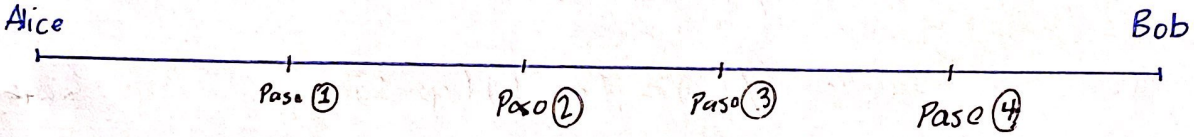
RSA

2019-11-06

Sistema criptográfico RSA:

- Inventado en 1978
  - Rivist, Shamir, Adleman
- Mas ampliamente utilizado



Paso 1: Generación de llaves

Paso 2: Distribución

Paso 3: Encriptación

Paso 4: Desencripción

Generación de la llave:

- Elegimos dos números primos p & q.
  # Estos los elige Alice, solo ella lo sabe.
- Se mantienen en secreto. # Eligen numeros de 1024 bits.
- Calculamos: $n = p \cdot q$ (parte de la clave)

[!] Encontrar p & q a partir de n es computacionalmente inviolable.

- Calculamos $\phi(n) = (p-1)(q-1)$

$\phi(n)$: totiente de Euler [1]

[1] devuelve el número de primos relativos con n menores que n.

$\{1, 2, 3, 4\}$

$\phi(5) = 4$

$\phi(11) = 10$

$\phi(a \cdot b) = \phi(a) \cdot \phi(b)$

$\therefore \quad \phi(n) = (p-1)(q-1)$

- Elegimos un entero e:

  1) $e < \phi(n)$

  2) $mcd(e, \phi(n)) = 1$

e & n es la llave pública.

- La clave pública (n, e)

- Calculamos otro entero d:

$$e \cdot d \equiv 1 \pmod{\phi(n)}$$

! "d" es el inverso multiplicativo de $e \cdot mod(\phi(n))$.

- Clave privada (n, d).

Alice ⟶ Publica la llave

Alice (n, d)

Encryptación:

$M = m_1 m_2 m_3 m_4 \cdots m_K$

mensaje

Calcula (Bob):

$$C_i = (m_i)^e \pmod{n}$$

$C = C_1 C_2 C_3 \cdots C_K$

Mensaje encriptado

Alice ⟵ Bob

Bob manda $C_i = m_i^e \bmod(n)$

Desencripta:

$C = c_1 c_2 c_3 \dots c_k$

Calculamos (Alice):

$m_i = (c_i)^d \bmod(n)$

$M = m_1 m_2 m_3 \dots m_k$

$ed \equiv 1 \bmod(\phi(n)) \longrightarrow ed \equiv 1 \bmod(n)$

---

[1] Dado un diccionario:

$\sum = (0: \_, 1: A, 2: B \dots 27: Z)$

- Si el mensaje M se descompone en monograma (letra por letra) se escogen "p" y "q" de modo que $n = p \cdot q$ sea mayor que el monograma más grande en $\sum$.

- Si son bigramas $m_1 m_2$ (letras de dos en dos)

Z A N A H O R I A

Z → 2701

El bigrama más grande será 2727.