5. Mathematical Background II

5.1 Euler's Phi Function

Given Zm → Phi(m) calculate the number of coprime number.

$$\Phi(m) = \prod_{i=1}^{m} (p_i^{e_i} - p_i^{e_i - 1})$$

$$m = p_1^{e_1} \cdot p_2^{e_2} \cdot \ldots \cdot p_n^{e_n}$$

5.2 Fermat's little theorem

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

$$a \cdot a^{p-2} \equiv 1 \bmod p$$
$$a^{-1} \equiv a^{p-2} \bmod p$$

p is prime

We can use this theorem in some cases to compute the multiplicative inverse, instead of using EEA.

5.3 Euler's theorem

$$a^{\Phi(m)} \equiv 1 \bmod m$$

a and m coprime; if p is prime then Phi(p) = p-1

Compute:

$$5^{200} \bmod 23$$

Since 23 is prime then:

$$\Phi(23) = 22$$
$$200 = 9 \cdot 22 + 2$$
$$5^{200} = 5^{22^9} \cdot 5^2$$
$$5^{\Phi(23)} = 5^{22} \equiv 1 \bmod 23$$
$$5^{200} \equiv 1^9 \cdot 5^2 \bmod 23 \equiv 25 \bmod 23 \equiv 2 \bmod 23$$

5.4 Chinese Remainder Theorem (CRT)

N pairwise coprime A integers.

has a unique solution modulo N = n1* … *nk.

$$x \equiv a_1 \bmod n_1$$
$$x \equiv a_2 \bmod n_2$$
$$\cdots$$
$$x \equiv a_k \bmod n_k$$

This can be used to “split” a modular problem on pq
over two “simpler” modular problems on p and q.

Given p and q coprime, if:

$$x \equiv a \bmod p$$
$$x \equiv a \bmod q$$

then:

$$x \equiv a \bmod p \cdot q$$

5.5 Orders

The Order of a finite group G is the cardinality.

Z* is the set of positive numbers smaller than m that are relatively coprime to m. The cardinality is thus equal to Phi(m).

ord(a) of an element a in a group G is the smaller positive integer k such that:

$$a^k = \underbrace{a \circ a \circ \ldots \circ a}_{k\ times} = 1,$$

5.6 Cyclic Group

A group G which contains an element a with maximum order ord(a) = |G| is said to be cyclic.

Elements with maximum order are called primitive elements or generators.

If |G| is prime, then all elements a != 1 ∈ G are primitive.

- Let (G, ∘) be a cyclic group then every element a ∈ G with ord(a) = s is the primitive element of a cyclic subgroup with s elements.
- [Lagrange’s Theorem] Let H be a subgroup of G. Then |H| divides |G|.

5.7 Discrete Logarithm Problem (DLP) on Z*p

problem of determining the integer 1 ≤ x ≤ p − 1 such that: a^x = b mod p
x must exists since a is a generator.

Another hard-to-solve problem used in cryptography is **IFP** (Integer Factorization Problem): given an integer n, we want to find integer a and b such that n = a*b

5.8 Exponentiation

How to compute x^a? e.g., x^4

1. naive approach: (((x * x) * x) * x) [4 multiplications]
2. faster approach: ((x * x) * (x * x)) [2 multiplications (only with power of 2)]

5.9 Square-and-Multiply (s-a-m) Algorithm

Our goal is to obtain exponent 11010 starting with exponent 1:

$$(x^{1_2})^2 = x^{10_2}$$
$$(x^{10_2}) \cdot x = x^{11_2}$$
$$(x^{11_2})^2 = x^{110_2}$$
$$(x^{110_2})^2 = x^{1100_2}$$
$$(x^{1100_2}) \cdot x = x^{1101_2}$$
$$(x^{1101_2})^2 = x^{11010_2}$$

```
static long fastExp(int base, int exp) {
    long f = 1;
    long b = base;
    while(exp > 0) {
        int lsb = 0x1 & exp;
        exp >>= 1;
        if(lsb) f *= b;
        b *= b;
    }
    return f;
}
```

1. in every iteration we square
2. if current bit is one, then we multiply by x