Une “mini-PKI” simplifiée

GS15 - A23 - Projet Informatique

Rapport et codes à rendre avant le 07/01
Soutenances entre le 08/01 et le 12/01

---

1 Description du projet à réaliser

Le but de ce projet informatique est de vous faire créer une technologique similaire à une «mini PKI», *Public Key Infrastructure*, ou Infrastructure (de gestion) de clés publiques. L'idée étant de vous faire développer un outil permettant de mettre en oeuvre des communications sécurisés entre deux utilisateurs.

La description du projet est volontairement «peu détaillée» afin **de laisser, dans ce projet, une très grande liberté quant à l'implémentation pratique des algorithmes, l'architecture du projet** et tous les autres spécificités d'implémentation. Les contraintes et exigeances que votre projet doit respecter sont explicitées dans la section 1.1.
La présence section introductive décrit le but et l'architecture générale du projet à réaliser.

La mise en place d'une communication sécurisée entre deux utilisateurs quelconques, sans aucune connaissance préalable, sur un réseaux non sécurisé nécessitent les opérations suivantes :

1. Authentification de la personne : pour cela il faut recevoir un ***certificat*** qui, d'une part, doit être signée par une tierce personne et, d'autre part, doit pouvoir identifier la personne (dans la vraie, le nom de domaine, dans notre cas, le numéro de téléphone ou l'email ou autre ...).

2. Vérifier la validité du certificat en interrogeant un ***dépot*** de certificat et de clé publique.

3. Enfin sécurisé la communication en utilisant (1) pour la confidentialité, des clés secrètes éphémères pour le chiffrement symétrique et (2) pour l'intégrité, une fonction de hashage, (Hmac ou simple Hash).

Il découle de la présente description des étapes que les protagonistes suivants seront impliqués :

1. Utilisateurs : leur rôle est simplement d'utiliser votre système ; attention pour cela ils doivent tout de même s'inscrire en générant un couple de clés publique / privée (1024 bits minimum) et en donnant un identifiant (email, téléphone,).

2. Autorité de confiance : c'est celle auprès de laquelle les utilisateur s'enregistre ; c'est également elle qui signe les certificats et enregistres ces données dans le ***dépot***.

3. Autorité de sequestre : c'est celle qui gère le ***dépot*** des certifications en y associant une date de validité.

Pour aller plus loin, il vous ai également demandé d'implémenter les fonctions suivantes :

1. La communication sécurisé peut se faire en mode «asynchrone». C'est à dire que Alice peut échanger un message à Bob si même si ce derner n'est pas connecté. Dans ce cas il faut pouvoir remédier à la phase de partage de clé secrète. Cette clé ne peut (évidemment !!) pas être directement envoyée avec le message que Bob consultera plus tard (sinon tout le monde notamment l'autorité de confiance) peut lire ce message. Il faut alors mettre en place un mécanisme de «partage de secret» de «partage asynchrone d'une clé secrète» entre Alice et Bob et faire en sorte que cette clé change.

2. Enregistrée de façon sécurisé des documents publiques «non fongibles» dans une chaîne sécurisée. Cette chaîne doit être vérifiable par tous et tout le monde peut demander à y ajouter un document qui n'est pas nécessairement confidentiel.

3. De vérifier la clé publique d'un utilisateur sans compromission de la clé privée. Chaque doit pouvoir demander à n'importe quel autre utilisateur de prouver qu'il a bien la clé privée correspond à la clé publique enregistrée par l'autorité de sequestre. Cela se fera soit par résolution d'un challenge (demande de chiffrement d'un message, protocole "Ali Baba" ou preuve de connaissance de Schnorr).

La très grande majorité des algorithmes sont décrits de façon très succincte car ils ont été (ou seront) étudiés en cours. Les algorithmes un peu plus détaillé sont ceux demdandés dans la section 1.1.
Encore une fois, vous devez répondre à certaines éxigeances et faire le minimum ou en faire beaucoup plus.

1.1 Exigences

Commençons par rappeler qu'un projet est un examen, toute fraude sera punie (en particulier la réutilisation / reproduction de sources, y compris entre groupe de GS15, sans les mentionner clairement). Toute utilisation de package python pour la cryptographie est prohibée ; les seules librairies autorisées sont celle de portée générales (*e.g.* os, bitarray, numpy, ...).

Notons également qu'il vous ait demandé d'utiliser, pour le chiffrement symétrique, l'algorithme *SERPENT* qui ne sera pas vu en cours et, par ailleurs, qui est légèrement modifié par les soins de votre professeur pervers et sadique.

Il vous est demandé de faire un menu permettant de tester individuellement les algorithmes implémentés; votre programme doit donc, typiquement, afficher lors de l'exécution le menu suivant:

```
Bonjour ô maître Rémi ! Que souhaitez vous faire aujourd'hui ?
->1<- Chiffrer / déchiffrer des message.
->2<- Créer un couple de clé publique / privée (générer un grand nombre premier
->2<- Signer un certificat.
->3<- Vérifier un certificat.
->4<- Enregistrer un document dans le coffre fort.
->5<- Envoyer un message (asyncrhone).
->6<- Demander une preuve de connaissance.
->0<- I WANT IT ALL !! I WANT IT NOW !! SecCom from scratch?.
```

Les algorithmes que vous devez développer, pour chacun des choix possibles, sont décrits ci-dessous (en commençant pas ceux que vous pouvez le plus implémenter le plus tôt dans le semestre) :
Choix `->1<-` : chiffrement et deéchifrfrement symétrique **??** pour la création d'un couple de clés publique / privée (Choix `->4<-`), **??** pour le hashage (Choix `->4<- et ->5<-`) et la preuve de travail (Choix `->6<-`), **??** pour la génération et la vérification d'une signature (Choix `->7<-`).

Enfin le principe de fonctionnement d'une block-chain est présenté dans la section **??**. Cela vous sera utile autant pour votre "culture" personnelle que pour l'implémentation des opérations de création, incrémentation et vérification d'une block-chain (Choix `->8<- et ->9<-`).

Il est conseillé de réutiliser les fonctions données / écrites pour les devoirs, notamment pour la lecture et l'écriture des fichiers, ainsi que les fonctions arithmétiques (tests de Rabin Miller, exponentiation rapide, etc. ...).

De nombreux points sont laissés à votre discrétion. En revanche il y a également de nombreuses consignes à respecter. Ci-dessous sont rappelées les **principales consignes que vous devez obligatoirement respecter** :

1. réaliser votre projet en utilisant le **langage python** ;
2. **respecter les consignes** données dans la section 3 du présent document ;
3. **chaque section contient un paragraphe "exigences" que vous devez suivre** (par exemple utiliser des clés publiques/privées de 1024 bits au moins, écrire les transactions dans des fichiers, etc. ...) ainsi qu'une section "recommandations" dans laquelle des suggestions sont proposées pour aller plus loin.
4. **vous devez rendre un rapport court, répondant uniquement (et pas plus)** aux exigences de la section 3 ; les soutenances auront lieu la soutenance précédent les finaux ; **vous devez réserver un créneau de soutenance**.

Enfin, je vous informe que la notation est faire afin que:

- un programme qui fonctionne et respecte l'ensemble des exigences se voit attribuer un 16/20 ;
- le respect, en sus, de l'ensemble des "recommandations" garantit un 20/20
- toutes les initiatives personnelles seront appréciées et valorisées (mais il est plus important de respecter les consignes)
- les projets de GS15 sont assez complets et chronophages ; **commencez en avance** et, si vous le faites bien, **utilisez les sur votre CV** pour montrer vos compétences dans le domaine de la crypto !

2 Chiffrement symétrique SERPENT

3 Questions pratiques et autres détails

Il est impératif que ce projet soit réalisé en binôme. Tout trinôme obtiendra une note divisée en conséquence (par $3/2$, soit une note maximale de $13,5$).
Encore une fois votre enseignant n'étant pas omniscient et ne connaissant pas tous les langages informatiques du monde ; aussi le language de programmation est imposé : *python*. Par ailleurs, les librairies / packages liés à la cryptographie sont interdites (les modules généraux, *bitarray*, *numpy*, *os*, etc ... peuvent être utilisés).
Par ailleurs, votre code devra être commenté (succinctement, de façon à comprendre les étapes de calculs, pas plus).
De même les soutenances se font dans mon bureau (H109). Vous devriez pouvoir exécuter votre code *python* sur mon PC. Dans tous les cas (notamment si vous utilisez plusieurs bibliothèques, dont certaines non-usuelles) amenez si possible votre machine afin d'assurer de pouvoir exécuter votre code durant la présentation.

Votre code doit être a minima capable de prendre en entrée un texte (pour le chiffrement, la signature, le hash, la block-chain, etc. ...) ; vous pouvez aussi vous amuser à assurer la prise en charge d'image pgm comme en TP, de fichiers binaires, etc. .... mais la prise en charge des textes est le minimum souhaité.

Un rapport très court est demandé : Par de formalisme excessif, il est simplement attendu que vous indiquiez les difficultés rencontrées, les solutions mises en oeuvre et, si des choix particuliers ont été faits (par exemple utilisation d'une bibliothèque très spécifique, quelle fonction de signature, quelle fonction de hashage, quelles modifications ont été nécessaires) les justifier brièvement. Faites un rapport très court (environ 2 pages) ce sera mieux pour moi comme pour vous. Le rapport est à envoyer avec les codes sources.

La présentation est très informelle, c'est en fait plutôt une discussion autour des choix d'implémentation que vous avez faits avec démonstration du fonctionnement de votre programme.

Vous avez, bien sûr, le droit de chercher des solutions sur le net dans des livres (ou, en fait, où vous voulez), par contre, essayez autant que possible de comprendre les éléments techniques trouvés pour pouvoir les présenter en soutenance, par exemple comment trouver un entier premier sécurisé, comment trouver un générateur, etc. ...

Enfin, vous pouvez vous amuser à faire plus que ce qui est présenté dans ce projet ... cela sera bienvenu, mais assurez-vous de faire *a minima* ce qui demandé, ce sera déjà très bien.

Je réponds volontiers aux questions (surtout en cours / TD), mais ne ferais pas le projet à votre place ... bon courage !