Introduction to Cryptography

We finish DLP/ Elgamal

PCMI 2022 - Undergraduate Summer School

① Solution to DLP: Baby steps, giant steps

$G = \langle g \rangle$ $\quad\quad h = g^x$ ← find x

Adversary E

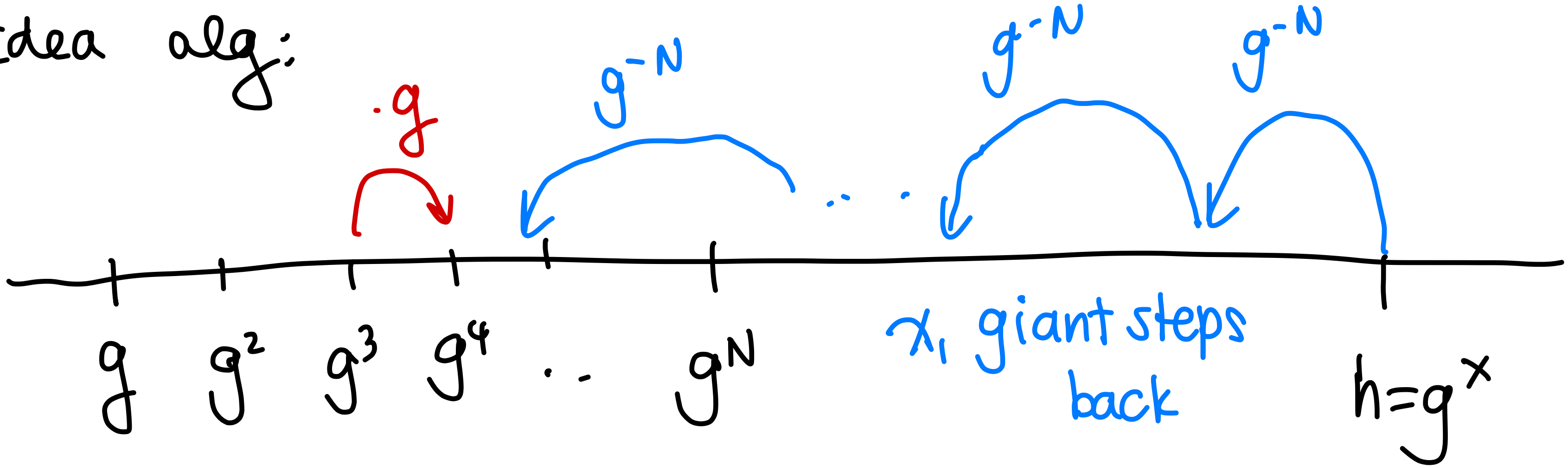
- choose N such that $\#G \leq N^2$

  (but close to $\sqrt{\#G}$)

If $x < \#G$ then $x = N x_1 + x_0$ (quotient x_1, remainder x_0)

$0 \leq x_0, x_1 < N.$

Idea alg:



Baby steps: multiplication by g

Giant steps: multiplication by g^N

① E compute + store $g, g^2, g^3 \dots g^{N-1}$

g^{x_0} is among these

① E compute + store $1, g, g^2, g^3, \dots g^{N-1}$

g^{x_0} is among these

② E computes $h \cdot g^{-N}$, hg^{-2N}, ...

until E gets a match with the list above

This will happen in x_1 steps.

③ $x = x_0 + Nx_1$

$x_0 =$ index of the match

$x_1 =$ how many giant steps.

Number of operations to carry out attack

- $\approx N$ group mult, $N \approx \sqrt{\#G}$

- computing g^N is polynomial using fast exponentiation (or one more mult.) inversion is also fast.

- $\approx \frac{N}{2}$, at worst N group mult

So overall this algorithm takes about ($\sim$)

$\sqrt{\#G}$ steps

This is exponential in $k = \log_2 \#G$ (size $\#G$)

$\#G = 2^k$ so $\sqrt{\#G} = 2^{k/2}$

This is an exponential time attack, so DLP is hard in general.

Security levels in crypto

Measured in bits:

"n bits of security" means takes 2^n steps to break / solve at least.

- 128 bits internet today
- 256 bits top secret

For "generic" G, or G s.t. we don't know how to use extra info, to get 128 bits of security, we need $\# G \simeq 2^{256}$

Such a G is an elliptic curve $/\mathbb{F}_p$

then $\# G = \# E(\mathbb{F}_p) \approx p$

$E \; y^2 = x^3 + Ax + B$, $A, B \in \mathbb{F}_p$

Next attack: index calculus is only for $G = (\mathbb{Z}/p\mathbb{Z})^\times$

old: discrete log

Adversary E chooses a bound $B \approx 2^{\sqrt{\log p \log\log p}} < 2^{\log_2 p}$

and computes + stores the primes less than B

$$\{\ell_1, \ell_2, \ldots, \ell_r\}$$

We call these primes the factor base.

① E to compute $\log_g l_j$ for each l_j in the factor base.

To do this, E chooses random integers i, computes

$$g^i \equiv g_i \mod p$$

least residue modulo p

and then checks if $g_i \in \mathbb{Z}$ is divisible only by primes in the factor base.

If so, E saves $g_i = \prod l_j^{e_j(i)}$, if not keep going.

E keeps going until they have $\approx r$ (# of primes in the factor base) equations

$$g^i \equiv g_i = \prod \ell_j^{e_j(i)}$$

Each of these equations give an equation

$$i \equiv \sum_{j=1}^{r} e_j(i) \boxed{\log_g \ell_j} \mod p-1$$

unknowns

Now E solves their r equations in r unknowns,

(2) E takes random values u and computes

$$h \cdot g^{-u} \equiv h_u \mod p$$

and checks if $h_u \in \mathbb{Z}$ is divisible only by primes in the factor base.

As soon as one such u is found, E is done.

$$h_u = \prod l_j^{e_j(u)}$$

Then we have

known from step 1

$$\log_g h - u \equiv \sum_{j=1}^{r} e_j(u) \boxed{\log_g l_j} \mod p-1$$

(Remember that

$$h \cdot g^{-u} \equiv h_u \mod p \quad \text{and} \quad h_u = \prod l_j^{e_j(u)} \quad)$$

This attack takes $\sim 2^{(\log p)^{1/3}(\log\log p)^{2/3}}$ steps

which is subexponential in $\log p$ (size of p)

Accordingly, for 128 bits of security

need $p \approx 2^{1024}, 2^{2048}, 2^{3072}$

That's all for now!