$x^2 \equiv 197 \pmod{19 \cdot 23}$

$3 \pmod 4$.

Empezamos

$y^2 \equiv 197 \equiv 7 \pmod{19}$ y $z^2 \equiv 197 \equiv 13 \pmod{23}$

$y \equiv (7)^{\frac{19+1}{4}} \pmod{19}$

$\equiv \pm 8 \pmod{19}$

$z \equiv (13)^{\left(\frac{23+1}{4}\right)} \equiv \pm 6 \pmod{23}$

Ahora usando CRT: $x \equiv 8 \pmod{19}$

$x \equiv 6 \pmod{23}$

nos da $x \equiv 236 \pmod{437}$ como una solución.

Obs: La solución no es única: ej $-236 \equiv 201 \pmod{437}$ también es una solución

En particular se puede ver que hay 4 soluciones.

El algoritmo de Pohlig-Hellman:

El CRT es más que un teorema, es una filosofía. una forma de ser

DLP: $g^x \equiv h \pmod p$: CRT no tiene nada que ver. Pero les recuerdo que x es definido mód $p-1$. Y entonces, el CRT se aplica con respecto a la factorización de $p-1$.

Teorema: (Pohlig-Hellman) G un grupo y supongamos que tenemos un algoritmo que resuelve el DLP en G para cualquier elemento con orden una potencia de un primo: supongamos: (9)

$g \in G$ tiene orden $q^e \Rightarrow$ se puede resolver el DLP en $\mathcal{O}(S_{q^e})$ pasos.

Sea g un elemento de orden $N = q_1^{e_1} q_2^{e_2} \cdots q_t^{e_t}$.

Entonces $g^x = h$ se puede resolver en

$$\mathcal{O}\left(\log N + \sum_{i=1}^{t} S_{q_i^{e_i}} \right)$$ pasos

usando el siguiente algoritmo:

(1) Para cada $1 \leq i \leq t$ sea

$$g_i = g^{N/q_i^{e_i}} \text{ y } h_i = h^{N/q_i^{e_i}}.$$

Ahora cada g_i tiene orden $q_i^{e_i}$ y usando el algoritmo dado para resolver $g_i^y = h_i$ (*). Sea y_i una solución a (*)

(2) usar CRT para resolver

$$x \equiv y_1 \pmod{q_1^{e_1}}$$

$$\vdots$$

$$x \equiv y_t \pmod{q_t^{e_t}}.$$

Obs: Reduce el DLP módulo compuesto al DLP módulo una potencia de un primo. Hay otra variante que lo reduce al DLP módulo primo

Dem: Primero probamos que los pasos nos dan una solución.

Sea x una solución al sistema en paso (2)

Entonces para cada i ~~tenemos~~ escribimos

$$x = y_i + q_i^{e_i} z_i$$

Entonces podemos calcular

$$\begin{aligned}
(g^x)^{N/q_i^{e_i}} &= g^{(y_i + q_i^{e_i} z)N/q_i^{e_i}} \\
&= g^{(N/q_i^{e_i}) y_i} g^{N z_i} \quad (\text{orden de } g = N) \\
&= \left(g^{N/q_i^{e_i}}\right)^{y_i} \\
&= g_i^{y_i} \\
&= h_i \\
&= h^{N/q_i}
\end{aligned}$$

O sea, tenemos

$$\frac{N}{q_i^{e_i}} x \equiv \frac{N}{q_i^{e_i}} \log_g(h).$$

Ahora podemos encontrar $c_1, c_2, \ldots, c_t$ t.q.

$$c_1 \frac{N}{q_1^{e_1}} + \cdots + c_t \frac{N}{q_t^{e_t}} = 1$$

porque los $N/q_i^{e_i}$ son coprimos.

(21)

Entonces como $\frac{N}{q^{e_i}} x \equiv \frac{N}{q_i^{e_i}} \log_g(h) \pmod N$. ($\frac{N}{q_i^{e_i}}$ no sea coprimo con N)

$$\Rightarrow \sum_{i=1}^{t} \frac{N}{q_i^{e_i}} c_i x \equiv \sum_{i=1}^{t} \frac{N}{q_i^{e_i}} c_i \log_g(h) \pmod N$$

Ahora como $\sum \frac{N}{q_i^{e_i}} c_i = 1 \Rightarrow$

$$x \equiv \cancel{\sum \frac{N}{q_i^{e_i}} c_i} \log_g(h).$$

Nos falta hablar de la complejidad:

Claramente paso (1) toma $\Theta(S_{q_i^{e_i}})$ para cada primo $q_i | N$ y $\Theta(\sum S_{q_i^{e_i}})$ en total.

Ahora el paso 2: ~~resulta de~~ el CRT es el resultado de aplicar XGCD varias veces y cada la i-ésima aplicación requiere

~~$\Theta(\log q_i^{e_i})$ pasos y $\log t$~~

$\Theta(\log q_1^{e_1} \cdots q_r^{e_r})$ pasos. Y el último paso requiere

$$\Theta(\log \prod q_i^{e_i}) = \Theta(\log N)$$

$\square$

pasos

__Obs:__ • Si ~~G tiene o~~ el orden de G se factoriza en potencias de primos pequeños $\Rightarrow$ DLP para G es inseguro.

• En particular para el DLP en $\mathbb{F}_p^\times$, lo mejor posible sería elegir un primo de Sophie Germain: un primo $p = 2q+1$ donde q es otro primo.

• Si g tiene orden q^e, entonces $g^{q^{e-1}}$ tiene orden q. Así cualquier algoritmo que resuelve el DLP en $O(S_{q^e})$ lo puede resolver en $O(S_q)$.

__Prop:__ G un grupo, q un primo. Supongamos que conocemos un algoritmo que resuelve el DLP $g^x = h$ para g de orden q en $O(S_q)$ pasos. Sea $g \in G$ un elemento de orden q^e con $e \geq 1$. Entonces podemos resolver el DLP

$$g^x = h \quad \text{en } O(eS_q) \text{ pasos.}$$

__Dem:__ Escribimos

$$X = x_0 + x_1 q + \cdots + x_{e-1} q^{e-1}$$

y determinamos los x_i sucesivamente. Observemos que el orden de $g^{q^{e-1}}$ es q y se puede calcular, ~~en $O(S_q)$ pasos~~.

$$h^{q^{e-1}} = (g^x)^{q^{e-1}}$$

$$= \left(g^{x_0+x_1q+\cdots+x_{e-1}q^{e-1}}\right)^{q^{e-1}}$$

$$= g^{x_0q^{e-1}} \cdot \left(g^{q^e}\right)^{x_1+x_2q+\cdots+x_{e-1}q^{e-2}}$$

$$= g^{x_0q^{e-1}} \quad \boxed{q^e \text{ es el orden de } g.}$$

$$= \left(g^{q^{e-1}}\right)^{x_0}$$

Como $g^{q^{e-1}}$ es un elemento de orden q en G, la ecuación

$$\left(g^{q^{e-1}}\right)^{x_0} = h^{q^{e-1}}$$

se puede resolver en $O(S_q)$ pasos. O sea, conocemos el x_0 t.q.

$$g^{x_0q^{e-1}} = h^{q^{e-1}} \text{ en } G.$$

Ahora repetimos el proceso para levantando cada lado a la potencia q^{e-2}.

29

$$h^{q^{e-2}} = (g^x)^{q^{e-2}}$$

$$= \left(g^{(x_0 + x_1 q + x_2 q^2 + \cdots + x_{e-1} q^{e-1})}\right)^{q^{e-2}}$$

$$= \underbrace{g^{x_0 q^{e-2}}}_{\text{conocido}} \cdot g^{x_1 q^{e-1}} \underbrace{\left(g^{q^e}\right)^{x_2 + x_3 q + \cdots + x_{e-1} q^{e-3}}}_{1}$$

Para encontrar x_1 tenemos que resolver el DLP

$$\left(g^{q^{e-1}}\right)^{x_1} = \left(h \cdot g^{-x_0}\right)^{q^{e-2}}$$

algo que se puede hacer en $\Theta(S_q)$ pasos. O sea, hemos encontrado x_0 y x_1 en $\Theta(2S_q)$ pasos. Repitiendo ~~nos da~~ este proceso nos da el resultado □

~~Ej. $5448^x = 6909$ en $\mathbb{F}_{11251}^{\times}$.~~

RSA y factorización

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

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

¿Cómo se generaliza este teorema? En particular, que 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,

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

(2)

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

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

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

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

$a^x \equiv b \pmod{p}$

DH

ElGamal

dificultad de logaritmos

$x^e \equiv c \pmod{N}$

RSA

dificultad de tomar raíces.

Raíces modulo un primo?

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

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

$$(c^d)^e \equiv c^{de} \equiv c^{1+k(p-1)}$$
$$\equiv c\,(c^{p-1})^k$$
$$\equiv c \pmod{p}.$$

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

$$mcd(e, (p-1)(q-1)) = 1$$

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.

$\square$

<u>Dem:</u> Casi igual a la anterior

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 $g = mcd(p-1, q-1)$ y supongamos que se resuelve:

$$de \equiv 1 \left(\text{mód } \frac{(p-1)(q-1)}{g}\right).$$

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.