Binome :
Adam MIHOUBI
Alparslane YUKSEL

Rapport de Projet d'initiation à la Sécurité : Client/Serveur FTP Sécurisé

1/- Client/Serveur FTP :

Pour l'architecture, nous avons utilisé un model MVC. Par contre le client est interactif, en effet ayant suivi la première spécification fournies lors du TP et du CM nous pensions qu'il fallait le faire de manière interactive. Le client nous propose de faire des choix pour les transfert. Nous n'avons pas implémenté ça en ligne de commande, car nous avons prit connaissance du sujet sur la page lacl trois jour avant les dates de soumission, nous avions déjà terminé le projet et ils nous aurait fallut repenser entièrement l'architecture du code.

2/- Connexion et échanges des certificats :

Chaque Client ou serveur lors d'une connexion possède son propre certificat ainsi que le certificat auto-signé de l'autorité, ils procèdent tout deux à l'envoie de leur certificat respectif, et vérifie à l'aide la clé publique du CA si le certificat qui leur ai envoyé par l'autre entité a bien été signé par la même autorité.

3/- Création de la clé de session :

Lorsque la connexion est établie, le serveur génère une clé de session symétrique avec l'algorithme AES, la stock dans un TreeMap et l'envoie au client, chaque client possède donc une clé de session unique pour communiquer avec le serveur.

4/-Choix d'envoi ou de récupération de fichier sur le serveur par le client :

Comme notre client est dynamique, on entre au clavier get pour télécharger du serveur ou send pour lui envoyer un fichier.

5/-Création des CSR et gestion des Révocation :

Nous avons créer une classe qui crée le CSR du client. Pour les révocation chaque client se connecte à la CA pour vérifier si son certificat n'est pas révoqué toutes les 5 secondes.

6/- Opérations de base et version évoluée :

Le Client propose a l'utilisateur de choisir le type d'envoi ou de téléchargement du fichier (Crypté au Signé), dans le cas d'un get, le serveur envoie la liste de fichiers existant dans son répertoire Le client choisi par la suite un fichier à télécharger. Dans le cas d'un put, le client entre send, ensuite le nom du fichier qu'il désire télécharger.

7/- Protocole d'Authentification :

S ----> CA : CSR { Local }
S ---> CA : S
CA ---> S : Cert (S) , Cert (CA)
C ---> CA : CSR { Local }
C ---> CA : C
CA ---> C : Cert(C) , Cert(CA)
C ---> S : C , Cert(C)
S ---> C : S , Cert(S)
S ---> C : { $\{K\}_{Kc}$, $\{\{K\}_{Kc}\}_{Ks-1}$ }