

---

# Cryptographie à clé publique– Solutions feuille de TD 1

27/01/2023

---

Retrouvez le sujet du TD et d'autres exercices à l'adresse :

[www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/clef-publique.html](http://www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/clef-publique.html)

(★) exercice fondamental      (★★) pour s'entraîner      (★★★) pour aller plus loin      ☞ sur machine

---

## Exercice 1. (★) Application du protocole de Diffie–Hellman.

On exécute le protocole de Diffie–Hellman dans le groupe multiplicatif  $G = \mathbb{F}_p^\times$  avec les paramètres suivants :

- $p = 19$ ,
- le générateur du groupe est  $g = 2$ ,
- la valeur secrète d'Alice est  $a = 4$ ,
- la valeur secrète de Bob est  $b = 6$ .

**Question 1.**– Quelle est la valeur commune obtenue par Alice et Bob ?

On souhaite maintenant exécuter le protocole de Diffie–Hellman dans le sous-groupe des résidus quadratiques de  $\mathbb{F}_p^\times$ , que l'on note  $QR_p^\times$ . Pour cela on garde la valeur  $p = 19$ .

**Question 2.**– L'élément  $g = 2$  reste-t-il un générateur de  $QR_p^\times$  ?

**Question 3.**– Démontrer que 5 est un générateur de  $QR_p^\times$ .

**Question 4.**– Exécuter le protocole de Diffie–Hellman dans  $QR_p^\times$  avec les valeurs suivantes :

- $p = 19$ ,
- $g = 5$ ,
- la valeur secrète d'Alice  $a = 4$ ,
- la valeur secrète de Bob est  $b = 6$ .

## Solutions de l'Exercice 1.

**Solution Q1.** La valeur commune est :

$$g^{ab} = 2^{4 \times 6} \pmod{19}$$

Plus précisément, Bob reçoit  $g^a = 2^4 = 16$  et opère

$$16^6 \equiv (-3)^6 \equiv 27^2 \equiv 8^2 \equiv 64 \equiv 7 \pmod{19}.$$

**Solution Q2.** Vérifions d'abord si 2 est un résidu quadratique modulo  $p$ . D'après le critère d'Euler, cela revient à vérifier si  $2^{(p-1)/2}$  vaut 1. On a :

$$2^{(p-1)/2} = 2^9 = (2^4)^2 \times 2 = 16^2 \times 2 \equiv (-3)^2 \times 2 = 18 \equiv -1 \pmod{19}$$

Donc 2 n'est pas un résidu quadratique modulo  $p$ ; il ne peut donc pas en être un générateur.

**Solution Q3.** Tout d'abord, on vérifie que 5 est un carré modulo  $p$ . On a :

$$5^{(p-1)/2} = 5^9 = (5^2)^4 \times 5 \equiv 6^4 \times 5 = 36^2 \times 5 \equiv (-2)^2 \times 5 = 20 \equiv 1 \pmod{19}$$

Donc  $5 \in \text{QR}_{19}^\times$ . Pour démontrer que c'en est un générateur, il faut montrer que son ordre est maximal, donc égal à 9. Pour cela, il suffit ici de démontrer que  $5^3 \not\equiv 1 \pmod{19}$ , car les ordres possibles sont les diviseurs de 9. Or,

$$5^3 = 25 \times 5 \equiv 6 \times 5 = 30 \equiv 11 \pmod{19}.$$

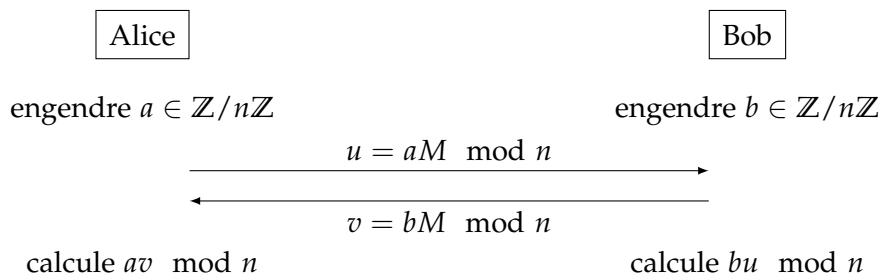
donc 5 est bien un générateur de  $\text{QR}_{19}^\times$ .

**Solution Q4.** Grâce aux calculs précédents, on obtient :

$$g^a = 5^4 \equiv -2 \pmod{19} \quad \text{puis} \quad g^{ab} = (-2)^6 = 64 \equiv 7 \pmod{19}$$

## **Exercice 2. (\*\*) Protocole de Diffie–Hellman dans $(\mathbb{Z}/n\mathbb{Z}, +)$ .**

On considère une variante additive du protocole de Diffie–Hellman dans le groupe fini  $G = (\mathbb{Z}/n\mathbb{Z}, +)$  où  $n$  est un très grand nombre entier. Pour cela, on fixe un générateur  $M \neq 0$  du groupe  $G$ , que l'on publie. Dans cette version « additive », le protocole devient donc :



**Question 1.–** Donner une caractérisation simple des générateurs du groupe additif  $(\mathbb{Z}/n\mathbb{Z}, +)$ . Puis, proposer un exemple de générateur qui convient pour tout  $n$ .

**Question 2.–** Vérifier que lorsqu'ils suivent le protocole ci-dessus, Alice et Bob détiennent bien un secret commun.

**Question 3.–** Rappeler un algorithme qui calcule l'inverse modulaire dans  $\mathbb{Z}/n\mathbb{Z}$  (autrement dit, qui effectue l'opération  $x \mapsto x^{-1} \pmod n$ ). La complexité de cet algorithme est-elle polynomiale en  $\log n$  ?

**Question 4.–** Selon vous, la variante du protocole de Diffie–Hellman proposée dans cet exercice est-elle sûre ? Justifier.

### **Solutions de l'Exercice 2.**

**Solution Q1.** Les générateurs du groupe additif  $(\mathbb{Z}/n\mathbb{Z}, +)$  sont les entiers compris entre 1 et  $n - 1$  qui sont premiers avec  $n$ . Par exemple : 1.

**Solution Q2.** À la fin de l'exécution du protocole, Alice détient

$$av \equiv abM \pmod{n}$$

et Bob détient

$$bu \equiv baM \pmod{n}$$

Ces deux valeurs sont bien égales.

**Solution Q3.** Pour calculer l'inverse modulaire de  $x$  dans  $\mathbb{Z}/n\mathbb{Z}$ , l'algorithme est le suivant :

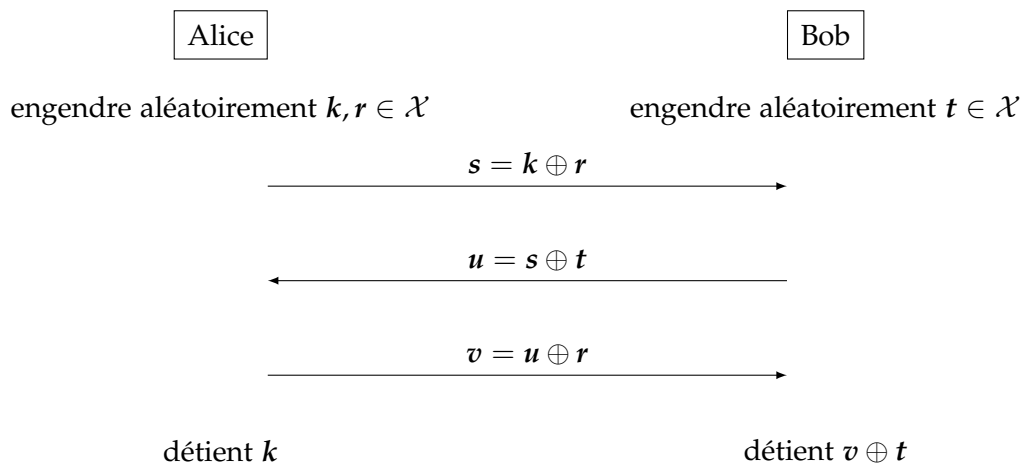
1. Exécuter l'algorithme d'Euclide étendu sur les entiers  $x$  et  $n$ , de manière à obtenir  $u$  et  $v$  tels que  $xu + nv = 1$  (en effet, pour que  $x$  soit inversible modulo  $n$ , il faut que  $x$  et  $n$  soient premiers entre eux).
2. Retourner  $u$ .

La complexité de l'algorithme d'Euclide étendu est de  $O(\log(n))$  opérations élémentaires (additions et multiplications modulo  $n$ ). Le calcul de l'inverse modulaire est donc efficace.

**Solution Q4.** Le protocole n'est pas sûr. En effet, un attaquant peut simplement calculer l'inverse modulaire  $M^{-1} \pmod{n}$ . Puis, comme  $aM$  est transmis d'Alice à Bob, l'attaquant obtient  $a$  en calculant  $aM \times M^{-1} \pmod{n}$ . Ensuite, il lui suffit de calculer  $a \times bM = abM \pmod{n}$  (car  $bM$  est également connu de l'attaquant, car il est transmis de Bob à Alice).

### **Exercice 3. (\*\*) Échange de clefs et masque jetable.**

On note  $\mathcal{X} = \mathbb{F}_2^n$  l'ensemble des chaînes de bits de longueur  $n$ , munies de l'opération de xor bit-à-bit, notée  $\oplus$ . On considère le protocole d'échange de clefs suivant.



**Question 1.–** Vérifier qu'Alice et Bob détiennent bien un secret commun.

**Question 2.–** Le protocole est-il sûr ? Justifier.

### **Solutions de l'Exercice 3.**

**Solution Q1.** À la fin de l'exécution du protocole, Alice détient la chaîne  $k$  et Bob détient

$$v \oplus t = u \oplus r \oplus t = s \oplus t \oplus r \oplus t = k \oplus r \oplus r = k.$$

C'est bien la même valeur.

**Solution Q2.** Alice et Bob s'échangent les données suivantes :

$$\begin{cases} s &= k \oplus r \\ u &= s \oplus t = k \oplus r \oplus t \\ v &= u \oplus r = k \oplus t \end{cases}$$

Un observateur qui a accès à ces données peut donc retrouver le secret commun  $k$  en calculant

$$s \oplus u \oplus v = 3k \oplus 2r \oplus 2t = k.$$

#### **Exercice 4. (\*\*\* ) Problèmes DH et sqDH.**

Soit  $G$  un groupe cyclique de générateur  $g$ . On rappelle que le problème calculatoire de Diffie–Hellman dans  $G$  est :

CDH : étant donné  $(g, g^a, g^b)$ , calculer  $g^{ab}$ .

On définit maintenant le problème de « Diffie–Hellman carré » dans  $G$  comme :

sqDH : étant donné  $(g, g^a)$ , calculer  $g^{a^2}$ .

**Question 1.–** Expliquer en quoi sqDH peut être vu comme un sous-problème de CDH.

**Question 2.–** Supposons que l'on détienne un algorithme  $A$  qui résout efficacement sqDH. Démontrer qu'à partir d'une instance  $(g, g^a, g^b)$  de CDH, on peut calculer  $g^{2ab}$ .

Pour simplifier, on suppose ici que l'ordre  $r$  du groupe  $G$  est un entier impair et connu.

**Question 3.–** On dit que  $x$  est un carré dans  $G$  s'il existe un  $y \in G$  tel que  $x = y^2$ . Décrire un algorithme qui calcule une racine carrée d'un carré  $x \in G$ .

**Question 4.–** En conclure que CDH se réduit à sqDH (dans le cas où l'ordre  $r$  est impair et connu).

#### **Solutions de l'Exercice 4.**

**Solution Q1.** Pour  $a = b$ , le problème CDH correspond à calculer  $g^{a^2}$  à partir de  $(g, g^a, g^a)$ . C'est donc exactement sqDH.

**Solution Q2.** Soit  $A$  un algorithme qui résout sqDH. On souhaite résoudre une instance  $(g, g^a, g^b)$  de CDH, c'est-à-dire calculer  $g^{ab}$  à partir de ces valeurs.

Observons que l'on peut calculer les valeurs suivantes :

- $g^{a^2}$  grâce à  $A(g, g^a)$ ,
- $g^{b^2}$  grâce à  $A(g, g^b)$ ,
- $g^{(a+b)^2}$  grâce à  $A(g, g^a g^b)$ .

On obtient alors

$$\frac{g^{(a+b)^2}}{g^{a^2} g^{b^2}} = g^{a^2+b^2+2ab-a^2-b^2} = g^{2ab}.$$

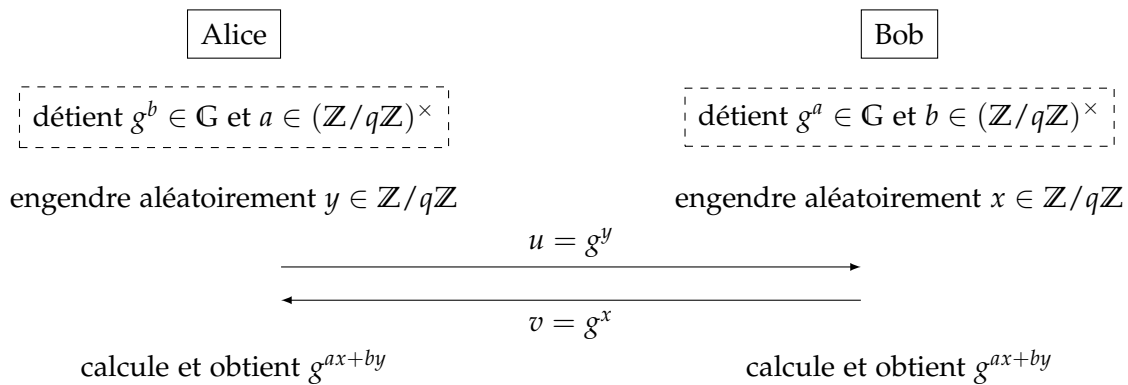
**Solution Q3.** Soit  $x = y^2$  un carré dans  $G$ . Comme  $r$  est impair et connu, on peut commencer par calculer  $m = 2^{-1} \pmod r$ . Ce calcul est une simple inversion modulaire. Puis, on calcule  $x^m = y^{2m} = y$  grâce à une exponentiation rapide.

**Solution Q4.** Avec la question 2, on obtient  $g^{2ab} = (g^{ab})^2$ , duquel on peut extraire  $g^{ab}$  grâce au calcul de racine carrée de la question 3.

**Exercice 5. (\*\*) Une variante de Diffie–Hellman.**

Alice et Bob souhaitent échanger un secret commun. Pour cela, ils mettent en place une variante du protocole de Diffie–Hellman. On se place donc dans un groupe cyclique  $G$  d'ordre  $q$ . **On suppose que  $q$  est premier.** Enfin, un générateur  $g$  de  $G$  est donné publiquement.

On suppose qu'Alice a engendré une valeur secrète aléatoire  $a \in (\mathbb{Z}/q\mathbb{Z})^\times$ , et a publié  $g^a \in G$ . De même, Bob a engendré une valeur secrète  $b \in (\mathbb{Z}/q\mathbb{Z})^\times$  et publié  $g^b \in G$ . Le protocole d'échange de secret commun est décrit ci-dessous.



**Question 1.–** Détailler les calculs permettant à Alice et Bob de calculer la valeur commune  $g^{ax+by}$ , à partir des valeurs que chacun détient. En donner la complexité.

**Question 2.–** Une fois que les entiers  $a$  et  $b$  sont fixés, quelle est la taille de l'ensemble des valeurs communes possibles ?

**Question 3.–** Pourquoi l'attaque de la personne au milieu (*man-in-the-middle*), telle que décrite dans le cours pour le protocole de Diffie–Hellman, ne fonctionne-t-elle pas dans le cas présent ?

**Question 4.–** Supposons que l'on détienne un algorithme  $\mathcal{A}$  qui sache résoudre efficacement le problème CDH (Diffie–Hellman calculatoire). Démontrer qu'on peut alors retrouver la valeur commune détenue par Alice et Bob en observant leurs échanges (c'est-à-dire en procédant à une attaque passive).

**Solutions de l'Exercice 5.**

**Solution Q1.** Alice obtient de Bob la valeur  $v = g^x$ , et elle connaît déjà  $g^b$  (la valeur précédemment publiée par Bob),  $a$  (sa valeur privée) et  $y$  (le secret qu'elle a engendré). Elle peut donc calculer :

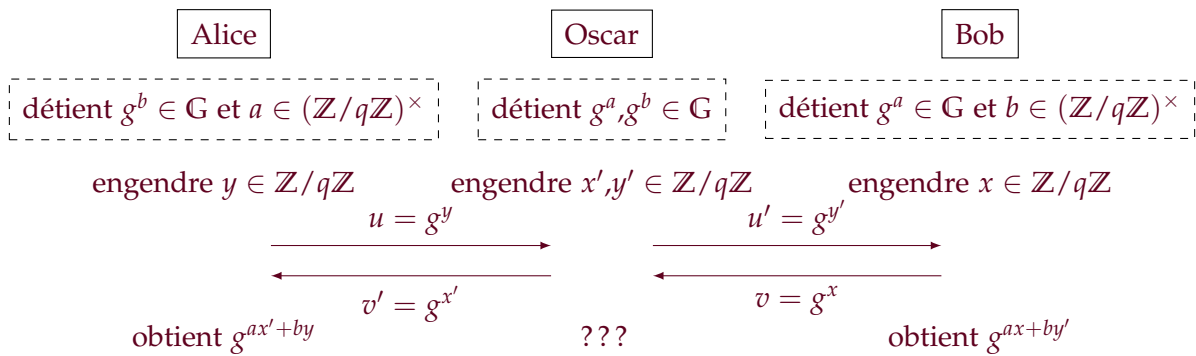
$$v^a (g^b)^y = g^{ax+by}.$$

De la même manière, Bob peut calculer  $u^b (g^a)^x = g^{ax+by}$ .

La complexité d'une exponentiation est en  $O(\log q)$  opérations dans  $G$ , donc la complexité du calcul complet (une multiplication + deux exponentiation) est également en  $O(\log q)$  opérations.

**Solution Q2.** Comme  $q$  est premier et  $a$  et  $b$  sont inversibles modulo  $q$ , l'entier  $ax + by$  peut atteindre n'importe quelle valeur dans  $\mathbb{Z}/q\mathbb{Z}$ . L'élément  $g^{ax+by}$  peut donc être n'importe quel élément du groupe  $G$ . On a donc  $q$  valeurs communes possibles.

**Solution Q3.** Si Oscar se place au milieu, on obtient le schéma suivant :



Comme Oscar ne connaît ni  $a$ , ni  $b$ , il ne peut pas calculer de valeur commune à Alice ou Bob.

**Solution Q4.** Supposons que l'algorithme  $\mathcal{A}$  sache résoudre CDH. Autrement dit, pour tous  $i, j \in \mathbb{Z}/q\mathbb{Z}$ , la sortie de  $\mathcal{A}(g, g^i, g^j)$  est  $g^{ij}$ .

On souhaite calculer  $g^{ax+by}$ . Pour cela, on connaît  $g^a$  et  $g^b$  (les valeurs publiques d'Alice et Bob) ainsi que  $g^x$  et  $g^y$  (les échanges entre Alice et Bob).

À partir de  $g^a$  et  $g^x$ , on peut donc obtenir  $g^{ax} = \mathcal{A}(g, g^a, g^x)$ . De manière similaire, on obtient  $g^{by} = \mathcal{A}(g, g^b, g^y)$ . En multipliant ces deux valeurs, on trouve  $g^{ax+by}$ .

### Exercice 6. (★★) $\square$ Implantation de Diffie–Hellman dans $\text{QR}_p^\times$ .

Dans cet exercice on suppose que  $p$  est un nombre premier de taille importante **tel que**  $p' = (p - 1)/2$  **est également un nombre premier**. Un tel nombre premier  $p$  est appelé **nombre premier sûr**, tandis que  $p'$  est un **nombre premier de Sophie Germain**.

**Question 1.–** Trouver dans les bibliothèques `random` et `math` de python comment :

- tirer uniformément un entier entre 1 et  $x$ ,
- calculer efficacement une puissance modulaire,
- calculer efficacement un inverse modulaire.

**Question 2.–** Écrire une fonction `is_square(x, p)` qui teste si un entier  $x$  est un résidu quadratique modulo  $p$ .

**Question 3.–** Écrire une fonction `random_QR_generator(p)` qui retourne un générateur aléatoire du groupe des résidus quadratiques modulo  $p$ . On rappelle que l'on suppose que  $p' = (p - 1)/2$  est également un nombre premier.

**Question 4.–** Écrire les fonctions suivantes du protocole de Diffie–Hellman :

- une fonction `system_parameters(t)` qui définit les paramètres du protocole : un nombre premier  $p$  aléatoire de taille  $t$  bits tel que  $p' = (p - 1)/2$  est aussi premier, et le générateur aléatoire  $g$  de  $\text{QR}_p^\times$ ,
- une fonction `random_secret(p)` qui définit la valeur secrète  $x$  engendrée par un participant au protocole,
- une fonction `to_send(x, g, p)` qui déduit de la valeur secrète  $x$  et du générateur  $g$  la valeur à transmettre à l'autre participant,
- une fonction `shared_value(gy, x, p)` qui déduit de la valeur reçue  $gy (= g^y)$  et de la valeur secrète  $x$ , la valeur commune aux deux participants.

Pour l'implantation de la fonction `system_parameters(t)`, on pourra notamment s'aider de la fonction `Crypto.Util.number.getPrime(t)` de la bibliothèque `Crypto.Util`, voir :

<https://pycryptodome.readthedocs.io/en/latest/src/util/util.html#Crypto.Util.number.getPrime>

**Question 5.**– En choisissant une taille  $t = 64$  (exemple-jouet), tester vos fonctions en exécutant les actions successives d’Alice et Bob dans le protocole de Diffie–Hellman.

### **Solutions de l’Exercice 6.**

Voir script à l’adresse :

[https://www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/docs/CP/td/scripts/implem\\_DH\\_QRp.py](https://www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/docs/CP/td/scripts/implem_DH_QRp.py)

---