Projet H : Messagerie sécurisée de bout en bout par chiffrement symétrique AES 256 bits avec possibilité de transfert de fichiers sans limite de taille

Objectifs :

Ces dernières années, la question du chiffrement est largement revenue sur le devant de la scène. Il faut dire que suite aux révélations d'Edward Snowden, nombreux sont ceux qui ont commencé à prendre sérieusement conscience des problématiques concernant la sécurisation de leurs données et de leurs échanges.

Nous voulons donc créer une messagerie instantanée, sans serveur intermédiaire, parfaitement anonyme, chiffrée de bout en bout, utilisant l'un des protocoles actuels de chiffrement symétrique le plus rapide et le plus sûr : l'AES (ADVANCED ENCRYPTION STANDARD) 256 bits, certifié et très utilisé par la NSA.

Pourquoi ce type de chiffrement et pas un chiffrement asymétrique ?

L'utilisation d'un système de chiffrement symétrique ou asymétrique dépend des tâches à accomplir. La cryptographie asymétrique présente deux intérêts majeurs : elle **supprime le problème de transmission sécurisée de la clé**, et elle permet la signature électronique. Elle ne remplace cependant pas les systèmes symétriques car ses **temps de calcul sont nettement plus longs** et la cryptographie asymétrique est de par sa nature même plus vulnérable (repose sur la difficulté à factoriser voir cours d'openclassrooms).

Nous allons donc supposer que la transmission de la clé n'est pas un problème pour l'instant (par la suite possibilité de créer un système hybride symétrique aes et asymétrique rsa).

Par un souci de temps de transfert nous avons donc choisi le chiffrement symétrique AES 256 bits. Pourquoi le AES 256 bits ? Parce que c'est le protocole symétrique qui offre le plus grand nombre de combinaisons et la meilleure résistance au crack (voir tableaux en annexe).

L'intérêt du chiffrement bout en bout (chiffrement des données avant l'envoi et après la réception) est conçu pour résister à toute tentative de surveillance ou de falsification, car aucun tiers ne peut déchiffrer les données communiquées ou stockées.

Réalisation :

Notre projet fera appel à deux principaux programmes :

-chiffrement.py (pour chiffrer les fichiers à envoyer avant l'envoi sur l'autre ordinateur puis pour les déchiffrer après leur arrivée), composé de deux fonctions encrypt(nom_du_fichier, clé_de_chiffrement) et decrypt(nom_du_fichier, clé_de_déchiffrement). Dans le programme chiffrement.py nous utiliserons le module Crypto.Cipher.AES (fiche de documentation en pdf qui renvoie à un autre de 47 pages expliquant tout le fonctionnement de ce protocole s'appuyant sur des permutations matricielles…) afin de chiffrer nos données et nos fichiers à envoyer et le module os pour lire nos fichiers en binaire puis écrire ces fichiers une fois déchiffrés.

-envoi.py (pour envoyer les fichiers et d'éventuels messages instantanés via le protocole TCP/IP)

Dans le programme envoi.py, le module socket sera utilisé pour envoyer les données chiffrées par le protocole TCP/IP.

Déroulement de l'envoi d'un fichier de l'ordinateur d'Alice à celui de Bob (les éternels amoureux de la cryptographie !) :

1-Alice lance le programme d'envoi.py principal

2-Elle doit rentrer l'adresse IP de l'ordinateur de Bob

3-La connexion s'ouvre sur un port préalablement choisi (en dehors de ceux réservés aux applications militaires ou au gouvernement ou au fonctionnement de certaines applications)

4-Alice peut donc envoyer des messages à Bob et Bob peut envoyer des messages à Alice. Mais avant chaque envoi le programme envoi.py importe la fonction encrypt du programme chiffrement.py et à la réception idem pour la fonction decrypt. Les messages sont donc chiffrés avant l'envoi avec une clé préalablement choisie et déchiffrés après la réception.

5-Si Alice ou Bob écrit le message « envoyer fichier ». Le programme demande à l'utilisateur le fichier qu'il souhaite envoyer et avec quelle clé de chiffrement… Le programme fait appel aux fonctions encrypt et decrypt pour chiffrer le fichier en question puis le programme envoi.py lit en binaire le fichier chiffré et le transmet à l'autre ordinateur qui par la suite le déchiffre…

6- Une fois qu'Alice et Bob ont discuté sans qu'aucune personne n'ai pu lire leurs messages, ils ont juste à appuyer sur CMD+Q pour quitter le programme.

Remarque législative : voir annexe

Une interface graphique est envisageable avec Tkinter…