Outline

Recap

Discrete Logarithm problem:

Given $g, h \in G$, find exponent x (an integer) such that $g^x = h$.

Known time and space complexity:

$N = \mathrm{ord}(g)$

Brute-force = $O(N)$ step → (step is multiplication)
$O(1)$ space
→ (this is exponential in the number of bits to store N)

Baby-step - Giant-Step = $O(\sqrt{N} \log N)$ step (still exponential)
$O(\sqrt{N})$ space

Diffie-Hellman key exchange

Properties of g when $G = \mathbb{Z}_p^*$ where p is prime

Fermat Little Theorem

Pohlig - Hellman algorithm

Diffie-Hellman key exchange protocol

public : p large prime

g large prime order in $\mathbb{Z}_p^*$

---

private : Alice : a

Bob : b

---

computation : Alice : $A = g^a \bmod p$

Bob : $B = g^b \bmod p$

Exchange : Alice $\xrightarrow{A}$ Bob

Alice $\xleftarrow{B}$ Bob

---

computation : Alice : $B^a \bmod p$

Bob : $A^b \bmod p$

---

① Alice and Bob compute the same secret shared value :

$$B^a = g^{ba} = g^{ab} = A^b \bmod p$$

② Eve knows A, B, g, p, and that $\exists\, a, b$ such that $A = g^a \bmod p$

$B = g^b \bmod p$.

Eve needs to find a, b.

This is equivalent to solve Discrete Logarithm.

Properties of g

The time and space complexity of Discrete Logarithm Algorithm depends on $ord(g)$.

We know $ord(g) \mid |G|$.

When $G = \mathbb{Z}_p^*$ where p is prime.

$|G| = p-1$

$ord(g) \mid p-1$ (This gives fermat's little theorem)

Is there $g \in \mathbb{Z}_p^*$ such that $ord(g) = p-1$?

In other words, is $\mathbb{Z}_p^*$ cyclic?

yes. $\mathbb{Z}_p^*$ is cyclic. The proof is quite involved and will be discussed later.

How to find g such that $ord(g) = p-1$?

In other words, how to find a generator of $\mathbb{Z}_p^*$?

No known deterministic polynomial algorithm.

Trial and Error.

Generators are common and easy to test.

If g is a generator, then g^i is also a geneator if $\gcd(i, \overset{ord(g)}{\overset{\parallel}{p-1}}) = 1$. (see next page for details)

Hence, there are $\varphi(p-1)$ generators.

Fermat's little theorem:

Let p be a prime.

For all $g \in \mathbb{Z}$, if p does not divide g,

then $g^{p-1} \equiv 1 \mod p$

Exponent of g lives in mod $p-1$.

That is, if $g^x = h \mod p$ then

$$g^{x \bmod p-1} = h \mod p$$

Application of Fermat's little theorem

① primarity test

② $a^{p-1} \equiv 1 \mod p$ $\longrightarrow$ $a^{p-1} \equiv 1 \mod p$

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

$a^{-1} = a^{p-2} \mod p$

$a^{p-2} \equiv a^{-1} \mod p$

Properties of g^i

$$\text{ord}(g^i) = \frac{\text{ord}(g)}{\gcd(i, \text{ord}(g))}$$

(a) if $\gcd(i, \text{ord}(g)) = 1$, then $\text{ord}(g^i) = \text{ord}(g)$

(b) Let the unique prime factorization of $\text{ord}(g)$ be

$\text{ord}(g) = N = q_1 q_2 \cdots q_n$ where $q_i = p_i^{e_i}$,

and $p_1, p_2, \ldots, p_n$ are distinct primes.

Let $N_i = \dfrac{N}{q_i} = q_1 q_2 \cdots q_{i-1} q_{i+1} q_{i+2} \cdots q_n$

$\text{ord}(g^{N_i}) = q_i$

Pohlig-Hellman Algorithm

Solves $g^x = h$ where $\mathrm{ord}(g) = N = q_1 q_2 \cdots q_n$

where $q_i = p_i^{e_i}$ and $p_1, p_2, \ldots, p_n$ are distinct primes.

For $i = 1, 2, \ldots, n$

$$N_i = \frac{N}{q_i}$$

$$g_i = g^{N_i}, \quad \mathrm{ord}(g_i) = q_i$$

$$h_i = h^{N_i}$$

Solve x_i such that $g_i^{x_i} = h_i$

Solve x such that

$$\left.\begin{aligned} x &\equiv x_1 \bmod q_1 \\ x &\equiv x_2 \bmod q_2 \\ &\vdots \\ x &\equiv x_n \bmod q_n \end{aligned}\right\} \text{By Chinese Remainder Theorem}$$

Remarks

① Pohlig-Hellman algorithm reduces the discrete logarithm problem for g of arbitrary order to the discrete logarithm for g^i of prime power order.

e.g: $S_{p^e} = \sqrt{p^e}$

② Suppose we can solve $g^x = h$ for $ord(g) = p^e$ in $O(S_{p^e})$ steps.

Then using Pohlig-Hellman algorithm, we can solve $g^x = h$ for $ord(g) = N = p_1^{e_1} p_2^{e_2} \cdots p_n^{e_n}$ in

$$O\left(\sum_{i=1}^{n} O(S_{p_i^{e_i}}) + \log N\right)$$

③ Pohlig-Hellman algorithm tells us that the discrete logarithm problem is easy to solve if $ord(g)$ is a product of small prime powers.

In particular, Diffie-Hellman is easy to break if $p-1$ is a product of small prime powers

Hence, for Diffie-Hellman exchange protocol, we should choose p such that $p = 2q+1$ where q is prime and use g such that $ord(g) = q$.

Such prime p is called safe prime.

Using Baby-step-Giant-step.

To solve $g^x = h$ for $ord(g) = p^e$

will require $O(p^{e/2})$ steps

Refinement algorithm can solve this in

$O(e S_p)$ steps

where $O(S_p)$ steps is require to solve

$g^x = h$ for $ord(g) = p$