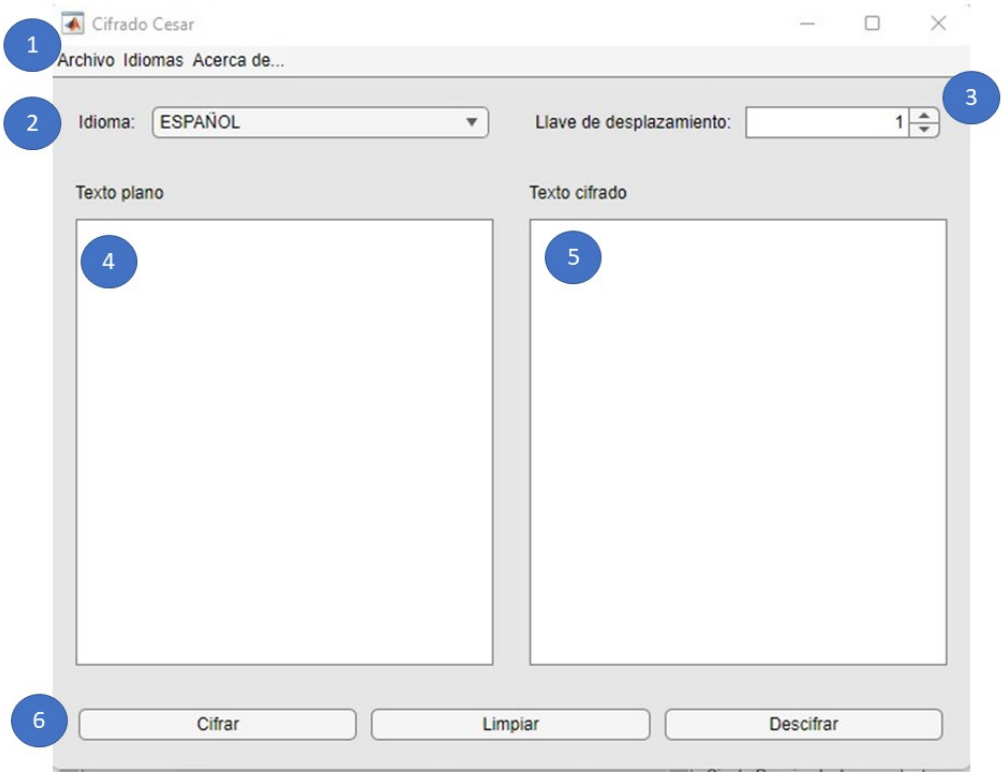

Fig. 1. GUI para el cifrado

V. Conclusiones

El método de cifrado cesar es una de las formas mas simples que tenemos de cifrar un texto y sorprende la antigüedad del mismo, en tiempos donde el análisis de frecuencias no era una realidad debido a los alcances de los medios impresos este cifrado resultó medianamente efectivo pues recordemos que aún es susceptible a ataques de fuerza bruta o de ingeniería social para conocer la llave de desplazamiento Asimismo, se deberían plantear cuestiones que resulten, al menos, interesantes para realizar como trabajos futuros.

REFERENCIAS

[1] Gómez Hernández, S. (2010). Análisis de textos cifrados de los siglos XVI y XVII (Bachelor's thesis).
[2] Paredes, G. G. (2006). Introducción a la Criptografía.