Autenticación y Firma Digital

(Tema 6)

Confidencialidad v/s integridad

- Confidencialidad
  - Para lograrla se cifra el mensaje M obteniendo un criptograma C.
- Integridad
  - Para lograrla se firma un hash del mensaje h(M), añadiendo una marca al mensaje o criptograma.

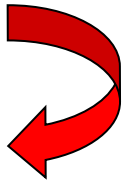
Si bien en ciertos escenarios es muy importante mantener el secreto de la información, si ésta lo requiere, en muchos casos tiene quizás más trascendencia el poder certificar la autenticidad entre cliente y servidor como ocurre en Internet.

Primer escenario de integridad

Escenario de desconfianza

1ª Solución. Uso de una tercera parte de confianza activa. Un juez tendrá una clave K_A con la que se comunica con A y una clave K_B con la que se comunica con B.

Usará criptografía simétrica



A envía un mensaje M a B:

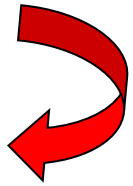
A cifra M con la clave $K_A \Rightarrow E_{KA}(M)$ y lo envía al juez. Este comprueba la integridad de A, lo descifra y envía a B, cifrado con K_B, el mensaje M, la identidad de A y la firma $E_{KA}(M)$: $E_{KB}\{M, A, E_{KA}(M)\}$. Ambos confían en el juez y ante cualquier duda éste puede desvelar la identidad de A descifrando $E_{KA}(M)$.

Segundo escenario de integridad

Escenario de desconfianza

2ª Solución. Uso de una tercera parte de confianza no siempre activa. Esta parte sólo actúa cuando se produce un conflicto entre los interlocutores, quienes se autentican a través de ella que les certifica.

Usará criptografía asimétrica



En este caso la figura del juez se conoce como una Autoridad de Certificación

Habrá una aceptación del sistema de autenticación por convencimiento propio y la confianza en los algoritmos.

Autenticación con sistemas asimétricos

Al existir una clave pública y otra privada que son inversas, se autentica el mensaje y al emisor.

Permite la firma digital, única para cada mensaje

Problema:

Los sistemas de cifra asimétricos son muy lentos y el mensaje podría tener miles o millones de bytes ...

Solución:

Se genera un resumen del mensaje, representativo del mismo, con una función hash imposible de invertir. La función hash comprime un mensaje de longitud variable a uno de longitud fija y pequeña.

Características de una firma digital

Requisitos de la firma digital:

a) Debe ser fácil de generar.
b) Será irrevocable, no rechazable por su propietario.
c) Será única, sólo posible de generar por su propietario.
d) Será fácil de autenticar o reconocer por su propietario y los usuarios receptores.
e) Debe depender del mensaje y del autor. → Esta última propiedad es muy importante pues protege la falsificación de un mensaje

Son condiciones más fuertes que la de una firma manuscrita.

Firma digital RSA de A hacia B

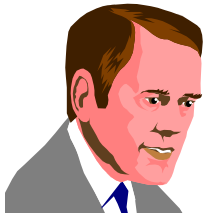
Clave Pública (n_A, e_A) Clave Privada (d_A)

Adela

Algoritmo:

Rúbrica: $r_A h(M) = h(M)^{dA} \bmod n_A$

A envía el mensaje M en claro (o cifrado) al destinatario B junto a la rúbrica: $\{M, r_A h(M)\}$



Benito

El destinatario B tiene la clave pública e_A, n_A de A y descifra $r_A h(M) \Rightarrow \{(h(M)^{dA})^{eA} \bmod n_A\}$ obteniendo así h(M). Como recibe el mensaje M', calcula la función hash h(M') y compara:

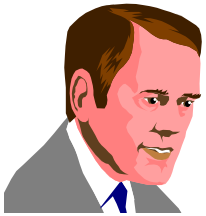
Si h(M') = h(M) se acepta la firma.

Valores y tamaños típicos de firmas

- En los siguientes ejemplos, por limitación del tamaño de los primos elegidos, se firmarán sólo bloques de una cantidad determinada de bits en función del cuerpo de trabajo.

- No obstante, en los sistemas reales esto no es así puesto que las funciones hash ya vistas entregarán -por lo general- resúmenes comprendidos entre 128 y 160 bits y, por otra parte, el cuerpo de trabajo de la cifra asimétrica para la firma digital será como mínimo de 512 bits (si bien en la actualidad se recomienda al menos 1.024). Por lo tanto el resumen que se firma es menor que el cuerpo de cifra o, lo que es lo mismo, es parte del conjunto de restos del grupo.

Ejemplo de firma digital RSA (B → A)



Benito

Hola. Te envío el documento. Saludos, Beni.

Adela

Sea h(M) = F3A9 (16 bits)

Claves Benito

$n_B = 65.669$

$e_B = 35$, $d_B = 53.771$

$2^{16} < 65.669 < 2^{17}$

Forzaremos firmar bloques de 16 bits

Claves Adela

$n_A = 66.331$

$e_A = 25$, $d_A = 18.377$

Firma

$h(M) = F3A9_{16} = 62.377_{10}$

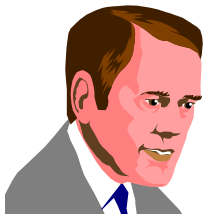
$r_{h(M)} = h(M)^{dB} \bmod n_B$

$r_{h(M)} = 62.377^{53.771} \bmod 65.669 = 24.622$

Benito envía el par (M, r) = (M, 24.622)

Nota: los primos que usa Benito son 97, 677 y Adela 113, 587

Comprobación la firma RSA por A



Benito

Claves Benito

$n_B = 65.669$

$e_B = 35$, $d_B = 53.771$

Claves Adela

$n_A = 66.331$

$e_A = 25$, $d_A = 18.377$

Adela

Teníamos que: $h(M) = F3A9_{16} = 62.377_{10}$

$r_{h(M)} = h(M)^{dB} \bmod n_B \; r_{h(M)} = 62.377^{53.771} \bmod 65.669 = 24.622$

Benito había enviado el par (M, r) = (M, 24.622)

Adela recibe un mensaje M' junto con una rúbrica r = 24.622:

- Calcula $r^{eB} \bmod n_B = 24.622^{35} \bmod 65.669 = 62.377$.
- Calcula el resumen de M' es decir h(M') y lo compara con h(M).
- Si los mensajes M y M' son iguales, entonces h(M) = h(M') y se acepta la firma como válida.
- NOTA: No obstante, h(M) = h(M') no implica que M = M'.

Firma digital ElGamal de A hacia B

Adela

ElGamal: El usuario A generaba un número aleatorio a (clave privada) del cuerpo p. La clave pública es $\alpha^a \bmod p$, con α generador.

Firma: (r, s)

Algoritmo de firma:

1º El usuario A genera un número aleatorio h, que será primo relativo con $\phi(p)$: $h / mcd\ \{h, \phi(p)\} = 1$

2º Calcula $h^{-1} = inv\ \{h, \phi(p)\}$

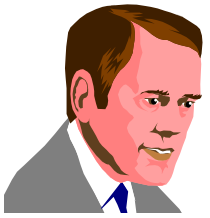
3º Calcula $r = \alpha^h \bmod p$

4º Resuelve la siguiente congruencia:

$M = a*r + h*s \bmod \phi(p) \quad \therefore$

$s = (M - a*r)*inv[h,\phi(p)] \bmod \phi(p)$

Comprobación de firma ElGamal por B



Benito

Conoce: p y (α^a) mod p

Algoritmo comprobación de firma:

1º El usuario B recibe el par (r, s) y calcula:

r^s mod p y $(\alpha^a)^r$ mod p

2º Calcula $k = [(\alpha^a)^r * r^s]$ mod p

Como r era igual a α^h mod p entonces:

$k = [(\alpha^{ar} * \alpha^{hs}]$ mod p = $\alpha^{(ar + hs)}$ mod p = α^β mod p

3º Como $M = (a*r + h*s) \bmod \phi(p)$ y α es una raíz primitiva de p se cumple que:

$\alpha^\beta = \alpha^\gamma$ ssi $\beta = \gamma \bmod (p-1)$

4º Comprueba que $k = \alpha^M$ mod p →

Si $k = [(\alpha^a)^r * r^s]$ mod p es igual a α^M mod p ...

Se acepta la firma

Ejemplo de firma ElGamal (B → A)

Benito

¡Hola otra vez! Soy Benito de nuevo. Saludos.

Adela

Sea h(M) = A69B (16 bits)

Claves Benito

$p_B = 79.903 \quad \alpha = 10$

$\alpha^b \bmod p = 3.631$

$b = 20, \quad h = 31$

$2^{16} < 79.903 < 2^{17}$

Forzaremos firmar bloques de 16 bits

Firma

1) $h^{-1} = inv[h, \phi(p)] = inv\ (31, 79.902) = 5.155$
2) $r = \alpha^h \bmod p = 10^{31} \bmod 79.903 = 11.755$
3) $s = [h(M) - b*r]*[inv(h,\phi(p)] \bmod \phi(p)$ $\quad h(M) = A69B_{16} = 42.651_{10}$
4) $s = [42.651-20*11.755]*5.155 \bmod 79.902$
5) $s = 68.539$ Luego, la firma será ⟶ (r, s) = (11.755, 68.539)

Comprobación de firma ElGamal por A

Benito

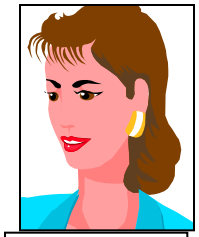
Claves Benito

$p_B = 79.903 \quad \alpha = 10$

$\alpha^b \bmod p = 3.631$

$b = 20, \quad h = 31$

$h(M) = A69B = 42.651$



Adela

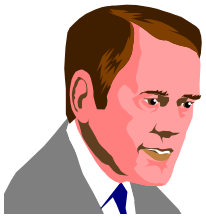
Adela recibe el par (r, s) = (11.755, 68.539)

Comprobación de la firma:

1) $r^s \bmod p = 11.755^{68.539} \bmod 79.903 = 66.404$
2) $(\alpha^b)^r \bmod p = 3.631^{11.755} \bmod 79.903 = 12.023$
3) $(\alpha^b)^r * r^s \bmod p = (12.023 * 66.404) \bmod 79.903 = 64.419 = k$
4) $\alpha^{h(M)} \bmod p = 10^{42.651} \bmod 79.903 = 64.419$

Como hay igualdad se acepta la firma

Importancia de α en la firma de ElGamal



Benito

Claves Benito

$p_B = 79.903 \quad \alpha = 10$

$\alpha^b \bmod p = 3.631$

$b = 20, \qquad h = 31$

$\alpha = 10$ es un generador del cuerpo $p = 79.903$ puesto que:

$p-1 = 79.902 = 2*3^2*23*193$

$q_1 = 2; \; q_2 = 3; \; q_3 = 23; \; q_4 = 193$

y se cumple $10^{(p-1)/qi} \bmod p \neq 1$

$10^{39.951} \bmod 79.903 = 79.902$

$10^{26.634} \bmod 79.903 = 71.324$

$10^{3.474} \bmod 79.903 = 2.631$

$10^{414} \bmod 79.903 = 41.829$

Si se elige $\alpha = 11$ que no es raíz, para el exponente 39.951 se obtiene el valor 1. No nos servirá para la firma porque será <u>imposible</u> comprobarla mediante la ecuación $k = \alpha^M \bmod p$.

Estándares de firma digital

El peor inconveniente de la firma propuesta por ElGamal es que duplica el tamaño del mensaje M al enviar un par (r, s) en Z_p y $\phi(p)$. No obstante, se solucionará con el algoritmo denominado DSS.

1991: National Institute of Standards and Technology (NIST) propone el DSA, Digital Signature Algorithm, una variante de los algoritmos de ElGamal y Schnoor.

1994: Se establece como estándar el DSA y se conoce como DSS, Digital Signature Standard.

1996: La administración de los Estados Unidos permite la exportación de Clipper 3.11 en donde viene inmerso el DSS, que usa una función hash de tipo SHS, Secure Hash Standard.

Digital Signature Standard DSS

Parámetros públicos de la firma:

- Un número primo p (512 bits < p < 1024 bits)
- Un número primo q (160 bits) divisor de p-1
- Un generador α "de orden q" del grupo p

Generador de orden q es aquella raíz α en el cuerpo Z_p de forma que q es el entero más pequeño que verifica:

$\alpha^q \bmod p = 1$ Así, para todo t: $\alpha^t = \alpha^{t \,(\bmod q)} \bmod p$

Elección de parámetros: primero se busca el primo p, luego un primo q que sea divisor de (p-1) y luego un valor h en p, de forma que si $\alpha = h^{(p-1)/q} \bmod q \neq 1$, éste será el generador.

Elección de parámetros en DSS

- Elegir un número primo 2^{159} bits < q < 2^{160} bits.
- Elegir $0 \leq t \leq 8$ y encontrar un número primo p que esté en $2^{511+64t} \leq p \leq 2^{512+64t}$ y que además q divida a (p-1).
- Elegir un generador α de orden q de la siguiente forma:
    - Elegir h, un elemento de p, de forma que se cumpla la condición $\alpha = h^{(p-1)/q} \bmod q \neq 1$.
- Elegir como clave privada un valor aleatorio a dentro del cuerpo del primo q.
- Calcular la clave pública $y = \alpha^a \bmod p$.
- Hacer público los parámetros q, p, α, y.

Generación de firma DSS (A $\rightarrow$ B)

- Valores públicos de A: primos p, q y el generador α
- Clave secreta de la firma: a $(1 \leq a \leq q)$ aleatorio
- Clave pública de la firma: $y = \alpha^a \bmod p$
- Para encontrar α y firmar un mensaje $1 \leq M \leq p$, A ya ha elegido un valor aleatorio $1 \leq h \leq q$.

Luego el firmante A calcula:

- $r = (\alpha^h \bmod p) \bmod q$
- $s = [(M + a*r) * inv\ (h,q)] \bmod q$

- La firma digital de A sobre M será el par (r, s)

Comprobación de firma DSS por B

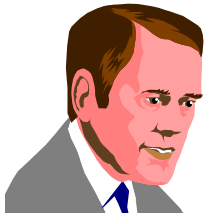
B recibe el par (r, s)

- Luego calcula:
  - $w = inv(s, q)$
  - $u = M * w \bmod q$
  - $v = r * w \bmod q$
- Comprueba que se cumple:
  - $r = (\alpha^u y^v \bmod p) \bmod q$
- Si se cumple, se acepta la firma como válida.

La firma DSS tendrá un tamaño menor que q al reducirse (r, s) a dicho módulo, siendo $q << p$.

Observe que la comprobación de la firma se hace sobre r, un valor en el que no interviene M.

Ejemplo de firma DSS (B → A)



Benito

Hola Adela, soy Benito y firmo con DSS.

Sea h(M) = 1101000 = 104 (un elemento de p_B)

Adela

Compruebe que son correctos α, h, p, q

Claves Benito

$p_B = 223$ $q_B = 37$ $\alpha = 17$

$y = \alpha^b \bmod p = 30$

$b = 25$, $h = 12$

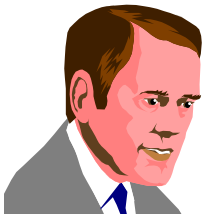
$2^8 < p_B = 223 < 2^7$

Forzaremos firmar bloques de 7 bits

Firma

1) inv (h,q) = inv (12, 37) = 34
2) $r = (\alpha^h \bmod p) \bmod q = (17^{12} \bmod 223) \bmod 37 = 171 \bmod 37 = 23$
3) $s = [h(M)+b*r]*[inv\ (h,q)] \bmod q = [104+25*23]*34 \bmod 37 = 35$
4) La firma digital de h(M) = 104 será: (r, s) = (23, 35)
5) Benito transmite a Adela el bloque: (M, r, s) = (M, 23, 35)

Comprobación de la firma DSS por A



Benito

Claves Benito

$p_B = 223$ $q_B = 37$ $\alpha = 17$

$y = \alpha^b \bmod p = 30$

$b = 25$, $h = 12$

Adela recibe el bloque:

$(M, r, s) = (M, 23, 35)$

Adela

¿Igualdad?

Comprobación de firma

1) $w = inv\ (s, q) = inv\ (35, 37) = 18$
2) $u = M*w \bmod q = 104*18 \bmod 37 = 22$
3) $v = r*w \bmod q = 23*18 \bmod 37 = 7$
4) $¿(\alpha^u\ y^v \bmod p) \bmod q = r\ ?$
5) $[(17^{22}30^7) \bmod 223] \bmod 37 = 23$

En caso afirmativo, se acepta la firma

Y el tamaño será menor que $q_B = 37$ es decir << $p_B = 223$ que era precisamente el punto débil del sistema de ElGamal.

Seguridad de los 160 bits de q

- Podríamos pensar que al bajar el número de bits de 1024 en la firma de ElGamal a sólo 160 (el valor de q) en la firma DSS la seguridad de dicha firma está comprometida.
- No obstante, la firma DSS tiene la misma fortaleza que la de ElGamal ya que q es un subgrupo de p. Eso quiere decir que para resolver el problema del logaritmo discreto en q, habrá que hacerlo obligatoriamente en p.
- Para evitar diversos ataques tanto en la firma ElGamal como en DSS, deberá firmarse siempre una función hash.
- DSS requiere el uso del hash SHA-1 sobre M, h(M).

Mensajes que no pueden firmarse

No todos los valores h(M) podrán firmarse con DSS.

- Para comprobar la firma en recepción se calcula el valor w = inv (s, q), donde s = [h(M)+b∗r]∗[inv (h,q)] mod q. Luego, debe existir dicho inverso.
- Si s = 0 no existe el inverso. Luego esta condición deberá comprobarse en emisión antes de proceder a la firma.
- No obstante, la probabilidad de que se dé esta situación es muy baja, del orden de 0.5^{160}.
- Así mismo, en emisión deberá verificarse que r ≠ 0. En ambos casos se elegirá un nuevo valor de h.