Criptografía de clave pública

Criptografía asimétrica

Fundamentos de los Criptosistemas de Clave Pública

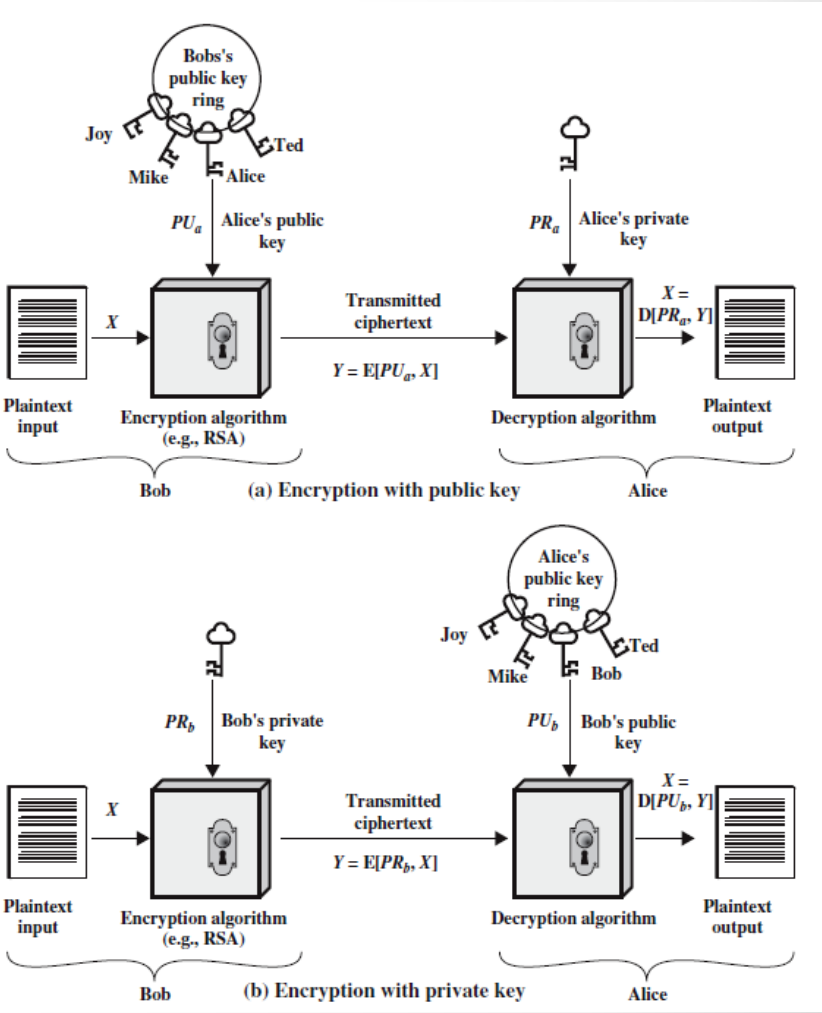
Criptosistemas de Clave Pública

- Conceptos **erróneos**:
  - Clave pública es más segura que criptografía simétrica
    - La **seguridad** depende de la **longitud de clave** y el **coste computacional** asociado
  - Clave pública es de propósito general y hace la criptografía simétrica obsoleta
    - Es **extremadamente lenta** y sólo se utiliza para distribuir claves y firmar digitalmente
  - La distribución de claves es trivial en clave pública
    - Es necesario un **protocolo** para la **distribución** de **claves**

- Surge para atajar dos de los problemas más difíciles de la criptografía simétrica:
  - Distribución de claves
  - Firma digital
- Diffie y Hellman
  - Proponen las funciones unidireccionales, fáciles de calcular en una dirección pero muy costosas en la otra; descubrimiento rompedor en 1976 que soluciona ambos problemas
- Se usan dos claves: una clave para el cifrado y otra, distinta pero relacionada, para el descifrado

Criptosistemas de Clave Pública

- Características:
  - Es difícil determinar la clave de descifrado a partir de la de cifrado (y viceversa)
  - En algunos algoritmos, cualquiera de las dos claves puede ser usada para el cifrado y la otra para el descifrado



Criptosistemas de Clave Pública

1. Cada usuario genera un par de claves asociadas
2. Cada usuario pone una de sus claves en un fichero público y mantiene en secreto la otra
3. Si Bob desea enviar un mensaje a Alice, lo cifra con su clave pública
4. Cuando Alice recibe el mensaje, lo descifra con su clave privada; nadie más puede descifrar el mensaje

- Con este enfoque, todos los participantes tienen acceso a las claves públicas de los demás
- Sólo el propio usuario tiene acceso a su clave privada
- Un usuario puede remplazar su clave privada y publicar su clave pública asociada en cualquier momento
  (recomendable de forma periódica)

Criptosistemas de Clave Pública

Cifrado simétrico

- Necesario para funcionar:
  1. Se usa el mismo algoritmo con la misma clave para el cifrado y el descifrado
  2. Emisor y receptor deben compartir algoritmo y clave
- Necesario por seguridad:
  1. La clave debe mantenerse en secreto
  2. Debe ser impráctico descifrar el mensaje si no se conoce nada más
  3. El conocimiento del algoritmo y de textos cifrados debe ser insuficiente para determinar la clave

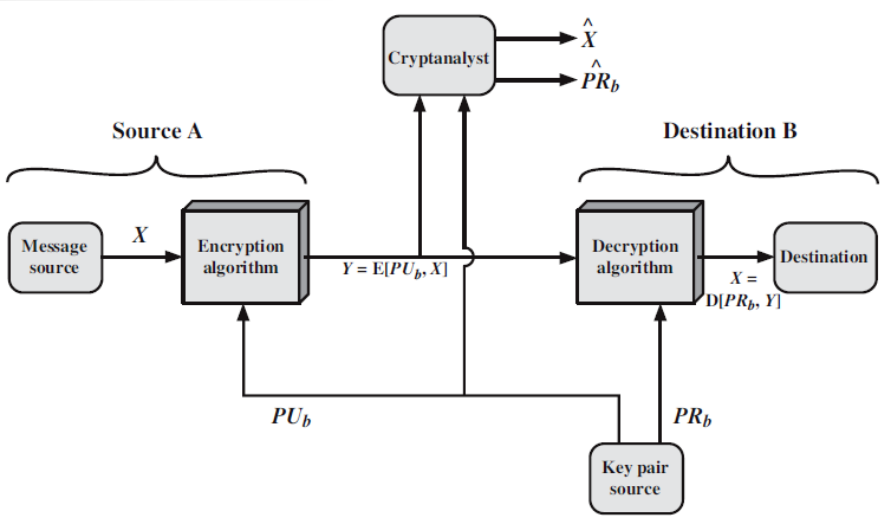
Cifrado de clave pública

- Necesario para funcionar:
  1. Se usa un algoritmo con dos claves distintas, una para el cifrado y otra para el descifrado
  2. Emisor y receptor deben tener una de las claves asociadas (no la misma)
- Necesario por seguridad:
  1. Una de las dos claves debe mantenerse en secreto
  2. Debe ser impráctico descifrar el mensaje si no se conoce nada más
  3. El conocimiento del algoritmo, una de las claves y de algunos textos cifrados debe ser insuficiente para determinar la otra clave

Criptosistemas de Clave Pública

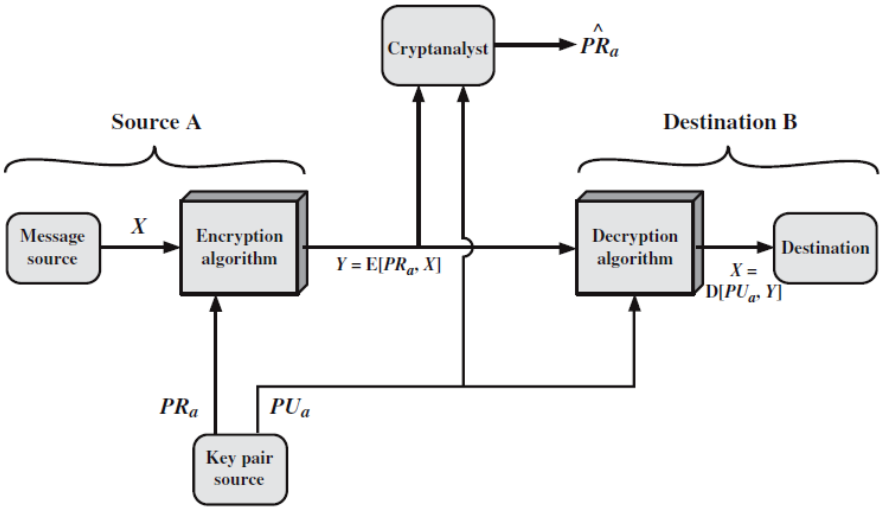
Privacidad

se cifra con la clave pública de B (destino),
se descifra con la clave privada de B



Autentificación

se cifra con la clave privada de A (origen),
se descifra con la clave pública de A



Aplicaciones

- Cifrado/descifrado
  - El emisor cifra un mensaje usando la clave pública del receptor
- Firma digital
  - El emisor firma un mensaje con su clave privada
- Intercambio de claves
  - Ambas partes cooperan para establecer una clave de sesión

- Algunos algoritmos sirven para las 3 aplicaciones mientras que otros sólo para algunas
- Dependiendo de la aplicación, el emisor utiliza su clave privada, la clave pública del receptor o ambas para realizar algún tipo de función criptográfica

Requisitos

1. Es fácil para B generar una pareja de claves (PU_b, PR_b)

2. Es fácil para A, conociendo la clave pública y el mensaje a cifrar, generar el texto cifrado $C = E(PU_b, M)$

3. Es fácil para B descifrar dicho criptograma usando su clave privada para recuperar M
   $M = D(PR_b, C)$

4. Es impráctico para un adversario, conociendo la clave pública PU_b, determinar la clave privada PR_b

5. Es impráctico para un adversario, conociendo la clave pública PU_b y el criptograma C, determinar el mensaje M

6. *Útil pero no necesario*: las claves se pueden aplicar en cualquier orden
   $M=D[PU_b,E(PR_b,N)]$
   $=D[PR_b,E(PU_b,M)]$

Criptoanálisis

- Son **vulnerables** a ataques por **fuerza bruta**
  - Se ha de emplear claves largas (dependen del algoritmo en concreto)
  - Existe un equilibrio entre el tamaño de clave y el coste computacional asociado
  - El tamaño de clave actual (y coste) implica que no son válidos para propósito general

- **Derivar la clave** privada a partir de la clave pública
  - No se ha demostrado matemáticamente que este ataque sea impráctico para ningún algoritmo de clave pública (incluido RSA)
  - La historia indica que un problema que parece irresoluble puede ser resuelto en el futuro desde otra perspectiva

- Ataque de **mensaje probable**
  - Suponiendo que se enviara una clave de DES (56 bits), el atacante sólo tiene que hacer una fuerza bruta de 56 bits para encontrar la clave, cifrándolas todas con la clave pública

El Algoritmo RSA

RSA

- Diseñado por Rivest, Shamir y Adleman en 1978
- Se apoya en que la **exponenciación modular** es una función unidireccional bajo ciertas condiciones
- Se basa en el problema de la **factorización** de un número con un gran número de cifras en sus factores primos
- La seguridad de RSA radica precisamente en la dificultad de la factorización de números grandes:
  - Es fácil saber si un número es primo *(de forma probabilística)*, pero muy costoso computacionalmente obtener la factorización en números primos de una potencia

RSA

- De entre todos los algoritmos asimétricos, quizá sea el más sencillo de comprender e implementar
- Sus claves sirven indistintamente tanto para cifrar como para autentificar
- Estuvo bajo patente de RSA Labs hasta el año 2000
- Nadie ha conseguido demostrar o rebatir su seguridad
- Se considera uno de los algoritmos más seguros
- Las claves pública y privada se calculan a partir de un número que se obtiene como producto de dos primos grandes
- Un atacante se enfrenta a un problema de factorización de un coste computacional muy elevado

RSA

- Para generar un par de claves se eligen dos primos grandes p y q y se calcula $n = pq$
- Se escoge un número natural $e, 0 < e < \Phi(n)$, primo con $\Phi(n)$:
$$mcd\big(e, \Phi(n)\big) = 1$$
- Sabemos que existe un número d que es el inverso de e
$$d \cdot e \; mod \; \Phi(n) = 1$$
- La clave pública es (n, e) y la función de cifrado
$$c = E_k(m) = m^e \; mod \; n$$
- La clave privada es (n,d) y la función de descifrado
$$m = D_k(c) = c^d \; mod \; n$$
- Han de permanecer en secreto también p y q

RSA

- En la práctica, se calculan las claves en secreto en la máquina en la que se va a guardar la clave privada
- Conviene cifrar la clave privada con algún criptosistema simétrico
- RSA permite longitud de clave variable, siendo recomendable 1024 bits o más
  (512 bit se rompe fácil)

- Existen dos posibles ataques:
  - Fuerza bruta, probar todas las claves privadas posibles
  - Factorizar n en números primos
- RSA presenta todas las ventajas de los criptosistemas de clave pública
  - Intercambio de claves
  - Firma digital
  - Cifrado y descifrado

Otros Algoritmos de Clave Pública

Algoritmo Diffie-Hellman

- El objetivo del algoritmo es permitir el intercambio de claves de forma segura
- Se basa en la dificultad de calcular el logaritmo discreto

$$b = a^i \ mod \ p, \qquad 0 \leq i \leq p-1$$

- Se conoce a i como el logaritmo discreto de b con base a, módulo p

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

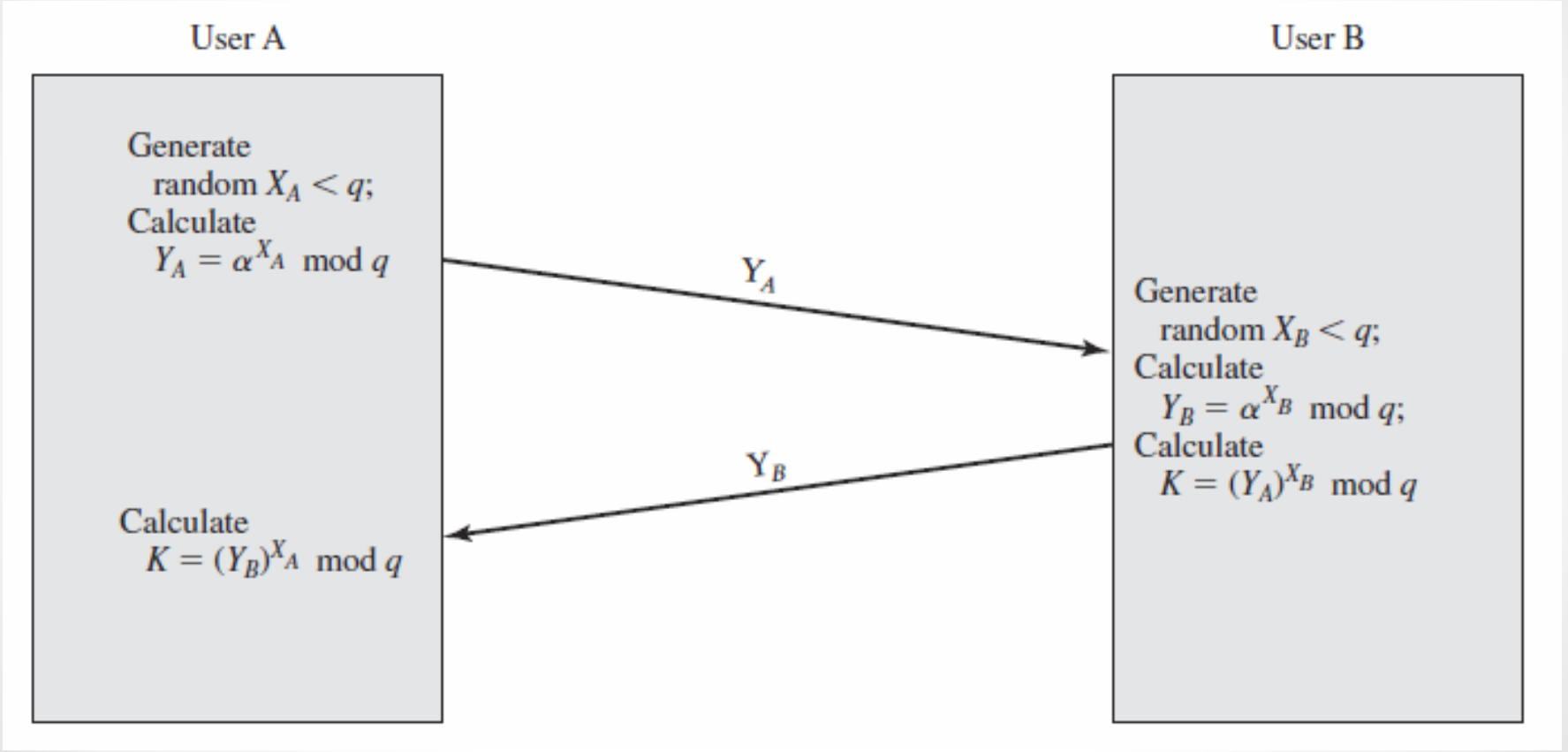
User A Key Generation	
Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{XA} \bmod q$

User B Key Generation	
Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{XB} \bmod q$

Calculation of Secret Key by User A
$K = (Y_B)^{XA} \bmod q$

Calculation of Secret Key by User B
$K = (Y_A)^{XB} \bmod q$

Esquema Diffie-Hellman



Ataque MITM

1. D prepara un ataque, generando dos claves privadas y sus claves públicas asociadas
2. A transmite Y_A a B
3. D intercepta Y_A y su clave alternativa (Y_{DA}) a B.
4. B recibe Y_{DA} y calcula K_B
5. B transmite Y_B a A
6. D intercepta Y_B y transmite Y_{DA} a A.
7. A recibe Y_{DA} y calcula K_A

- En este punto A y B creen que comparten una clave pero en realidad la comparten con D
- D puede renviar los mensajes leyéndolos, o modificándolos a su antojo
- El protocolo de intercambio de claves es vulnerable porque no autentifica a las partes, requiriendo elementos externos:
  - Firmas digitales
  - Autoridades certificadores
  - Información precompartida

ElGamal

- Anunciado por T. ElGamal en 1984
- Relacionado con Diffie-Hellman, basado en logaritmos discretos
- Se utiliza en varios estándares: DSS, S/MIME, etc.

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

Key Generation by Alice	
Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{XA} \bmod q$
Public key	$PU = \{q, \alpha, Y_A\}$
Private key	X_A

Encryption by Bob with Alice's Public Key	
Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)^k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Decryption by Alice with Alice's Private Key	
Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{XA} \bmod q$
Plaintext:	$M = (C_2 K^{-1}) \bmod q$

Curvas Elípticas

- Se basan en una aritmética especial sobre cierto tipo de expresiones matemáticas que permite una reducción significativa del tamaño de clave necesario.

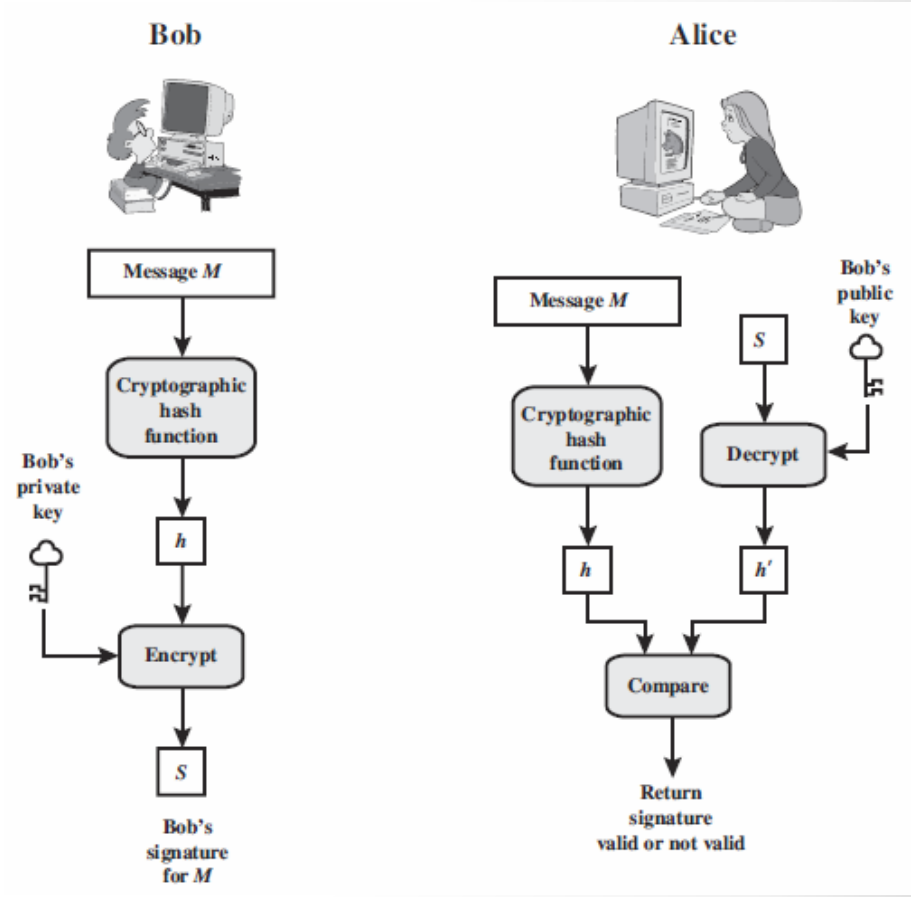
- Recientemente, se ha descubierto que la NSA ha propiciado el uso de curvas elípticas débiles en algunos estándares, dañando la imagen de seguridad de los algoritmos basados en esta técnica.

Symmetric Scheme (key size in bits)	ECC-Based Scheme (size of *n* in bits)	RSA/DSA (modulus size in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

Firma Digital

Propiedades

- Verifica el autor, así como la fecha y hora de la firma
- Autentifica el contenido en el momento de la firma
- Es verificable por terceras partes para resolver disputas



Ataques

- Sólo con clave
  - C sólo conoce la clave pública de A
- Mensaje conocido
  - C conoce un conjunto de mensajes y sus firmas
- Mensaje elegido genérico
  - C elige una lista de mensajes y obtiene sus firmas. No depende de la clave pública de A
- Mensaje elegido dirigido
  - Similar al genérico pero los mensajes se eligen una vez se conoce la clave pública de A
- Mensaje elegido adaptativo
  - C utiliza A como *oráculo*. C puede obtener firmas de mensajes que dependen de otros mensajes y firmas anteriores

Se define el éxito en el ataque como

- Rotura Total
  - C obtiene la clave privada de A
- Falsificación universal
  - C encuentra un algoritmo de firma eficiente que es equivalente a la firma de A
- Falsificación selectiva
  - C logra falsificar la firma para un mensaje en concreto de su elección
- Falsificación existencial
  - C falsifica la firma de un mensaje pero no tiene elección del mismo

Requisitos

- La firma debe ser un patrón de bits que dependa del mensaje firmado
- La firma debe usar información exclusiva del emisor para evitar la falsificación y el repudio
- Debe ser fácil producir la firma digital
- Debe ser fácil reconocer y verificar la firma digital
- Debe ser impráctico falsificar una firma digital, bien construyendo un mensaje para una firma concreta o falsificando la firma para un nuevo mensaje
- Debe ser práctico almacenar una copia de la firma digital

Firma Digital Directa

- Consiste en el esquema de firma digital que involucra únicamente a las partes comunicantes *(fuente, destino)*
- La confidencialidad se puede lograr cifrando el mensaje junto con su firma mediante criptografía simétrica
- La validez de este esquema depende de la seguridad de la clave privada. El emisor puede declarar que fue robada *(repudio)*
- Si un atacante roba la clave privada, puede generar mensajes firmados con anterioridad
- La solución
  - Autoridades certificadoras

Estándar de Firma Digital (DSS)

- El NIST propone en 1991 un estándar de firma digital (DSS) y su algoritmo correspondiente (DSA)
- El DSA propuesto es una variante de ElGamal
- El DSS, es parte de un proyecto más amplio del NIST y la NSA (CAPSTONE)
- CAPSTONE es un proyecto para desarrollar un estándar de clave pública, consta de 4 elementos:
  - Algoritmo de cifrado, Skipjack (no es público)
  - Estándar de firma digital (DSS y DSA)
  - Un protocolo de intercambio de clave (no anunciado)
  - Estándar de función hash (SHS y SHA)

Estándar de Firma Digital (DSS)

- El DSA propuesto es una variante de ElGamal que evita la duplicación del tamaño del mensaje

- Cada usuario elige los parámetros:
  - Un primo grande p (512 bits) entre 2^{511} y 2^{512}
  - Un primo q (160 bits) divisor de p-1 entre 2^{159} y 2^{160}
  - Una raíz α
  - Su clave privada es *x*, un entero entre 0 y q
  - Su clave pública es $y = \alpha^x \ mod \ p$

Generación y Verificación (DSS)

- Sea h una función hash y m el mensaje a firmar
- Para **generar** la firma digital, se elige un entero k entre 1 y q y se calculan

$$r = \left(\alpha^k mod\ p\right) mod\ q$$
$$s = [h(m) + x \cdot r] \cdot k^{-1}\ mod\ q$$

- La firma digital es el par (r, s)
- Para **verificar** la firma digital se calcula

$$w = s^{-1}\ mod\ q$$
$$u = h(m) \cdot w\ mod\ q$$
$$v = r \cdot w\ mod\ q$$

- Y se comprueba

$$r = (\alpha^u y^v\ mod\ p)\ mod\ q$$

Ampliación

Otros materiales

- Se puede consultar los capítulos 12, 17 y 18 del libro de Lucena
  (en los materiales de UACloud)

- También se puede consultar los capítulos 8 y 11 de *"Handbook of Applied Cryptography"*
  (más avanzado y en inglés)

Cuestiones

- ¿Por qué no usamos criptografía de clave pública para propósito general y qué alternativas tenemos a nuestra disposición?

- ¿Qué papel tienen las funciones hash en los protocolos de firma digital?

- ¿Qué diferencia hay entre un certificado y un par de claves pública/privada?