Number Theory and Cryptography

Michele Amoretti, University of Parma

Modular Arithmetics

Let us consider a and r integers, n positive integer, with $0 \leq r < n$. We write

$$r = a \bmod n \tag{1}$$

meaning that n divides $a - r$ (i.e., $n|a - r$), or equivalently that r is the remainder of the division between a and n. For example, $11 \mod 7 = 4$ and $-11 \mod 7 = 3$. We denote n as the **modulus**. It has the following properties:

$$[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n \tag{2}$$
$$[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n \tag{3}$$
$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n \tag{4}$$

We say that integers a and b are **congruent** modulo n if

$$a \bmod n = b \bmod n \tag{5}$$

and we write it as follows:

$$a \equiv b \pmod{n} \tag{6}$$

For example: $73 \equiv 4 \pmod{23}$ and $21 \equiv -9 \pmod{10}$. Properties:

$$a \equiv b \pmod{n} \text{ iff } n|a - b \tag{7}$$
$$a \equiv b \pmod{n} \text{ iff } b \equiv a \pmod{n} \tag{8}$$
$$\text{if } a \equiv b \pmod{n} \text{ and } b \equiv c \pmod{n}, \text{ then } a \equiv c \pmod{n} \tag{9}$$

Note that $a \equiv 0 \pmod{n}$ is equivalent to $n|a$ (i.e., n divides a).

Let us introduce $\mathbb{Z}_n = \{0, .., n-1\}$, where n is a positive integer. It is noteworthy that sums and products between numbers in $\mathbb{Z}_n$, modulo n, always give numbers in $\mathbb{Z}_n$. The following property also holds:

$$ab \bmod n = 1 \text{ iff } ab \equiv 1 \pmod{n} \tag{10}$$

Instead, the following equivalence is not always true:

$$ab \equiv 1 \bmod n \text{ iff } b \equiv a^{-1} \pmod{n} \tag{11}$$

Indeed, the inverse element $a^{-1} \in \mathbb{Z}_n$ of $a \in \mathbb{Z}_n$ exists only if $GCD(a, n) = 1$.

Application: RSA

Using RSA for confidentiality purposes requires the following operations:

$$C = M^e \bmod n \quad \text{encryption, performed by the sender} \tag{12}$$

$$M = C^d \bmod n = M^{ed} \bmod n \quad \text{decryption, performed by the recipient} \tag{13}$$

where

- M is a message block to be encrypted (plaintext)
- C is the corresponding cyphertext
- $\{e, n\}$ is the public key of the recipient
- $\{d, n\}$ is the private key of the recipient
- n is a very large positive integer, resulting from the product of two large primes p and q selected by the recipient
- $2^i < n \leq 2^{i+1}$, where i is the size of the block in terms of number of bits
- $\phi(n) = (p-1)(q-1)$ is the trapdoor key
- e is such that $1 < e < \phi(n)$ and $GCD(\phi(n), e) = 1$
- d is such that $d \equiv e^{-1} \pmod{\phi(n)}$
- equivalently, $de \equiv 1 \pmod{\phi(n)}$
  Proof: If a is relatively prime to n (i.e., $GCD(a, n) = 1$), then $(a \times b) \equiv (a \times c)(\text{mod } n)$ is equivalent to $b \equiv c \pmod{n}$. Let us replace a with e, b with d, c with e^{-1}, and n with $\phi(n)$ to complete the proof.

To compute d, the **extended Euclid algorithm** can be used.

Encryption and decryption require to calculate powers of integers, modulo n. Let us recall eq. (4):

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n.$$

To calculate $a^b \bmod n$, we use the latter property and the following fact:

$$a^b = \prod_{b_i \neq 0} a^{2^i} \tag{14}$$

where b_i is the ith bit of b in binary form. For example:

$$\begin{aligned} b = 7 \quad & (111 \text{ in binary form}) \\ \Longrightarrow a^7 &= a^{2^2} \times a^{2^1} \times a^{2^0} \end{aligned}$$

Therefore

$$a^b \bmod n = \left[\prod_{b_i \neq 0} a^{2^i}\right] \bmod n = \left(\prod_{b_i \neq 0} a^{2^i} \bmod n\right) \bmod n \tag{15}$$

Discrete Logarithm

If a is a **primitive root** of the prime number p, then

$$\begin{aligned} a \bmod p \\ a^2 \bmod p \\ .. \\ a^{p-1} \bmod p \end{aligned}$$

is an arrangement of the integers $1, .., p-1$. For example, let us consider $p = 19$: its primitive roots are 2, 3, 10, 13, 14, 15.

For any integer b and primitive root a of the prime number p, there is one and only one i such that $b \equiv a^i \pmod{p}$. We denote i as the **discrete logarithm** of b for the basis a, module p.

So far, no one was able to design a classical (i.e., not quantum) algorithm to find i for any pair (a, p) in polynomial time with respect to $\log p$.

Application: Diffie-Hellman

1. Alice and Bob share
   - a prime number q,
   - an integer α such that $\alpha < q$ and α is a primitive root of q.
2. Alice generates a private key $X_A < q$, while Bob generates a private key $X_B < q$.
3. Alice calculates the public key $Y_A = \alpha^{X_A} \bmod q$, while Bob calculates the public key $Y_B = \alpha^{X_B} \bmod q$.
4. Alice sends Y_A to Bob. Bob sends Y_B to Alice.
5. Alice calculates $K = (Y_B)^{X_A} \bmod q$, and Bob calculates $K = (Y_A)^{X_B} \bmod q$. The two keys are equal! Proof:

$$\begin{aligned} K &= (Y_B)^{X_A} \bmod q \\ &= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\ &= (\alpha^{X_B})^{X_A} \bmod q \\ &= \alpha^{X_B X_A} \bmod q \\ &= (\alpha^{X_A})^{X_B} \bmod q \\ &= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\ &= (Y_A)^{X_B} \bmod q \end{aligned}$$

It is not computationally feasible to obtain X_A from Y_B (X_B from Y_B), even if the pair (α, q) is known.

Finite Fields

A **field** is a set in which we can do addition, subtraction, multiplication, and division without leaving the set.

A **finite field** is a field with a finite number of elements. Such a number is called **order** of the finite field.

A finite field of order m exists if and only if

$$m = p^n \tag{16}$$

where n is a positive integer and p is prime.

A finite field of order p^n is denoted as $GF(p^n)$, where GF stands for Galois Field (an alternative name for finite field). We are particularly interested in $GF(2^n)$.

The elements of $GF(p^n)$ can be represented by means of polynomials:

$$A(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + .. + a_1x + a_0 \tag{17}$$

where $a_i \in \{0, 1, .., p-1\}$ for any i. For $GF(2^n)$, the coefficients $a_i \in \{0, 1\}$ are the bits of the binary representation of the considered element in $GF(2^n)$.

Addition in $GF(2^n)$

Given two numbers $A, B \in GF(2^n)$, their sum $C = A + B$ has binary representation $(c_{n-1}, c_{n-1}, .., c_1, c_0)$, where $c_i = a_i \oplus b_i$ for any $i \in \{0, 1, .., n-1\}$.

Example:

$$\begin{aligned} A(x) &= x^6 + x^4 + x + 1 \quad 01010011 \\ B(x) &= x^7 + x^6 + x^3 + x \quad 11001010 \\ C(x) &= x^7 + x^4 + x^3 + 1 \quad 10011001 \end{aligned}$$

Attention! The addition in $GF(2^n)$ produces a result that is different from the one produced by the addition modulo 2^8. For example, if $A = 83$ and $B = 202$, their sum is $C = 153$, which is different from $(83 + 202) \bmod 2^8$.

Multiplication in $GF(2^n)$

To compute the product of $A, B \in GF(2^n)$, use the following formula:

$$C(x) = A(x)B(x) \bmod P(x) \tag{18}$$

where $P(x)$ is an **irreducible** polynomial of degree n, i.e., a polynomial that cannot be decomposed as the product of two polynomials of lower degree.

Example:

$$\begin{aligned}
A(x) &= x^6 + x^4 + x^2 + x + 1 \\
B(x) &= x^7 + x + 1 \\
P(x) &= x^8 + x^4 + x^3 + x + 1 \\
C(x) &= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \bmod P(x) \\
&= x^7 + x^6 + 1
\end{aligned}$$

Inverse in $GF(2^n)$

All non-zero elements of a field (or finite field) have a **multiplicative inverse**. Formally, if a belongs to a field and $a \neq 0$, there is an element a^{-1} belonging to the same field such that $aa^{-1} = a^{-1}a = 1$.

The inverse of $A \in GF(2^n)$ is the number A^{-1} whose associate polynomial $A^{-1}(x)$ is such that $A^{-1}(x)A(x) \bmod P(x) = 1$, where $P(x)$ is an irreducible polynomial of degree n. To compute $A^{-1}(x)$, the most common (and efficient) approach is the Extended Euclid Algorithm ([2], p.163).

The inverse element of $a \in \mathbb{Z}_m = \{0, 1, .., m-1\}$ exists if and only if $GCD(a, m) = 1$. If m is prime, then $GCD(a, m) = 1$ for any $a \in \mathbb{Z}_m$. Therefore $\mathbb{Z}_p$ with p prime is a finite field. Instead, $\mathbb{Z}_{p^n}$ is not a finite field. Thus, for any $a \in \mathbb{Z}_{p^n}$, there is no $a^{-1} \in \mathbb{Z}_{p^n}$. For example, let us take $p = 2$ and $n = 3$ so that $\mathbb{Z}_{p^n} = \mathbb{Z}_8 = \{0, 1, .., 7\}$. Take $a = 4$, then $GCD(a, p^n) = 2$. Indeed, there is no $a^{-1} \in \mathbb{Z}_8$ such that $a^{-1}a \bmod 8 = 1$.

Arithmetics in $GF(2^n)$ vs. Arithmetics Modulo 2^n

Why using arithmetics in $GF(2^n)$ rather than arithmetics modulo 2^n (that is, arithmetics in $\mathbb{Z}_{2^n}$)? Suppose we wish to use 3-bit blocks for our encryption algorithm and use only the operations of addition and multiplication. If we compare the multiplication table for $GF(8)$ ([2], p.150) with that for $\mathbb{Z}_8$ ([2], p.159), we may observe that the number of occurrences of the resulting nonzero integers is uniform only in the $GF(8)$ case:

Integer	1	2	3	4	5	6	7
Occurrences in $\mathbb{Z}_8$	4	8	4	12	4	8	4
Occurrences in $GF(8)$	7	7	7	7	7	7	7

This is true for any n. Therefore, using arithmetics in $\mathbb{Z}_{2^n}$, a clever cryptanalysis technique could leverage the fact that some integers are more frequent than others in the ciphertext.