Criptografía Asimétrica III

M.I. GONZÁLEZ VASCO / GRADO EN INGENIERÍA DE LA CIBERSEGURIDAD

UNIVERSIDAD REY JUAN CARLOS

Qué vamos a aprender

1. Agunos ejemplos nuevos de cifrado
2. Firma digital

(Capítulos 11 (ejemplos), 19 (sección 2, Merkle-Hellman y 14 (firmas) del Smart)

1. Otros ejemplos de cifrado

Cifrado de El Gamal

- Gen(1^n)
  - Se eligen (G,q,g) como en el intercambio de claves de Diffie-Hellman.
  - $x \leftarrow Z_q$
  - $h := g^x$
  - pk:=(G,q,g,h)
  - sk:=(G,q,g,x)

Cifrado de El Gamal

- $Enc_{pk}(m)$
  - $y \leftarrow Z_q$
  - $c_1 := g^y$
  - $c_2 := h^y\ m$
  - Devuelve $c := (c_1, c_2)$
- $Dec_{sk}(c)$
  - Devuelve $m := c_2 / c_1^x$

Cifrado de El Gamal: corrección

Supongamos $(c_1,c_2)= Enc_{pk}(m)=(g^y, h^y\ m)$

Entonces

$c_2/c_1^x = h^y\ m\ /\ (g^y)^x = (g^x)^y\ m\ /\ (g^y)^x = m$

¿Seguridad?

- ElGamal es seguro IND-CPA (bajo la hipótesis DDH)
- ElGamal no es seguro en el sentido CCA2

Esquema Merkle-Hellman

- Basado en el problema “Knapsack” o “de la mochila” (seguridad OW-CPA solo..)
- No es IND-CPA
- No es OW-CCA2

Esquema Merkle-Hellman

- Gen(1^n)
  - Se elige una secuencia de enteros super creciente $a_1 \ldots a_n$
  - Se transforma en una secuencia "normal", eligiendo N y M, primos entre sí y calculando

    $$b_i = Na_i \mod N$$
  - La clave pública pk es la secuencia $b_1 \ldots b_n$
  - sk:=($a_1 \ldots a_n$) (o bien N,M)

Cifrado

- $Enc_{pk}(m)$
  - Rompemos m en bloques de n bits, supongamos que uno de salida es: $m_1 \ldots m_n$
  - Construimos un entero C sumando los elementos de la secuencia pública con bit asociado igual a 1

  Es decir, $c = m_1 b_1 + \ldots + m_n b_n$
- $Dec_{sk}(c)$; recuperar la secuencia de bits original utilizando la secuencia super-creciente (y la solución de su knapsack asociado)
  - ver ejemplo (Smart, p. 305)

2. Firma Digital

Uso

- Sirve para garantizar la **integridad** de los mensajes, es decir, da una prueba al receptor de que el mensaje no ha sido modificado por un adversario.
- Alice genera un par (pk,sk). Para firmar un mensaje m que quiere enviar, usa sk para producir una firma $\sigma(m)$.
- Cualquiera que conozca pk puede comprobar si $\sigma(m)$ es una firma válida de m relativa a pk.

Sintaxis de un esquema de firma

- Consiste en tres algoritmos: (Gen, Sign, Vrfy)
- $(pk,sk) \leftarrow Gen(1^n)$
- $\sigma \leftarrow Sign_{sk}(m)$
- $b:=Vrfy_{pk}(m, \sigma)$, donde

  b=1 significa válido

  b=0 significa no válido
- Corrección: definición estándar

Firma RSA de libro de texto

- $Gen(1^n)$
  - $(N,e,d) \leftarrow GenRSA(1^n)$
  - $pk:=(N,e)$
  - $sk:=(N,d)$
- $\sigma = Sign_{sk}(m):=[m^d \bmod N]$
- $Vrfy_{pk}(\sigma,m)=1$ si y solo si

$$m=[\sigma^e \bmod N]$$

Ataques

- **No-message attack**:
  - Elegir σ cualquiera
  - Calcular $m := \sigma^e$
  - (σ,m) es un par firma-mensaje válido
- Un adversario (con acceso a un oráculo de firma) quiere firmar m:
  - Obtiene σ1 firma de m1
  - Obtiene σ2 firma de m2:=m/m1 mod N
  - σ:= σ1·σ2 es una firma válida de m

Paradigma *hash-and-sign*

- Idea: cualquier esquema de firma se puede modificar con una función hash resistente a colisiones.
- Antes de firmar: calcular el hash del mensaje y firmar ese valor.
- Para verificar: calcular el hash del mensaje y ejecutar el algoritmo de verificación del esquema original con ese valor.
- Ventajas del paradigma:
  - Eligiendo bien la función hash, al menos mantiene (y puede mejorar) la seguridad del esquema original.
  - Permite firmar mensajes de longitud arbitraria.

RSA Full Domain Hash (RSA-FDH)

- Se dispone de una función hash $H:\{0,1\}^* \rightarrow Z_N^*$
- Para firmar $m \in \{0,1\}^*$ se calcula

  $\sigma := [H(m)^d \bmod N]$
- Para verificar se comprueba si

  $\sigma^e = [H(m) \bmod N]$

DSA

- Gen(1^n) produce:
  - p y q primos, $|q|=n$, $p=2q+1$
  - g generador de un subgrupo de orden q de Z_p^*
  - $H:\{0,1\}^* \rightarrow Z_q$ una función hash
  - $x \in Z_q$
  - $y:=g^x$
  - pk:=(H,p,q,g,y)
  - sk:=(H,p,q,g,x)

DSA

- $Sign_{sk}(m)$ calcula
  - $k \in Z_q$
  - $r:=[[g^k \bmod p] \bmod q]$
  - $s:=[(H(m)+x \cdot r)/k \bmod q]$
  - $\sigma:=(r,s)$

DSA

- $Vrfy_{pk}(m,\sigma)$ calcula
  - $\sigma = (r,s)$
  - $u:=[H(m)/s \bmod q]$
  - $v:=[r/s \bmod q]$

  y comprueba si

  $[[g^u y^v \bmod p] \bmod q]=r$

  Si se cumple devuelve válido, en caso contrario, devuelve no válido.

Recomendaciones

Table 5.5: Public Key Based Scheme Summary Table

Scheme	Classification Legacy	Classification Future	Notes
Public Key Encryption/Key Encapsulation			
RSA-OAEP	✓	✓	See text
RSA-KEM	✓	✓	See text
PSEC-KEM	✓	✓	See text
ECIES-KEM	✓	✓	See text
RSA-PKCS# 1 v1.5	✗	✗	
Public Key Signature Schemes			
RSA-PSS	✓	✓	See text
ISO-9796-2 RSA-DS2	✓	✓	Message recovery variant of RSA-PSS
PV Signatures	✓	✓	ISO 14888-3 only defines these for a finite field
(EC)Schnorr	✓	✓	See text
(EC)KDSA	✓	✓	See text
XMSS	✓	✓	See text
RSA-PKCS# 1 v1.5	✓	✗	No security proof
RSA-FDH	✓	✗	Issues in instantiating the required hash function
ISO-9796-2 RSA-DS3	✓	✗	Similar to RSA-FDH
(EC)DSA,(EC)GDSA	✓	✗	Weak provable security guarantees
(EC)RDSA	✓	✗	Weak provable security guarantees
ISO-9796-2 RSA-DS1	✗	✗	Attack exists (see notes)

Tamaño de claves

Table 4.6: Key Size Analysis. A * notes the value could be smaller due to specific protocol or system reasons, the value given is for general purposes.

			Future System Use	
	Parameter	Legacy	Near Term	Long Term
Symmetric Key Size	k	80	128	256
Hash Function Output Size	m	160	256	512
MAC Output Size*	m	80	128	256
RSA Problem	$\ell(n) \geq$	1024	3072	15360
Finite Field DLP	$\ell(p^n) \geq$	1024	3072	15360
	$\ell(p), \ell(q) \geq$	160	256	512
ECDLP	$\ell(q) \geq$	160	256	512
Pairing	$\ell(p^{k \cdot n}) \geq$	1024	6144	15360
	$\ell(p), \ell(q) \geq$	160	256	512