CÓDIGOS Y CRIPTOGRAFÍA

Cifrado RSA.
Autentificación de firma

RSA: Cifrado de clave pública

- En 1978 Rivest, Shamir y Adleman idearon un sistema criptográfico que se conoce como RSA en su honor.

RSA: Cifrado de clave pública

- En 1978 Rivest, Shamir y Adleman idearon un sistema criptográfico que se conoce como RSA en su honor.
- Es un sistema de clave pública y por tanto asimétrica.

RSA: Cifrado de clave pública

- En 1978 Rivest, Shamir y Adleman idearon un sistema criptográfico que se conoce como RSA en su honor.
- Es un sistema de clave pública y por tanto asimétrica.
- El receptor genera tanto la clave pública como la privada y el emisor sólo conocerá la pública.

RSA: Cifrado de clave pública

- En 1978 Rivest, Shamir y Adleman idearon un sistema criptográfico que se conoce como RSA en su honor.
- Es un sistema de clave pública y por tanto asimétrica.
- El receptor genera tanto la clave pública como la privada y el emisor sólo conocerá la pública.
- La función matemática de un sólo sentido en la que se basa es la descomposición en factores primos.

RSA: Cifrado de clave pública

- En 1978 Rivest, Shamir y Adleman idearon un sistema criptográfico que se conoce como RSA en su honor.
- Es un sistema de clave pública y por tanto asimétrica.
- El receptor genera tanto la clave pública como la privada y el emisor sólo conocerá la pública.
- La función matemática de un sólo sentido en la que se basa es la descomposición en factores primos.
- Supongamos que A es el receptor y B el emisor.

RSA: Cifrado de clave pública

- En 1978 Rivest, Shamir y Adleman idearon un sistema criptográfico que se conoce como RSA en su honor.
- Es un sistema de clave pública y por tanto asimétrica.
- El receptor genera tanto la clave pública como la privada y el emisor sólo conocerá la pública.
- La función matemática de un sólo sentido en la que se basa es la descomposición en factores primos.
- Supongamos que A es el receptor y B el emisor.

Pasos del método

1. Generación de claves (A)
2. Cifrado (B)
3. Descifrado (A)

RSA: Generación de claves (A)

- A elige dos números primos grandes p y q.
  Los va a mantener en secreto.

RSA: Generación de claves (A)

- A elige dos números primos grandes p y q.
  Los va a mantener en secreto.
- Halla $n = p \cdot q$.

RSA: Generación de claves (A)

- A elige dos números primos grandes p y q.
  Los va a mantener en secreto.
- Halla $n = p \cdot q$.
- Calcula $\varphi(n) = (p-1)(q-1)$.

RSA: Generación de claves (A)

- A elige dos números primos grandes p y q.
  Los va a mantener en secreto.
- Halla $n = p \cdot q$.
- Calcula $\varphi(n) = (p-1)(q-1)$.
- Elige $e < \varphi(n)$ de modo que $M.C.D.(e, \varphi(n)) = 1$.
  O, equivalentemente, $M.C.D.(e, p-1) = M.C.D.(e, q-1) = 1$.

RSA: Generación de claves (A)

- A elige dos números primos grandes p y q.
  Los va a mantener en secreto.
- Halla $n = p \cdot q$.
- Calcula $\varphi(n) = (p-1)(q-1)$.
- Elige $e < \varphi(n)$ de modo que $M.C.D.(e, \varphi(n)) = 1$.
  O, equivalentemente, $M.C.D.(e, p-1) = M.C.D.(e, q-1) = 1$.
- Halla d tal que $1 < d < \varphi(n)$ de modo que $d = e^{-1} \bmod \varphi(n)$.

CLAVE PÚBLICA: (n, e).

CLAVE PRIVADA: (n, d).

RSA: Generación de claves (A)

- A elige dos números primos grandes p y q.
  Los va a mantener en secreto.
- Halla $n = p \cdot q$.
- Calcula $\varphi(n) = (p-1)(q-1)$.
- Elige $e < \varphi(n)$ de modo que $M.C.D.(e, \varphi(n)) = 1$.
  O, equivalentemente, $M.C.D.(e, p-1) = M.C.D.(e, q-1) = 1$.
- Halla d tal que $1 < d < \varphi(n)$ de modo que $d = e^{-1} \bmod \varphi(n)$.

CLAVE PÚBLICA: (n, e).

CLAVE PRIVADA: (n, d).

* Difícil calcular d.
* Clásicamente: $e = F_4 = 1 + 2^{2^4} = 65537$ (Primo de Fermat).

RSA: Generación de claves (A)

- A elige dos números primos grandes p y q. $(p = 643$ y $q = 11)$
  Los va a mantener en secreto.
- Halla $n = p \cdot q$. $(n = 7073)$
- Calcula $\varphi(n) = (p-1)(q-1)$. $(\varphi(n) = 6420)$
- Elige $e < \varphi(n)$ de modo que $M.C.D.(e, \varphi(n)) = 1$.
  O, equivalentemente, $M.C.D.(e, p-1) = M.C.D.(e, q-1) = 1$.
  $(e = 31)$
- Halla d tal que $1 < d < \varphi(n)$ de modo que $d = e^{-1} \bmod \varphi(n)$.
  $(d = 2071)$

CLAVE PÚBLICA: (n, e). $((7073, 31))$

CLAVE PRIVADA: (n, d). $((7073, 2071))$

- Difícil calcular d.
- Clásicamente: $e = F_4 = 1 + 2^{2^4} = 65537$ (Primo de Fermat).

RSA: Cifrado (B)

- Supongamos que B quiere enviar un mensaje M a A.

RSA: Cifrado (B)

- Supongamos que B quiere enviar un mensaje M a A.
- Convierte numéricamente cada caracter de M, usando la identificación con $\mathbb{Z}_{27}$, pero empleando dos dígitos por caracter.

a	b	c	d	e	f	g	h	i	j	k	l	m	n
00	01	02	03	04	05	06	07	08	09	10	11	12	13
$\tilde{n}$	o	p	q	r	s	t	u	v	w	x	y	z	
14	15	16	17	18	19	20	21	22	23	24	25	26	

RSA: Cifrado (B)

- Supongamos que B quiere enviar un mensaje M a A.
- Convierte numéricamente cada caracter de M, usando la identificación con $\mathbb{Z}_{27}$, pero empleando dos dígitos por caracter.

a	b	c	d	e	f	g	h	i	j	k	l	m	n
00	01	02	03	04	05	06	07	08	09	10	11	12	13
$\tilde{n}$	o	p	q	r	s	t	u	v	w	x	y	z	
14	15	16	17	18	19	20	21	22	23	24	25	26	

- Agrupa en bloques, asegurando que el número de cada bloque sea menor que n (por ejemplo dígitos$(n) - 1$).

RSA: Cifrado (B)

- Supongamos que B quiere enviar un mensaje M a A.
- Convierte numéricamente cada caracter de M, usando la identificación con $\mathbb{Z}_{27}$, pero empleando dos dígitos por caracter.

a	b	c	d	e	f	g	h	i	j	k	l	m	n
00	01	02	03	04	05	06	07	08	09	10	11	12	13
$\tilde{n}$	o	p	q	r	s	t	u	v	w	x	y	z	
14	15	16	17	18	19	20	21	22	23	24	25	26	

- Agrupa en bloques, asegurando que el número de cada bloque sea menor que n (por ejemplo dígitos$(n) - 1$).
- Completamos el último bloque con 0 o 30s para fijar el tamaño del bloque.

RSA: Cifrado (B)

- Supongamos que B quiere enviar un mensaje M a A. (Viernes)
- Convierte numéricamente cada caracter de M, usando la identificación con $\mathbb{Z}_{27}$, pero empleando dos dígitos por caracter.

a	b	c	d	e	f	g	h	i	j	k	l	m	n
00	01	02	03	04	05	06	07	08	09	10	11	12	13
$\tilde{n}$	o	p	q	r	s	t	u	v	w	x	y	z	
14	15	16	17	18	19	20	21	22	23	24	25	26	

(v i e r n e s $\Rightarrow$ 22 08 04 18 13 04 19).

- Agrupa en bloques, asegurando que el número de cada bloque sea menor que n (por ejemplo dígitos$(n) - 1$).

(220 804 181 304 19$\underline{0}$)

- Completamos el último bloque con 0 o 30s para fijar el tamaño del bloque.

RSA: Cifrado (B)

- Cada bloque M_i lo cifra mediante la fórmula

$$C_i = M_i^e \bmod n.$$

RSA: Cifrado (B)

- Cada bloque M_i lo cifra mediante la fórmula

$$C_i = M_i^e \bmod n.$$

- Envía $C = [C_1 \quad C_2 \quad C_3 \quad \dots \quad]$ a A.

RSA: Cifrado (B)

- Cada bloque M_i lo cifra mediante la fórmula

$$C_i = M_i^e \bmod n.$$

$$220^{31} = 6809 \bmod 7073, \quad 804^{31} = 6623 \bmod 7073$$
$$181^{31} = 60 \bmod 7073, \quad 304^{31} = 469 \bmod 7073$$
$$190^{31} = 6196 \bmod 7073$$

- Envía $C = [C_1 \quad C_2 \quad C_3 \quad \dots \quad]$ a A.

RSA: Cifrado (B)

- Cada bloque M_i lo cifra mediante la fórmula

$$C_i = M_i^e \bmod n.$$

$$220^{31} = 6809 \bmod 7073, \quad 804^{31} = 6623 \bmod 7073$$
$$181^{31} = 60 \bmod 7073, \quad 304^{31} = 469 \bmod 7073$$
$$190^{31} = 6196 \bmod 7073$$

- Envía $C = [C_1 \quad C_2 \quad C_3 \quad \dots \quad]$ a A.

$$C = [6809 \quad 6623 \quad 60 \quad 469 \quad 6196]$$

RSA: Descifrado (A)

- A opera cada bloque C_i mediante $M_i = C_i^d \bmod n$, recuperando así los bloques originales.

RSA: Descifrado (A)

- A opera cada bloque C_i mediante $M_i = C_i^d \bmod n$, recuperando así los bloques originales.

- Completa los bloques añadiendo 0's al principio si es necesario hasta que sean de tamaño dígitos$(n) - 1$.

RSA: Descifrado (A)

- A opera cada bloque C_i mediante $M_i = C_i^d \bmod n$, recuperando así los bloques originales.

- Completa los bloques añadiendo 0's al principio si es necesario hasta que sean de tamaño dígitos$(n) - 1$.

- Concatena los bloques M_i, agrupa de 2 en 2 y vuelve a pasar a caracteres, recuperando M. Elimina los $30's$ y/o el 0 del final, si existen.

RSA: Descifrado (A)

- A opera cada bloque C_i mediante $M_i = C_i^d \bmod n$, recuperando así los bloques originales.

  $6809^{2071} = 220 \bmod 7073$, $6623^{2071} = 804 \bmod 7073$
  $60^{2071} = 181 \bmod 7073$, $469^{2071} = 304 \bmod 7073$
  $6196^{2071} = 190 \bmod 7073$

- Completa los bloques añadiendo 0's al principio si es necesario hasta que sean de tamaño dígitos$(n) - 1$.

  No hace falta

- Concatena los bloques M_i, agrupa de 2 en 2 y vuelve a pasar a caracteres, recuperando M. Elimina los $30's$ y/o el 0 del final, si existen.

22	08	04	18	13	04	19	0
v	i	e	r	n	e	s	

RSA: ¿Por qué funciona lo anterior?

- Sabemos que e y d cumplen $ed \equiv 1 \pmod{\varphi(n)}$, es decir, $ed - 1 = k\varphi(n)$ para cierto k.

RSA: ¿Por qué funciona lo anterior?

- Sabemos que e y d cumplen $ed \equiv 1 \pmod{\varphi(n)}$, es decir, $ed - 1 = k\varphi(n)$ para cierto k.

- Por lo tanto, usando el Teorema de Euler:

$$C_i^d = M_i^{ed} = M_i^{1+k\varphi(n)} = M_i \left(M_i^{\varphi(n)}\right)^k \equiv M_i \pmod{n}.$$

RSA: ¿Por qué funciona lo anterior?

- Sabemos que e y d cumplen $ed \equiv 1 \pmod{\varphi(n)}$, es decir, $ed - 1 = k\varphi(n)$ para cierto k.

- Por lo tanto, usando el Teorema de Euler:

$$C_i^d = M_i^{ed} = M_i^{1+k\varphi(n)} = M_i \left(M_i^{\varphi(n)}\right)^k \equiv M_i \pmod{n}.$$

- Problema: Para que eso funcione, cada M_i debe ser coprimo con n.
  - Podemos pensar que es poco probable haber cogido un bloque así...
  - Pero se puede probar que incluso en el caso de no ser coprimos, lo anterior sigue funcionando.
    EJERCICIO: Comprobadlo, recordando que $n = pq$ con p y q primos.

RSA: Varias optimizaciones

- Optimización 1: En lugar de calcular d inverso modular de e módulo $\varphi(n)$, calculamos

$$d_p \equiv e^{-1} \pmod{p-1}$$
$$d_q \equiv e^{-1} \pmod{q-1}$$

RSA: Varias optimizaciones

- Optimización 1: En lugar de calcular d inverso modular de e módulo $\varphi(n)$, calculamos

$$d_p \equiv e^{-1} \pmod{p-1}$$
$$d_q \equiv e^{-1} \pmod{q-1}$$

- Optimización 2: En lugar de calcular $C^d \pmod{n}$, vamos a calcular:

$$C^d \equiv C^{d_p} \equiv C_p^{d_p} \pmod{p}, \quad \text{donde } C_p \equiv C \pmod{p}$$
$$C^d \equiv C^{d_q} \equiv C_q^{d_q} \pmod{q}, \quad \text{donde } C_q \equiv C \pmod{q}$$

RSA: Varias optimizaciones

- Optimización 1: En lugar de calcular d inverso modular de e módulo $\varphi(n)$, calculamos

$$d_p \equiv e^{-1} \pmod{p-1}$$
$$d_q \equiv e^{-1} \pmod{q-1}$$

- Optimización 2: En lugar de calcular $C^d \pmod{n}$, vamos a calcular:

$$C^d \equiv C^{d_p} \equiv C_p^{d_p} \pmod{p}, \quad \text{donde } C_p \equiv C \pmod{p}$$
$$C^d \equiv C^{d_q} \equiv C_q^{d_q} \pmod{q}, \quad \text{donde } C_q \equiv C \pmod{q}$$

- Si $C^d \equiv x_1 \pmod{p}$ y $C^d \equiv x_2 \pmod{q}$, el teorema Chino de los restos nos dice que

$$C^d = qC_p^{d_p} \left(q^{p-2} \pmod{p}\right) + pC_q^{d_q} \left(p^{q-2} \pmod{q}\right) \pmod{n}$$

RSA: Varias optimizaciones

- Optimización 1: En lugar de calcular d inverso modular de e módulo $\varphi(n)$, calculamos

$$d_p \equiv e^{-1} \pmod{p-1}$$
$$d_q \equiv e^{-1} \pmod{q-1}$$

- Optimización 2: En lugar de calcular $C^d \pmod{n}$, vamos a calcular:

$$C^d \equiv C^{d_p} \equiv C_p^{d_p} \pmod{p}, \quad \text{donde } C_p \equiv C \pmod{p}$$
$$C^d \equiv C^{d_q} \equiv C_q^{d_q} \pmod{q}, \quad \text{donde } C_q \equiv C \pmod{q}$$

- Si $C^d \equiv x_1 \pmod{p}$ y $C^d \equiv x_2 \pmod{q}$, el teorema Chino de los restos nos dice que

$$C^d = qC_p^{d_p} \left(q^{p-2} \pmod{p}\right) + pC_q^{d_q} \left(p^{q-2} \pmod{q}\right) \pmod{n}$$

- Optimización 3: Fórmula de Garner...