13-02-21

Il nonno di Heidi

June 2021

1 Cyclic Groups and Discrete Logarithm

1.1 Groups and Cyclic Group

A group (G, *) is a set of elements with 2 operation: * [1] and the inversion, in which:

- for $a, b \in G \rightarrow a * b \in G$
- Associative property : $a * (b * c) = (a * b) * c$
- exist a neutral element $e \in G$ such that for each $a \in G \rightarrow a * e = e * a = a$
- for $a \in G$, exist the inverse $a^{-1} \in G$ in which: $a * a^{-1} = e$.

The cardinality of a Group $|G|$ is called **order of the group G**. A Group can have an *order* $\rightarrow \infty$ or not; in Cryptography, we always use finite groups.

An example of a Group is $\Re, +$, in fact:

- for $a, b \in \Re \rightarrow a + b \in \Re$
- $a + (b + c) = (a + b) + c$
- neutral element is 0
- for $a \in \Re$, the inverse is $-a$.

A Group $(G, *)$ is is cyclic if exists an element $a \in G$ in which each element $b \in G$ is generated by it:

$$b = a * a * a * a * a ... * a \tag{1}$$

a is called **generator of G**.

A Cyclic **Subgroup** is a subset of G in which, chosen $a \in G$:

$$< a >= \{..., a^{-2}, a^{-1}, 1, a^1, a^2, a^3, ...\} \tag{2}$$

[1] * is a generic operation that respects the properties ahead

Each element in ¡a¿ is unique, but is not necessary true that all elements in G are into ¡a¿. An example of cyclic group is $(Z_n, +)$ with generator = 1 and neutral element = 0; another interesting group is $(Z_n^*, *)$, where $Z_n^* = Z_n \backslash \{0\}$. Is not ever a cyclic one, but Gauss shows that is true for n = prime number. The generator in this case is called **primitive element**.

Let's consider a group Z_5^*, = {1, 2, 3, 4}; a generator of this group is 2, in fact:

$$< 2 >= \{1, 2, 2 * 2 = 4, 2^3 = 8 mod 5 = 3\}$$

1.2 Lagrange's Theorem

Let's consider a cyclic subgroup ¡a¿, with $a \in Z_n^*$, we can prove that whatever generator we choose, the order (cardinality) of ¡a¿ is ever a divisor of the order of Z_n^*. a simple proof of the theorem is the following:

we calculate:

$$< a >= \{1, a^1, a^2, ..., a^{\omega-1}\}$$

where ω is the cardinality of ¡a¿.

For construction, each element of ¡a¿ is unique and they are all elements of Z_n^*. now we have to verify if ¡a¿ is a cyclic group of G.

1. if this is true, then we finish, because $|Z_n^*| = | < a > | \rightarrow \omega$ is a divisor of Z_n^*
2. if not, means that at least one element $b \in Z_n^*$ is not in ¡a¿; in this case, we compute:

   $$< a > *b = \{b, a^1 * b, a^2 * b, ..., a^{\omega-1} * b\}$$

   also in this case, we have ω elements, that are disjoint to the original subgroup $< a > \rightarrow | < a > \cup < a > *b| = 2\omega$
3. verify that $< a > \cup < a > *b$ is exactly Z_n^* and reiterate the procedure if is not true.

In any case, we will found an order nω exactly equal to the order of Z_n^*.

1.3 Square and multiply algorithm

This method is used to calculate in an efficient way power of numbers in any kind of groups. For simplicity, consider $(Z^+ \backslash \{0\}, *)$ all positive numbers starting from 1

We mention it because it is used to Diffie Hellman key exchange method, because the complexity of this is lower, $O(\log_2(n))$, respect to the simpler approach to multiply the base n times.

1.3.1 How it works?

Let's consider $a, b \in Z^+$, and we want to calculate a^b;

- Setup: convert b into binary number $\rightarrow$ $b_{\text{base10}} = (d_{t-1}...d_2 d_1 d_0)_{\text{base2}}$ where t is number of bits needed to represent b ($d_{t-1} = 1$ for construction). declare a variable T = e the neutral number (in our case, e = 1).
- Loop: for each bit of b_{base2}, starting from d_{t-1},

  $$T = T * T$$

  - if $d_i = 1 \rightarrow T = T * a$
- at the end, we obtain the right value.

6.1.9 Efficient computation of powers: **square-and-multiply algorithm**

```
T=e
For i=t downto i=0
    T = T * T
    if d_i = 1
       T = T * a
return(T)
```

Figure 1: Implementation of square and multiply algorithm in python