# Carnet de Travaux Libres
# Sécurité
# Licence 3 Informatique

Julien Bernard

## Table des matières

## Exercice 1 : Calcul de PGCD

**Question 1.1** Calculer les PGCD des nombres suivants à l'aide de l'algorithme d'Euclide basé sur le lemme d'Euclide :

— $(a, b) = (120, 55)$
— $(a, b) = (49, 38)$
— $(a, b) = (42, 56)$

**Question 1.2** Reprendre la question précédente pour trouver les cœfficients de Bezout à l'aide de l'algorithme de Bezout.

## Exercice 2 : Inverses modulaires

**Question 2.1** Résoudre les équations suivantes :

1. $17x \equiv 10 \mod 50$
2. $35x \equiv 10 \mod 50$
3. $35y \equiv 11 \mod 50$

## Exercice 3 : Solution du théorème des restes chinois

**Question 3.1** Soit $\hat{n}_i = \frac{n}{n_i}$. Justifier que $\hat{n}_i$ est inversible modulo $n_i$.

**Question 3.2** On note $\hat{n}_i^{-1}$ cet inverse et on pose $e_i = \hat{n}_i \times \hat{n}_i^{-1}$. Calculer $e_i \mod n_j$ pour tout $1 \leq j \leq k$

**Question 3.3** En déduire une solution du théorème des restes chinois.

## Exercice 4 : Le problème des pirates

Une bande de 17 pirates s'est emparée d'un butin composé de pièces d'or d'égale valeur. Ils décident de se les partager également et de donner le reste au cuisinier chinois. Celui-ci recevrait trois pièces. Mais les pirates se querellent et six d'entre eux sont tués. Le cuisinier recevrait alors 4 pièces. Survient alors un naufrage et seuls 6 pirates, le trésor et le cuisinier sont sauvés. Le partage laisserait 5 pièces d'or à ce dernier.

**Question 4.1** Quelle est alors la fortune minimale que peut espérer ce dernier s'il décide d'empoisonner le reste des pirates ? On pourra utiliser les résultats suivants :

— $17 \times 11 \times 6 = 1122$ et $66 = 3 \times 17 + 15$
— $8 \times 66 \times 3 = 1584$ et $16 \times 102 = 1632$
— $4151 = 3 \times 1122 + 785$

## Exercice 5 : Indicatrice d'Euler

**Question 5.1** Pour $n = p^k$ où $p$ est premier et $k > 0$, montrer que $\varphi(n) = \left(1 - \frac{1}{p}\right) \times n$.

**Question 5.2** On admet que si $n_1$ et $n_2$ sont premiers entre eux, alors $\varphi(n_1 \times n_2) = \varphi(n_1) \times \varphi(n_2)$. Montrer que si $n = p_1^{\alpha_1} \times \ldots \times p_k^{\alpha_k}$, alors

$$\varphi(n) = n \times (1 - \frac{1}{p_1}) \times \ldots \times (1 - \frac{1}{p_k})$$

**Question 5.3** À l'aide du théorème d'Euler, proposer un algorithme de calcul de l'inverse modulo $n$.

**Question 5.4** Application : calculer (le plus vite possible) $22^{-1} \mod 63$ et $5^{2001} \mod 24$. On pourra utiliser : $22^2 = 43 \mod 63$ et $22^4 = 22 \mod 63$.

**Question 5.5** Donner trois algorithmes différents pour calculer l'inverse de $y$ modulo $n = p_1^{\alpha_1} \times \ldots \times p_k^{\alpha_k}$, où les $p_i$ sont des entiers premiers distincts.

## Exercice 6 : Chiffrement et déchiffrement RSA

Dans toute la suite, on pourra utiliser les résultats numériques suivants :
— $319 = 11 \times 29$ ; $10^{11} = 263 \mod 319$ ; $263^2 = 216 \times 319 + 265$ ;
— $133^3 = 12 \mod 319$ ; $133^{25} = 133 \mod 319$ ;
— $11^2 = 121 \mod 280$ ; $11^4 = 81 \mod 280$ ; $11^8 = 121 \mod 280$ ; $11^{16} = 81 \mod 280$ ;
— $95 = 64 + 31$ ; $81 \times 11 = 51 \mod 280$ ; $81 \times 121 = 1 \mod 280$.

On considère la clef publique RSA (11, 319), c'est-à-dire pour $n = 319$ et $e = 11$.

**Question 6.1** Quel est le chiffrement avec cette clé du message $M = 100$ ?

**Question 6.2** Calculer $d$ la clé privée correspondant à la clé publique $e$.

**Question 6.3** Déchiffrer le message $C = 133$.

**Question 6.4** Le message codé 625 peut-il résulter d'un codage avec la clé publique ? Même question avec la clé privée.

## Exercice 7 : Cryptographie RSA et authentification

Un professeur envoie ses notes au secrétariat de l'université par mail. La clef publique du professeur est (3, 55), celle du secrétariat (3, 33).

**Question 7.1** Déterminer la clé privée du professeur et du secrétariat de l'université.

**Question 7.2** Pour assurer la confidentialité de ses messages, le professeur chiffre les notes avec la clef RSA du secrétariat. Quel message chiffré correspond à la note 12 ?

**Question 7.3** Pour assurer l'authenticité de ses messages, le professeur signe chaque note avec sa clé privée et chiffre le résultat avec la clef RSA du secrétariat. Le secrétariat recoit ainsi le message 23. Quelle est la note correspondante ?

## Exercice 8 : Chiffrement de Vigenère et RSA

On considère un chiffrement opérant sur l'alphabet $\{A, B, C, \ldots, Z, \_\}$ (où _ désigne l'espace) dont chaque symbole est désigné par un nombre compris entre 0 et 26 :

| A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| O | P | Q | R | S | T | U | V | W | X | Y | Z | _ | |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |

**Question 8.1** Rappeler la méthode de chiffrement de Vigenère d'un texte $M$ avec la clef $K$ qui donne le texte chiffré $C$. On pourra considérer $M$, $C$ et $K$ comme des tableaux de taille respective $m$, $c$ et $k$ commençant à l'indice 0.

**Question 8.2** Oscar et Eve interceptent un message d'Alice à Bob chiffré à l'aide de Vigenère :

```
HWQIOQVPIF
TDIHTY_WAF
NGY_FCOMVI
CGEVZCVIAF
JDFZKYLYHG
YGEHRSHMMX
CVHBFAJYKN
ZIXHPZHEQY
YJRHTYWMUK
YKPBYYGEHA
G_DY_YWDTF
MHFZKYZYHX
CISVICHIVZ
```

Ils travaillent sur la cryptanalyse de ce texte. Oscar envoie à Eve la taille de la clef qu'il a réussi à déterminer. Pour cela, il utilise la clef publique RSA de Eve $(n, e) = (35, 5)$. Ivan intercepte le message chiffré RSA de la longueur de la clef : il obtient 10. Quelle est la longueur de la clef ?

**Question 8.3** Eve a réussi à déchiffrer $K[1]$ et $K[2]$. Elle les envoie à Oscar en utilisant la clef publique RSA d'Oscar $(n, e) = (65, 7)$. Ivan intercepte ainsi le chiffré de $K[1] : 48$ ; et le chiffré de $K[2] : 4$. Quelles sont les valeurs de $K[1]$ et $K[2]$ ? On pourra utiliser les résultats suivants : $48^2 = 29 \mod 65$ et $48^5 = 3 \mod 65$.

**Question 8.4** Trouver les éléments manquants de la clef en précisant la méthode utilisée. Préciser la clef utilisée pour le chiffrement et le déchiffrement (sous forme de chaîne de caractères). Décrypter le message. On donne la répartition des symboles utilisés dans un texte français :

| | |
|---|---|
| _ | 18,35% |
| E | 14,86% |
| S | 6,97% |
| A | 6,40% |
| N | 6,23% |
| ... | ... |

## Exercice 9 : Factorisation de $n$ à partir de $n$, $e$ et $d$ pour $e$ petit

On suppose disposer dans un système RSA de $(n, e, d)$ avec $e$ «petit».

**Question 9.1** Montrer qu'il existe $k \in \mathbb{Z}$ tel que $ed-1 = k \mod n-(p+q)+1$

**Question 9.2** On suppose $p$ et $q$ différent de 2 et 3. Montrer $k \leq 2e$

**Question 9.3** Proposer alors un algorithme permettant de factoriser $n$

## Exercice 10 : Attaque RSA par exposant commun

William, Jack et Averell ont respectivement les clefs RSA publiques $(3, n_W)$, $(3, n_J)$ et $(3, n_A)$. Joe envoie en secret à chacun d'eux le même message $x$ avec $0 \leq x < \min(n_W, n_J, n_A)$.

**Question 10.1** Montrer que Lucky Luke, qui voit passer sur le réseau $c_W = x^3 \mod n_W$, $c_J = x^3 \mod n_J$ et $c_A = x^3 \mod n_A$ peut facilement calculer $x$. On rappelle (ou on admettra !) que pour $a$ et $k$ entier, la méthode de Newton permet de calculer très rapidement $\lfloor a^{1/k} \rfloor$, en temps $O(\log^2 a)$.

## Exercice 11 : Attaque RSA par module commun

Une implémentation de RSA donne à deux personnes (Alice et Bob) le même nombre $n$ (produit de deux nombres premiers) mais des couples de clefs $(e_A, d_A)$ et $(e_B, d_B)$ différents. On suppose de plus que $e_A$ et $e_B$ sont premiers entre eux (ce qui est le plus général). Supposons alors que Alice et Bob chiffrent un même message $m$ et que Oscar intercepte les deux messages $c_A = m^{e_A} \mod n$ et $c_B = m^{e_B} \mod n$ qu'il sait être deux chiffrements du même message $m$.

**Question 11.1** Montrer qu'Oscar peut alors très facilement découvrir le message $m$.

## Exercice 12 : Attaque sur RSA

Soit $n = pq$ un modulo RSA de 2048 bits, avec $p$ et $q$ premiers grands, mais secrets. Connaissant $n$ mais pas $p$ et $q$, cet exercice construit une attaque pour calculer $p$ et $q$ dans le cas où $|p - q|$ n'est pas trop grand. On dispose des opérations ($+$, $-$, $\times$, pgcd et inverse) modulo $n$ et des opérations :

- `Racine(m)` : retourne l'entier $\lfloor \sqrt{m} \rfloor$
- `EstCarre(m)` : retourne VRAI s'il existe un entier $t$ tel que $t^2 = m$ et FAUX sinon.

Toutes ces opérations sont très rapides pour tout entier $m$ qui a moins de 2048 bits.

**Question 12.1** Montrer que $n = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$

**Question 12.2** Soit $1 < a < \sqrt{n}$ un entier et soit $b = n + a^2$. On suppose qu'il existe un entier $c$ tel que $c^2 = b$. En déduire alors la valeur de $p$ et $q$ à partir de $a$ et $c$.

On considère l'algorithme suivant :

```
a := 0
Répéter
  a := a + 1
  b := n + a*a
Jusqu'à EstCarré(b)
c := Racine(b)
Retourner a et c
```

**Question 12.3** Justifier que, théoriquement, cet algorithme s'arrête et que les valeurs de $a$ et $c$ retournées permettent de calculer rapidement les facteurs $p$ et $q$ de $n$.

**Question 12.4** Combien de fois le corps de la boucle `Répéter` est-il exécuté ?

**Question 12.5** En utilisant 1000 unités arithmétiques, le corps de la boucle prend un temps moyen $\tau \approx 10^{-8}$ secondes $< 10^{-13}$ jours. Combien de jours faut-il avec cet algorithme pour factoriser $n$ si $|p - q| < 10^{14}$ ?

**Question 12.6** Moralité : comment proposez-vous de choisir $p$ et $q$ pour résister à cette attaque ?

## Exercice 13 : Attaque RSA par texte chiffré bien choisi

Eve intercepte le message $c$ chiffré envoyé par Bob à Alice : $c = m^{e_A} \mod n_A$. Pour déchiffrer $c$, Eve procède ainsi :

1. Eve choisit un entier $0 < r < n_A$ au hasard et calcule $x = r^{e_A} \mod n_A$ ;
2. Eve calcule $y = x.c \mod n_A$
3. Eve demande à Alice de signer $y$ avec sa clef privée.

**Question 13.1** Justifier que Eve peut bien procéder comme indiqué, c'est-à-dire qu'elle a bien toutes les données et qu'elle peut faire tous les calculs en temps raisonnable.

**Question 13.2** Quelle est la valeur de $u$, le message renvoyé par Alice à Eve ?

**Question 13.3** Montrer en justifiant de manière précise qu'Eve peut alors facilement découvrir le message $m$ émis par Bob. On pourra calculer $u.r^{-1} \mod n_A$.

**Question 13.4** Est-ce qu'Alice a un moyen de savoir qu'elle signe un message frauduleux ? Autrement dit, peut-elle retrouver $m$ à partir de $y$ ?

**Question 13.5** Quelle est la moralité de cette histoire ?

## Exercice 14 : Factorisation de $n$ par la connaissance de $\phi(n)$

**Question 14.1** Soient $a$ et $b$ deux entiers, et $S = a + b$ et $P = a \times b$, la somme et le produit de $a$ et $b$. Quels sont les racines du polynôme $X^2 - S.X + P$ ?

**Question 14.2** Comment trouver $p$ et $q$ à partir de la connaissance de $\phi(n)$ et de $n$ ?

**Question 14.3** Quelle mesure prendre pour empêcher un attaquant d'obtenir $\phi(n)$ ?

## Exercice 15 : RSA et messages longs

Alice possède la clef publique $(5, 91)$.

**Question 15.1** Bob souhaite lui envoyer le message $M = 667$. Comment Bob va-t-il découper son message selon la méthode vue en cours pour pouvoir l'envoyer à Alice ?

**Question 15.2** Quels sont les messages chiffrés que Bob va envoyer à Alice ?

**Question 15.3** Alice reçoit les messages suivants de Charlie : $\{2, 3\}$. Quel est le message d'origine de Charlie ? On pourra utiliser les calculs suivants :

$$2^{12} \equiv 1 \mod 91,\ 3^6 \equiv 1 \mod 91$$

## Exercice 16 : Chiffrement symétrique

On considère un algorithme $E$ de chiffrement symétrique utilisant des clefs $K$ de $n$ bits.

**Question 16.1** Qu'appelle-t-on complexité d'une attaque d'un algorithme de chiffrement ?

**Question 16.2** On suppose que l'algorithme $E$ est sûr. Quelle est la complexité de la meilleure attaque ?

**Question 16.3** On veut renforcer la sécurité en créant à partir de $E$ l'algorithme Double$E$ tel que, pour tout message $M$ :

$$\text{Double}E_{(K_1,K_2)}(M) = E_{K_1}(E_{K_2}(M))$$

où $K_1$ et $K_2$ sont deux clefs de l'algorithme $E$. Quelle est la complexité de l'attaque par force brute sur cet algorithme ?

**Question 16.4** Décrire précisément l'attaque de la rencontre au milieu dans le cas de Double$E$. Quelle est la classe de cette attaque ? Quelle est la complexité de cette attaque ?

**Question 16.5** Quelle modification pouvez-vous apporter pour contrer l'attaque décrite précédemment ?

**Question 16.6** Quelle est alors la complexité de la meilleure attaque ?

**Question 16.7** On suppose désormais que $E$ possède la propriété suivante : «Soit $K_1$ et $K_2$ deux clefs de l'algorithme $E$, alors il existe une clef $K_3$ tel que, pour tout message $M$, $E_{K_1}(E_{K_2}(M)) = E_{K_3}(M)$». Que dire de la sécurité de Double$E$ ? Que dire de la sécurité de votre modification ?

## Exercice 17 : Propriété de complémentation de DES

On cherche à montrer que $E_K(M) = \overline{E_{\overline{K}}(\overline{M})}$ où $\overline{X}$ est le complémentaire de $X$, c'est-à-dire $X$ avec tous ses bits inversés.

**Question 17.1** Rappeler la méthode pour chiffrer avec un réseau de Feistel.

**Question 17.2** Montrer que $\overline{X} \oplus Y = \overline{X \oplus Y}$.

**Question 17.3** La figure 1 montre la fonction $f$ de DES. Justifier que la fonction $f$ de DES a la propriété : $f(\overline{H}, \overline{S}) = f(H, S)$.

**Question 17.4** On admet que pour une clef $K' = \overline{K}$, alors les sous-clefs ont la propriété suivante : $K'_i = \overline{K_i}$. On pose $L'_0 = \overline{L_0}$ et $R'_0 = \overline{R_0}$. Montrer qu'alors, pour tout $i > 0$ on a : $L'_i = \overline{L_i}$ et $R'_i = \overline{R_i}$.

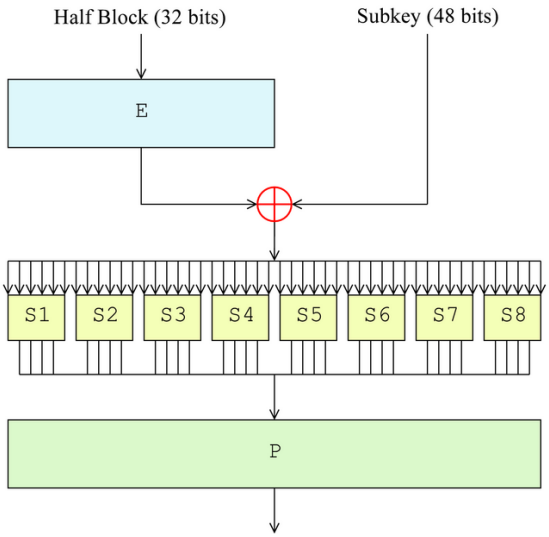


FIGURE 1 – La fonction $f$ de DES

**Question 17.5** En déduire la propriété de complémentation.

**Question 17.6** On construit une attaque pour trouver une clef $K$. Étant donné un message $M$, on demande $C_1 = E_K(M)$ et $C_2 = E_K(\overline{M})$. Comment peut-on faire alors pour ne parcourir que la moitié des clefs pour trouver la clef $K$ ?

**Question 17.7** Quelle est la classe de cette attaque ? Quelle est sa complexité ?

## Exercice 18 : Propriétés des fonctions de hachage cryptographiques

On rappelle les trois propriétés attendues des fonctions de hachage cryptographiques :

1. **Résistance à la préimage** : étant donné $y$, on ne peut pas trouver en temps raisonnable un $x$ tel que $H(x) = y$
2. **Résistance à la seconde préimage** : étant donné $x$, on ne peut pas trouver en temps raisonnable $x' \neq x$ tel que $H(x) = H(x')$
3. **Résistance aux collisions** : on ne peut pas trouver en temps raisonnable $x$ et $x'$ tels que $H(x) = H(x')$

**Question 18.1** Montrer, par contraposée, que la résistance aux collisions implique la résistance à la seconde préimage qui implique la résistance à la préimage.

## Exercice 19 : La fonction identité comme fonction de hachage

On considère la fonction identité, c'est à dire la fonction qui à $x$ associe lui-même.

**Question 19.1**

1. Cette fonction est-elle résistante aux collisions ?
2. Cette fonction est-elle résistante à la seconde préimage ?
3. Cette fonction est-elle résistante à la préimage ?

**Question 19.2** En dehors des trois propriétés, pourquoi la fonction identité n'est-elle pas une fonction de hachage ?

## Exercice 20 : Une mauvaise fonction de hachage

Soit $f : \{0,1\}^m \to \{0,1\}^m$, une fonction quelconque. On propose comme fonction de hachage à itérer $g : \{0,1\}^{2m} \to \{0,1\}^m$, la fonction telle que pour un $x$ de $2m$ bits, découpé en deux blocs de $m$ bits $x_L$ et $x_R$, on ait :

$$g(x) = g(x_L||x_R) = f(x_L \oplus x_R)$$

où $||$ désigne la concaténation.

**Question 20.1** Montrer que $g$ n'est pas résistante à la seconde préimage.

## Exercice 21 : Une autre mauvaise fonction de hachage

Soit $g$ une fonction de hachage résistante aux collisions qui produit une empreinte de $n$ bits. On considère la fonction de hachage suivante :

$$h(x) = \begin{cases} 1||x & \text{si } x \text{ possède } n \text{ bits} \\ 0||g(x) & \text{sinon} \end{cases}$$

où $||$ désigne la concaténation.

**Question 21.1** Montrer que $h$ est résistante aux collisions.

**Question 21.2** Montrer que $h$ n'est pas résistante à la préimage.

## Exercice 22 : Le paradoxe des anniversaires

On considère la fonction de hachage $H : \{0,1\}^t \to \{0,1\}^m$ avec $t > m$ et on pose $n = 2^m = |\{0,1\}^m|$. On dispose de $k$ messages $\{x_i\}_{1 \leq i \leq k}$ aléatoires ($k \ll n$). On supposera les hachages $z_i = H(x_i)$ comme aléatoires. On souhaite déterminer la probabilité pour obtenir au moins une collisions à partir des messages $x_i$.

**Question 22.1** On note $p_k$ la probabilité que les $\{z_i\}_{1 \leq i \leq k}$ soient tous différents. Montrer que :

$$p_k = \prod_{i=1}^{k-1} \left(1 - \frac{i}{n}\right)$$

**Question 22.2** Montrer que $\forall x \in \mathbb{R}, e^{-x} \geq 1 - x$

**Question 22.3** La probabilité d'obtenir au moins une collision est $1 - p_k$. Soit $0 < \alpha < 1$. Montrer que si on veut $1 - p_k \geq \alpha$, alors il suffit que :

$$k \geq \sqrt{2n \ln\left(\frac{1}{1-\alpha}\right)} = \mathcal{O}(\sqrt{n})$$

**Question 22.4** Application numérique : On considère une assemblée de $k$ personnes. Quelle est la probabilité qu'au moins deux d'entre-elles aient leur anniversaire le même jour (en ne tenant pas compte de l'année) ? Pour quelle valeur minimale de $k$ cette probabilité est elle supérieure à $\frac{1}{2}$ ? à 99% ?

## Exercice 23 : Un autre test de primalité

On se donne une fonction $J$ qui à tout couple d'entiers associe un entier. Cette fonction peut se calculer avec l'algorithme suivant, lorsque le second argument est impair :

**Algorithme 1** Calcul de $J(a, b)$ pour $b$ impair

```
if a = 1 ou b = 1 then
  return 1
end if
if a%b = 0 ou b%a = 0 then
  return 0
end if
if a est pair then
  return (-1)^((b^2-1)/8) × J(a/2, b)
else
  if a > b then
    return J(a%b, b)
  else
    return (-1)^((a-1)(b-1)/4) × J(b, a)
  end if
end if
```

**Question 23.1** Calculer $J(6, 7)$ et $J(12, 11)$.

**Question 23.2** Justifier que l'algorithme donné ci-dessus termine toujours.

**Question 23.3** Justifier que le résultat de l'algorithme ci-dessus est toujours 1 ou $-1$ si $a$ et $b$ sont premiers entre eux (on pourra utiliser le fait que $\text{pgcd}(a, b) = \text{pgcd}(a\%b, b)$).

**Question 23.4** On admet que si $b$ est premier alors $J(a, b) = a^{\frac{b-1}{2}} \mod p$. On admet aussi que si $a$ et $b$ sont premiers entre eux (i.e. leur pgcd vaut 1), que $b$ est impair, et que $a$ a été choisi aléatoirement et equiprobablement inférieur $b$ alors :

- Si $J(a, b) = a^{\frac{b-1}{2}} \mod b$, la probabilité que $b$ soit premier est supérieur à $\frac{1}{2}$.
- Si $J(a, b) \neq a^{\frac{b-1}{2}} \mod b$, alors on est sûr que $b$ n'est pas premier.

En déduire un algorithme probabiliste de test de primalité (on pourra utiliser une fonction calculant le PGCD de deux entiers sans la définir).

## Exercice 24 : Génération aléatoire et sécurité

**Question 24.1** Donner des exemples d'utilisation de génération aléatoire en sécurité. Si le générateur est mauvais (à définir), quels risque cela peut-il poser ?

**Question 24.2** Une des façons de simuler la génération aléatoire de nombres est la méthode dite de *congruence linéaire*. C'est notamment celle utilisée par la fonction `rand(3)` en C. On utilise pour cela une suite :

$$x_{n+1} = (ax_n + c) \mod m$$

avec $a$ (multiplicateur), $c$ (incrément), $x_0$ (graine), et $m$ qui sont des entiers positifs non nuls (sauf $x_0$ éventuellement qui peut être nul). Chaque $x_i$ est une nouvelle valeur pseudo-aléatoire. La valeur $x_0$ sert à initialiser le générateur. En C, cette valeur est donnée grâce à la fonction `srand(3)`.

Essayer avec les valeurs suivantes de calculer les premiers termes. Commenter les résultats obtenus.

| $a$ | $c$ | $x_0$ | $m$ |
|---|---|---|---|
| 12 | 32 | 16 | 64 |
| 11 | 25 | 16 | 64 |

**Question 24.3** Justifier qu'un tel générateur est périodique. Quelle est sa plus longue période ? D. Knuth a donné des critères, nécessaires et suffisants, pour que la suite de valeurs obtenue ait la période maximale :

— $c$ et $m$ doivent être premiers entre eux,
— $a - 1$ doit être un multiple de $p$, pour tout $p$ nombre premier diviseur de $m$,
— $a - 1$ doit être divisible par 4 si $m$ est divisible par 4.

Vérifier que $a = 31415821$, $c = 1$ et $m = 10^8$ et $x_0 = 0$ satisfont ces critères.

**Question 24.4** On admet qu'avec les valeurs précédentes, la suite obtenue est : 1, 31415822, 40519863, 62952524, 25482205, 90965306, 70506227, 6817368, 12779129, 29199910, 45776111, 9252132, 22780373, 20481234, 81203115, ...

En utilisant cette suite, commenter la page de manuel de la fonction `rand(3)` qui dit :

```
Si vous désirez engendrer un entier aléatoire entre 1 et 10, vous
devez toujours procéder en utilisant les bits de poids forts,
comme dans :

  j=1+(int) (10.0*rand()/(RAND\_MAX+1.0));

et jamais ainsi :

  j=1+(rand() \% 10);

(car cette dernière version utilise les bits de poids faibles).
```

## Exercice 25 : Échelle de Montgomery

L'Échelle de Montgomery est une technique utilisée pour faire de l'exponentiation rapide, c'est-à-dire pour calculer $x^n$ avec $x$ et $n$ deux entiers. Étant donné la représentation binaire de $n = n_{k-1} \dots n_0$ avec $n_{k-1} = 1$, on définit l'algorithme MONTGOMERY suivant :

**Algorithme 2** Calcul de $x^n$

```
1: a ← x
2: b ← x^2
3: for i = k − 2 to 0 do
4:    if n_i = 0 then
5:       b ← a ∗ b
6:       a ← a^2
7:    else
8:       a ← a ∗ b
9:       b ← b^2
10:   end if
11: end for
12: return a
```

**Question 25.1** Rappeler l'algorithme récursif d'exponentiation rapide vu en cours. Rappeler, sans le justifier, son coût en nombre de multiplications.

**Question 25.2** Quel est le coût de l'algorithme MONTGOMERY en nombre de multiplications ? Justifier.

**Question 25.3** Dérouler l'algorithme pour le calcul de $3^{13}$. Indiquer la valeur de $a$ et $b$ au début de chaque boucle et le cas utilisé dans la boucle. Indication : $13 = 1101_2$.

**Question 25.4** On remarque que cet algorithme effectue un nombre constant de multiplication pour un $k$ donné. Est-ce le cas pour l'algorithme d'exponentiation rapide vu en cours ? Quel avantage peut-il y avoir selon vous à utiliser l'algorithme MONTGOMERY ?

## Exercice 26 : Signatures

Lorsqu'Alice souhaite envoyer un message $M$ à Bob en utilisant le système RSA, on propose qu'Alice envoie en fait le message :

$$A \to B : \{\{M\}_{K_A^{-1}}\}_{K_B}$$

**Question 26.1** En terme de temps de chiffrement, qu'est-ce que cela implique ?

**Question 26.2** Comment fait Bob pour trouver le message d'Alice ? Qu'est-ce que cela implique en terme de temps de déchiffrement ?

**Question 26.3** Pourquoi Bob peut-il être convaincu que le message a bien été écrit par Alice ? En supposant que le message $M$ ne contient pas de données cohérentes (par exemple juste un nombre aléatoire), quel problème peut-il se poser ?

**Question 26.4** On propose de modifier l'envoie du message par :

$$A \to B : \{A, \{M\}_{K_A^{-1}}\}_{K_B}$$

À votre avis pourquoi ?

**Question 26.5** Montrer que le protocole tel quel ne permet pas d'authentifier à qui est destiné le message.

**Question 26.6** Une solution possible est alors d'envoyer le message :

$$A \to B : \{A, \{B, M\}_{K_A^{-1}}\}_{K_B}$$

En quoi cela est-il une solution ?

**Question 26.7** Si le message envoyé par Alice est anodin et que Bob ne connaît pas la clé publique d'Alice (il doit aller la chercher sur un serveur par exemple), quel problème y a-t-il ?

**Question 26.8** Une solution possible est alors d'envoyer le message :

$$A \to B : \{A, M, \{B, M\}_{K_A^{-1}}\}_{K_B}$$

En quoi cela est-il une solution ?

**Question 26.9** Ici, la signature est aussi longue que le message initial. Voyez-vous un moyen d'en réduire la taille en utilisant une fonction à sens unique ?

## Exercice 27 : Protocole WEP

Le protocole WEP (*Wired Equivalent Privacy*) est un protocole assurant la protection des données dans des transmissions sans fil. Ce protocole utilise deux fonctions, la fonction $C$ qui est une fonction de détection d'erreurs et une fonction $RC4$ qui est un générateur pseudo-aléatoire (comme `rand()` en `C`), mais à deux arguments.

Lorsqu'Alice veut envoyer un message à Bob, elle envoie le message suivant :

$$N_B, ([M, C(M)] \oplus RC4(N_B, K_{AB}))$$

où $K_{AB}$ est la clé partagé entre Alice et Bob.

**Question 27.1** Comment fait Bob pour déchiffrer le message envoyé par Alice ?

**Question 27.2** On suppose que la fonction $C$ vérifie :

$$C(x \oplus y) = C(x) \oplus C(y)$$

Montrer que ce protocole n'est alors pas sûr vis-à-vis de l'intégrité.

## Exercice 28 : Protocole à quatre

On considère un protocole contenant quatre participants, Alice, Bob, Carol et Dave. Chaque membre possède une clef publique et connaît un entier : Alice connaît $S_A$, Bob $S_B$, Carol $S_C$ et Dave $S_D$. Les étapes du protocole sont les suivantes :

1. $A \to B : A, \{N_A + S_A\}_{K_B}$
2. $B \to C : A, B, \{N_A + S_A + S_B\}_{K_C}$
3. $C \to D : A, B, C, \{N_A + S_A + S_B + S_C\}_{K_D}$
4. $D \to A : A, B, C, D, \{N_A + S_A + S_B + S_C + S_D\}_{K_A}$
5. $A \to B, C, D : S_A + S_B + S_C + S_D$

**Question 28.1** On suppose que $S_A = 5$, $S_B = 12$, $S_C = 1$ et $S_D = 2$ et que, par ailleurs, $N_A = 10$. Dérouler le protocole. Que connaissent respectivement $A$, $B$, $C$ et $D$ à la fin du protocole ?

**Question 28.2** À votre avis, à quoi peut servir ce protocole, éventuellement dans une version généralisée ? Montrer que si $B$ est malhonnête, il peut connaître $S_A$.

## Exercice 29 : Partage de secret

On considère le problème suivant : Charlie connaît un secret industriel, par exemple la recette d'un soda révolutionnaire. Il souhaite partager ce secret avec Alice et Bob, au cas où un malheur lui arrive. Cependant il ne leur fait pas totalement confiance et ne souhaite pas que l'un d'eux lance une boisson concurrente avec sa recette. Il veut donc partager le secret entre Alice et Bob, sans qu'aucun n'ait d'information sur la recette sans l'autre. On suppose que la recette de Charlie est un message $R$ codé sur $n$-bits.

**Question 29.1** On suppose que Charlie coupe la chaîne $R$ en deux morceaux de même longueur, $R_1$ et $R_2$ sans les chiffer ($R = \{R_1, R_2\}$). Quel danger cela représente-t-il pour le secret ?

**Question 29.2** Charlie propose d'utiliser le protocole 1 suivant :

1. Il génère un nombre aléatoire $S$ de la même taille que $R$ (en nombre de bits).
2. Il donne $S$ à Alice et $R \oplus S$ à Bob.

Justifier qu'avec le protocole 1 ni Alice ni Bob n'ont d'information sur la recette.

**Question 29.3** Comment peuvent faire Alice et Bob pour retrouver ensemble la recette avec le protocole 1 ?

**Question 29.4** On suppose maintenant que Charlie souhaite partager le secret entre trois personnes : Alice, Bob, et Eve de telle sorte qu'ils n'aient seuls ou à deux aucune information sur la recette, mais qu'à eux trois ils puissent reconstituer le secret. Pour cela il utilise le protocole 2 suivant :

1. Il génère deux nombres aléatoires $S$ et $T$ de la même taille que $R$ (en nombre de bits).
2. Il donne $S$ à Alice, $T$ à Bob et $R \oplus S \oplus T$ à Eve.

Justifier que ce protocole satisfait bien les exigences souhaitées.

**Question 29.5** On suppose maintenant que Charlie souhaite partager le secret entre quatre personnes : Alice, Bob, Eve et Fionna de telle sorte qu'ils n'aient seuls à deux ou à trois aucune information sur la recette, mais qu'à eux quatre ils puissent reconstituer le secret. Proposer un protocole 3 adaptés des protocoles 1 et 2 (on ne demande pas de justification).

**Question 29.6** On est toujours dans le cadre où Charlie souhaite partager le secret entre quatre personnes : Alice, Bob, Eve et Fionna de telle sorte qu'ils n'aient seuls ou a deux aucune information sur la recette. Cependant, sachant qu'il sera difficile pour ces quatre personnes de ce réunir ensembles, il souhaite que si trois d'entre elles (n'importe lesquelles) au moins se réunissent, elle puissent reconstituer la recette, mais pas à deux. Par exemple, Alice, Eve et Fionna doivent pouvoir reconstituer la recette à elle trois, mais aussi Alice, Bob et Eve (n'importe quelle combinaison de trois personnes). En revanche, Alice et Eve seuls ne doivent par exemple tirer aucune information à elles seules. Pourquoi le protocole 3 ne marche plus dans ce cadre ?

**Question 29.7** Pour résoudre le problème ci-dessus, Charlie utilise le protocole 4 :

1. Il génère deux nombres aléatoires $a$ et $b$ non nuls
2. Il construit le polynôme $P(x) = ax^2 + bx + R$.
3. Il tire aléatoirement quatre entiers distincts non nuls $\alpha_1$, $\alpha_2$, $\alpha_3$ et $\alpha_4$.

4. Il envoie à Alice $\alpha_1$ et $P(\alpha_1)$, à Bob $\alpha_2$ et $P(\alpha_2)$, à Eve $\alpha_3$ et $P(\alpha_3)$, et à Fionna $\alpha_4$ et $P(\alpha_4)$.

En supposant que $R = 10$, $a = 1$, $b = 3$ et $\alpha_2 = 2$, quels messages sont envoyés à Bob ?

**Question 29.8** Dans le protocole 4, comment font Alice, Eve et Fionna pour retrouver $R$ ?

**Question 29.9** Dans le protocole 4, pourquoi deux individus ne peuvent-ils pas découvrir la recette ?

**Question 29.10** Après avoir exécuté le protocole 4, Charlie souhaite faire aussi partager le secret avec Gustave, tout en gardant la règle des trois participants pour découvrir le secret. Comment peut-il le faire facilement, sans rien envoyer de nouveau aux anciens participants ?

## Exercice 30 : Pile ou face équitable

Alice et Bob utilisent un protocole pour jouer à pile ou face via un canal numérique.

On considère le protocole naïf suivant : Alice tire au hasard une valeur $x_0 \in \{0, 1\}$, puis Bob envoie à Alice son pari $b \in \{0, 1\}$. Alice vérifie si $b = x_0$. Si oui, elle informe Bob qu'il a gagné et inversement.

**Question 30.1** Expliquer comment Alice peut facilement tricher.

Alice possède une clef RSA. On considère le protocole suivant :

1. Alice tire au hasard $x$ et le chiffre avec sa clef publique pour obtenir $c$. Elle envoie $c$ à Bob.
2. Bob reçoit $c$ et choisit $b \in \{0, 1\}$ et l'envoit à Alice.
3. Alice reçoit $b$ et envoie $x$ à Bob.
4. Si $x \mod 2 = b$ alors Bob a gagné, sinon il a perdu.

**Question 30.2** Quels sont les calculs effectués par Alice à l'étape 1.

**Question 30.3** Justifier qu'Alice ne peut pas tricher. On précisera comment à l'étape 4 Bob vérifie que les valeurs de $x$ et $c$ transmises par Alice sont bien cohérentes.

**Question 30.4** À quelle condition peut-on dire que Bob ne peut pas tricher ?

**Question 30.5** Si Bob a perdu, il peut tricher en prétendant ne pas avoir voulu jouer : il argumente que quelqu'un a joué à sa place à l'étape 2. Modifier le protocole pour empêcher Bob de tricher.

**Question 30.6** Est-ce que votre solution permet le rejeu ? Si oui, expliquer comment et proposer une solution. Si non, expliquer pourquoi.

**Question 30.7** Que penser du titre de cet exercice ?

## Exercice 31 : Distribution de clés privées avec un serveur central

On propose un protocole de distribution de clés secrètes centralisé. Chaque individu partage une clé secrète avec le serveur (Alice partage la clé $K_{AS}$ et Bob la clé $K_{BS}$). Supposons qu'Alice veuille envoyer un message $M$ à Bob, le protocole est le suivant :

1. $A \to S : A, \{B, K\}_{K_{AS}}$
2. $S \to B : \{A, K\}_{K_{BS}}$
3. $A \to B : \{M\}_K$

Alice commence par envoyer au serveur un message disant qu'elle souhaite communiquer avec Bob en utilisant la clé secrète $K$ qu'elle vient de générer. Le serveur envoie à Bob de façon secrète cette clé en lui disant de l'utiliser pour déchiffrer les messages d'Alice. Alice peut alors envoyer le message à Bob, chiffré avec $K$.

**Question 31.1** À votre avis, pourquoi Alice n'envoie pas directement au Serveur le message $\{A, B, M\}_{K_{AS}}$ pour que celui-ci renvoie à Bob le message $\{A, B, M\}_{K_{BS}}$ ?

**Question 31.2** Supposons que Charlie soit malhonnête et soit capable d'écouter le réseau et d'y envoyer des messages (mais pas, pour le moment, de les intercepter). On suppose aussi qu'Alice et Charlie sont tous les deux vendeurs de glaces et que Bob est leur fournisseur de glace à la rhubarbe. Le protocole décrit sert à faire les commandes sécurisées entre Alice et Bob. Comment Charlie peut-il obliger Alice à commander beaucoup plus de glace à la rhubarbe qu'elle ne le souhaite (en supposant qu'Alice ait déjà fait une commande) ?

**Question 31.3** On remplace la dernière étape du protocole par :

3. $A \to B : \{M, N\}_K$

où $N$ est une nonce. En quoi cela évite-t-il la faille ci-dessus ?

**Question 31.4** On suppose que Charlie est capable d'intercepter des messages et que la première étape du protocole est :

1. $A \to S : A, B, \{K\}_{K_{AS}}$

Montrer une faille.

**Question 31.5** Si $n$ personnes cherchent à communiquer avec Bob à la fois, quelle difficulté a-t-il pour déchiffrer les messages ? Comment ce problème peut-il être frauduleusement exploité pour gêner Bob ?

**Question 31.6** Remplacer la dernière étape du protocole par :

3. $A \rightarrow B : A, \{M, N\}_K$

règle-t-il le problème ? On met de côté ce problème (question 5), qui peut apparaître sur de nombreux protocoles.

**Question 31.7** Plusieurs algorithmes de chiffrement par clés secrètes utilisent des blocs de chiffrement et satisfont la propriété :

$$\{M_1, M_2\}_K = \{M_1\}_K \{M_2\}_K$$

En utilisant cette propriété, trouver une faille au protocole.

**Question 31.8** On suppose que le serveur, un peu occupé, met du temps à envoyer le message à Bob et que ce dernier reçoive le message d'Alice avant d'avoir reçu la clé $K$. Que se passe-t-il ? Quel problème cela pose-t-il ?

**Question 31.9** Supposons qu'un pirate casse la sécurité du serveur, que se passe-t-il ? Faire une comparaison avec ce qui se passerait dans le cas de NSPK-Lowe avec Serveur.

**Question 31.10** Que pensez-vous de ce protocole vis-à-vis des attaques par cryptanalyse statistique ?

## Exercice 32 : Alerte de sécurité

Une alerte de sécurité est un document émis par un organisme d'État ou une entreprise qui a pour but de prévenir de la présence d'une faille de sécurité dans un logiciel ou un protocole informatique.

Parmi les organismes d'État qui publient des alertes de sécurité, on trouve :

— Le Centre d'Expertise Gouvernemental de Réponse et de Traitement des Attaques informatiques (CERTA) en France

— United States Computer Emergency Readiness Team (US-CERT) aux États-Unis

Les alertes de sécurité reçoivent un nom dans le dictionnaire Common Vulnerabilities and Exposures (CVE) qui leur permet d'être identifié de manière unique chez tous les fournisseurs d'alertes.

**Question 32.1** Pour chacune des alertes de sécurité que vous allez étudier, identifiez et faites la liste des informations techniques et non-techniques concernant l'alerte.

**Question 32.2** Précisez les mesures à prendre en tant qu'utilisateur puis en tant que responsable d'un système d'information.

## Exercice 33 : Politique de sécurité

La politique de sécurité des systèmes d'information (PSSI) est un plan d'actions définies pour maintenir un certain niveau de sécurité. C'est un ensemble de règle à suivre par tous les intervenants dans une organisation (entreprise, administration). Pour construire une politique de sécurité, on procède en plusieurs étapes :

1. Une analyse des menaces, avec pour chaque cas la probabilité et la criticité de la menace.
2. L'établissement de règles et d'actions pour chacun des intervenants dans l'organisation, que ce soit en prévention (à travers de la formation par exemple) ou en cas de gestion des incidents (à travers un plan de continuité d'activité par exemple).

Le but du TD est de définir une politique de sécurité pour une petite entreprise (fictive). L'entreprise est une usine de câbles RJ45 en plein essor qui emploie 300 personnes. Sur le site, le personnel est réparti aux machines et à l'administration. Les 250 ouvriers ont accès à des ordinateurs dans l'usine pour lire leur courrier électronique professionnel, tandis que l'administration (50 personnes) gère toute l'usine (commande, client, vente, etc). L'usine possède un site de vente en ligne géré par l'administration. Le site de l'usine est situé entre une route nationale et une rivière et est entourée d'un simple grillage de 2 mètres de haut.

**Question 33.1** Faire l'analyse des menaces.

**Question 33.2** Écrire la politique de sécurité.

## Exercice 34 : Virus shell

**Remarque importante : Diffuser des codes malveillants, comme des virus, est strictement interdit.**

Un virus est un programme capable de se recopier lui même.

**Question 34.1** Que pensez-vous du code suivant :

```
for i in *.sh
   if test "./$i"!="$0" ; then
      tail -n 5  $0 | cat >> $i;
   fi
done
```

**Question 34.2** Le modifier pour qu'il affiche un message «Bonjour, je suis le gentil virus» lors de son exécution.

**Question 34.3** Le modifier pour qu'il ne se recopie pas dans un fichier où il est déjà présent.

**Question 34.4** Comment lutter contre ce type de virus ?

## Exercice 35 : Système de Merkle-Hellman

On considère le problème suivant, appelé problème du *sac à dos*.
— Entrées : $k$ entiers positifs $n_1, \ldots, n_k$ et un entier positif $w$.
— Sortie : $(\alpha_1, \ldots, \alpha_k)$ tels que les $\alpha_i$ appartiennent à $\{0, 1\}$ et vérifient

$$\sum_{i=1}^{i=k} \alpha_i n_i = w$$

si c'est possible, $-1$ sinon.

Par exemple, si $n_1 = 2, n_2 = 3, n_3 = 1$ et $w = 3$, des solutions du problème sont $(1, 0, 1)$ car $n_1 + n_3 = 3$ ou $(0, 1, 0)$ car $n_2 = 3$. En revanche, pour les mêmes entrées $n_1, n_2, n_3$ mais pour $w = 7$, la sortie du problème est $-1$.

**Question 35.1** En supposant que $n_1 = 2$, $n_2 = 6$, $n_3 = 3$, $n_4 = 8$ et $w = 13$, donner une sortie du problème.

**Question 35.2** En supposant que $n_1 = 2$, $n_2 = 6$, $n_3 = 3$, $n_4 = 8$ et $w = 7$, justifier que la sortie du problème est $-1$.

**Question 35.3** Pour une instance générale, justifier que si $w > \sum_{i=1}^{i=k} n_i$, alors la sortie du problème est $-1$.

**Question 35.4** On suppose que les $n_i$ forment une suite finie *super-croissante*, c'est-à-dire que pour tout $k \geq j > 1$, $n_j > \sum_{i=1}^{j-1} n_i$. Expliquer dans ce cas comment le problème peut se résoudre en temps polynomial. On admettra que dans le cadre général le problème du sac à dos ne peut pas se résoudre en temps polynomial.

**Question 35.5** Dans le système de Merkle-Hellman, une clé privée est un tuple $(n_1, \ldots, n_k, q, r)$ tel que les $n_i$ forment une suite super-croissante d'entiers, $q$ est un nombre premier supérieur à la somme des $n_i$ et $r < q$. La clé publique associé est $(\beta_1, \ldots, \beta_k)$ où $\beta_i = n_i r \mod q$. Justifier que $(2, 3, 6, 12, 25, 53, 46)$ est bien une clef privée.

**Question 35.6** En gardant les valeurs de la question précédente, que valent $\beta_1$ et $\beta_2$ ? On admettra que $\beta_3 = 11$, $\beta_4 = 22$ et $\beta_5 = 37$.

**Question 35.7** Pour chiffrer un message de $k$ bits $m = m_1 \ldots m_k$ ($m_i \in \{0, 1\}$), un agent calcule $e = \sum_{i=1}^{i=k} m_i \beta_i$. L'entier $e$ est le message chiffré. En reprenant les valeures de la question précédente, comment est chiffré le message 01100 ? et le message 10011 ?

**Question 35.8** Lorsque qu'un agent reçoit un message $c$ chiffré qui lui est destiné, il calcule $c' = cr^{-1} \mod q$ où $r^{-1}$ est un inverse de $r$ modulo $q$. Justifier que dans le cas des valeurs des questions précédents, $r^{-1} = 15$ convient.

**Question 35.9** Comment fait alors l'agent pour retrouver le message intial efficacement ? En gardant les valeurs précédentes, en supposant que $c = 119$, et en admettant que $119.15 = 36 \mod 53$, déchiffrer le message $c$.

**Question 35.10** Justifier brièvement en quoi cette approche est un bon candidat pour un système à clé publique.

Remarque culturelle : le système de Merkle-Hellman a une faille ; les messages peuvent être déchiffrés en temps polynomial (ça n'a rien d'évident).