COMPTE RENDU
PROJET CRYPTOGRAPHIE

BENDER Quentin

GOUIRAN Hugo

Sommaire

01. ACR & ACI

Concernant l'ACR, nous avons choisis une durée de 10 ans car cela permet une longue utilisation sans avoir à redéfinir les clé et certificats. Pour sa clé, on utilise l'algorithme ECC 384 pour assurer une haute protection à cette clé.

Le choix de l'ECC donne une clé plus courte qu'un chiffrement RSA mais offre une même résistance. Cela permet de générer des clés plus rapidement et d'utiliser moins de stockages.

Commande :

openssl ecparam -genkey -name secp384r1 -out ../ACR/root_ca.key

openssl req -new -x509 -key ../ACR/root_ca.key -out ../ACR/root_ca.crt -days 7200

De ce qu'y est de l'ACI, nous avons mis une durée de 2 ans qui permet une certaine longueur tout en permettant un renouvellement en cas de vol/déchiffrement de clé et certificat. Pour la clé, on la chiffre avec un ECC 256 qui permet aussi une certaine protection (plus légère que l'ACR étant donné le renouvellement toutes les 2 années).

Commande :

openssl ecparam -genkey -name prime256v1 -out ../ACI/intermediate_ca.key

openssl req -new -key ../ACI/intermediate_ca.key -out ../ACI/intermediate_ca.csr

openssl x509 -req -in ../ACI/intermediate_ca.csr -CA ../ACR/root_ca.crt -CAkey ../ACR/root_ca.key -CAcreateserial -out ../ACI/intermediate_ca.crt -days 730 -extfile ../ACI/intermediate_ca.cnf -extensions v3_intermediate_ca

02. CREATION DU CERTIFICAT USER

L’utilisateur va inscrire son adresse électronique et par la suite recevoir un code unique lié à son mail. Si l’adresse est vérifiée, cela sera stocké dans la base de données au format JSON. Par la suite, il devra spécifier les champs afin de générer son certificat.

Lors de la génération, on créé une clé privé en prime256 puis on créé le certificat qui est chiffré en sh256 qui offre une bonne protection sans demander trop de moyen. Il a une durée de 1 an qui est une durée raisonnable pour un certificat d’utilisateur car il devra le renouveler et ça évite les risques de vols de certificats.

page de renseignement e-mail

page de vérification du code renseigné par mail

Une fois que la vérification d'un code est fait par email on permet à l'utilisateur de rentrer ces informations pour la demande de CSR.

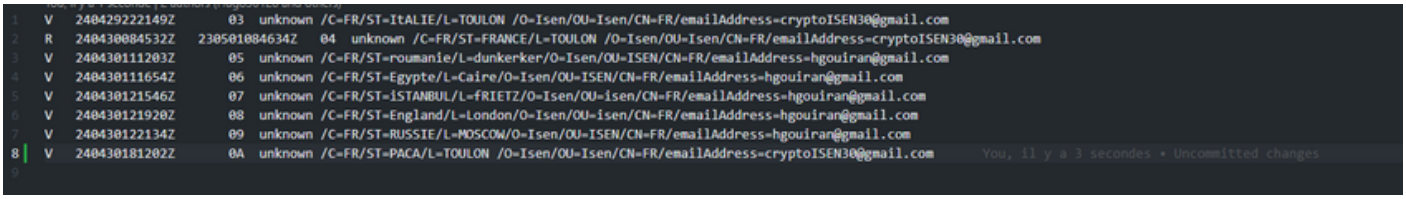
page permettant de renseigner les champs d'informations de la CSR

Des que tous les champs sont renseignés la signature de la CSR par l'ACI se fait automatiquement et du coup et l'user atterit sur une page pour télécharger un pkcs de son certificat pour le renseigner sur Thunderbird

03. REVOCATION CERTIFICAT USER

Ici on demande dans un premier temps l'ID du certificat que veut révoquer l'utilisateur.

Si l'ID existe on arrive sur une page qui nous demande quelle est la raison de la révocation ainsi que l'email associé aux certificats pour bien confirmer la révocation.



Si tous les informations sont correctes alors le certificat est révoqué et les informations sont stockés dans un fichier .db qui nous permet de voir si le statut du certificat est V (validé) ou R (révoqué).

Dans ce fichier se trouve le serial qui correspond à l'ID du certificat ainsi que la date de quand à été crée ou révoqué le certificat et enfin les informations contenues dans le certificat.

Et ensuite avec un serveur OCSP nous pourrons envoyer des requetes à ce serveur qui sera lié à la database de l'ACI qui répondra si le certificat est valide ou non

04. CREATION SERVEUR OCSP

Création d'un serveur OCSP signé par l'ACI et en utilisant quasiment le fichier de configuration que pour l'ACI

Commande :

```
openssl ecparam -genkey -name prime256v1 -out ../OCSP/ocsp.key

openssl req -new -key ../OCSP/ocsp.key -out ../OCSP/ocsp.csr

openssl x509 -req -in ../OCSP/ocsp.csr -CA ../ACI/intermediate_ca.crt -CAkey
../ACI/intermediate_ca.key -CAcreateserial -out ../OCSP/ocsp.crt -days 730
```

Ensuite après avoir crée une AC pour l'OCSP on peut lancer le serveur via cette commande :

Commande :

```
openssl ocsp -port 8081 -index ../ACI/intermediate_ca.db -rsigner
../ACI/intermediate_ca.crt -rkey ../ACI/intermediate_ca.key -CA
../ACI/intermediate_ca.crt -text -crl_check
```

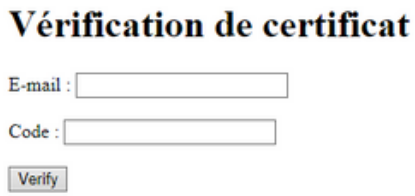
Ici je lance mon serveur OCSP sur le port 8081 et du coup écoute les requetes OCSP sachant qu'il est relié à une database ici il s'agit de celle de l'ACI.

Ci dessous une requete typique OCSP :

Commande :

```
openssl ocsp -issuer ../ACI/intermediate_ca.crt -cert
.\usercertificate\cryptoISEN30@gmail.com\26649\certificate.crt -u
http://localhost:8081 -text
```

05. VERIFICATION OCSP



page de vérification via OCSP

Pour vérifier le statut d'un certificat OCSP on interroge du coup le serveur OCSP en envoyant une requête avec le certificat qui est identifié par l'Email et le code du certificat envoyé par mail.

Tout ca fait automatiquement que ce soit d'allumer le serveur OCSP et d'envoyer la requete OCSP avec les informations qui seront renseignés

```
usercertificate/cryptoISEN30@gmail.com/12101/certificate.crt: revoked
```

exemple d'un certificat révoqué

```
usercertificate/cryptoISEN30@gmail.com/29467/certificate.crt: good
```

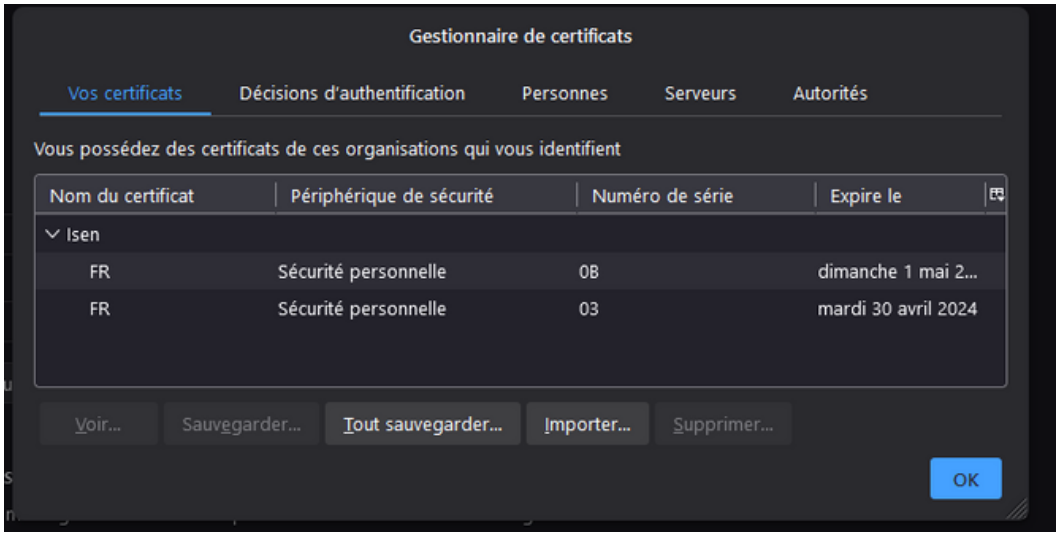
exemple d'un certificat valide

06. SCENARIO ATTAQUE

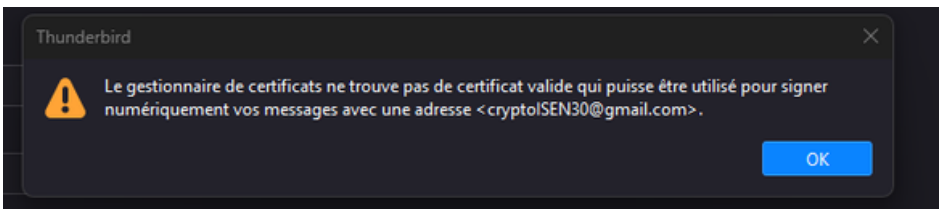
Scénario avec un certificat expiré

Nous avons essayé de mettre un certificat utilisateur avec une date de validation expirée ici normalement le certificat est déjà expiré depuis le 30 avril 2022 :

openssl ca -batch -config ../ACI/intermediate_ca_copy.cnf -in .\usercertificate\cryptoISEN30@gmail.com\12101\csr.csr -out .\usercertificate\cryptoISEN30@gmail.com\12101\certificate_revoked.crt -enddate 220430235959Z



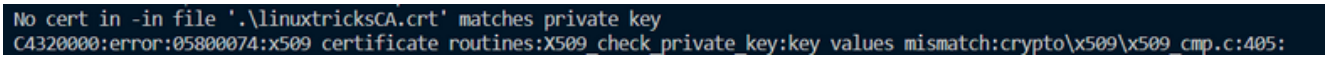
Ici j'ai importé les deux certificats : un valide et l'autre qui est expirée et ensuite quand j'essaye d'utiliser ce certificat pour signer les mails avec S/MIME il me renvoie cette réponse



Parfois c'est meme possbile qu'on ne puisse pas le certificat car non valide

Autre scénario d'attaque (Fausse AC)

Nous pouvons créer la fausse AC avec la clé et le certificat d'un utilisateur. Cependant, quand on essaie de générer un certificat en utilisant notre fausse AC, une erreur apparait pour préciser qu'il y a un problème au niveau des clés car différentes entre le vrai AC et notre fausse AC



Création de l'AC

```
openssl req -x509 -new -nodes -key
..\usercertificate\hgouiran@gmail.com\24727\private.key -sha256 -days
10000 -out linuxtricksCA.pem
```

```
openssl x509 -in linuxtricksCA.pem -inform PEM -out
linuxtricksCA.crt
```

Génération du certificat

```
openssl pkcs12 -inkey
.\usercertificate\hgouiran@gmail.com\24727\private.key -in
.\linuxtricksCA.crt -export -out itconnect.pfx
```

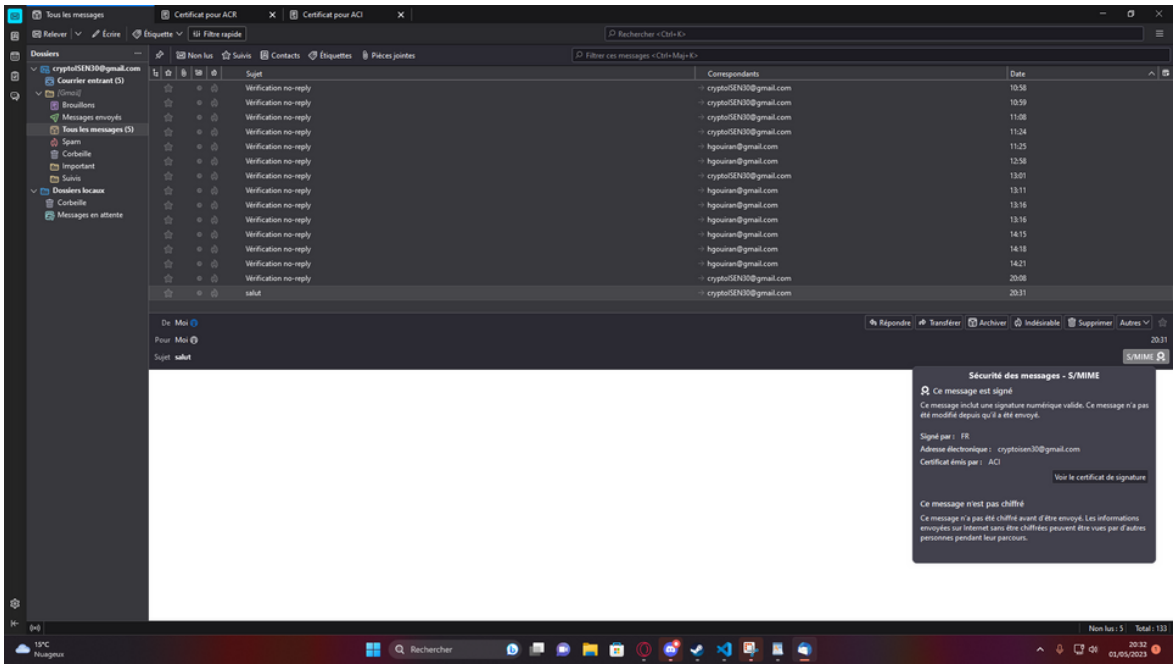
07. THUNDERBIRD

Ici on renseigne les certificats de l' ACI et de l'ACR pour que ensuite on puisse renseigner fournir des certificats utilisateurs signés par nos autorités importés

Certificat ACR

Certificat ACI

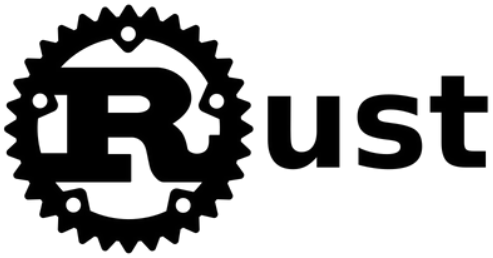
Après avoir renseigné les autorités on peut renseigner les certificats utilisateurs qui vont nous permettre de signer nos mails



Sur cette capture j'ai réussi à signer un mail avec le certificat utilisateur et je me suis envoyé mon propre mail et je peux voir le certificat de l'émetteur.

Malheureusement sur ThunderBird nous n'avons pas réussi à utiliser la vérification OCSP .

08. TECHNOLOGIE UTILISE



Pour ce projet nous avons décidé d'utiliser **Rust** qui est un language qu'on utilise tous les jours pour notre master-projet donc nous avons de bonne connaissances sur ce language sachant que nous utilisons des frameworks comme **Actix-Web** qui nous permettent de créer une interface web facilement.

A cela s'ajoute aussi les commandes openssl qui nous permet de gérer les opérations sur les certificats malheureusement comme nous sommes sur Windows nous n'avons pas utilisé les librairies **Openssl** qui sont déjà dans rust c'est pourquoi nous exécutons des commandes openssl dans le rust.

Nous avons utilisé les protocoles **SMTP** pour envoyer des emails et créer des adresses mail test.