Métropole – 2025 – sujet2 - Correction

Exercice 3 (8 points)

PARTIE A – Méthode du masque jetable

1. Chiffrement du message LIBRE avec la clé EYQMT.

On convertit chaque lettre en son rang dans l'alphabet, on additionne message et clé puis on applique le modulo 26.

LIBRE → [11, 8, 1, 17, 4]
EYQMT → [4, 24, 16, 12, 19]

Somme modulo 26 → [15, 6, 17, 3, 23] soit le message chiffré : PGRDX.

2. Fonction indice

```
def indice(L, element):
    for i in range(len(L)):
        if L[i] == element:
            return i
```

Commentaire : on parcourt la liste jusqu'à trouver l'élément recherché.

3. Fonction lettres_vers_indices

```
def lettres_vers_indices(chaine):
    indices = []
    for lettre in chaine:
        indices.append(indice(alphabet, lettre))
    return indices
```

4. Complétion de la fonction chiffrement

```
def chiffrement(msg, cle):
    assert len(cle) >= len(msg), 'impossible'
    indices_msg = lettres_vers_indices(msg)
    indices_cle = lettres_vers_indices(cle)
    n = len(msg)
    indices_msg_chiffre = []

    for k in range(n):
        ind = indices_msg[k] + indices_cle[k]
        if ind >= 26:
```

```
            ind = ind - 26
        indices_msg_chiffre.append(ind)

    msg_chiffre = indices_vers_lettres(indices_msg_chiffre)
    return msg_chiffre
```

5. Appel chiffrement('RESEAU','GFTZ')

Une erreur se produit car la clé est plus courte que le message, ce qui déclenche l'assertion.

6. Déchiffrement de GMEDH avec la clé FVEIT

On soustrait les indices de la clé à ceux du message chiffré (modulo 26). Le message obtenu est SALUT.

7. Principe du déchiffrement

Le déchiffrement consiste à effectuer l'opération inverse du chiffrement : soustraction des indices et modulo 26.

8. Fonction dechiffrement

```
def dechiffrement(msg, cle):
    assert len(cle) >= len(msg), 'impossible'
    indices_msg = lettres_vers_indices(msg)
    indices_cle = lettres_vers_indices(cle)
    n = len(msg)
    indices_msg_dechiffre = []

    for k in range(n):
        ind = indices_msg[k] - indices_cle[k]
        if ind < 0:
            ind = ind + 26
        indices_msg_dechiffre.append(ind)

    msg_dechiffre = indices_vers_lettres(indices_msg_dechiffre)
    return msg_dechiffre
```

PARTIE B – Sécurisation des communications

9. Chiffrement symétrique / asymétrique

Le **chiffrement symétrique** utilise **une seule clé secrète**, partagée par l'émetteur et le récepteur, pour chiffrer et déchiffrer les messages.
Le **chiffrement asymétrique** repose sur **une paire de clés** :

- une **clé publique** pour chiffrer,
- une **clé privée** pour déchiffrer.

*Commentaire :L'élève doit impérativement mentionner le **nombre de clés** et leur **rôle**. Confondre clé publique et clé privée est une erreur classique.*

10. Déchiffrement par Bob

Bob déchiffre le message à l'aide de **sa clé privée**, correspondant à la clé publique utilisée par Alice pour chiffrer le message.

Commentaire :La clé privée ne doit jamais être transmise : elle garantit la confidentialité du message.

11. Risque si l'identité n'est pas vérifiée

Une personne malveillante peut se faire passer pour Alice et envoyer un message chiffré à Bob.

Commentaire : Il s'agit d'une ***usurpation d'identité*** (attaque de type *man-in-the-middle*).

12. Principe du protocole HTTPS

HTTPS repose sur le protocole **TLS** :

utilisation du chiffrement asymétrique pour authentifier le serveur et échanger une clé secrète ;

utilisation du chiffrement symétrique pour sécuriser les échanges de données.

Commentaire : Cette phase hybride permet de concilier ***sécurité*** *et* ***efficacité****.*

13. Pourquoi ne pas utiliser uniquement l'asymétrique ?

Le chiffrement asymétrique est **coûteux en calcul** et lent. Il n'est donc utilisé que pour l'échange de clés.

Commentaire ***:*** *Cette question fait le lien entre* ***performances*** *et* ***sécurité****, point clé du programme NSI.*

PARTIE C – Réseaux

14. Pourquoi la commande `ping` échoue ?

La commande `ping` échoue car l'adresse IP utilisée est incorrecte.
Marc a utilisé l'adresse **192.168.100.115** alors que l'adresse correcte de Bob est **192.168.110.115**.

Commentaire : Une erreur sur un seul octet de l'adresse IP suffit à empêcher toute communication.

15. Écriture décimale du masque de sous-réseau

Le masque binaire :

`11111111.11111111.11111111.11100000`
correspond au masque décimal :
255.255.255.224.

Commentaire : Chaque groupe de 8 bits est converti séparément en décimal.

16. Nombre d'adresses disponibles

Avec ce masque, il y a **5 bits pour les hôtes**, soit : $\mathbf{2^5}$=32 adresses

Parmi elles :

- 1 adresse réseau,
- 1 adresse de broadcast,

Il reste donc **30 adresses utilisables**.

17. Écriture binaire de 134

Le nombre décimal **134** s'écrit en binaire :

$$134 = 10000110_2$$

Commentaire : Il est important de vérifier la position des bits de poids fort.

18. Communication entre Zoé et Bob

Zoé et Bob peuvent communiquer car ils appartiennent au **même sous-réseau**.

La commande correcte est donc :

ping 192.168.110.115

Commentaire : Deux machines d'un même sous-réseau peuvent communiquer directement sans passer par un routeur.