2. Mathematical Background I

2.1 Modular Arithmetic

Computations in finite sets

Crypto algorithms are based on finite sets that permits to study important properties.

Modulo operator

Let a, m, r in Z with m>0 : a = r mod m

m is *modulus,* r is **remainder (**not unique**)**

Or a = q * m + r

Congruence

2 number are congruent if: a = b mod *n*

- Properties

  - invariance over addition: $a \equiv b \bmod n \Leftrightarrow (a+c) \equiv (b+c) \bmod n$
  - invariance over multiplication: $a \equiv b \bmod n \Leftrightarrow (a \cdot c) \equiv (b \cdot c) \bmod n$
  - invariance over exponentiation: $a \equiv b \bmod n \Leftrightarrow a^k \equiv b^k \bmod n$

Modular reduction

- **Hard approach:** 3^8 = 6561 = 2 mod 7
- **Easy approach:** 3^8 = 3^4*3^4 = 81*81 = 4 * 4 mod 7 = 16 mod 7 = 2 mod 7

Modular division:

b/a = b*a^-1 mod *m*

The **inverse of a** is: a * a^-1 = 1 mod m

e.g.: 5/7 = 5*4 = 20 = 2 mod 9 (since 7*4 =28) = 1 mod 9

The inverse of a number *a mod m* *exists only if:* gcd(a,m) = 1

gcd = greatest common divisor

Algebraic strctures

Group

set of element and one group operator "cerchio" such that:

1. **closure:** $\forall a, b \in G : a \circ b \in G$
2. **associativity** $\forall a, b, c \in G : (a \circ b) \circ c = a \circ (b \circ c)$
3. **neutral element** $\exists 1 \in G$ such that $\forall a \in G : a \circ 1 = a$
4. **inverse element** $\forall a \in G, \exists a^{-1} \in G : a \circ a^{-1} = 1$

If abelian group then also:

5. **commutative** $\forall a, b \in G : a \circ b = b \circ a$

Ring

A ring has addition and multiplication

Hence, there is closure (the result is in the ring).

Informally, a ring is a structure in which we can always add, subtract and multiply, but we can only divide by certain elements (namely by those for which a multiplicative inverse exists).

An element a has a multiplicative inverse if and only if: gcd(a,m) = 1
We say that a is coprime or relatively prime to m.

Field

A **field** is a structure in which we can always add, subtract, multiply and divide. (not in tings or groups).

0 does not need to have the inverse

A **finite field** (Galois Field) is a field with a finite set of elements.

- **Properties:**
  - **Theorem:** Let F be a field, then for any non negative element a the inverse is unique

    Proof by contradiction. Suppose a-1 and b are two inverses of a then:
    ab = 1
    a^-1 ab = 1 a^-1
    (a^-1 a)b = 1 a^-1
    1 b = 1 a^-1
    b = a^-1 (hence the two inverses must be equal and cannot be different)

Finite fields (or Galois Fields)

Th.: Zp is a field if **p** is prime.

Th.: Finite field Exists only if they have p^m elements, with p prime and m integer

GF(2^8) is used by AES

The elements of a prime field GF(p) are the integers {0, 1, 2, ..., p-1}

- Addition: $a + b \equiv c \bmod p$
- Subtraction: $a - b \equiv d \bmod p$
- Multiplication: $a \cdot b \equiv e \bmod p$
- division (multiplicative inverse):

$$\forall a \in GF(p) : \exists\, a^{-1} : a \cdot a^{-1} \equiv 1 \bmod p$$

Extension field GF(2^m) Arithmetic

Elements are polynomial

$$a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_1 x + a_0 = A(x) \in GF(2^m)$$

E.g., Let's consider $GF(2^3)$:

$$A(x) = a_2 x^2 + a_1 x + a_0$$

Which are the elements of $GF(2^3)$?

$$\{0, 1, x, 1+x, x^2, 1+x^2, x+x^2, 1+x+x^2\}$$

We can generate them by consider all the possible combination of three bits:

$$\{000_2, 001_2, 010_2, 011_2, 100_2, 101_2, 110_2, 111_2\}$$

Multiplication alert: the result after the multiplication has to be in GF(2^n); so we have to perform **modulo reduction against** an irreducible polynomial

$$P(x) = x^3 + x + 1$$

Hence, multiplication:

$$C(x) = A(x) \cdot B(x) \bmod P(x)$$

In our example:

$$\begin{array}{rl} & (x^4 + x^3 + \qquad x + 1) : (x^3 + x + 1) = x \\ - & \underline{(x^4 + \qquad x^2 + x)} \\ & \qquad x^3 + x^2 + \qquad 1 \end{array}$$

Still not in GF(2^3)

Iteratively if needed

N.B.: For every field GF(2^m), there are several irreducible polynomials.

N.B.: the result depends on the P(x) used.

Multiplicative inverse → *Extended Euclidean Algorithm*

A(x)*A(x)^-1 = 1 mod P(x)

Euclidean algorithm

compute the greatest cdg(r0,r1)

E.g. r0 = 84 = 2 * 2 * 3 * 7

r1 = 30 = 2 * 3 * 5
gcd(r0, r1) = gcd(84, 30) = 6

Observation: gcd(r0, r1) = gcd(r0 - r1, r1) and more in general gcd(r0, r1) = gcd(r0 mod r1, r1)

- reduce the problem finding the gcd of two smaller numbers
- repeat recursively
- stop when r0 - r1 is zero (eventually switch to continue)

```
DO
  i = i+1
  ri = r(i-2) mod r(i-1)
WHILE ri != 0
RETURN
  gcd(r0,r1) = r(i-1)
```

E.g., gcd(973, 301) = ?

$$\overset{r_0}{973} = 3 * \overset{r_1}{301} + \overset{r_2}{70}$$

$$301 = 4 * 70 + \overset{r_3}{21}$$

$$70 = 3 * 21 + \overset{r_4}{7}$$

$$21 = 3 * 7 + 0$$

$$\gcd(973, 301) = 7$$

Extended Euclidean Algorithm (EEA)

Compute gcd(r0,r1) and mult. inverse with gcd(r0,r1) = 1

Three main steps:

1. Compute gcd(r0,r1) using EA
2. Compute r0 = q1*r1 + r2
3. Compute r2 = s2*r0 + t2*r1
   Repeat this process recursively.

See example p.36 or on the book

Input: positive integers r_0 and r_1 with $r_0 > r_1$
Output: $\gcd(r_0, r_1)$, as well as s and t such that $\gcd(r_0, r_1) = s \cdot r_0 + t \cdot r_1$.

Initialization:

$s_0 = 1 \qquad t_0 = 0$
$s_1 = 0 \qquad t_1 = 1$
$i = 1$

Algorithm:

1 DO
1.1 $\quad i = i + 1$
1.2 $\quad r_i = r_{i-2} \bmod r_{i-1}$
1.3 $\quad q_{i-1} = (r_{i-2} - r_i)/r_{i-1}$
1.4 $\quad s_i = s_{i-2} - q_{i-1} \cdot s_{i-1}$
1.5 $\quad t_i = t_{i-2} - q_{i-1} \cdot t_{i-1}$
WHILE $r_i \neq 0$
2 RETURN
$\quad \gcd(r_0, r_1) = r_{i-1}$
- $s = s_{i-1}$
$\quad t = t_{i-1}$

Main application of EEA is computing the inverses modulo n.

$$a \cdot a^{-1} \equiv 1 \bmod n$$

$$gcd(n, a) = 1 = s \cdot n + t \cdot a \equiv t \cdot a \bmod n$$

existence condition of inverse

EEA

inverse

For polynomial in GF(28), EEA works similarly. However, s and t must be replaced by two polynomials s(x) and t(x), where t(x) will be the inverse that we are looking for.

E.g., Inverse of $A(x) = x^2$ in GF(2^3) with $P(x) = x^3 + x + 1$

Iteration	$r_{i-2}(x)$	$= [q_{i-1}(x)]\, r_{i-1}(x) + [r_i(x)]$	$t_i(x)$
2	$x^3 + x + 1$	$= [x]\, x^2 + [x + 1]$	$t_2 = t_0 - q_1 t_1 = 0 - x1 \equiv x$
3	x^2	$= [x]\, (x + 1) + [x]$	$t_3 = t_1 - q_2 t_2 = 1 - x(x) \equiv 1 + x^2$
4	$x + 1$	$= [1]\, x + [1]$	$t_4 = t_2 - q_3 t_3 = x - 1\,(1 + x^2)$ $t_4 \equiv 1 + x + x^2$
5	x	$= [x]\, 1 + [0]$	Termination since $r_5 = 0$

Initially, we assume: $t_0(x) = 1,\ t_1(x) = 0$

FOR SMALL SIZE FINITE FIELD, A PRECOMPUTE LOOKUP TABLE IS THE MOST EFFICIENT METHOD FOR IMPLEMENTING MULTIPLICATION.