RSA ← Criptografía simétrica

Clave pública del destinatario

Clave privada del destinatario

Generación de claves

1. Primos $\{p, q\}$ para calcular $n = p \cdot q$
2. Calcular $\phi(n) = (p-1)(q-1)$
3. Calcular la clave pública e:
   1) $1 < e < \phi(n)$
   2) e primo relativo de $\phi(n) \rightarrow mcd(e, \phi(n)) = 1$
4. Calcular la clave privada d como el inverso de la pública en $\phi(n)$

   $d = inv(e, \phi(n))$, es decir, $d \cdot e \bmod(\phi(n)) = 1$
5. Hacer <u>pública</u> $\boxed{e, n}$ y mantener <u>secreto</u> $\boxed{d, p, q, \phi(n)}$

$$M^{Ke} \bmod(n) = M'$$
$$M'^{Kd} \bmod(n) = M$$

$K_e \{e, n\}$ → $K_d \{d, n\}$

Inverso (Paso 4)

Ejemplo $p = 11$ $q = 7$

AEE (Algoritmo de euclides extendido)

$n = p \cdot q \rightarrow n = 11 \cdot 7 = 77$

$\phi(n) = (p-1)\cdot(q-1) = 10 \cdot 6 = 60$

donde $mcd(e, 60) = 1$

60	5·2
6	3·2
1	

Como no aparece en la factorización el m.c.d entre ellos es 1.

$1 < e < \phi(n) \rightarrow 1 < e < 60 \rightarrow$ Por ejemplo $e = 13$

Queremos $d = inv(e, \phi(n)) \rightarrow e \cdot d \bmod(\phi(n)) = 1$

	$\phi(n)$	e	restos					
	60	13	8	5	3	2	1	0
q_i		4	1	1	1	1	1	/
X_i	1	0						/
Y_i	0	1	-4	5	-9	14	-23	/

Parte del algoritmo

$$Y_{i+2} = Y_i - Y_{i+1} \cdot q_{i+1}$$

$= 0 - 1 \cdot 4 = -4$

$= 1 - (-4 \cdot 1) = 1 + 4$

$= -4 - (5 \cdot 1) = -4 - 5 = -9$

$= 5 - (-9 \cdot 1) = 5 + 9 = 14$

$d = -9 - (14 \cdot 1) = -9 - 14 = -23$

$d = -23 \quad e = 13 \quad \phi(n) = 60$

$13 \cdot (-7) \bmod(60) = 1$? ✓

$-23 \bmod(60) = 60 - 23 = 37 \rightarrow$ $13 \cdot 37 \bmod(60) = 1$

37 (d) inverso de 13 (e)

Funciones Hash

← No son funciones de cifrado, es computacionalmente imposible volver al mensaje original del hash.

Función que asocia a cualquier archivo M de cualquier tamaño, un resumen h(M) suyo, de longitud fija y supuestamente único.

El tamaño de h(M) dependerá del algoritmo (no del mensaje) utilizado.

Propiedades

1. Facilidad de cálculo
2. Propiedad de unidireccionalidad. ← Computacionalmente imposible encontrar el mensaje M a partir de h(M).
3. Propiedad de compresión. ← De un M de longitud variable se obtiene h(M) de longitud fija.
4. Propiedad de difusión. ← Si se modifica un solo bit del mensaje M, h(M) debería cambiar en la mitad de sus bits aproximadamente
5. Resistencia simple a colisiones. ← Conocido un h(M) encontrar un $h(M') = h(M)$ requiere 2^{n-1} intentos.
6. Resistencia fuerte a colisiones. ← Encontrar un par al azar M, M' que $h(M) = h(M')$ requiere $2^{n/2}$ intentos.

Por la paradoja del cumpleaños.

- MD5
  - Bloques de 512 bits
  - Salida h(M) de 128 bits.
- SHA1
  - Bloques de 512 bits.
  - Salida h(M) de 160 bits.
- SHA2
  - salida h(M)
    - 224
    - 256 ← Usado actualmente
    - 384
    - 512