Criptografía y Cifrado de Información 2021

Lab 02

22.julio.2021

En este laboratorio implementaremos funciones que convierten cadenas a bits y a Base 64. Además, investigamos una propiedad estadística de la función XOR.

Base 64 es un sistema de numeración posicional que usa 64 como base. Es la mayor potencia que puede ser representada usando únicamente los caracteres imprimibles de ASCII:

```
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
```

Para transformar datos en una codificación Base 64, la cadena se convierte a bits (Ejercicio 1). Luego, el flujo de bits se divide en bloques de 3 bytes (24 bits), y cada bloque a su vez se subdivide en 4 grupos de 6 bits cada uno. Los bloques de 6 bits se convierten a número decimal y el resultado se imprime como el caracter en la posición indicada por dicho decimal. Si hay menos de 3 bytes por codificar, el resto del búfer se rellena con ceros a la derecha.

Después de codificar los datos, si en el paso anterior quedaban 2 octetos por codificar, entonces se añade el carácter '=' al final de la salida; si solo quedaba un octeto, se concatenarán dos caracteres '='.

Por ejemplo el texto: "Man is distinguished, not only by his reason, but by this singular passion from other animals, which is a lust of the mind, that by a perseverance of delight in the continued and indefatigable generation of knowledge, exceeds the short vehemence of any carnal pleasure." se codifica en base64 como sigue:

```
TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieSB0aGlz
IHNpbmd1bGFyIHBhc3Npb24gZnJvbSBvdGhlciBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
dGhlIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbiB0aGUgY29udGlu
dWVkIGFuZCBpbmRlZmF0aWdhYmxlIGdlbmVyYXRpb24gb2Yga25vd2xlZGdlLCBleGNlZWRzIHRo
ZSBzaG9ydCB2ZWhlbWVuY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS4=
```

La siguiente figura muestra un ejemplo.

Texto de entrada	M								a								n							
ASCII	77								97								110							
Bits	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Índice	19				22				5				46											
Resultado en Base64	T				W				F				u											

1. Implementar funciones para convertir una cadena de caracteres a bits. Esto es, por cada caracter de la cadena, la función debe encontrar la representación en 8 bits del valor ASCII de dicho caracter. Luego, la función debe devolver la concatenación de todos estos bits, sobre todos los caracteres de la cadena.
   Implementar también la función que hace lo contrario: dada una cadena de bytes, devuelve el texto correspondiente.

   Muestre 3 ejemplos sencillos de cadenas, para verificar que sus funciones convierten dicha cadena a bytes, y luego recupera el texto original.

2. Implementar funciones para convertir una cadena de caracteres a Base 64. Implementar también la función que hace lo contrario: dada una cadena de bits en Base 64, devuelve el texto correspondiente.

   Muestre 3 ejemplos sencillos de cadenas, para verificar que sus funciones convierten dicha cadena a Base 64, y luego recupera el texto original.

3. **Propiedad de la función XOR.**

a) Implementar una función que haga la operación XOR, bit a bit, con dos cadenas de bits.

b) Dada una cadena de bits X (por ejemplo, una que proviene de convertir un texto a bits). Calcular la distribución de probabilidad de ocurrencia sus bits '0' y '1', y mostrarlo en un histograma. ¿Son iguales?

c) Repetir lo mismo para los bigramas '00', '01', '10', '11'; y para los trigramas '000', '001', '010', '011', '100', '101', '110', '111'.
¿Qué ocurre con la distribución de probabilidades?

d) Ahora, generar una cadena de bits Y de forma aleatoria, que tenga la misma longitud de X, y hacer la operación $Z = XOR(X; Y) = X \oplus Y$.
Investigar la distribución de probabilidad de ocurrencia para los bits, los bigramas y trigramas de Z, y graficar en histogramas. Comparar estas distribuciones con el valor original en X.