Projet sécurité
Implémentation d'un protocole de type FTP sécurisé

Vous devez implémenter en Java une version simplifiée d'un outil de transfert de fichiers sécurisé. Votre implémentation doit satisfaire les spécifications suivantes :

1. L'architecture du protocole devrait être de type client-serveur. Les serveurs sont de type interactif, mais les clients s'exécutent à la demande, sous forme d'une commande avec options, sans interaction avec l'utilisateur.
2. Les clients se connectent sur le serveur, en s'authentifiant avec des certificats signés par une authorité unique. Les serveurs aussi s'authentifient auprès des clients en utilisant des certificats signés par la même authorité.
3. Lors d'une connexion, après vérification de l'authenticité du client, le serveur crée une clé de session pour un algorithme symétrique de votre choix et l'envoie (cryptée et signée !) au client. C'est cette clé qui sera utilisée pour les communications ultérieures, à l'intérieur de cette session, jusqu'à ce que le client se déconnecte.
4. Le lancement du client se fait avec des options pour télécharger vers ou depuis le serveur des fichiers désignés – il s'agit donc des options en ligne de commande à passer au programme qui implémente votre client.
5. La version de base de votre système devrait permettre au serveur d'authentification de créer *hors ligne* des certificats pour des clients, à partir des fichiers qui contiennent toutes les informations nécessaires. Vous pouvez utiliser la classe `PKCS10CertificationRequest` de BouncyCastle.

   La version évoluée devrait permettre la gestion par le serveur d'authentification d'une liste de certificats révoqués, qui peut être consultée par les serveurs ou clients avant de vérifier l'authenticité de leurs interlocuteurs.
6. Les opérations de base sont :
   (a) Le téléchargement cryptée des fichiers (en utilisant la clé de session) dans les deux sens (upload et download).
   (b) L'affichiage des informations sur un répertoire et la modification du répertoire de travail.

   Dans la version évoluée, vous devez permettre aussi le téléchargement des fichiers sans chiffrement mais accompagnés d'un moyen d'authentification (signature, MAC créé avec la clé de session).

La gestion des comptes de téléchargement pourrait représenter une version encore plus évoluée.

7. Le protocole d'authentification entre un client et un serveur devrait être présenté dans le rapport accompagnant votre projet, avec analyse de l'implémentabilité.

Vous devez fournir les sources et des jars (mais sans les jars pour BouncyCastle !) pour l'évaluation de votre projet, ainsi qu'un rapport détaillant comment vous avez abordé chacun des points ci-dessus.