Séance 8 - Travaux Dirigés et Pratiques
Arithmétique et Programmation

Yann Rotella

2026

Partie 1 - Théorie rappels

Exercice 1. *Calculs modulaires rappels.*
Calculer :

(1) $24000 \mod 24$
(2) $38 \mod 13$
(3) $14 \cdot 17 \mod 15$
(4) $3 \cdot (1 + 22) \mod 11$

Exercice 2. *Algorithme d'Euclide étendu.*
Soient $a = 1234$ et $b = 357$.

(1) Calculer le $\text{pgcd}(a, b)$ en utilisant l'algorithme d'Euclide.
(2) Calculer $u, v \in \mathbb{Z}$ tels que $u \cdot a + v \cdot b = \text{pgcd}(a, b)$.
(3) Calculer l'inverse de b modulo a.

Exercice 3. *Diffie-Hellman - algorithmique.*
L'objectif de cet exercice consiste à regarder la *complexité* des calculs réalisés lors de l'échange de clefs Diffie-Hellman. On se place dans $\mathbb{Z}_p$ avec $\log_2(p) = n$.

(1) On suppose que l'on a un entier M de ℓ bits et un entier N de k bits, avec $k < \ell$. Combien coûte la réduction modulaire, algorithmiquement de M par N ?
(2) Combien coûte la multiplication de deux entiers algorithmiquement ?
(3) Comment calculer efficacement $X^t \mod p$? Détaillez votre réponse.
(4) Donner alors la complexité totale du coût de l'algorithme d'échange de clefs Diffie-Hellman.
(5) Même question pour ElGamal.

Partie 2 - Programmation

L'avantage de Python est sa gestion native des entiers de taille arbitraire. Pour comprendre la complexité et l'intérêt des différents algorithmes cryptographiques, nous allons, au fur et à mesure des différents TDs, réimplémenter les différents cryptosystèmes.

Dans toute cette partie, on n'utilise pas de librairie, on recode tout avec des fonctions arithmétiques élémentaires

Exercice 4. *Algorithme d'euclide étendu.*
But de l'exercice :

(1) On s'échauffe : implémenter l'algorithme d'Euclide étendu. Testez sur des petites valeurs.

Algorithm 1 Euclide étendu

Input: a, b deux entiers naturels
Output: r, u, v tels que $r = \text{pgcd}(a, b)$ et $r = au + bv$

```
r = a, r' = b, u = 1, v = 0, u' = 0, v' = 1
while r' ≠ 0 do
    q = r/r'
    r_s = r, u_s = u, v_s = v
    r = r', u = u', v = v'
    r' = r_s − q r', u' = u_s − q u', v' = v_s − q v'
end while
return (r, u, v)
```

(2) Implémenter une fonction inverse, qui prend deux entiers (x, n) et renvoie y le plus petit entier positif tel que $yx = 1 \mod n$ si un tel y existe et -1 sinon.

Exercice 5. *L'échange de clefs Diffie-Hellman.*
Programmez l'échange de clefs de Diffie et Hellman sur $\mathbb{Z}_p$ avec p premier. Vérifier que les calculs des deux parties donnent à la fin le même résultat. Testez sur des petits entiers premiers.

Exercice 6. *ElGamal.*
Programmez les fonctions de chiffrement et de déchiffrement du cryptosystème d'ElGamal.

Exercices complémentaires

Exercice 7. *Diffie-Hellman.*
Alice et Bob souhaitent échanger une clé secrète en utilisant le protocole d'échange de clés Diffie-Hellman. Ils se mettent d'accord sur le nombre premier $p = 17$. Afin d'exécuter le protocole, Alice et Bob ont également besoin de se mettre d'accord sur un élément générateur de $\mathbb{Z}_{17}^*$, qu'on notera α.

(1) Calculer le plus petit élément générateur de $\mathbb{Z}_{17}^*$.

(2) Alice choisit comme clé secrète $a = 5$ tandis que Bob choisit comme clé secrète $b = 7$. Calculer la clé publique A d'Alice et la clé publique B de Bob, en utilisant l'élément générateur α calculé dans la question précédente.

(3) Calculer la clé secrète commune k_{AB} qu'établissent Alice et Bob après l'exécution du protocole.

Exercice 8. *Diffie-Hellman, valeurs faibles.*
Pour l'échange de clés Diffie-Hellman, les clés privées sont choisies dans l'ensemble $\{2, ..., p-2\}$. Pourquoi, sont les valeurs 1 et $p - 1$ exclues ? Décrire leur faiblesse.