Examen du 25 juin 2021, 16h45, durée 2 heures.
Calculatrice autorisée, feuille A4 recto-verso manuscrite autorisée.

1. BASE 16. (4 POINTS)

L'application `openssl` permet d'afficher des entiers utilisés pour le cryptage RSA, en base 16 par groupe de 2 chiffres séparés par le signe `:`. Ainsi l'entier $n = 42348$ en base 10 sécrit `a5:6c`.

(1) Déterminer en base 10 l'entier n représenté par `04:23:48`

(2) Déterminer la repésentation de l'entier n dont l'écriture en base 10 est 123456789.

(3) Pour afficher un résultat facile à lire, la représentation des grands entiers se fait sur plusieurs lignes, chaque ligne contient au plus 15 groupes de 2 chiffres. Le début de l'affichage d'un entier n de 2048 bits $2^{2047} \leq n < 2^{2048}$ est le suivant :

```
f2:0e:d4:9d:44:04:c4:c8:6a:5b:c6:9a:d6:df:9c:
f5:56:f2:0d:ad:6c:34:b4:48:f7:a7:a8:27:a0:c8:
```
...

Combien de lignes faut-il pour reprśenter n ? La dernière ligne est-elle complète ? Sinon, combien de groupes de deux chiffres contient la dernière ligne ?

2. CODAGE (4 POINTS)

On considère un code linéaire de matrice génératrice à coefficients modulo 2 :

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \pmod 2$$

(1) Quel est le mot de code correspondant à $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$?

(2) On reçoit `1000000000`, déterminer le mot du code le plus proche du mot reçu.

(3) Quel est la distance de Hamming de ce code ? En déduire t, le nombre maximal d'erreurs qu'on peut espérer corriger.

(4) Donner une méthode simple pour corriger au plus t erreurs.

(5) Discuter l'efficacité de ce code par rapport à un code de répétition (on rappelle qu'il s'agit d'un code où on répète plusieurs fois de suite chaque bit).

3. CRYPTOGRAPHIE RSA (12 POINTS)

3.1. **Un exemple.** On rappelle que pour crypter un message a à un destinataire dont la clef publique est (c,n), il faut lui envoyer $b = a^c \pmod{n}$. Pour pouvoir faire les calculs, on suppose dans cette question que $c = 17, n = 55$.

(1) Crypter le message $a = 3$ en donnant le détail des calculs par la méthode de la puissance rapide.

(2) Déterminer $\phi(n)$, le nombre d'éléments inversibles dans $\mathbb{Z}/n\mathbb{Z}$ pour $n = 55$.

(3) Déterminer d l'inverse de 17 modulo $\phi(55)$ en donnant les étapes intermédiaires.

(4) Déterminer $b^d \pmod{n}$.

(5) Pourquoi ne doit-on pas prendre $n = 55$ si on souhaite que le message codé reste confidentiel ?

3.2. **Utiliser 3 comme clef publique.** Ici n n'est plus égal à 55, c'est le produit de deux nombres premiers quelconques. On se propose de prendre $c = 3$.

(1) À quelle condition sur $\phi(n)$ peut-on prendre $c = 3$?

(2) Comparer le nombre d'opérations nécessaires au cryptage d'un message lorsque $c = 3$ avec $c = 17$.

(3) Écrire un algorithme en langage naturel ou en C ou en Python permettant de connaitre le nombre d'opérations nécessaires au cryptage en fonction de c.

(4) Un espion envoie le même message a à trois destinataires différents, ayant chacun leur clef publique $c = 3, n_1 = 187$, $c = 3, n_2 = 46$ et $c = 3, n_3 = 253$ (on a pris des petites valeurs de n pour que les calculs soient faisables à la calculatrice). Les services de contre-espionnage arrivent à intercepter les trois messages codés :

$$b_1 = 98 \pmod{187}, \quad b_2 = 15 \pmod{46}, \quad b_3 = 126 \pmod{145}$$

Il s'agit de déterminer *a sans chercher à factoriser les entiers* n_i.

(a) Montrer qu'il existe un unique entier $b \in [0, n_1 n_2 n_3[$ tel que

$$b = b_1 \pmod{n_1}, \quad b = b_2 \pmod{n_2}, \quad b = b_3 \pmod{n_3}$$

(b) Expliquer comment calculer b, et donner le détail des calculs si vous avez le temps,

(c) On trouve $b = 9261$, en déduire a sachant que $a \in [0, n_1[$.