Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

IC-1801 Taller de Programación

Prof. Mauricio Avilés

Proyecto Programado 0 – Criptografía y cifrado

Motivación

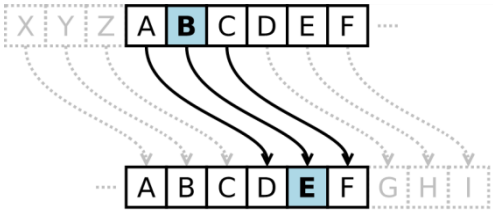
La criptografía es la técnica, ciencia o arte de la escritura secreta. El principio básico de la criptografía es mantener la privacidad de la comunicación entre dos personas, alterando el mensaje original de modo que sea incomprensible a toda persona distinta del destinatario.

Se puede decir que la criptografía es tan antigua como la civilización, cuestiones militares, religiosas o comerciales impulsaron desde tiempos remotos el uso de escrituras secretas. Los antiguos *egipcios* usaron métodos criptográficos, mientras el pueblo utilizaba la lengua demótica, los sacerdotes usaban la escritura hierática (jeroglífica) incomprensible para el resto. Los antiguos *babilonios* también utilizaron métodos criptográficos en su escritura cuneiforme.

En la actualidad existen diversos usos de técnicas criptográficas, se han extendido desde las redes de computadoras hasta el almacenamiento de la información. Algunos de los métodos básicos serán implementados como parte de este proyecto programado.

1. Cifrado César

El cifrado César, también conocido como cifrado por desplazamiento, es una técnica de codificación muy simple que fue utilizada por el dictador romano Julio César para comunicarse con sus generales. Consiste en sustituir cada letra del texto original, reemplazándola por otra letra que se encuentre un número fijo de posiciones más adelante en el alfabeto.



Ejemplo con desplazamiento = 3:

Alfabeto:	`abcdefghijklmnñopqrstuvwxyz`
Alfabeto codificación:	`defghijklmnñopqrstuvwxyzabc`
Texto:	`espero tus problemas se acaben`
Texto codificado:	`hvshur wxv sureñhodv vh dfdehp`

El mensaje cifrado se consigue adelantando tres letras cada una de las letras del mensaje, sin cambiar los espacios en blanco. Si la letra resultante es mayor que la Z, entonces debe continuarse al inicio del abecedario. El desplazamiento a utilizar es variable, puede usarse cualquier valor entero, incluso negativos.

Decodificación

Para la decodificación de un mensaje se requiere el mensaje codificado y el desplazamiento. El proceso es el mismo que con la codificación pero interpretando el desplazamiento como negativo. De esta forma, en el ejemplo anterior cada letra del mensaje codificado se desplaza tres espacios hacia la izquierda, produciendo el mensaje original nuevamente.

2. Cifrado monoalfabético con palabra clave

En este cifrado se mapea cada letra con otra letra del alfabeto. El orden del alfabeto cifrado se obtiene a partir de una palabra clave. Supongamos que se utiliza la palabra clave "`configurables`". En general, la palabra clave se usa como inicio del alfabeto y el resto del alfabeto utiliza las letras que faltan en el mismo orden en que aparecen en el alfabeto normal.

Alfabeto original: `abcdefghijklmnñopqrstuvwxyz`

Alfabeto codificación: `configurablesdhjkmñpqtvwxyz`

Esta palabra clave no contiene letras repetidas, pero en caso de que las contenga deben eliminarse. Por ejemplo, si la palabra clave fuera "`calabaza`", al eliminar las letras repetidas, la verdadera palabra clave a utilizar sería "`calbz`", y por lo tanto el alfabeto de codificación sería "`calbzdefghijkmnñopqrstuvwxy`".

Para codificar el texto se realiza la sustitución de cada letra del mensaje por la que corresponde en el alfabeto de codificación:

Texto: `formemos una banda de rocanrol`

Texto codificado: `gjñsisjp tdc ocdfc fi ñjncdñje`

Decodificación

Para la decodificación del mensaje se requiere el mensaje codificado y la palabra clave utilizada. Para obtener el mensaje original debe realizarse el proceso inverso que en la codificación. Se genera el alfabeto de codificación y se sustituye cada letra del mensaje codificado por la que se encuentra en la misma posición en el alfabeto original.

3. Cifrado Vigenère

Este cifrado también utiliza una palabra clave para realizar el proceso de cifrado. Para realizar el proceso se utiliza la siguiente correspondencia de valores en las letras del alfabeto:

a	0	f	5	k	10	o	15	t	20	y	25
b	1	g	6	l	11	p	16	u	21	z	26
c	2	h	7	m	12	q	17	v	22		
d	3	i	8	n	13	r	18	w	23		
e	4	j	9	ñ	14	s	19	x	24		

Para codificar un texto, a la primera letra del texto a modificar se le suma el valor de la primera letra de la palabra clave, esto da el valor de la letra en el texto codificado, el código resultante no puede ser mayor que 26, por lo que deben utilizarse las primeras letras del alfabeto si es mayor (puede utilizarse la operación de módulo). Luego, para a siguiente letra, se le suma el valor de la siguiente letra de la palabra clave. Cuando ya se utilizaron todos los valores de la palabra clave, se repiten. Esto se hace hasta codificar todo el texto.

Ejemplo con palabra clave "amigo":

Texto: `hoy celebraremos como familia`

Valores de la palabra clave:

a	m	i	g	o
0	12	8	6	15

Valores del mensaje, alineamiento y suma con los valores de la palabra clave:

h	o	y		c	e	l	e	b	r	a	r	e	m	o	s		c	o	m	o		f	a	m	i	l	i	a
7	15	25		2	4	11	4	1	18	0	18	4	12	15	19		2	15	12	15		5	0	12	8	11	8	0
+																													
a	m	i		g	o	a	m	i	g	o	a	m	i	g	o		a	m	i	g		o	a	m	i	g	o	a
0	12	8		6	15	0	12	8	6	15	0	12	8	6	15		0	12	8	6		15	0	12	8	6	15	0
=																													
h	a	g		i	s	l	p	j	x	o	r	p	t	u	h		c	a	t	u		t	a	x	p	q	w	a
7	0	6		8	19	11	16	9	24	15	18	16	20	21	7		2	0	20	21		20	0	24	16	17	23	0

Texto codificado: `hag islpjxorptuh catu taxpqwa`

Decodificación

Para la decodificación se requiere el mensaje codificado y la palabra clave. Se hace el proceso inverso, al código de cada letra del mensaje se le resta el código de la letra de la palabra clave correspondiente y se obtiene el código de la letra del mensaje original.

h	a	g		i	s	l	p	j	x	o	r	p	t	u	h		c	a	t	u		t	a	x	p	q	w	a
7	0	6		8	19	11	16	9	24	15	18	16	20	21	7		2	0	20	21		20	0	24	16	17	23	0
-																													
a	m	i		g	o	a	m	i	g	o	a	m	i	g	o		a	m	i	g		o	a	m	i	g	o	a
0	12	8		6	15	0	12	8	6	15	0	12	8	6	15		0	12	8	6		15	0	12	8	6	15	0
=																													
h	o	y		c	e	l	e	b	r	a	r	e	m	o	s		c	o	m	o		f	a	m	i	l	i	a
7	15	25		2	4	11	4	1	18	0	18	4	12	15	19		2	15	12	15		5	0	12	8	11	8	0

4. Cifrado PlayFair modificado

Este cifrado consiste en una técnica de sustitución dinámica que también utiliza una palabra clave. Con la palabra clave se genera una matriz de letras que se usa para sustituir las letras de dos en dos, en vez de individualmente.

Supongamos la palabra clave "`profundizaste`". Con esta palabra se genera una matriz de seis filas y cinco columnas con todas las letras del abecedario. Si la palabra clave tuviera letras repetidas, deben eliminarse.

```
p r o f u
n d i z a
s t e b c
g h j k l
m ñ q v w
x y 1 2 3
```

Dado que la cantidad de letras no alcanza para completar la matriz, utilizaremos los números 1, 2 y 3 para completarla.

Ahora supongamos que la palabra a codificar es: "`murcielagos`". Si la palabra a codificar tuviera alguna letra repetida consecutivamente (por ejemplo "`accion`"), las separamos con un 1 ("`ac1cion`"). Luego, hay que

hacer es agrupar las letras en parejas: `mu rc ie la go s1`. Si la cantidad de letras es impar, como en este caso, la última letra se agrupa con un número 1. Si la cantidad de letras es par, no es necesario agregar nada.

Seguidamente procedemos a codificar cada par de letras. Para cada par de letras pueden darse tres casos según la fila y la columna donde se encuentren, y en cada caso hacemos la codificación de forma diferente:

1. Están en diferente fila, diferente columna

   Cada letra se sustituye por la que se encuentra en la misma fila y en la columna de la otra letra.

   ```
   p r o f u
   n d i z a
   s t e b c
   g h j k l
   m ñ q v w
   x y 1 2 3
   ```

   Por ejemplo, si la pareja es “pe”, la “p” se sustituye por “o” y la “e” por “s”.

2. Misma fila, diferente columna

   Cada letra se sustituye por la que se encuentra en la posición contigua a la derecha. Si la letra está en la columna de más a la derecha, se sustituye por la que está en la primera columna.

   ```
   p r o f u
   n d i z a
   s t e b c
   g h j k l
   m ñ q v w
   x y 1 2 3
   ```

   Por ejemplo, si la pareja es “ru”, la “r” se sustituye por “o” y la “u” por “p”.

3. Diferente fila, misma columna

   Cada letra se sustituye por la que se encuentra en la posición contigua hacia abajo. Si la letra está en la última fila, se sustituye por la que está en la primera fila.

   ```
   p r o f u
   n d i z a
   s t e b c
   g h j k l
   m ñ q v w
   x y 1 2 3
   ```

   Por ejemplo, si la pareja es “yh”, la “y” se sustituye por “r” y la “h” se sustituye por “ñ”.

Continuando con el ejemplo:

`mu→wp (caso 1)`

`rc→ut (caso 1)`

`ie→ej (caso 3)`

`la→wc (caso 3)`

`go→jp (caso 3)`

`s1→ex (caso 1)`

Texto codificado: “wputejwcjpex”

Para codificar un texto completo, se aplica el proceso anterior a cada una de las palabras del mensaje. Por ejemplo, al codificar el mensaje “guitarras guardadas en el placard”, procesamos cada palabra por separado.

Palabra 1: “guitarras” → ”guitar1ras” → “gu it ar 1r as”

gu→lp (caso 1)

it→de (caso 1)

ar→du (caso 1)

1r→yo (caso 1)

as→nc (caso 1)

Palabra 1 codificada: “lpdeduyonc”

Palabra 2: “guardadas” → “gu ar da da s1”

gu→lp (caso 1)

ar→du (caso 1)

da→in (caso 2)

da→in (caso 2)

s1→ex (caso 1)

Palabra 2 codificada: “lpduininex”

Palabra 3: “en” → “en”

en→si (caso 1)

Palabra 3 codificada: “si”

Palabra 4: “el” → “el”

el→cj (caso 1)

Palabra 4 codificada: “cj”

Palabra 5: “placard” → “pl ac ar d1”

pl→ug (caso 1)

ac→cl (caso 3)

ar→du (caso 1)

d1→iy (caso 1)

Palabra 4 codificada: “uglduiy”

El resultado completo de la codificación de la frase con la palabra clave “profundizaste”:

“guitarras guardadas en el placard”

“lpdeduyonc lpduininex si cj uglduiy”

Decodificación

La decodificación de este método se realiza casi de la misma forma que la codificación, pero los casos 2 y 3 deben hacerse restando posiciones a las filas y columnas en vez de sumar (en sentido contrario). Por todo lo demás, es básicamente el mismo algoritmo.

Las palabras del mensaje a decodificar todas deben ser de un largo par, de otro modo, el mensaje no se encontraría bien codificado. Esto debería chequearse como parte de las restricciones a la hora de desencriptar un mensaje.

Al decodificar una palabra, la misma puede contener números, estos deben eliminarse.

5. Cifrado Rail Fence

En el cifrado ***Rail Fence*** (que significa “cerca de palos de madera”), el texto original se escribe en zigzag, imitando la posición inclinada de los palos de madera en una cerca de este tipo. Luego, los caracteres se agrupan por filas, generándose la nueva posición de cada carácter dentro del mensaje. Luego el texto resultante se divide en grupos de 5 caracteres.

Observe el siguiente ejemplo de la codificación. Se quiere codificar el mensaje M.

`M = “Espero que este día nunca termine”`

1. Debido a que se van a usar 3 líneas para construir el zigzag, el mensaje debe ajustarse para que tenga una cantidad de caracteres que sea múltiplo de 4. El mensaje tiene 33 letras. El múltiplo de 4 mayor más cercano es el 36, por lo que se agregan 3 espacios en blanco al final para que el mensaje tenga este largo.
   `M = “Espero que este día nunca termine   ”`

2. El siguiente paso consiste en cambiar los espacios en blanco por un carácter diferente, ya que el espacio en blanco será utilizado con otro propósito. En este caso utilizaremos guiones.

   `M = “Espero-que-este-día-nunca-termine---”`

3. Seguidamente, los caracteres del mensaje se representan con la siguiente disposición en zigzag.

E				r				u				s				d				n				a				r				e			
	s		e		o		q		e		e		t		-		í		-		u		c		-		e		m		n		-		-
		p				-				-				e				a				n				t				i				-	

Observe que la línea del medio tiene el doble de caracteres que las otras. Entonces la línea de arriba tiene 36/4 = 9 caracteres, la del medio 36/2=18 y la de abajo 9, también. Los caracteres se agrupan por líneas y se obtiene siguiente texto.

`M = “Erusdnareseoqeet-í-uc-emn--p--eanti-”`

4. Como último paso para la codificación, se divide el texto en grupos de 5 caracteres y se separan con espacios. El último grupo puede tener menos caracteres, si no alcanzan para formar un grupo de 5 letras.

M = “Erusd nares eoqee t-í-u c-emn --p-- eanti -”

De esta forma, M contiene el resultado de la codificación del mensaje.

Decodificación

A continuación, se muestra el proceso de decodificación, que es básicamente el inverso de la codificación. Para decodificar un mensaje, este debe tener un largo que sea múltiplo de 4, sin contar los espacios que separan los grupos.

1. Primero se eliminan los espacios que separan los grupos de caracteres. El resultante debe tener una cantidad de caracteres que sea múltiplo de 4.

   M = “Erusdnareseoqeet-í-uc-emn--p--eanti-”

2. Se construye la línea de zigzag con los caracteres del mensaje. Se separa el mensaje en 4 partes iguales, dejando una en la primera línea, dos en la segunda y una en la tercera.

E				r				u				s				d				n				a				r				e			
	s		e		o		q		e		e		t		-		í		-		u		c		-		e		m		n		-		-
		p				-				-				e				a				n				t				i				-	

   Siguiendo la línea de zigzag, se reconstruye el mensaje original, tomando a la vez un carácter de la primera línea, la segunda, la tercera y la segunda.

   M = “Espero-que-este-día-nunca-termine---”

3. Seguidamente se cambian los guiones por espacios en blanco.

   M = “Espero que este día nunca termine   ”

4. Finalmente se eliminan los espacios en blanco que se ubican al final del mensaje.

   M = “Espero que este día nunca termine”

De esta forma, M contiene el mensaje original que se encontraba codificado.

Software a desarrollar

El objetivo de esta tarea es programar todos los algoritmos de codificación y decodificación.

Para todos los casos se trabajará con el siguiente alfabeto: “abcdefghijklmnñopqrstuvwxyz”. Deben eliminarse los caracteres no alfabéticos de los mensajes a codificar, exceptuando los espacios. Los acentos deben sustituirse por las vocales sin acento.

Deben implementarse las funciones necesarias en Python para cumplir con los requerimientos de esta tarea. Cada una de estas funciones retorna un string con el texto codificado. Las funciones deben producir una excepción en caso de que no se cumplan las restricciones.

1. cesarCod(texto, desplazamiento) → codifica usando cifrado césar con el desplazamiento indicado
2. cesarDec(texto, desplazamiento) → decodifica usando cifrado césar con el desplazamiento indicado
3. monoCod(texto, palabra) → codifica usando cifrado monoalfabético con la palabra clave indicada
4. monoDec(texto, palabra) → decodifica usando cifrado monoalfabético con la palabra clave indicada
5. vigenereCod(texto, palabra) → codifica el texto usando cifrado Vigenere con la palabra clave indicada

6. vigenerDec(texto, palabra) → decodifica el texto usando cifrado Vigenere con la palabra clave indicada
7. playfairCod(texto, palabra) → codifica el texto usando cifrado PlayFair con la palabra clave indicada
8. playfairDec(texto, palabra) → decodifica el texto usando la palabra clave indicada
9. railfenceCod(texto) → codifica el texto usando cifrado Rail Fence
10. railfenceDec(texto) → decodifica el texto usando cifrado Rail Fence

Además de las funciones, debe crearse un programa principal que se encargue de implementar la interfaz de usuario. Dicho programa debe cumplir con lo siguiente:

1. Mensaje de bienvenida y descripción breve del programa.
2. Mostrar un menú principal con opciones para elegir uno de los cinco métodos de encripción o salir del programa.
3. En cada método de encripción, una opción para elegir si se desea codificar o decodificar
4. Solicitar al usuario el mensaje a procesar y cualquier otro valor necesario (palabra clave o desplazamiento)
5. Mostrar al usuario el resultado de la codificación/decodificación
6. Volver al menú principal para continuar utilizando el programa

Aparte de las subrutinas sugeridas, pueden crearse las subrutinas que el equipo de trabajo considere necesario.

El programa debe ser lo más robusto posible, ser resistente ante cualquier valor "extraño" que el usuario ingrese, detectar estas situaciones y volver a solicitar los datos que sean necesarios. El programa nunca debería terminar con un error.

Documentación

Interna

Toda subrutina debe llevar como documentación interna comentarios con lo siguiente:

- Descripción breve de la función
- Entradas
- Salidas
- Restricciones

Externa

Debe entregarse un manual de usuario en formato PDF que explique a cualquier persona cómo instalar y utilizar. Con este manual, cualquier persona debe ser capaz de poner a funcionar el programa.

Debe incluirse una portada con los datos básicos del grupo de trabajo y el proyecto. El formato del manual es libre e informal, pueden utilizarse imágenes, colores y cualquier temática que el equipo de trabajo elija.

El manual debe incluir instrucciones para instalar el intérprete de Python, poner a funcionar el programa y cómo utilizar sus opciones. Incluya imágenes y ejemplos de la utilización del programa y los diferentes métodos de encripción.

Por favor evitar copiar partes de este enunciado en el manual de usuario, este debe ser 100% original.

Entrega

El tiempo asignado para la tarea programada es de 2 semanas. El trabajo se realizará en parejas o tríos.

La entrega debe hacerse en la sección de evaluaciones del curso en el TEC-Digital. Debe entregarse un archivo comprimido .ZIP con el archivo .py de la solución y un archivo .PDF con la documentación.

Evaluación

La tarea tiene un valor de 20% de la nota final, en el rubro de Proyectos Programados.

Desglose de la evaluación de la tarea programada:

Documentación:	20%
Programación:	80%

Recomendaciones adicionales

Antes de escribir código entienda muy bien el problema y trate de diseñar los algoritmos a utilizar. Esto mediante un análisis donde tome el problema y lo subdivide en problemas más pequeños y fáciles de solucionar, como se ha visto en clases.

Piense en las acciones que un usuario podría hacer para causar fallos en las funciones y tome las medidas necesarias.

Si bien es cierto que lo más importante es que el programa funcione, no descuide la documentación, ya que una mala documentación podría causar que su sistema sea malinterpretado, obteniendo una nota que no merezca.

Cualquier actividad fraudulenta será procesada según el Reglamento de Enseñanza-Aprendizaje del Instituto Tecnológico de Costa Rica.