CONTROL DE TRANSMISIÓN DE DATOS. **5 de Diciembre de 2002**
Grupos 10 y 30

Notas Importantes:

1. Los resultados no justificados, no serán tenidos en cuenta.
2. Los problemas se entregarán por separado, poniendo su nombre y apellidos en cada hoja, y numerándolas.
3. Un error conceptual grave, puede anular todo el problema.

Problema 1 (50%)

Considere que se realiza un experimento que consiste en una transmisión de datos mediante un módem que utiliza una constelación PAM-16, tiene un factor de *roll-off* del 25% y el canal tiene un ancho de banda de 5 Khz.

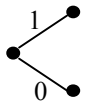
a) ¿Cuál es la cantidad de información máxima que emite la fuente (la llamaremos $Fuente_1$) en 10 segundos?. Considere que los símbolos fuente son equiprobables.

b) Si durante esos 10 segundos, se han emitido 10000 símbolos fuente, ¿cuál es la entropía máxima de dicha $Fuente_1$?

c) ¿Cuántos símbolos tendría la fuente si alcanza la entropía máxima con símbolos equiprobables?

Independientemente del resultado anterior, suponga que la fuente consta de 4 símbolos equiprobables {A, B, C, D}. Ahora se utiliza un código aritmético para codificar las secuencias de símbolos de dicha fuente. Se desea decodificar un valor real positivo que se ha recibido. La longitud de la secuencia que se transmitió, se ha enviado a parte.

Las posibles longitudes de las secuencias codificadas mediante el codificador aritmético, pueden ser uno de estos valores que emite la $Fuente_2$: {1, 2, 3, 4, 5, 6, 7, 8}. Las probabilidades asociadas a estas longitudes son {0.4, 0.2, 0.1, 0.1, 0.05, 0.05, 0.05, 0.05}. Se utiliza un código *Huffman* para codificar dichas longitudes. Considere que el árbol binario utilizado tiene a 1 sus ramas hacia arriba y a 0 sus ramas hacia abajo.



d) Halle el código Huffman utilizado para la $Fuente_2$.

e) Calcule la entropía de dicha $Fuente_2$.

f) Calcule la longitud media utilizada.

g) Calcule la eficiencia del código.

La longitud de la secuencia de símbolos de la fuente, que se ha enviado mediante el código *Huffman*, ha sido 11100. El valor real recibido ha sido 0.0868. Se desea decodificar dicho valor.

h) ¿Cuál es la secuencia de símbolos que se emitió, mediante la codificación aritmética anterior?

Considere ahora que en lugar del codificador aritmético, se utiliza un codificador LZW, con un diccionario cargado inicialmente con A, B, C, D en las posiciones 1, 2, 3, 4.

i) ¿Cuál sería entonces la secuencia codificada mediante el codificador LZW, para la secuencia de símbolos que ha decodificado en el apartado anterior?

j) ¿Qué código le parece más adecuado utilizar para nuestra fuente, el código aritmético o el código LZW? ¿Le parece más adecuado otro código? Razone su respuesta.

k) Calcule, o en su defecto acote, la entropía conjunta de las fuentes *Fuente*$_1$ y *Fuente*$_2$.

Problema 2 (50%)

Sea un sistema de clave pública RSA donde todos los usuarios utilizan e=23. Se dispone de una autoridad de certificación (CA) que genera certificados con un formato muy sencillo: CERTIFICADO := {ID_USUARIO (1 byte), N (1 byte), firma (1 byte)}. La CA utiliza las claves (N=143; e=23; d=47). Para la firma se utiliza la función de hash H(M), calculada de la siguiente forma. Considere k=4:

1. Se añade al final del mensaje el número de ceros necesario para que la longitud sea múltiplo de k.
2. Se divide el mensaje resultante en n bloques de k bits, mi, $0 \le i \le n-1$
3. H(M) se calcula iterativamente de la siguiente manera:

$$h_0 = m_0$$
$$h_{i+1} = h_i \oplus m_{i+1} \quad 0 \le i \le n-2 \quad \text{(XOR bit a bit)}$$
$$H(M) = h_{n-1}$$

a) Genere un par de claves para el usuario A. Utilice p=7, q=17.

b) Si A utiliza como identificador 00000010, indique todos los bits que forman el certificado de A. **Nota**: Recuérdese que la firma de un certificado, engloba a todos los campos de dicho certificado.

c) Un usuario B, que no comparte ningún secreto con A, quiere enviar una clave de sesión de 1 byte a A. Indique un protocolo mediante el cual puede enviar a A la clave k=00001010. Obtenga también la codificación de dicha clave. **Nota**: suponga que B confía en CA y que B conoce la clave pública de CA.

d) Codifique el mensaje M=11001100 mediante un cifrado de Vernan con la clave de sesión negociada en el apartado anterior. Justifique si el sistema es incondicionalmente seguro (considere también la parte de gestión de claves).

e) Sea un algoritmo de cifrado simétrico en bloque de 8 bits, donde el mensaje de entrada se coloca como estado inicial de un LFSR primitivo de grado 8, y el criptograma de salida se obtiene como el estado del LFSR al cabo del número de iteraciones que indique la clave. Si k=253, cifre el mensaje M=0001010000011000 en modo ECB.

Resolución problema 1

a) $W=(1+\alpha)/2 * v_m \rightarrow 5000=(1.25/2) * v_m \rightarrow v_m=8000$ símbolos/seg

PAM-16 → A=16 símbolos del modulador → $q=\log_2 A=4$ bits/símbolo

v_t [bits/seg]= q [bits/símbolo]* v_m [símbolos/seg]→ $v_t = 8000*4=32000$ bits/seg

En 10 segundos, se emiten como máximo (para el mejor código posible, el más eficiente) I=32000 bits/seg *10 seg = 320000 bits

b) La longitud media del código $\bar{L} \geq H$ es la entropía máxima de la fuente, $H = \bar{I}$ [bits/símbolo] = 320000 bits/10000 símbolos = 32 bits/símbolo

c)

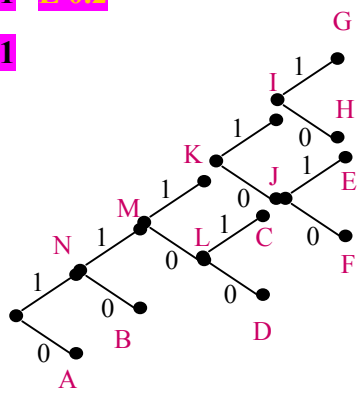
$$H(F_{\text{sin memoria}}) = \sum_{i=1}^{F} P(S_i) \cdot \log_2 \left(\frac{1}{P(S_i)} \right) = \bar{I}$$

para $P(S_i) = \frac{1}{F}, \forall i \Rightarrow H(F) = \log_2 F$

$32 = \log_2 F \Rightarrow F = 2^{32}$ símbolos

d) Código Huffman para *Fuente*$_2$.

1	A 0.4	A 0.4	A 0.4	A 0.4	A 0.4	A 0.4	A 0.4	A 0.4	A 0.4	N 0.6
2	B 0.2	B 0.2	B 0.2	B 0.2	B 0.2	B 0.2	B 0.2	**M 0.4**	**N 0.6**	A 0.4
3	C 0.1	C 0.1	C 0.1	C 0.1	K 0.2	**K 0.2**	**M 0.4**	**B 0.2**		
4	D 0.1	D 0.1	D 0.1	D 0.1	**C 0.1**	**L 0.2**				
5	E 0.05	E 0.05	I 0.1	**I 0.1**	**K 0.2**	**D 0.1**				
6	F 0.05	F 0.05	**E 0.05**	**J 0.1**						
7	**G 0.05**	**I 0.1**	**F 0.05**							
8	**H 0.05**									



A longitud=1 → 0

B longitud=2 → 10

C longitud=3 → 1101

D longitud=4 → 1100

E longitud=5 → 11101

F longitud=6 → 11100

G longitud=7 → 11111

H longitud=8 → 11110

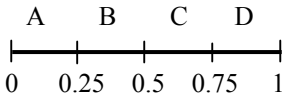
e)

$$H(F_2) = 0.4 \cdot \log_2 \frac{1}{0.4} + 0.2 \cdot \log_2 \frac{1}{0.2} + 2 \cdot 0.1 \cdot \log_2 \frac{1}{0.1} + 4 \cdot 0.05 \cdot \log_2 \frac{1}{0.05} = 2.521928 \ \ bits/símbolo$$

f) $\overline{L} = 0.4 + 0.2 \cdot 2 + 4 \cdot 0.1 \cdot 2 + 5 \cdot 0.05 \cdot 4 = 2.6 \, bits/símbolo$

g) $E = \frac{H}{\overline{L}} = \frac{2.521928}{2.6} = 0.97$

h) La longitud de la secuencia de símbolos fuente, se ha recibido que es el valor (su codificación en *Huffman*) 11100. Se corresponde con una longitud de la secuencia emitida por la fuente igual a **6**.



$0.0868 \Rightarrow$ A

$\frac{0.0868 - 0}{0.25} = 0.3472 \Rightarrow$ B

$\frac{0.3472 - 0.25}{0.25} = 0.3888 \Rightarrow$ B

$\frac{0.3888 - 0.25}{0.25} = 0.5552 \Rightarrow$ C

$\frac{0.5552 - 0.5}{0.25} = 0.2208 \Rightarrow$ A

$\frac{0.2208 - 0}{0.25} = 0.8832 \Rightarrow$ D

La secuencia emitida fue ABBCAD.

i)

1	**A**	**ABBCAD**
2	**B**	**1 2 2 3 1 4**
3	**C**	
4	**D**	
5	AB	
6	BB	
7	BC	
8	CA	
9	AD	

j) El código LZW tiene sentido para secuencias más largas, en que hayan repeticiones, para fuentes con memoria. El LZW implica gestionar un diccionario. El código aritmético necesita enviar la longitud de la secuencia y almacenar y enviar números reales con muchos decimales. En este caso parece más sencillo un Huffman que al ser símbolos equiprobables tendrá longitud fija 2 bits/símbolo.

k) $H(F_1, F_2) \leq H(F_1)+H(F_2) = H(F_1)+H(F_2)$ para Fuentes independientes, como es el caso que nos ocupa.

$$H(F_1, F_2) = H(F_1)+H(F_2) = 32 + 2.521928 = 34.521928 \text{ bits/símbolo}$$

Resolución Problema 2

a) $N = p \cdot q = 7 \cdot 17 = 119$,

$\Phi(N) = (p-1)\cdot(q-1) = 6 \cdot 16 = 96$. Compruebo que efectivamente $\text{mcd}(\Phi(N), e) = 1$. Entonces :

$$d = \frac{k \cdot \Phi(N)+1}{e} = \frac{k \cdot 96+1}{23} = \frac{k \cdot (23 \cdot 4+4)+1}{23} = k \cdot 4 + \frac{4 \cdot k+1}{23} \Rightarrow \text{con } k = 17,\ d \text{ es entero.}$$

$d = 71$

Clave pública = (e, N_A) = (23, 119) Clave privada = d = 71

b) Certificado_A = Identificador_A, N_A, Firma = 00000010 01110111 00000111, ya que:

$N_A = 119 \equiv 01110111$

Firma => Quien forma es la Autoridad de Certificación (CA).

- Primero genero los campos del certificado, que es lo que la CA ha de firmar: M=00000010 01110111.
- Y ahora este M se ha de firmar, con la clave privada de la CA.
- Para ello primero genero su función resumen H(M) = 0000⊕0010⊕0111⊕0111 = = 00000010 ≡ 2
- Y CA genera su $Firma = (H(M))^{d_{CA}} \bmod N_{CA} = 2^{47} \bmod 143 = 7 \equiv 00000111$

c) Primero A enviará a B su certificado, que le da la autoridad de certificación: CERTIFICADO_{CA} [A, N_A, FIRMA(A, N_A)]

CA obtiene la FIRMA(A, NA) con su clave privada, d_{CA} y con N_{CA}.

Como B sabe la clave pública de CA (e_{CA}, N_{CA}) podrá averiguar H(M). Lee la clave pública de A (N_A) a partir del CERTIFICADO_{CA}. Genera M y calcula H(M). Si coinciden, queda autenticado.

Y como todos los usuarios, A también tiene como exponente público *e*. Así, B ya conoce la clave pública de A, (e, N_A).

El criptograma correspondiente a la clave k=00001010 es: $C = k^e \bmod N_A$

$C=10^{23} \bmod 119 = 5 \equiv 101$

d) Cifrado de Vernan => C=M⊕K, siendo M el mensaje y K la clave. Es incondicionalmente seguro si la longitud de la clave K, coincide con la del mensaje M. La codificación sí que es incondicionalmente segura. Pero la gestión de claves no, ya que RSA no lo es.

e) Modo ECB quiere decir que el mensaje se parte en bloques y cada bloque se codifica con la misma clave, en este caso de 8 bits cada uno: M= 00010100 00011000.

El polinomio de conexiones es primitivo de grado 8, por lo que $L=2^8-1=255$. El estado 253 es el estado dos veces anterior al inicial:

- Estado dos veces anterior al 00010100 = 01010000
- Estado dos veces anterior al 00011000 = 01100000
- Por lo que el criptograma es 01010000 01100000