CRISPEL Esteban
DEVAUX Baptiste

Vérification de chaînes de certificats

Projet de cryptographie

2023-2024

Sommaire :

Introduction :

La validation des certificats revêt une importance capitale dans le domaine de la sécurité informatique. Les certificats X.509 sont utilisés pour garantir l'authenticité lorsque l'on consulte un site web.

La vérification des certificats implique plusieurs étapes pour garantir la confiance et la sécurité du système :

- **Authenticité de l'émetteur :** Il est essentiel de s'assurer que le certificat a été émis par une autorité de certification légitime et digne de confiance.
- **Validité temporelle :** La période de validité d'un certificat doit être vérifiée pour éviter son utilisation après expiration ou avant sa date d'émission.
- **Intégrité des données :** La vérification de l'intégrité du certificat garantit qu'il n'a pas été altéré depuis sa délivrance. Cela implique la validation de la signature numérique du certificat à l'aide de la clé publique de l'autorité de certification.
- **Chemin de certification :** Dans le cas des certificats intermédiaires ou de chaînes de confiance, il est crucial de vérifier la validité de tous les certificats dans la chaîne, en s'assurant que chaque certificat est signé par l'autorité de certification précédente jusqu'à atteindre l'autorité racine.
- **Révocation :** Enfin, la vérification du statut de révocation d'un certificat est essentielle pour détecter les certificats qui ont été révoqués avant leur expiration prévue, en raison de compromissions de clés privées ou d'autres incidents de sécurité.

Choix d'outils :

Environnement de développement : VsCode et IntelliJ sur Linux parce qu'on a l'habitude d'utiliser ces IDE.
Langage : Java, les librairies étant donné en Java, pas besoin d'en chercher d'autres
Librairies : java.security, math.BigInteger et bouncycastle.

Étapes implémentées :

3.1 Validation d'un certificat d'autorité racine

Test sur des certificats valides et invalides, pour chaque certificat:

- Vérification de la signature
- Vérification de la clé publique
- Vérification de KeyUsage
- Vérification de la période de validité

3.2 Validation d'une chaîne de certificats

Pour vérifier la chaîne de certificat, on va reprendre les vérifications précédentes et en ajouter de nouvelles :

- Vérification de la signature
- Vérification de la clé publique
- Vérification de KeyUsage
- Vérification de la période de validité
- Vérification de l'émetteur
- Vérification des basic
- Vérification du statut de révocation

Nous avons testé la vérification de chaîne sur plusieurs site, voici les résultats :

- Amazon : Toute la chaîne est valide
- Tbs : Toute la chaîne est valide
- Certificat expiré : Le programme détecte bien qu'un certificat est expiré
- Facebook : Toute la chaîne est valide

Structure du programme :

Trois fichiers:

- *validateCert.java*
- *validateCertChain.java*
- consoleColors.java

Le plus gros fichier (*validateCertChain*) est décomposé en plusieurs fonctions permettant la simplification du code et la réutilisation de celles-ci.

ConsoleColors permet de mieux présenter les différentes vérifications du programme. Ce fichier appartient à Brandan LE-GALL, il nous l'a gentiment donné.

Difficultés :

- Utilisation de la lib BigInteger
- Utilisation de la lib BouncyCastle : pour le téléchargement de la CRL, il y a l'erreure "unknown object in getInstance: org.bouncycastle.asn1.DEROctetString" que nous n'avons pas réussis à fixer. Cette erreur n'apparaît pas pour les RCA.

Sources :

- Docs des librairies utilisées
- LLMs (gpt 3.5, claude 3)
- Stackoverflow
- Le cours fournis : A1 - Intro to crypto - Crypto in Java.pdfr