# RSA y Factorización



DH
ElGamal ← como base tienen el hecho que es fácil calcular potencias $a^n$ mód $p$ pero difícil encontrar el $n$ sólo sabiendo el $a^n$ (módulo $p$).

El pequeño teorema de Fermat tiene un rol importante en esto.

¿Cómo se generaliza este teorema? En particular, qué potencia $x$ nos da $a^x \equiv 1 \pmod{pq}$.

<u>Thm</u>: $p, q$ primos distintos y $g = \gcd(p-1, q-1)$.

Entonces

$$a^{(p-1)(q-1)/g} \equiv 1 \pmod{pq} \quad \forall a : \operatorname{mcd}(a, pq) = 1$$

En particular si $p$ y $q$ son impares,

$$a^{(p-1)(q-1)/2} \equiv 1 \pmod{pq} \quad \forall a : \operatorname{mcd}(a, pq) = 1.$$

Dem: Se ve que $p \nmid a$ y $g \mid q-1$. Entonces

$$a^{(p-1)(q-1)/g} = (a^{p-1})^{\underbrace{(q-1)/g}_{\in \mathbb{Z}}}$$

$$\equiv (1)^{(q-1)/g} \pmod p$$

$$\equiv 1 \pmod p.$$

También se ve

$$a^{(p-1)(q-1)/g} \equiv 1 \pmod q.$$

$$\Rightarrow p \mid a^{(p-1)(q-1)/g} - 1 \quad y \quad q \mid a^{(p-1)(q-1)/g} - 1$$

$$\Rightarrow pq \mid a^{(p-1)(q-1)/g} - 1$$

$$\Rightarrow a^{(p-1)(q-1)/g} \equiv 1 \pmod{pq}. \quad \square$$

---

| | |
|---|---|
| $a^x \equiv b \pmod p$ | $x^e \equiv c \pmod N$ |
| DH<br>ElGamal<br>dificultad de logaritmos | RSA<br>dificultad de tomar raíces. |

Raíces modulo un primo?

**Prop:** $p$ primo $e \geq 1$ un entero que verifica $\mathrm{mcd}(e, p-1) =$

Se sabe que $e$ tiene un inverso $d$ módulo $p-1$ y entonces la ecuación

$$x^e \equiv c \pmod p$$

tiene la ~~única~~ solución $x \equiv c^d \pmod p$ y es única.

**Dem:** Si $c \equiv 0 \pmod p$, $x \equiv 0 \pmod p$ es la única solución.

Entonces supongamos que $c \not\equiv 0 \pmod p$:

$$de \equiv 1 \pmod{p-1} \Rightarrow \exists k \; de = 1 + k(p-1)$$

Ahora

$$\begin{aligned}(c^d)^e \equiv c^{de} &\equiv c^{1+k(p-1)} \\ &\equiv c\,(c^{p-1})^k \\ &\equiv c \pmod{\text{módulo } p}.\end{aligned}$$

Es única: $x_1$ y $x_2$ dos soluciones: $x_1^e \equiv c \equiv x_2^e \pmod p$.

$$x_1 \equiv x_1^{de} \equiv (x_1^e)^d \equiv c^d \equiv (x_2^e)^d \equiv x_2^{de} \equiv x_2 \pmod p.$$

$$\Rightarrow x_1 = x_2. \quad \square$$

¿Y qué pasa si miramos módulos compuestos?

<u>Prop</u>: $p, q$ primos distintos, $e \geq 1$ t.q

$$\operatorname{mcd}(e, (p-1)(q-1)) = 1$$

Sabemos que hay un inverso $d$ de $e$ módulo $(p-1)(q-1)$

Entonces la ecuación $x^e \equiv c \pmod{pq}$ tiene la solución

$x \equiv c^d \pmod{pq}$ y es única.

<u>Dem</u>: Casi igual a la anterior $\square$

Prop La proposición nos da un algoritmo para ~~hallar la raíz~~.

resolver

$$x^e \equiv c \pmod{pq}.$$

1). Hallar $d$, el inverso de $e$ módulo $(p-1)(q-1)$

2) calcular $c^d \pmod{pq}$.

Podemos hacer la cuenta más rápido usando un valor de $d$ más pequeño: Sea $g = \operatorname{mcd}(p-1, q-1)$ y supongamos que se resuelve:

$$de \equiv 1 \pmod{\frac{(p-1)(q-1)}{g}}.$$

Ahora

$$a^{(p-1)(q-1)/g} \equiv 1 \pmod{pq}$$

$\Rightarrow$ Si escribimos $de = 1 + k(p-1)(q-1)/g$, tenemos

$$(c^d)^e = c^{de} = c \cdot c^{((p-1)(q-1)/g)k} \equiv c \pmod{pq}.$$

El sistema RSA:

- Setup: $p, q$ primos grandes, $N = pq$, $e, c \in \mathbb{Z}$.
- Problema: Resolver $x^e \equiv c \pmod{N}$
- Fácil: Bob, sabiendo $p$ y $q$ puede resolver la ecuación fácilmente (encontrando el inverso de $e$ mód $(p-1)(q-1)$.
- Difícil: Si saber $p$ ni $q$, un adversario no puede hallar $x$.
- Dicotomía: Resolviendo $x^e \equiv c \pmod{N}$ es fácil sabiendo extra información (p.ej., $d$ or $p$ y $q$)

pero difícil para otros.

Esquema del Sistema



| Bob | Alice |
| --- | --- |
| **Creación de la clave** | |
| Elige $p$ y $q$ secreto | |
| Elige $e$ con $mcd(e, (p-1)(q-1)) = 1$ | |
| Publica $N = pq$ y $e$. | |
| **Encriptar** | |
| | ~~Elige~~ Escribe y codifica un mensaje $m$ |
| | Calcula $c \equiv m^e \pmod{N}$ |
| | y lo manda a Bob |
| **Desencriptar** | |
| Calcula $d$ t.q $ed \equiv 1 \pmod{(p-1)(q-1)}$ | |
| Calcular $m' \equiv c^d \pmod{N}$. | |

Notes:



- $N$ - el módulo
  $e$ - la potencia de encriptar
  $d$ - la potencia de desencriptar

- intuitivamente uno debería elegir $d$ y $e$ pequeño para que los procesos de encriptar y desencriptar sean rápidos.
  Pero como $de \equiv 1 \pmod{(p-1)(q-1)}$ se espera que los dos no pueden ser pequeños.
- Se cree que eligiendo $e = 3$ es tan seguro que eligiendo un $e$ grande. Otra elección común para $e$ es $2^{16}+1$ porque calculando $m^{65537}$ requiere una multiplicación y 16 cuadrados.
- Si Bob elige un $d$ pequeño puede causar que la implentación de RSA sea insegura: si $d < N^{1/4}$ usando fracciones continuas se puede romper RSA.

- Sabiendo

$(p-1)(q-1)$ un adversario puede ~~romper~~ resolver $x^e \equiv c \pmod{N}$

y puede desencriptar mensajes dirigidos a Bob

Pero despejando

$$(p-1)(q-1) = pq - (p+q) + 1$$
$$= N - (p+q) + 1$$

$\Rightarrow$ sabiendo $(p+q)$ deja el adversario desencriptar

Además sabiendo $(p+q)$ deja el adversario hallar $p$ y $q$:

$$X^2 - (p+q)X + pq$$

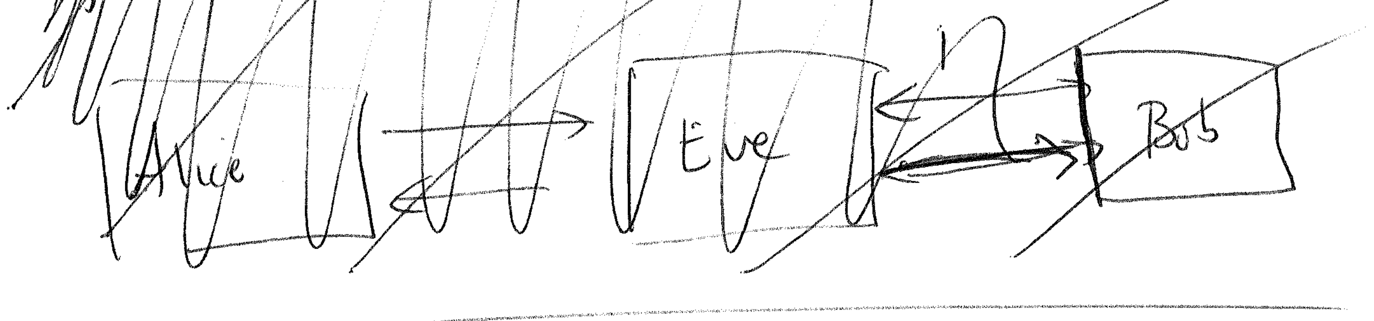
tiene raíces $p$ y $q$.

- ~~Un adversario no tiene que~~

- Hemos visto no es más fácil para un adversario determinar $(p-1)(q-1)$ que factorizar $N$ pero esto no quiere decir que hay que factorizar $N$ para desencriptar.

~~Implementación y Seguridad~~ OK

~~Ej Man-in-the-middle.~~




Ej Supongamos que Eve convence a Alice que Alice debería desencriptar mensajes aleatorios mandados por Eve.

Esto es ~~ad~~ una situación razonable: una forma por la cual Alice puede autenticar su identidad como la dueña de la clave pública $(N, e)$ es desencriptar mensajes usando la $d$ privada. (Eve tiene acceso a un oráculo para RSA)

Eve hace lo siguiente: intercepta un texto cifrado $c$ mandado por Bob a Alice. Eve elige un $k \in \mathbb{Z}$ al azar.

y manda

$$c' \equiv k^e c \pmod{N}$$

a Alice

Alice desencripta $c'$ y devuelve $m'$ a Eve: (10)

$$m' \equiv (c')^d \equiv k^{ed} c^{ed} \equiv k^{ed} m^{ed}$$
$$\equiv (km)^{ed}$$
$$\equiv km \pmod{N}$$

Como ~~Eve~~ sabe el $k \Rightarrow$ Eve sabe el $m$ encriptado por Bob

- Eve puede descriptar mensajes secretos sin factorizar.
- Alice recibe mensajes aleatorios por la multiplicación ~~por~~ $k$ ~~m~~. $k^e c$.

Ej: Si Alice publica dos potencias $e_1$ y $e_2$ para usar con $N$ y Bob encripta un solo mensaje usando $e_1$ y $e_2$.

Si Eve intercepta
$$c_1 \equiv m^{e_1} \pmod{N}$$
$$c_2 \equiv m^{e_2} \pmod{N}$$

~~B~~ puede usar $e_1 \cdot u + e_2 \cdot v = mcd(e_1, e_2)$

para calcular

$$c_1^u c_2^v = m^{e_1 u + e_2 v} = m^{mcd(e_1, e_2)} \pmod{N}$$

Si $mcd(e_1, e_2) = 1$, Eve ha recuperado el mensaje original (si no es 1 pero es chico, todavía causa problemas).

Moraleja: Alice debería usar solamente una potencia para cada módulo $N$.

~~Pastar~~

## Testeando primalidad:

¿Cómo se arranca con una implementación de RSA?

¿Cómo se encuentran dos primos grandes?

Elige un número $n$ de, digamos, 50 dígitos.

1) Bob hace "trial division" y encuentra que $n$ no tiene factores más pequeños de 1000000.

2) Empieza creer que $n$ es primo. Calcular

$2^{n-1} \pmod n$ y resulta en algo distinto que 1.

$\Rightarrow$ $n$ es compuesto.

(contrarecíproco de Fermat).

$p$ primo $\Rightarrow a^{p-1} \equiv 1 \pmod p$

$\Updownarrow$

$\forall p \quad a^{p-1} \equiv a \pmod p$.

3) Bob elige otro $n$.

calcular $2^{n-2} \equiv 2 \pmod{n}$.

¿$n$ es primo? ¡NO! (tiene que ser por ~~...~~ el recíproco de Fermat es falso).

Pero si lo hace más probable que $n$ es primo.

<u>Def</u>: $n \in \mathbb{Z}$ fijo. Se dice que $a \in \mathbb{Z}$ es un testigo de la "compuestitud" de $n$ si

$$a^n \not\equiv a \pmod{n}.$$

<u>Obs</u>:
- La existencia de un solo testigo $\Rightarrow$ $n$ compuesto
- Si Bob testea con $a_1, a_2, a_3, \ldots$ y ninguno es un testigo Bob empieza pensar que $n$ es primo.
- Los números Carmichael son números compuestos sin testigos. Son escasos pero hay infinitos (Alford, Granville Pomerance 1994)
- Entonces Bob busca un mejor test para compuestitud.

El test de Miller-Rabin tiene como base: (14)

Prop: $p$ primo impar; $(p-1) = 2^k q$ ← $q$ impar.

Sea $a$ cualquier entero no divisible por $p$. Entonces una de las condiciones es verdad:

(i) $a^q \equiv 1 \pmod{p}$

(ii) uno de $a^q, a^{2q}, a^{4q}, \dots, a^{2^{k-1}q}$ es $\equiv -1 \pmod{p}$

Dem: Por Fermat

$$a^q, a^{2q}, a^{4q}, \dots, a^{2^{k-1}q}, \underbrace{a^{2^k q}}_{a^{p-1} \equiv 1 \pmod{p}}$$

Como cada ~~ele ter~~ elemento de la lista es el cuadrado del número anterior:

(i): $a^q \equiv 1 \pmod{p}$ (y así los otros son)

(ii) hay otro número $x$ en la lista t.q. $x \not\equiv 1 \pmod{p}$
$x^2 \equiv 1 \pmod{p}$
$\Rightarrow x \equiv -1 \pmod{p}$

$\square$