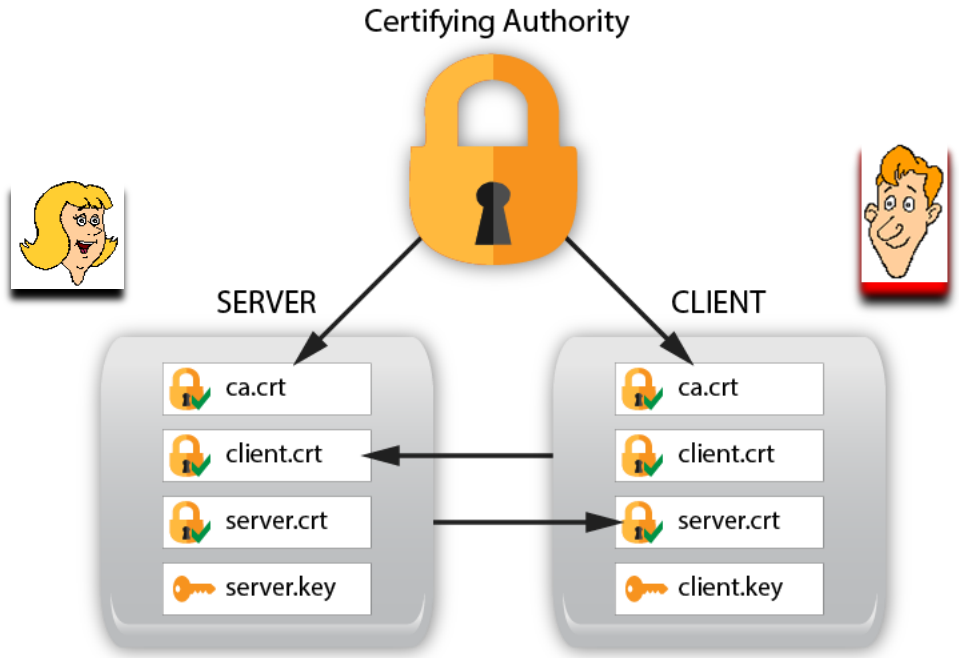
Travaux dirigés –TD
PKI et services de sécurité

Exercice 1 : Génération d'un certificat x509.

On suppose qu'Alice veut obtenir un certificat (**server.crt**) pour son serveur auprès d'une autorité de confiance (AC). On suppose que le client Bob veut aussi obtenir un certificat (**client.crt**) auprès de la même autorité de confiance (AC) pour l'utiliser quand il se connecte au serveur d'Alice. L'autorité de confiance a généralement un certificat appelé auto-signé (**ca.crt**) qui doit être utilisé par les clients pour la vérification des certificats serveurs et par ces derniers pour vérifier les certificats des clients.



Le problème, avec la cryptographie à clef publique est de s'assurer qu'on a bien les bonnes clefs publiques. L'une des solutions consiste à utiliser des certificats. Un certificat contient :

- Une identité (adresse email, URL d'un serveur, adresse IP d'une machine, . . .).
- Une clef publique associée à cette identité.
- Une signature des deux éléments ci-dessus par une autorité de certification

Dans cet exercice nous voulons modéliser les différents échanges entre les demandeurs de certificats et l'autorité de confiance. Pour ce faire, nous supposons que le serveur, le client et l'AC peuvent potentiellement utiliser un certain nombre de méthodes cryptographiques, qui sont décrites dans le tableau suivant :

M	Message contenant : nom, clé publique, adresse, email
K_c K^{-1}_c	Clé publique du client Clé privée du client
K_S K^{-1}_S	Clé publique du serveur Clé privée du serveur
K_{AC} K^{-1}_{AC}	Clé publique de l'autorité de confiance Clé privée de l'autorité de confiance
SHA	Fonction de hachage SHA-256
$Sign$	Signature numérique avec RSA

1. Expliquer comment peut-on obtenir un certificat pour sa clé publique sans divulguer sa clé privée ?

   Pour réaliser un certificat, on a besoin principalement de deux choses : la clé publique, et les informations d'identités. La clé privée n'est pas utile dans ce cas-là. On transmet les informations utiles à l'AC (signée avec la clé privée, pour prouver que l'on possède bien cette clé). L'AC crée un certificat, le signe, et le retourne au demandeur. La clé privée n'a jamais transitée, mais l'AC a pu constater que l'utilisateur en dispose (puisqu'il a signé).

   Les certificats sont signés par leur émetteur donc si la signature est vérifiée par l'AC, l'intégrité et l'authenticité du certificat sont assurées. Si l'on fait confiance à l'émetteur, cela signifie que les informations (clés publique et identité du détenteur) ont été vérifiées par l'émetteur, et donc que l'ensemble est authentique.

2. Pour obtenir un certificat le serveur commence par envoyer une requête de certification signée à l'autorité de confiance. Modéliser cette requête.

   Ms= Nom du serveur + la clé publique du serveur en clair + adresse + date
   Sign $_{Ks}^{-1}$[SHA(Ms)] + Ms en clair
   Le tout est encapsulé dans une enveloppe cryptographique pour la requête de certification définie dans le standard PKCS-10. C'est une requête .csr

3. Pour obtenir un certificat le client envoie une requête de certification signée à l'autorité de confiance. Modéliser cette requête.

   Mc= Nom du serveur + la clé publique du client en clair + adresse + date
   Sign $_{Kc}^{-1}$[SHA(Mc)] + Mc en clair
   Le tout est encapsulé dans une enveloppe cryptographique pour la requête de certification définie dans le standard PKCS-10. C'est une requête .csr

4. L'autorité de confiance doit, après vérification des informations dans les requêtes, signer les requêtes de certifications. Modéliser la signature des deux requêtes (serveur et client).

   Signature de la requête du serveur : Sign $_{Kac}^{-1}$[SHA(Ms)]
   Certificat envoyé du serveur : Sign $_{Kac}^{-1}$[SHA(Ms)] + Ms
   Signature de la requête du client : Sign $_{Kac}^{-1}$[SHA(Mc)]
   Certificat envoyé au client : Sign $_{Kac}^{-1}$[SHA(Mc)] + Mc

   Le certificat est mis sous une enveloppe cryptographique qui peut être du PKCS7 ou PKCS12. On appelle ce certificat un certificat X509.

5. Expliquer comment le client peut vérifier la validation du certificat du serveur. Modéliser le processus.

   Un certificat est valide si : 1. La date actuelle est dans son intervalle de validité, 2. La signature qu'il contient est vérifiée par une AC, 3. Le certificat de l'émetteur (AC) est lui aussi vérifié, ou est de confiance.

   Verif $_{Ks}$[server.crt)] = h1. On calcule SHA (Ms) on trouve h2. Si h1=h2 le certificat est valide.

6. Expliquer comment le serveur peut vérifier la validation du certificat du client. Modéliser le processus.

Un certificat est valide si : 1. La date actuelle est dans son intervalle de validité, 2. La signature qu'il contient est vérifiée par une AC, 3. Le certificat de l'émetteur (AC) est lui aussi vérifié, ou est de confiance.

Verif $_{Kc}$[client.crt)] = h1. On calcule SHA (Mc) on trouve h2. Si h1=h2 le certificat est valide.

7. Quel protocole déployé à grande échelle de nos jours utilise le système d'autorités de confiance ?

Le protocole SSL/TLS, utilisé par exemple par le protocole HTTPS, utilise un système d'autorités de confiance pour valider les certificats. C'est le protocole le plus utilisé sur le web pour sécuriser les échanges entre un client et un serveur.

Exemple d'un message échangé :

Si Alice souhaite signer le hash de son message avec sa clé privée, elle transmettra par exemple : Sign $_{K}^{-1}$[SHA(M)].

Application numérique

La clé publique RSA du serveur (n_s=221,e_s=5) et sa clé privée est d_s=77.
La clé publique RSA du client est (n_c=91,e_c=5) et sa clé privée est d_c=29.
La clé publique RSA de l'autorité est (n_{ac}=299, e_{ac}=5) et sa clé privée est d_{ac}=53.

Nous supposons que le contenu de la requête de certification envoyée par le serveur est M1=s||221||date => M1= 115 221 17102019

Nous supposons que le contenu de la requête de certification envoyée par le client est M2=c||91||date => M2= 99 91 17102019

L'algorithme de hachage appliqué sur les données numériques fonctionne de la manière suivante : on divise les éléments de données en groupes de deux chiffres. Ensuite, on calcule la somme des groupes. Exemple: si le message est 4432, la somme résultante sera 76 car 44+32=76). La valeur de hachage du message complet M1 est calculée en additionnant les valeurs de hachage de chaque élément de données et en appliquant l'opération de modulo 91.

Calculer les valeurs des opérations cryptographiques (signature et vérification) dans les questions 2 à 6.

Pour la requête du serveur :
Le hash du M1=11+52+21+17+10+20+19 mod 91 = 150 mod 91=59
S=M^{ds} mod n_s = 59^77 mod 221 = 128
L'autorité de confiance vérifie de la manière suivante:
V=S^{es}mod n_s = 128^5 mod 221 =59 et H(M1)= 11+52+21+17+10+20+19 mod 91 = 150 mod 91=59 donc c'est bon. Dans la pratique l'autorité réalise une enquête de vérification de toutes les informations fournies comme l'adresse, la sécurisation de la clé privée, la possession du nom du domaine etc. et ceci avant de signer le certificat.

Pour la requête du client :
Le hash du M2=99+91+17+10+20+19 mod 91 = 256 mod 91=74
S=M^{dc} mod n_c = 74^29 mod 91 = 16
L'autorité de confiance vérifie de la manière suivante:

V=S^{ec}mod n_c = 16^5 mod 91 =74 et H(M2)= 99+91+17+10+20+19 mod 91 = 256 mod 91=74
donc c'est bon. Dans la pratique l'autorité réalise une enquête de vérification de toutes les informations fournies comme l'adresse, la sécurisation de la clé privée, la possession du nom du domaine etc. et ceci avant de signer le certificat.

L'autorité de confiance signe le certificat du serveur :

L'autorité signe la demande du serveur avec S=$M1^{dac}$ mod n_{ac}=59^53 mod 299=141
Le certificat server.crt= Signature + M1, le tout en PKCS7 ou PKCS12.

L'autorité de confiance signe le certificat du client :

L'autorité signe la demande du client avec S=$M2^{dac}$ mod n_{ac}=74^53 mod 299=172
Le certificat client.crt= Signature + M2, le tout en PKCS7 ou PKCS12.

Le client vérifie le certificat du serveur :
L'autorité doit transmettre sa clé publique au client et serveur sous le nom ca.crt
Le client vérifie : H(M1)= 11+52+21+17+10+20+19 mod 91 = 150 mod 91=59
V=141^5 mod 299 = 59 donc c'est valide

Le serveur vérifie le certificat du client :
L'autorité doit transmettre sa clé publique au client et serveur sous le nom ca.crt
Le serveur vérifie : H(M2)= 99+91+17+10+20+19 mod 91 = 256 mod 91=74
V=172^5 mod 299 = 74 donc c'est valide

Exercice 2 : Chaînes de certification

Alice reçoit le certificat de Bob signé par l'autorité de certification TrustSign. Malheureusement Alice ne connaît pas la clef publique de TrustSign. Il se trouve que cette clef (i.e., clef publique de TrustSign) est certifiée par l'autorité de certification VeriSign (dite racine) dont Alice a entièrement confiance.

a) Dessinez le graphe hiérarchique des différents certificats, en indiquant à chaque fois les clés authentifiées.
b) Dessinez le graphe de la chaîne de confiance qui en résulte. Comment Alice pourra-elle vérifier le certificat de Bob ?
c) Que pouvez-vous dire de la validité de la clef contenu dans le certificat de Bob ?