Exercice 1 (6 points)

Cet exercice porte sur la sécurisation des communications, la représentation des données et la programmation en Python.

Alice et Bob cherchent à communiquer de manière sécurisée sur un réseau ouvert à tous. Eve veille et écoute tout ce qui passe sur le réseau. Mallory aimerait bien se faire passer pour quelqu'un d'autre. L'objectif de cet exercice est de s'intéresser à des protocoles de *chiffrements* et à un protocole de *signature* permettant d'authentifier l'auteur d'un message.

Alice, qui ne connaît Bob que par les réseaux sociaux, aimerait lui faire parvenir de manière secrète le message `m0` suivant :

```
m0 = 'Rendez-vous à 16h place de la liberté. Signé : Alice.'
```

Si Alice transmet directement ce message `m0` sur le réseau, Eve, qui écoute le réseau en permanence, pourra en prendre connaissance.

Partie A : Cryptographie symétrique

On se place dans le cadre d'un chiffrement symétrique avec une seule clé. On suppose disposer d'une fonction `code` en Python telle que `code(m, cle)` permet de chiffrer et de déchiffrer un message `m` à l'aide de la clé `cle`. Cette fonction prend en paramètres deux chaînes de caractères et renvoie une chaîne de caractères. On suppose que, pour tout message `m`, on a toujours : `code(code(m, cle), cle)` égal à `m`.

Ceci veut dire que l'on peut chiffrer un message à l'aide de la clé, puis le déchiffrer exactement de la même manière à l'aide de cette même clé.

Alice effectue donc l'instruction suivante :

```
m1 = code(m0, cle)
```

Elle transmet à Bob le message `m1` ainsi que la clé `cle` sur le réseau.

1. Donner l'instruction que doit écrire Bob pour déchiffrer le message d'Alice et affecter le résultat dans une variable `m2`.

Cependant, Eve dispose du message `m1` ainsi que de la `cle` qui ont tous les deux été transmis sur le réseau. Elle peut donc effectuer la même instruction que Bob et prendre connaissance du message secret.

Partie B : Cryptographie asymétrique

On se place maintenant dans le cadre d'un chiffrement asymétrique avec cette fois-ci une paire clé privée/clé publique. Dans ce système, chaque individu possède une paire de clés associées `(cle1, cle2)`. On suppose toujours disposer d'une

fonction `code` telle que `code(m, cle)` permet de chiffrer ou déchiffrer un message `m` à l'aide de la clé `cle`. On suppose cette fois-ci que, pour tout message `m` et pour toute paire de clés associées `(cle1, cle2)`, on a toujours : `code(code(m, cle1), cle2)` qui est égal à `code(code(m, cle2), cle1)` et qui sont tous les deux égaux à `m`.

Ceci veut dire que lorsque l'on transforme un message à l'aide d'une clé puis de l'autre clé associée on retrouve le message initial. On suppose que la connaissance d'une clé ne permet pas de trouver l'autre. On suppose aussi qu'il est impossible de retrouver un message chiffré par une clé sans connaître l'autre clé.

Alice et Bob ont tous les deux généré une paire de clés associées. On note `(cle1_a, cle2_a)` la paire de clés d'Alice et `(cle1_b, cle2_b)` la paire de clés de Bob. Alice diffuse sa clé `cle1_a` sur Internet mais pas sa clé `cle2_a` et de même Bob diffuse sa clé `cle1_b` sur Internet mais pas sa clé `cle2_b`.

Alice effectue l'instruction suivante, en utilisant la clé `cle1_b` de Bob qu'elle trouve sur Internet :

```
m1 = code(m0, cle1_b)
```

Elle transmet ensuite ce message chiffré `m1` sur le réseau.

2. Donner l'instruction que doit réaliser Bob pour déchiffrer le message d'Alice afin de connaître l'heure et le lieu du rendez-vous.

3. Justifier qu'il est désormais impossible pour Eve de prendre connaissance du contenu du message secret.

4. On parle de système de clé privée/clé publique. Dans l'échange précédent, indiquer quelle est la clé privée et quelle est la clé publique.

5. Bob souhaite accuser bonne réception de ce rendez-vous et transmettre à Alice le message suivant `'Bien reçu. Rendez-vous à 16h donc.'`, dont Eve ne doit pas pouvoir prendre connaissance. Donner, dans l'ordre, les instructions que doivent réaliser Bob et Alice pour sécuriser ce deuxième envoi.

6. Expliquer pourquoi il est nécessaire d'avoir les deux clés au lieu de n'avoir que la clé privée.

Partie C : Signature

Dans cette partie et la suivante, Alice et Bob ne cherchent plus à cacher le message et Alice transmet directement `m0` sur le réseau sans le chiffrer. Mallory souhaite envoyer le message suivant à Bob, en faisant croire que ce message provient d'Alice :

```
m3 = 'Rendez-vous à 15h rue de la dictature. Signé : Alice.'
```

Il intercepte le message `m0` transmis par Alice sur le réseau, le supprime, et le remplace par le message `m3` qu'il transmet à Bob, qui n'a aucun moyen de savoir que le message ne provient pas d'Alice.

Pour éviter cela, on propose un protocole de *signature* permettant à Alice de certifier qu'un message a été écrit par elle.

Alice chiffre son message `m0` avec sa deuxième clé `cle2_a`, c'est-à-dire elle calcule `m0_s = code(m0, cle2_a)` que l'on appelle la *signature* du message `m0`. Elle transmet à la fois le message `m0` et sa signature `m0_s` sur le réseau à Bob. De manière générale, on suppose qu'il n'est pas possible de construire `m0_s` à partir de `m0` sans connaître la clé à l'origine de cette transformation.

7. Expliquer en quoi les informations `m0`, `m0_s` et `cle1_a` connues de Bob permettent de garantir à la fois que le message provient bien d'Alice et que Mallory n'a pas pu envoyer le message `m3` à Bob en se faisant passer pour Alice.

Partie D : Signature par empreinte

Un des problèmes de l'approche précédente est qu'Alice doit transmettre à la fois `m0` et `m0_s`, ce qui double globalement la taille de chaque envoi. Pour résoudre ce problème, on va signer en utilisant une empreinte du message plutôt que le message lui-même.

On rappelle qu'un texte est représenté en machine à l'aide d'un encodage, par exemple l'encodage ASCII, que l'on supposera utilisé dans cet exercice. La fonction `ord` en Python permet d'obtenir le code d'un caractère.

Par exemple :

```
>>> ord('a')
97

>>> ord('b')
98

>>> ord('c')
99
```

On rappelle que la valeur absolue d'un nombre `x`, que l'on note `|x|` est la valeur de ce nombre sans tenir compte de son signe. On suppose disposer de la fonction `abs` en Python qui calcule la valeur absolue d'un nombre passé en paramètre.

Par exemple :

```
>>> abs(4)
4

>>> abs(-6)
6
```

On considère la fonction de réduction qui prend en paramètre une chaîne de caractères de longueur `n` et qui renvoie la somme, pour chaque indice `i` entre `1` et `n - 1`, du produit de l'indice `i` et de la valeur absolue de la différence entre le code du caractère d'indice `i` et celui d'indice `i - 1`.

On considère par exemple la chaîne de caractères `s = 'abca'`, représentée ci-dessous avec les indices :

0	1	2	3
a	b	c	a

Pour cette chaîne de caractères, on obtient donc une réduction :

```
1 * |98 - 97| + 2 * |99 - 98| + 3 * |97 - 99| = 9
```

8. Donner sans justifier la réduction de la chaîne de caractères `'bac'`.

9. Écrire en Python une fonction `reduction` qui calcule l'entier correspondant à la réduction d'une chaîne de caractères.

   Par exemple :

   ```
   >>> reduction('abca')
   9

   >>> reduction(m0)
   62073

   >>> reduction(m3)
   53681
   ```

Dans la suite, on peut utiliser la fonction `str` pour transformer l'entier correspondant à une réduction, en chaîne de caractères.

Alice calcule donc `m0_r = str(reduction(m0))` puis `m0_s = code(m0_r, cle2_a)` qui est la nouvelle signature de `m0`.

10. Décrire ce que doit désormais réaliser Bob pour vérifier l'authenticité du message, c'est-à-dire qu'il a bien été envoyé par Alice.

Pour réduire davantage encore la taille de la signature, on propose de n'utiliser que les dix premiers indices du message `m0` dans le calcul de la réduction plutôt que le message en entier.

11. Commenter cette approche.

Partie E : Chiffrement et signature

Dans le protocole proposé dans la dernière partie, le message `m0` est envoyé sans être chiffré et Eve peut en prendre connaissance.

12. Proposer un protocole qui permet à Alice d'envoyer un message à Bob de manière confidentielle en certifiant que ce message provient bien d'elle et de manière à ce que Eve ne puisse pas en prendre connaissance. Détailler également la procédure que doit suivre Bob pour déchiffrer le message et garantir qu'il provient d'Alice.