

MESSAGERIE CRYPTÉE

Projet hors norme. La documentation pdf sur le sujet est très fournie, ce qui est en accord avec la quantité, la longueur et la complexité des programmes : deux dossiers contenant chacun quatre programmes ! Les modules utilisés sont non standard mais adaptés au sujet (time, socket, os, sys, binascii quoique ce dernier semble inutilisé) et l'un d'entre eux nécessitent une importation : le module Crypto qui est accompagné de sa documentation (en anglais). Bref ce projet est difficile à appréhender, d'autant plus qu'il nécessite, pour le voir fonctionner, que deux ordinateurs soient connectés par un réseau local (LAN).

Personnellement, je n'ai pas découragé ce projet qui semblait tenir à cœur d'un des auteurs, mais je peux difficilement l'évaluer n'ayant pas moi-même les connaissances nécessaires ni non plus forcément le bon réseau local pour le tester... Je peux tenter d'en évaluer certains aspects mais cela reste superficiel et général.

Des parties de programme sont écrites en anglais, comme par exemple dans la fonction Main de code.py, cet extrait :

```
choice = input("Would you like to (E)ncrypt or (D)ecrypt?: ")
if choice == 'E':
    filename = input("File to encrypt: ")
    password = input("Password: ")
    encrypt(getKey(password), filename)
    print("Done.")
```

Les variables en anglais, c'est monnaie courante et même parfois conseillé en programmation, mais le texte qui s'affiche en console : quel besoin de le garder en anglais ? Est-ce la trace d'un emprunt (le programme ou la fonction recopiée d'un programme trouvé sur internet) et d'un oubli de traduire ces phrases, ou est-ce la volonté d'écrire un programme à vocation internationale ? Je penche pour la 1^{ère} possibilité et me demande jusqu'où va l'emprunt... Je comprends que l'utilisation du module Crypto étant particulière, et largement hors programme, elle soit confiée à des fonctions que l'on n'a ni à comprendre ni à évaluer. Mais ce point aurait mérité d'être éclairci.

Il est dit dans le compte-rendu « la dernière version de nos programmes est normalement traduite entièrement en français », dois-je en déduire que je n'ai pas la dernière version ?

Passons à un autre programme, envoi.py. Celui-ci est traduit en français et légèrement commenté, parfois avec des fautes d'orthographe monstrueuses comme « Boucle **temps** que l'ont est connecte » au lieu de « Boucle **tant** que l'on est connecte ». Mes observations s'arrêtent là car une fois encore le programme est trop complexe pour que je puisse l'appréhender en un temps raisonnable. Les autres élèves, non plus, c'est certain, n'en comprendront pas du tout le fonctionnement. Il y a une bonne maîtrise des fondamentaux, c'est une évidence, mais avec la restriction que je ne sais pas à qui attribuer le mérite de cette programmation. Visiblement, cependant, les principes sont compris, bien utilisés et parfois adaptés/améliorés pour une information complète de l'utilisateur potentiel.

Des éléments de programmation qui n'ont pas été abordés semblent bien maîtrisés :

- Les blocs try: except: , massivement utilisés.
- Certaines fonctions du module time comme strftime dans une instruction telle que :
`print(time.strftime(" >> Le %d/%m a %H:%M transfert termine !"))`
- La technologie des sockets qui pourrait être un peu expliquée, notamment la forme des messages envoyés ; ceci afin de mieux comprendre des parties telles que :
`nomFich = recu.split("NAME ")[1]`
`nomFich = nomFich.split("OCTETS ")[0]`
`taille = recu.split("OCTETS ")[1]`

Les programmes server.py et client.py sont assez courts (40 lignes environ chacun) et similaires, dans une boucle while True, les opérations ne sont pas commentées mais relativement éclairées par les messages (en français) qui sont écrits sur la console.

Le compte-rendu donne des explications techniques hors sujet, ou qui mentionne des possibilités non retenues par le projet. Par exemple, que faut-il penser de cette phrase : « Même quand on doit communiquer directement des paquets de données, on préférera utiliser des solutions comme ZeroMQ, bien plus fiables et faciles à mettre en œuvre. » ? Qu'est-ce que ZeroMQ ? Les appréciations sur ce module proviennent apparemment d'une source mais elle est non identifiée.

La notation de ce projet peut paraître arbitrairement basse, compte tenu du haut niveau de programmation nécessaire pour le comprendre, mais elle sanctionne légèrement ce trop grand niveau qui est mal adapté à la demande car trop largement hors sujet. L'idée étant de présenter en phase finale le projet à la classe.

Projet de note : 17/20

Détail de la note :

- préprojet : 2/2 (le projet est ambitieux mais déjà bien cerné et bien documenté)
- compte-rendu : 3/4 (dossier trop technique, surtout destiné à expliquer le fonctionnement de l'ensemble ; narration de quelques difficultés rencontrées et mentions de solutions trouvées ou pas sur internet)
- Programme : 12/14 (je suis témoin qu'il est fonctionnel mais je ne sais pas le faire fonctionner moi-même ; l'ensemble est trop complexe et d'un trop haut niveau ; un travail en équipe a-t-il été réalisé : ce n'est pas évident, même si le compte-rendu dit « nous » je n'ai vu qu'un « je » à l'œuvre)