Ana I. González-Tablas Ferreres
José María de Fuentes García-Romero de Tejada
Lorena González Manzano
Pablo Martín González
UC3M | GRUPO COMPUTER SECURITY LAB (COSEC)

EXAMEN FINAL

CONVOCATORIA ORDINARIA MAYO 2013

CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

Examen Final Convocatoria Ordinaria Mayo 2013

Apellidos: **Nombre:**

NIA: **Grupo reducido:**

PREGUNTAS DE RESPUESTA OBJETIVA (0,60 puntos)

Responda a las siguientes preguntas de respuesta objetiva. Solo hay una respuesta correcta.

1. **Sea *a* raíz primitiva o generador de Z_n, con *n* no primo. Seleccione la respuesta correcta:**
   a. Existe al menos un resto potencial de *a* respecto del módulo *n* igual a 1.
   b. El gaussiano de *a* respecto del módulo *n* es igual a $\Phi(n)$.
   c. Las potencias de *a* generan el conjunto reducido de restos Z^*_n, que incluye todos los elementos con inverso multiplicativo en Z_n.
   d. Todas las respuestas anteriores son ciertas.

2. **Indique cuál de las siguientes afirmaciones es cierta:**
   a. La entropía de una fuente de mensajes mide la incertidumbre que, a priori, tiene un observador acerca de la aparición de un mensaje.
   b. La entropía de una fuente de mensajes es la cantidad máxima de información transportada por un mensaje perteneciente a dicha fuente.
   c. La entropía tiende a infinito cuando la fuente de mensajes produce mensajes equiprobables.
   d. A menor entropía, mayor incertidumbre sobre M.

3. **Indique cuál de las siguientes afirmaciones NO es cierta:**
   a. Los cifradores simétricos de bloque son generalmente más lentos que los cifradores simétricos de flujo.
   b. Los cifradores simétricos de bloque son generalmente más lentos que los cifradores asimétricos.
   c. De los cifradores vistos en la asignatura, los cifradores simétricos de flujo son los cifradores más rápidos.
   d. De los cifradores vistos en la asignatura, los cifradores asimétricos son los más lentos.

4. **Indique cuál de las siguientes afirmaciones acerca del protocolo de acuerdo de claves Diffie-Hellman es cierta:**
   a. Permite establecer un canal autenticado y cifrado.
   b. Necesita intercambiar un secreto entre dos partes, por lo que es susceptible a ataques de escucha en el canal.
   c. Permite acordar una clave criptográfica entre dos partes y a través de un canal inseguro sin necesidad de intercambiar previamente ningún secreto.
   d. Se basa en el esquema de cifrado asimétrico ElGamal.

CUESTIONES (0,40 puntos)

1. **Exponga brevemente cuál sería, en su opinión, un criptosistema adecuado para el envío de información cifrada y autenticada desde A hasta B (se desea garantizar así mismo el no repudio de A), teniendo en cuenta todo lo visto en la asignatura, y que se desea enviar gran cantidad de información. Indique las fortalezas de su elección, y qué debilidades podría presentar.**

2. **Desarrolle, esquemáticamente, una clasificación de los métodos criptográficos clásicos, por un lado, y los métodos de cifra modernos, por otro. Para cada tipo de categoría, indique uno o más ejemplos de los vistos en la asignatura.**

CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

Examen Final Convocatoria Ordinaria Mayo 2013

Apellidos: **Nombre:**

NIA: **Grupo reducido:**

EJERCICIO 1 (1,8 puntos)

La entidad A desea obtener un certificado de clave pública de cierta Autoridad de Certificación AC. AC cuenta con dos parejas de claves: un par para firmar (algoritmo de firma ElGamal FEG con función resumen Hash1) y otro par para cifrar (algoritmo RSA). AC utiliza su par de claves de firma para generar certificados de clave pública, tanto para sí misma como para los usuarios de AC. Los usuarios de AC utilizan el par de claves de cifrado de AC para enviarle las peticiones cifradas a AC.

Los dos certificados de clave pública que AC ha generado sobre cada una de sus claves públicas se ponen a disposición de los usuarios a través de la página web. Además, para cada certificado, AC publica también su resumen (en hexadecimal) generado con la función resumen Hash2, para permitir a los usuarios verificar la integridad de los certificados una vez se los han descargado.

Especificación de las funciones resumen y el formato de los certificados:

- Hash1 es una función que acepta una lista de elementos codificados en decimal pertenecientes cada uno de ellos al rango [0, 255]. La salida de la función es el resultado de aplicar módulo 16 a la suma de todos los elementos de la lista de entrada. El resultado se codifica en decimal.

  Ejemplo Hash1: ListaEntrada = (35, 11). Hash1(35, 11) = (35 + 11) mod 16 = 14.

- Hash2 es una función que acepta una lista ordenada de elementos codificados en decimal pertenecientes cada uno de ellos al rango [0,255]. Cada elemento de la lista de entrada es codificado en binario (8 bits) y rotado circularmente a la derecha *i* bits, donde *i* es la posición que el elemento ocupa en la lista de entrada (primera posición = 1). La salida de la función es el resultado de aplicar la función XOR a todos los elementos una vez han sido rotados. El resultado se codifica en hexadecimal.

  Ejemplo Hash2: ListaEntrada = (35, 11).
  Hash2(35, 11) = RotarDchaCirc[1]($23_{(16)}$) XOR RotarDchaCirc[2]($0B_{(16)}$) = RotarDchaCirc[1](0010 $0011_{(16)}$) XOR RotarDchaCirc[2](0000 $1011_{(16)}$) = $53_{(16}$.

- Los certificados generados por AC constan de los siguientes elementos:
  - identidad del usuario poseedor del certificado;
  - elementos de la clave pública del usuario;
  - firma de AC sobre los elementos anteriores = (r, s). Como se ha comentado previamente, el algoritmo de firma es ElGamal (FEG) con función resumen Hash1.

DATOS:

Las identidades de A y AC se codifican ambas como cero: $ID_{AC} = 0$, $ID_A = 0$.

Claves de **firma** de AC: P = 17 (primo); g = 3 (generador); $X_{AC} = 4$; $Y_{AC} = 13$
$Cert_{AC}$-Firmar = (ID_{AC}; Y_A; $FEG_{AC}(Hash1(ID_{AC}, Y_A))$) = ($ID_{AC}$; Y_A; r, s) = (0; 13; 5, 5)

Claves de **cifrado** de AC: $n_{AC} = 33$ (módulo); $e_{AC} = 7$ (exponente público)
$Cert_{AC}$-Cifrar = (ID_{AC}; e_{AC}, n_{AC}; $FEG_{AC}(Hash1(ID_{AC}, e_{AC}, n_{AC}))$) = ($ID_{AC}$; e_{AC}, n_{AC}; r, s) = (0; 7, 33; 10, 0)

PARTE 1

A accede a la página web de AC y se descarga los certificados $Cert_{AC}$-Cifrar y $Cert_{AC}$-Firmar (véase el apartado Datos). Los valores hash publicados para cada uno de ellos son: Hash2($Cert_{AC}$-Cifrar) = $45_{(16}$ y Hash2($Cert_{AC}$-Firmar) = $B3_{(16}$

Se pide:

a) *(0,1 pts.)* Muestre cómo A verificaría la integridad de los certificados que se ha descargado comprobando los resúmenes publicados en la página web de AC, y realice los cálculos.
b) *(0,25 pts.)* Muestre cómo A verificaría el certificado $Cert_{AC}$-Firmar, y realice los cálculos.
c) *(0,3 pts.)* Muestre cómo A verificaría el certificado $Cert_{AC}$-Cifrar, y realice los cálculos.

PARTE 2

A continuación, A genera su par de claves de firma (algoritmo RSA –FRSA-- con función resumen Hash1). A elige como clave pública $(e_A, n_A) = (5, 21)$. Seguidamente, A genera una solicitud de generación de certificado para enviársela a AC. Dicha solicitud será la firma de A sobre los datos que A desea que se reflejen en el certificado:

$$Solicitud = (ID_A; e_A, n_A; FRSA_A(Hash1(ID_A, e_A, n_A)))$$

Antes de enviarla a AC, A cifra la solicitud para que solo AC pueda acceder a ella.

Se pide:

a) *(0,2 pts.)* Calcule d_A, es decir, la clave privada de A.
b) *(0,25 pts.)* Calcule $FRSA_A(Hash1(ID_A, e_A, n_A))$, es decir, la firma que genera A sobre su identidad y su clave pública.
c) *(0,1 pts.)* A debe enviar a AC la solicitud cifrada. Indique (sin realizar los cálculos) cómo A cifraría para AC la solicitud (si hay más de un elemento, considere que se cifra cada uno de ellos independientemente).

PARTE 3

A envía a AC la solicitud de generación de certificado cifrada (el resultado del apartado (c) de la parte 2).

Se pide:

a) *(0,1 pts.)* Indique (sin realizar los cálculos) cómo AC descifraría la solicitud cifrada recibida de A.
b) *(0,25 pts.)* Asuma que la solicitud que ha recibido AC de A, una vez descifrada, es:

$$Solicitud = (ID_A; e_A, n_A; FRSA_A(Hash1(ID_A, e_A, n_A))) = (0; 5, 21; 19).$$

Muestre cómo AC verificaría la solicitud que ha recibido de A.

c) *(0,25 pts.)* AC procede a generar el certificado de clave pública de A. Para realizar la firma, elige k=7. Calcule $Cert_A = (ID_A; e_A, n_A; FEG_{AC}(Hash1(ID_A, e_A, n_A)))$.

CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

Examen Final Convocatoria Ordinaria — Mayo 2013

Apellidos: **Nombre:**

NIA: **Grupo reducido:**

EJERCICIO 2 (1,2 puntos)

Alicia y Benito usan un cifrador Vernam para intercambiar datos confidenciales. Para producir la clave (serie cifrante) necesaria para dicho cifrador utilizan un LFSR de cuatro celdas, cuyo polinomio asociado es $f(x) = x^4 + x^3 + x^2 + 1$. Con el fin de acordar una semilla S de 4 bits ($S = a_1a_0b_1b_{0\,(2}$) con la que inicializar el LFSR, Alicia genera los dos bits más significativos de S (es decir, $S_A=a_1a_{0\,(2}$) y Benito los dos menos significativos ($S_B = b_1b_{0\,(2}$). Para intercambiarse de forma confidencial S_A y S_B, deciden utilizar el algoritmo de cifrado asimétrico RSA. Suponga que S_A es $10_{(2}$ y considere que Alicia recibe de Benito el valor $11000100_{(2}$. Suponga que la clave pública RSA de Alicia es $(e_A, N_A) = (147, 253)$ y la de Benito es $(e_B, N_B) = (7, 55)$.

a) *(0,3 pts.)* Calcule el valor que Alicia envía a Benito, expresado en decimal.
b) *(0,4 pts.)* Indique cómo Alicia obtendría la semilla S a partir de su parte de la semilla (S_A) y del valor que ha recibido de Benito. Realice los cálculos.
c) *(0,3 pts.)* Ignore el resultado anterior y suponga que la semilla que han convenido usar para inicializar el LFSR es $S = 0110_{(2}$. Benito envía el primer mensaje cifrado a Alicia. Considere que el mensaje cifrado que recibe Alicia es $C = 3BC_{(16}$. Calcule el mensaje en claro (descifrado) correspondiente y expréselo en hexadecimal.
d) *(0,2 pts.)* Indique si la secuencia (serie cifrante) generada por el LFSR tiene periodo máximo o no. Justifique su respuesta.

SOLUCIÓN

EJERCICIO 1 (1,5 puntos)

SOLUCIÓN:

PARTE 1

a) Verificación $Cert_{AC}$-Firmar

$Cert_{AC}$-Firmar = (ID_{AC}; Y_A; $FEG_{AC}(Hash1(ID_{AC}, Y_A))$) = (0; 13; 5, 5)

Hash2($Cert_{AC}$-Firmar) = RotarDchaCirc[1]($00_{(16}$) XOR RotarDchaCirc[2]($0C_{(16}$) XOR RotarDchaCirc[3]($05_{(16}$) XOR RotarDchaCirc[4]($05_{(16}$) = $B3_{(16}$

0000 0000 → 0000 0000
0000 1101 0100 0011
0000 0101 1010 0000
0000 0101 0101 0000

1011 0011 = $B3_{(16}$

$Cert_{AC}$-Cifrar = (ID_{AC}; e_{AC}, n_{AC}; $FEG_{AC}(Hash1(ID_{AC}, e_{AC}, n_{AC}))$) = (0; 7, 33; 10, 0)

Hash2($Cert_{AC}$-Cifrar) = RotarDchaCirc[1]($00_{(16}$) XOR RotarDchaCirc[2]($07_{(16}$) XOR RotarDchaCirc[3]($21_{(16}$) XOR RotarDchaCirc[4]($0A_{(16}$) XOR RotarDchaCirc[5]($00_{(16}$) = $45_{(16}$

0000 0000 → 0000 0000
0000 0111 1100 0001
0010 0001 0010 0100
0000 1010 1010 0000
0000 0000 0000 0000

0100 0101 = $45_{(16}$

Ambos resúmenes son los mismos que los publicados en la página web.

b) $Cert_{AC}$-Firmar = (ID_{AC}; Y_A; $FEG_{AC}(Hash1(ID_{AC}, Y_A))$) = ($ID_{AC}$; Y_A; r, s) = (0; 13; 5, 5)
Hash1(ID_{AC}; Y_A)= 0 + 13 mod 16 = 13
$V_1 = Y_A^{\,r} \cdot r^{\,s} \bmod p = 13^{\,5} \cdot 5^{\,5} \bmod 17 = 371293 \cdot 3125 \bmod 17 = 13 \cdot 14 \bmod 17 = 182 \bmod 17 = 12$
$V_2 = g^{\,H(M)} \bmod p = 3^{\,13} \bmod 17 = 1594323 \bmod 17 = 12$
El certificado está autofirmado con la pareja de la clave que se certifica, por tanto es un certificado raíz y no hay cadena de certificación.

c) (ID_{AC}; e_{AC}, n_{AC}; $FEG_{AC}(Hash1(ID_{AC}, e_{AC}, n_{AC}))$) = ($ID_{AC}$; e_{AC}, n_{AC}; r, s)=(0; 7, 33; 10, 0)
Hash1(ID_{AC}; e_{AC}, n_{AC})= 0 + 7 + 33 mod 16 = 40 mod 16 = 8
$V_1 = Y_A^{\,r} \cdot r^{\,s} \bmod p = 13^{\,10} \cdot 10^{\,0} \bmod 17 = 16 \cdot 1 \bmod 17 = 16$
$V_2 = g^{\,H(M)} \bmod p = 3^{\,8} \bmod 17 = 6561 \bmod 17 = 16$

CRIPTOGRAFÍA Y SEGURIDAD INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

Examen Final Convocatoria Ordinaria Mayo 2013

Apellidos: **Nombre:**

NIA: **Grupo reducido:**

El certificado está firmado por AC pero con otras claves, por tanto habría que verificar también $Cert_{AC}$-Firmar (cadena de certificación), que es lo que se pedía en el apartado anterior.

PARTE 2

a) $e_A \cdot d_A \bmod \Phi(n_A) = 1$
$5 \cdot d_A \bmod \Phi(21) = 1$
$\Phi(21) = \Phi(3) \cdot \Phi(7) = 2 \cdot 6 = 12$
$d_A = 5$ ($5 \cdot 5 \bmod 12 = 25 \bmod 12 = 1$)

b) $Hash1(ID_A; e_A, n_A) = 0 + 5 + 21 \bmod 16 = 26 \bmod 16 = 10$
$FRSA_A(Hash1(ID_A; e_A, n_A)) = H(M)^{dA} \bmod n_A = 10^5 \bmod 21 = 19$

c) Solicitud = $(ID_A; e_A, n_A; FRSA_A(Hash1(ID_A, e_A, n_A))) = (0; 5, 21; 19)$.
A debería cifrar para AC cada uno de los elementos de la solicitud. Si denominamos a cada elemento como M_i, su cifrado $C(M_i)$ se calcularía como:

$C(M_i) = M_i^{eAC} \bmod n_{AC} = M_i^7 \bmod 33$

PARTE 3

a) $M_i = C(M_i)^{dAC} \bmod n_{AC} = C(M_i)^{dAC} \bmod 33$

En caso de calcular d_{AC}, su valor es 3 ($3 \cdot 7 \bmod 20 = 1$)

b) Recuperamos el resumen a partir de la firma:
$Hash1(ID_A; e_A, n_A) = FRSA_A(Hash1(ID_A; e_A, n_A))^{eA} \bmod n_A = 19^5 \bmod 21 = 10$
Calculamos el resumen a partir de los datos:
$Hash1(ID_A; e_A, n_A) = 0 + 5 + 21 \bmod 16 = 26 \bmod 26 = 10$

Coinciden, por tanto, la firma enviada por A sobre los datos que desea que se certifiquen es correcta.

c) $r = g^k \bmod p = 3^7 \bmod 17 = 11$

$s = (H(M) - X_A \cdot r) \cdot k^{-1} \bmod (p-1)$
$H(M) == Hash1(ID_A; e_A, n_A) = 10$ (calculado en el apartado anterior)
$k^{-1} \bmod (p-1) = 7^{-1} \bmod 16 = 7^7 \bmod 16 = 7$
$s = (10 - 4 \cdot 11) \cdot 7 \bmod 16 = -34 \cdot 7 \bmod 16 = 14 \cdot 7 \bmod 16 = 2$

$Cert_A = (ID_A; e_A, n_A; FEG_{AC}(Hash1(ID_A, e_A, n_A))) = (0; 5, 21; 11, 2)$

EJERCICIO 2 (1,5 puntos)

Solución:

a)

$10_{(2} = 2_{(10}$

$2^7 \bmod 55 = 18$

b)

$11000100_{(2} = 196_{(10}$

Alicia descifra 196 con su clave privada

ed =1 mod fi(n)

147d =1 mod fi(253)

fi(253)= fi(11)fi(23)= 10*22= 220

147d = 1 mod 220

220 = 1*147 + 73

147 = 2*73 + 1

1= 147 – 2*73

1= 147 – 2*(220-147)

1= 147 – 2*220 + 2*147 = 3*147 – 2*220

d= 3

$196^3 \bmod 253$ =196*196*196-29761*253 =3

S = 1011

c)

0110

0<1100

1<1000

1<0001

0<0010

0<0101

0<1011

1<0110

La serie cifrante tiene un period de 7. La secuencia 0110001 se repite

$3BC_{(16} = 0011\ 1011\ 1100_{(2}$

xor 0110 0010 1100

$0101\ 1001\ 0000_{(2} = 590_{(16}$

Alternativamente

$3BC_{(16} = 0011\ 1011\ 1100_{(2}$

xor 0011 0100 0110

$0000\ 1111\ 1010 = 0FA_{(16}$

d) El period es 7 y no es máximo porque es menor que 2^4 -1 = 15 que sería el máximo periodo que se puede alcanzar con un LFSR de 4 celdas.