Criptografía de clave pública

Criptosistema RSA

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
KISA

Criptosistema RSA

- Desarrollado por R. Rivest, A. Shamir y L. Adleman (1977).
- Es uno de los criptosistemas de clave pública más sencillos y fáciles de entender.
- Se apoya en el **problema de factorización de números enteros:** Dado un entero grande n, hallar su factorización en factores primos.

Algunos resultados previos

- Dado un entero n, *función de Euler:*

$$\begin{aligned}\phi(n) = & \ \mathrm{card}(\mathbb{Z}_n^*) \\ = & \ \text{n}^0 \text{ de enteros } a \text{ t. q. } 0 < a < n \text{ y } \mathrm{mcd}(a, n) = 1.\end{aligned}$$

- Si n primo, $\phi(n) = n - 1$.
- Si $n = pq$, p, q primos distintos,

$$\begin{aligned}\phi(n) = & \ (p-1)(q-1) = pq + 1 - p - q \\ = & \ n + 1 - (p + q).\end{aligned}$$

- *Pequeño Teorema de Fermat*: si p es un número primo, para cualquier entero positivo a tal que $\mathrm{mcd}(a, p) = 1$,

$$a^{p-1} \equiv 1 \mod p.$$

Observación. Si $n = pq$ con p, q primos distintos, conocer $\phi(n)$ equivale a conocer p, q:

- Si conocemos p, q, podemos calcular

$$\phi(n) = (p-1)(q-1) = n + 1 - (p+q).$$

- Si conocemos $\phi(n)$, podemos obtener p y q resolviendo el sistema:

$$\begin{cases} pq = n, \\ p + q = n + 1 - \phi(n). \end{cases}$$

Generación de claves en RSA

Cada usuario A:

- Elige dos primos grandes p, q.
- Calcula $n = pq$.
- Calcula $\phi(n) = (p-1)(q-1) = n + 1 - p - q$.
- Elige un entero e, $1 < e < \phi(n)$, tal que $\mathrm{mcd}(e, \phi(n)) = 1$.
- Calcula $d = e^{-1} \mod \phi(n)$.

Clave pública: $K_{\mathrm{E}} = (n, e)$.

Clave privada: $K_{\mathrm{D}} = (p, q, d)$.

Observación:

$$\mathrm{mcd}(e, \phi(n)) = 1 \Longleftrightarrow \mathrm{mcd}(e, p-1) = \mathrm{mcd}(e, q-1) = 1.$$

Ejemplo

Para generar sus claves el usuario A:

- Elige dos primos, por ejemplo $p = 131$, $q = 223$.
- Calcula $n = pq = 131 \cdot 223 = 29213$.
- Calcula $\phi(n) = n + 1 - (p + q) = 29214 - 354 = 28860$.
- Elige un entero e tal que $1 < e < 28860$ y $\mathrm{mcd}(e, 130) = \mathrm{mcd}(e, 222) = 1$, por ejemplo $e = 1327$.
- Calcula $d = (e^{-1} \mod 28860) = 25663$.

Clave pública: $(n = 29213, e = 1327)$.

Clave secreta: $(p = 131, q = 223, d = 25663)$.

Funciones de cifrado y descifrado en RSA

- Función de cifrado (a partir de la clave pública $K_{\mathrm{E}} = (n, e)$):

$$\begin{array}{ccc} \mathbb{Z}_n & \xrightarrow{\mathrm{E}} & \mathbb{Z}_n \\ M & \mapsto & C = M^e \mod n. \end{array}$$

- Función de descifrado (a partir de la clave privada $K_{\mathrm{D}} = (p, q, d)$):

$$\begin{array}{ccc} \mathbb{Z}_n & \xrightarrow{\mathrm{D}} & \mathbb{Z}_n \\ C & \mapsto & M = C^d \mod n. \end{array}$$

Se cumple $\mathrm{D} = \mathrm{E}^{-1}$. Es decir, para cualquier $M \in \mathbb{Z}_n$,

$$\mathrm{D}(\mathrm{E}(M)) = M.$$

Justificación

Veamos que, efectivamente, para $M \in \mathbb{Z}_n$, $\mathrm{D}(\mathrm{E}(M)) = M$.

$$\mathrm{D}(\mathrm{E}(M)) \equiv \mathrm{D}(M^e) \equiv (M^e)^d \equiv M^{ed} \pmod{n}.$$

$$M^{ed} \stackrel{?}{\equiv} M \pmod{n}$$

$$ed \equiv 1 \pmod{\phi(n)} \Rightarrow \phi(n) \mid ed - 1 \Rightarrow (p-1)(q-1) \mid ed - 1$$

$$\Rightarrow \left\{ \begin{array}{l} (p-1) \mid ed - 1 \Rightarrow ed \equiv 1 \pmod{(p-1)} \\ (q-1) \mid ed - 1 \Rightarrow ed \equiv 1 \pmod{(q-1)} \end{array} \right\}$$

$$\Rightarrow \left\{ \begin{array}{l} M^{ed} \equiv M^1 \equiv M \pmod{p} \\ M^{ed} \equiv M^1 \equiv M \pmod{q} \end{array} \right\} \xrightarrow{p \neq q \text{ primos}} M^{ed} \equiv M \pmod{pq}.$$

Observaciones.

- Si p o q no son primos, el algoritmo no funciona.

- Resolviendo el problema de factorización de números enteros, RSA quedaría roto: si factorizamos n, podemos calcular p, q, d.

- Resolviendo el problema del logaritmo discreto podemos calcular d:
  Ciframos un mensaje arbitrario
  $$C = M^e \mod n$$
  y entonces, como
  $$M = C^d \mod n,$$
  se tiene que
  $$d = \log_C M \mod n.$$

Ejemplo

- Queremos enviar cifrado el mensaje $M = 1000$ a un usuario A.

  Supongamos que la clave pública de A es $(n, e) = (29213, 1327)$.
  Para cifrar el mensaje, calculamos

  $$1000^{1327} \equiv 26145 \mod 29213.$$

  Enviamos: $C = 26145$.

- El usuario A conoce su clave privada $(p, q, d) = (131, 223, 25663)$.

  A recibe $C = 26145$.

  Para descifrar, calcula $n = 131 \cdot 223 = 29213$ y

  $$26145^{25663} \equiv 1000 \mod 29213.$$

Descifrado utilizando el Teorema chino del resto

Para facilitar los cálculos en el descifrado, se puede utilizar el Teorema chino del resto.

Sean

$$M_1 \equiv C^d \mod p, \quad M_2 \equiv C^d \mod q.$$

El sistema

$$\left.\begin{array}{l} x \equiv M_1 \mod p, \\ x \equiv M_2 \mod q \end{array}\right\}$$

tiene una única solución módulo $n = pq$, que viene dada por

$$x \equiv qq_1 M_1 + pp_1 M_2 \mod n,$$

donde

$$p_1 \equiv p^{-1} \mod q, \quad q_1 \equiv q^{-1} \mod p.$$

Si la solución es x,

$$\left.\begin{array}{l} x \equiv M_1 \equiv C^d \mod p \\ x \equiv M_2 \equiv C^d \mod q \end{array}\right\} \overset{p \neq q \text{ primos}}{\Longrightarrow} x \equiv C^d \mod pq.$$

Por lo que

$$x \equiv M \mod n.$$

Para obtener M, podemos calcular

$$qq_1 M_1 + pp_1 M_2 \mod n,$$

donde

$$M_1 \equiv C^d \mod p, \quad M_2 \equiv C^d \mod q.$$

Por otra parte, si

$$C_p \equiv C \mod p, \quad d_p \equiv d \mod p-1,$$

se tiene que

$$M_1 \equiv C_p^{d_p} \mod p.$$

Análogamente, si

$$C_q \equiv C \mod q, \quad d_q \equiv d \mod q-1,$$

entonces

$$M_2 \equiv C_q^{d_q} \mod q.$$

Entonces, el procedimiento para descifrar consiste en

- Calcular

$$d_p \equiv d \mod p-1, \quad d_q \equiv d \mod q-1,$$
$$p_1 \equiv p^{-1} \mod q, \quad q_1 \equiv q^{-1} \mod p,$$
$$coef_1 = qq_1, \quad coef_2 = pp_1.$$

- Calcular

$$C_p \equiv C \mod p, \quad C_q \equiv C \mod q,$$
$$M_1 \equiv C_p^{d_p} \mod p, \quad M_2 \equiv C_q^{d_q} \mod q.$$

- Calcular

$$M \equiv coef_1 M_1 + coef_2 M_2 \mod n.$$

Con ello conseguimos que el módulo al calcular las potencias sea sensiblemente menor y, en consecuencia, los cálculos más rápidos.

Además, los valores d_p, d_q, $coef_1$ y $coef_2$ sólo dependen de la clave privada y no del mensaje cifrado, por lo que pueden calcularse previamente.

Ejemplo

Supongamos que la clave privada del usuario A sea:

$$p = 131, \quad q = 223, \quad d = 25663 \quad (n = 131 \cdot 223 = 29213).$$

A calcula:

$$\begin{gathered}
d_p \equiv 25663 \mod 130, \quad d_p = 53, \\
d_q \equiv 25663 \mod 222, \quad d_q = 133, \\
p_1 \equiv 131^{-1} \mod 223, \quad p_1 = 143, \\
q_1 \equiv 223^{-1} \mod 131, \quad q_1 = 47, \\
coef_1 = 223 \cdot 47 = 10481, \\
coef_2 = 131 \cdot 143 = 18733.
\end{gathered}$$

$$p = 131, \quad q = 223, \quad d = 25663 \quad (n = 131 \cdot 223 = 29213).$$

Una vez calculados

$$d_p = 53, \quad d_q = 133, \quad coef_1 = 10481, \quad coef_2 = 18733,$$

para descifrar $C = 26145$, calcula

$$C_p \equiv 26145 \mod 131, \quad C_p = 76,$$

$$C_q \equiv 26145 \mod 223, \quad C_q = 54,$$

$$M_1 \equiv 76^{53} \mod 131, \quad M_1 = 83,$$

$$M_2 \equiv 54^{133} \mod 223, \quad M_2 = 108,$$

$$M \equiv (10481 \cdot 83 + 18733 \cdot 108) \mod 29213, \quad M = 1000.$$

Vulnerabilidades del RSA

Claves demasiado cortas

La seguridad del criptosistema RSA está basada en el problema de la factorización de enteros grandes.

El entero n debe ser de al menos 1024 bits.

Exponentes bajos

Si el exponente de cifrado e o el de descifrado d son pequeños, existen métodos de ataque eficientes para romper el sistema.

Claves débiles

Existen casos para los cuales el algoritmo RSA deja el mensaje en claro sin cifrar:

$$M^e \equiv M \mod n.$$

Se puede probar que, si $n = pq$, el número de mensajes que quedan inalterados es

$$\sigma_n = (1 + \mathrm{mcd}(e-1, p-1))(1 + \mathrm{mcd}(e-1, q-1)).$$

Para evitar que σ_n sea grande, conviene elegir p, q *primos fuertes:*

$$p = 1 + 2p', \quad q = 1 + 2q',$$

con p' y q' primos grandes.

Ejemplo

Si $p = 11$, $q = 7$, $e = 5$, hay 9 mensajes (de los 77 posibles) que quedan sin cifrar. Por ejemplo $M = 22$.

$$22^5 \equiv 22 \mod 77.$$

Ataque de módulo común

Supongamos que una institución calcula $n = pq$ y reparte las claves de cifrado y descifrado (e_i, d_i).
Ciframos un mensaje empleando dos claves diferentes:

$$C_1 \equiv M^{e_1} \mod n, \quad C_2 \equiv M^{e_2} \mod n.$$

El atacante intercepta C_1 y C_2 y por tanto conoce n, e_1, e_2, C_1, C_2.

Si e_1 y e_2 son primos relativos, el algoritmo extendido de Euclides le permite encontrar u, v tales que

$$ue_1 + ve_2 = 1.$$

Entonces, calculando $C_1^u C_2^v \mod n$ puede recuperar el mensaje M:

$$C_1^u C_2^v \equiv M^{e_1 u} M^{e_2 v} \equiv M^{e_1 u + e_2 v} \equiv M^1 \mod n.$$

Ataque cíclico

Sea M un mensaje y C su cifrado con la clave pública (n, e):

$$C \equiv M^e \mod n.$$

El ataque cíclico consiste en hacer

- $C_0 = C$.
- Calcular sucesivos cifrados $C_i \equiv C_{i-1}^e \mod n$, hasta que $C_i = C$.
- Entonces $M \equiv C_{i-1} \mod n$.

Para prevenir los ataques, los primos p y q del RSA:

- deben tener más de 100 dígitos,
- deben ser tales que $p - q$ sea grande,
- deben ser de la forma $p = 2p' + 1$, $q = 2q' + 1$, con p' y q' primos grandes.

Fin de la sección