IFT-4100/7100 : Aspects pratiques de la chaîne de blocs

TP1 : Signatures basées sur le hachage

Date d'échéance : 18 février 2024

Préparation de l'environnement de travail :

1. Télécharger le fichier *projet_tp1.zip*
2. Créer un environnement virtuel
3. Installer dans l'environnement virtuel créé les packages Python définis dans le fichier *requirements.txt*

Pour ce faire, exécutez la suite de commandes suivante :

```
$ cd projet_tp1/
$ python -m venv .venv
$ .venv\Scripts\activate # (sur Windows)
$ source .venv/bin/activate # (sur Linux ou macOS)
(.venv)$ python -m pip install -r requirements.txt
```

Partie 1

Implémentez les fonctions *generate_keys()*, *sign()* et *verify()* dans *tp1/lamport.py*.

Lorsque vous les aurez implémentées correctement, vous devriez voir afficher "*Vérification de la signature : Bonne*" à l'exécution du programme. Vous pouvez à tout moment tester ceci en exécutant la commande ci-dessous :

```
(.venv)$ python -m tp1
```

Astuce : Vous devrez examiner les bits de chaque octet d'un hachage. Pour ce faire, vous pouvez utiliser des opérateurs de bits.

Assurez-vous que votre code passe les tests en exécutant la commande ci-dessous :

```
(.venv)$ pytest -vv
```

Partie 2

Il y a une clé publique et 5 signatures fournies dans le fichier *signatures.py*. En utilisant ces données, vous devriez être en mesure de contrefaire une autre signature de votre choix. Faites-en sorte que le message que vous signez contienne le mot "contrefait" ainsi que votre adresse e-mail ULAVAL. Il existe un fichier *test_forge.py* qui vérifiera la présence du mot "contrefait" dans le message signé.

5 signatures suffisent pour qu'une implémentation efficace soit relativement rapide. Pour vous assurer que vous êtes sur la bonne voie : sur un MacBook Pro 3,3 GHz Dual-Core Intel Core i7, mon implémentation "non optimisée" peut créer une signature contrefaite en environ 3 minutes.

Si vous réussissez à faire passer les tests dans *test_forge.py*, vous avez probablement réussi. Exécutez la commande :

```
(.venv)$ pytest -vv
```

et laissez-vous guider par les erreurs que vous obtiendrez.

Remarque : N'oubliez pas de formater votre code avant de le soumettre. Vous pouvez le faire en exécutant la commande suivante :

```
(.venv)$ black .
```