Instituto Politécnico Nacional

Centro de Estudios Científicos y Tecnológicos 9

“Juan de dios Bátiz”

Equipo
“GUIMO”

Protocolo de cifrado

Alumnos:

García Ambrosio Aldo

Hernández Vera Gabriel

Ponce Espino Miguel Ángel

Salinas Aquino Omar Iván

Profesor:

GUTIÉRREZ GALAN GERMÁN

Unidad de aprendizaje:

Introducción a los Sistemas Distribuidos

Grupo.

5IM10

Fecha de entrega:

12 de septiembre de 2023

Protocolo de Encriptación y Cifrado del equipo GUIMO

Resumen

A lo largo de los años el humano siempre ha buscado tener privacidad, principalmente en el envío de información, ante esta problemática diversos personajes crearon procesos para codificar esta información, un ejemplo es el Emperador Julio César con el cifrado César, pero este resulta ser muy fácil de quebrantar por lo cual creamos GUIMO encryption text protocol (GETP), este es un protocolo diseñado para encriptar (conversión de datos de un formato legible a un formato codificado) cualquier tipo de mensaje utilizando un cifrado simétrico el cual es cualquier técnica que utiliza la misma llave para cifrar y descifrar los datos. Todo esto con el fin de que solo el emisor y el receptor conozcan el mensaje que fue enviado. El documento actual proporciona una descripción general del proceso de encriptación y des-encriptación, así como el código de un programa que cuenta con la función de realizar las actividades anteriormente mencionadas, realizado con el lenguaje de programación “Python”.

Abstract

Over the years, humans have always sought to have privacy, mainly when sending information. Faced with this problem, various people created processes to encode this information. An example is the Emperor Julius Caesar with the Caesar cipher, but this turns out to be very easy. of breaking which is why we created GUIMO encryption text protocol (GETP), this is a protocol designed to encrypt (convert data from a readable format to an encoded format) any type of message using a symmetric encryption which is any technique that uses same key to encrypt and decrypt data. All this so that only the sender and the receiver know the message that was sent. The current document provides a general description of the encryption and decryption process used, as well as the code of a program that has the function of performing the aforementioned activities, carried out with the “Python” programming language.

TABLA DE CONTENIDOS

Introducción

En la era digital actual, la seguridad de la información es una preocupación primordial. La necesidad de proteger datos sensibles y confidenciales ha impulsado el desarrollo de técnicas avanzadas de encriptación y des-encriptación. Siendo la encriptación el proceso de convertir datos legibles en una forma ilegible (cifrado) utilizando un algoritmo y una clave secreta. Solo aquellos que poseen la clave adecuada pueden descifrar y acceder a la información original. Esta técnica es esencial para garantizar la privacidad y la seguridad de la información, ya sea en la transmisión a través de redes de comunicación o en el almacenamiento en dispositivos de almacenamiento.

En este contexto, este documento presenta un protocolo de encriptación y des encriptación implementada en el lenguaje de programación Python.

La elección de Python como lenguaje base para este protocolo se debe a su versatilidad, simplicidad y amplia comunidad de desarrolladores. Estas características permiten una implementación eficiente y un mantenimiento sostenible del protocolo a lo largo del tiempo.

El objetivo principal de este protocolo es proporcionar una solución robusta y confiable para asegurar la confidencialidad de los datos durante su transmisión y almacenamiento. Se ha diseñado considerando diversos escenarios de aplicación, desde la comunicación en redes hasta la protección de archivos individuales.

En las secciones siguientes, se detallarán los fundamentos teóricos de la encriptación y des-encriptación utilizada en este protocolo. Posteriormente, se describirá la implementación práctica en Python, abordando los aspectos técnicos y las consideraciones de seguridad pertinentes.

Es importante destacar que este protocolo se ofrece como una herramienta adicional en el arsenal de soluciones de seguridad digital y no pretende reemplazar estándares reconocidos de encriptación. A lo largo de este documento, se proporcionarán ejemplos de uso y se discutirán posibles escenarios de aplicación para ilustrar la eficacia y versatilidad de este protocolo de encriptación y des encriptación desarrollada en Python.

Encriptación

1.- Para el encriptado de información, primeramente se ingresará una cadena de texto, la cual se comprobará que no esté vacía, para luego mediante un ciclo for se convertirá carácter por carácter a su unidad en el código ASCII. Mediante la siguiente función de Python:

```
Cod_ASCII = ord(Letra)
```

¿Qué es el código ASCII?

> *"El código ASCII (American Standard Code for Information Interchange) es un sistema de codificación que asigna un valor numérico único a diferentes caracteres utilizados en la comunicación electrónica." (Martinez, 2023)*

ASCII extendido

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	└	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	┴	225	E1h	ß
130	82h	é	162	A2h	ó	194	C2h	┬	226	E2h	Ô
131	83h	â	163	A3h	ú	195	C3h	├	227	E3h	Ò
132	84h	ä	164	A4h	ñ	196	C4h	─	228	E4h	õ
133	85h	à	165	A5h	Ñ	197	C5h	┼	229	E5h	Õ
134	86h	å	166	A6h	ª	198	C6h	ã	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	Ã	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	╚	232	E8h	Þ
137	89h	ë	169	A9h	®	201	C9h	╔	233	E9h	Ú
138	8Ah	è	170	AAh	¬	202	CAh	╩	234	EAh	Û
139	8Bh	ï	171	ABh	½	203	CBh	╦	235	EBh	Ù
140	8Ch	î	172	ACh	¼	204	CCh	╠	236	ECh	ý
141	8Dh	ì	173	ADh	¡	205	CDh	═	237	EDh	Ý
142	8Eh	Ä	174	AEh	«	206	CEh	╬	238	EEh	¯
143	8Fh	Å	175	AFh	»	207	CFh	¤	239	EFh	´
144	90h	É	176	B0h	░	208	D0h	ð	240	F0h	
145	91h	æ	177	B1h	▒	209	D1h	Ð	241	F1h	±
146	92h	Æ	178	B2h	▓	210	D2h	Ê	242	F2h	‗
147	93h	ô	179	B3h	│	211	D3h	Ë	243	F3h	¾
148	94h	ò	180	B4h	┤	212	D4h	È	244	F4h	¶
149	95h	ò	181	B5h	Á	213	D5h	ı	245	F5h	§
150	96h	û	182	B6h	Â	214	D6h	Í	246	F6h	÷
151	97h	ù	183	B7h	À	215	D7h	Î	247	F7h	¸
152	98h	ÿ	184	B8h	©	216	D8h	Ï	248	F8h	°
153	99h	Ö	185	B9h	╣	217	D9h	┘	249	F9h	¨
154	9Ah	Ü	186	BAh	║	218	DAh	┌	250	FAh	·
155	9Bh	ø	187	BBh	╗	219	DBh	█	251	FBh	¹
156	9Ch	£	188	BCh	╝	220	DCh	▄	252	FCh	³
157	9Dh	Ø	189	BDh	¢	221	DDh	¦	253	FDh	²
158	9Eh	×	190	BEh	¥	222	DEh	Ì	254	FEh	■
159	9Fh	ƒ	191	BFh	┐	223	DFh	▀	255	FFh	

2.- Posteriormente, continuando en el mismo ciclo for se propuso una función cuadrática inyectiva, que haga una determinada operación que altere el número clave ASCII y así obtener una cantidad numérica diferente, dificultando la des-encriptación y mejorando de esta forma la encriptación.

¿Qué es una función inyectiva?

> *Se dice que una función $f : A \rightarrow B$ es inyectiva si y solo si se satisface que: si $a \neq b$ entonces $f(a) \neq f(b)$. En términos más simples quiere decir que para cada valor de "x" solamente existe un valor de "y".*

Por lo tanto, es posible afirmar que un número del código ASCII tiene solamente un número cifrado, que a su vez quiere decir, que el número cifrado tiene solamente un valor en el código ASCII, en consecuencia, no habrá alguna mala interpretación al cifrar y descifrar algún texto.

Además, también se evalúa que la función solo arroje números dentro de un rango que al transformarlos en binario tenga una longitud máxima de 15 dígitos
La función tomada en cuenta es la siguiente:

```
Func = ((Cod_ASCII - 32) % (2**15)) ** 2
```

Se evalúa el valor de la letra en código ASCII y se obtiene otro valor.

3.- Posteriormente, al obtener dicho valor, el último paso consta de transformar el valor obtenido a número binario mediante la función format() de python, asegurándose también que la longitud de este binario sea de 15 dígitos:

```
Binario = format(Func, '015b')
```

Decimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

¿Qué es el sistema binario?
El sistema binario o sistema diádico es un sistema de numeración fundamental en la computación e informática, en el cual la totalidad de los números pueden representarse empleando cifras compuestas por combinaciones de dos únicos dígitos. En el caso del código binario, los dígitos utilizados son ceros (0) y unos (1). (Sistema binario - qué es, concepto, aplicaciones y ejercicios, s. f.)
Para transformar números enteros a binario se puede recurrir a los siguientes pasos:

- *División sucesiva por 2: Divide el número decimal entre 2 y anota el cociente y el residuo. Luego, divide el cociente resultante entre 2 nuevamente y anota el nuevo cociente y residuo. Repite este proceso hasta que el cociente sea 0.*

- *Leer los residuos en orden inverso: Los residuos obtenidos en el paso anterior, leídos en orden inverso, representan el número en binario.*

4.- Finalmente, el binario obtenido se irá agregando a una cadena que, tras el término de la iteración del ciclo for, se imprimirá la encriptación final:

```
    Bin_Fin += Binario

print("Encriptación Exitosa: \n", Bin_Fin, "\n")
```

Des-encriptación

1.- Para la des-encriptación de la información, el programa tomará el valor otorgado en binario y siguiendo el empaquetamiento de la encriptación en binarios de 15 dígitos, se segmenta la cadena binaria en binarios con una extensión de 15 dígitos a través de un for, los cuales se irán agregando a una lista, esto mediante un def():

```
def Dividir_Binario(binario):
    Segmentos = [binario[i:i+15] for i in range(0, len(binario), 15)]
    return Segmentos
```

2.-. A través de un ciclo for se tomará una “n” cadena de binario de la lista Segmentos y luego mediante la función int() de python se convertirá a enteros:

```
Bin = segmentos[i]
fx = int(Bin,2)
```

Por otra parte el proceso para convertir binario a decimal es el siguiente:

- Asigna un valor a cada dígito en el número binario, comenzando desde la derecha y aumentando en potencias de 2. El dígito más a la derecha representa 2^0 (1). El siguiente dígito a la izquierda representa 2^1 (2). Y así sucesivamente, aumentando la potencia de 2 en cada posición a medida que avanzas hacia la izquierda.
- Multiplica cada dígito binario por su valor asignado y suma todos los resultados. El resultado de la suma es el equivalente decimal del número binario.

3.- Dentro del tercer paso, el valor será introducido en la función inversa de la función anteriormente empleada, con el objetivo de regresarlo a su valor original y así poder transformar los valores numéricos a caracteres dentro del sistema ASCII.

¿Qué es una función inversa?

> *Una función inversa o también llamada recíproca es aquella que cumple que el dominio es igual al recorrido de la función original y su recorrido es igual al dominio de la misma función.*

Entonces, el lenguaje algebraico, si tenemos una función:

$f: A \rightarrow B \; y \; x \rightarrow \; f(x) = y$

La función inversa será:

$f\text{-}1: B \rightarrow A \; y \; y \rightarrow \; f\text{-}1(y) = x$

No todas las funciones tienen una función inversa, para que una la función inversa exista, la función original tiene que ser biyectiva, lo que obliga que a todos los elementos de B llegue solo una flecha desde A (inyectiva y sobreyectiva a la vez).

Para lograr esto, la función requerida es:

```
Cod_ASCII= int((fx ** 0.5) + 32)
```

En este caso se emplea exponentes decimales **0.5 debido a que, para emplear la operación raíz en python se necesita emplear la librería sqrt

4.- Como se mencionó en el paso anterior, una vez obtenido el valor inicial, el número obtenido se convertirá nuevamente a caracteres mediante la función chr() de python, tomando como referencia el sistema ASCII.Posteriormente se construirá una cadena de caracteres sumando cada carácter obtenido, por iteración. Para finalmente imprimir la cadena:

```
    caracter = chr(Cod_ASCII)
    Cadena_Final += caracter

print("Descencriptación Exitosa: \n", Cadena_Final ,"\n")
```

Finalmente, será desplegado el mensaje descifrado.

Bibliografía:

- Martinez, D. G. (2023). ¿Qué es y para qué sirve el código ASCII? Blog. https://es.godaddy.com/blog/que-es-y-para-que-sirve-el-codigo-ascii/

- Sistema binario - qué es, concepto, aplicaciones y ejercicios. (s. f.). Concepto. https://concepto.de/sistema-binario/#ixzz8Ct81eJsy

- Números binarios (Artículo) | Khan Academy. (s. f.). Khan Academy. https://es.khanacademy.org/computing/ap-computer-science-principles/x2d2f703b37b450a3:digital-information/x2d2f703b37b450a3:binary-numbers/a/bits-and-binary

- Función inversa (portaleducativo.net)