Séance 4 : Arithmétique Modulaire et Applications Cryptographiques

Hervé Talé Kalachi

Résumé

Ce cours explore les fondements de l'arithmétique modulaire avec une emphase sur ses applications en cryptographie moderne.

1 Congruences

1.1 Définitions Fondamentales

Définition 1.1 (Congruence Modulo)**.** Pour $n \in \mathbb{N}^*$, deux entiers a et b sont congrus modulo n si :

$$n \mid (a - b) \iff \exists k \in \mathbb{Z},\ a = b + kn$$

On note $a \equiv b \pmod{n}$.

Théorème 1.1 (Propriétés des Congruences)**.** Pour $a \equiv b \pmod{n}$ et $c \equiv d \pmod{n}$:

- $a \pm c \equiv b \pm d \pmod{n}$
- $ac \equiv bd \pmod{n}$
- $a^k \equiv b^k \pmod{n}$ pour $k \in \mathbb{N}$

1.2 Classes d'Équivalence

Exemple 1.1 (Système complet de résidus)**.** Pour $n = 5$, les classes résiduelles sont :

$$\{[0], [1], [2], [3], [4]\} \quad \text{où } [k] = \{5m + k \mid m \in \mathbb{Z}\}$$

Application aux Horloges

L'arithmétique modulaire modélise naturellement les systèmes cycliques comme les heures (mod 12/24) ou les jours de la semaine (mod 7).

2 Inversion Modulaire et Théorème d'Euler

2.1 Théorie de l'Inversion

Théorème 2.1 (Existence d'Inverse)**.** Un entier a admet un inverse modulo n si et seulement si :

$$\gcd(a, n) = 1$$

Exemple 2.1 (Calcul d'Inverse avec Euclide Étendu)**.** Trouvons l'inverse de 7 modulo 26 :

$$\begin{aligned} 26 &= 3 \times 7 + 5 \\ 7 &= 1 \times 5 + 2 \\ 5 &= 2 \times 2 + 1 \end{aligned}$$

Remontée :

$1 = 5 - 2 \times 2 = 5 - 2 \times (7 - 5) = 3 \times 5 - 2 \times 7 = 3 \times (26 - 3 \times 7) - 2 \times 7 = 3 \times 26 - 11 \times 7$

L'inverse est $-11 \equiv 15 \mod 26$

2.2 Théorème d'Euler-Fermat

Théorème 2.2 (Théorème d'Euler)**.** Si $a \wedge n = 1$, alors :

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

où $\phi(n)$ est l'indicatrice d'Euler.

Remarque 2.1. Pour n premier, $\phi(n) = n - 1$ (Petit Théorème de Fermat)

3 Applications Cryptographiques

3.1 Cryptosystème RSA

- Clé publique : $(n = pq, e)$ avec $e \wedge \phi(n) = 1$
- Clé privée : $d = e^{-1} \mod \phi(n)$
- Chiffrement : $c \equiv m^e \mod n$
- Déchiffrement : $m \equiv c^d \mod n$

3.2 Échange de Clés Diffie-Hellman

1. Alice et Bob choisissent p premier et g générateur modulo p
2. Alice envoie $A = g^a \mod p$
3. Bob envoie $B = g^b \mod p$
4. Clé secrète partagée : $K = A^b \mod p = B^a \mod p$

4 Implémentations Algorithmiques

Listing 1 – Exponentiation Modulaire Rapide

```
def exp_mod(a, b, n):
    result = 1
    a = a % n
    while b > 0:
```

```python
        if b % 2 == 1:
            result = (result * a) % n
        a = (a * a) % n
        b = b // 2
    return result
```

Listing 2 – Inverse Modulaire avec Python

```python
def inverse_mod(a, n):
    g, x, y = euclide_etendu(a, n)
    if g != 1:
        return None  # Pas d'inverse
    else:
        return x % n
```

5 Exercices

Exercice 1 : Bases

Résoudre $12x \equiv 9 \pmod{15}$

Exercice 2 : Théorie des Nombres

Montrer que si $a \equiv b \pmod{n}$, alors $\gcd(a, n) = \gcd(b, n)$

Exercice 3 : Cryptographie

1. Générer des clés RSA pour $p = 5$, $q = 11$, $e = 3$*
2. Chiffrer le message $m = 12$

Défi :

Résoudre le système de congruences :

$$\begin{cases} x \equiv 2 \pmod{3} \\ x \equiv 3 \pmod{5} \\ x \equiv 2 \pmod{7} \end{cases}$$

Références

[1] Paar, Christof et Pelzl, Jan. *Understanding Cryptography*. Springer, 2010.

[2] Shoup, Victor. *A Computational Introduction to Number Theory and Algebra*. Cambridge, 2005.

[3] Menezes, Alfred J. *Handbook of Applied Cryptography*. CRC Press, 1996.