GÉNÉRER UN CERTIFICAT AUTO-SIGNÉ AVEC OPENSSL

Voici les étapes pour créer un certificat auto-signé et l'implémenter sur Node.js localement :

- Tout d'abord, vous devez générer une clé privée et un certificat auto-signé en utilisant OpenSSL. Si vous n'avez pas OpenSSL installé sur votre système, vous pouvez le télécharger et l'installer depuis le site web officiel.

- Ouvrez un terminal et exécutez les commandes suivantes :

```
openssl genrsa –out key.pem 2048
openssl req –new –key key.pem –out cert.csr
openssl x509 –req –in cert.csr –signkey key.pem –out cert.pem
```

Vous serez invité à saisir des informations de certificat telles que le nom de l'organisation, le nom commun, etc. Il est important de fournir des informations précises et correctes pour ces champs car elles seront incluses dans le certificat.

Une fois ces commandes exécutées, vous devriez avoir deux fichiers **: key.pem** (la clé privée) et **cert.pem (le certificat auto-signé).**

```
openssl genrsa –out key.pem 2048
```

Voici une explication détaillée de chaque paramètre :

- openssl est le nom du programme OpenSSL à exécuter. OpenSSL est une boîte à outils logicielle de sécurité qui fournit des implémentations de cryptographie, de certificats et de protocoles de sécurité.

- genrsa est une commande OpenSSL pour générer une paire de clés publique-privée pour un algorithme de cryptographie RSA.

- -out key.pem indique à OpenSSL d'enregistrer la clé privée générée dans un fichier nommé key.pem. Vous pouvez choisir un autre nom de fichier ou un autre emplacement pour le fichier de clé privée.

- 2048 est la longueur de la clé privée RSA, exprimée en bits. Une clé privée de 2048 bits est considérée comme suffisamment sûre pour la plupart des applications actuelles, mais vous pouvez choisir une longueur différente en fonction de vos besoins de sécurité.

```
openssl req -new -key key.pem -out cert.csr
```

- req est une commande OpenSSL pour créer une requête de signature de certificat (CSR). Une CSR est un message cryptographique qui contient les informations requises pour obtenir un certificat signé par une autorité de certification.

- -new indique à OpenSSL de créer une nouvelle CSR. Si vous avez déjà une CSR, vous pouvez utiliser l'option -out pour spécifier un nom de fichier différent pour la nouvelle CSR.

- -key key.pem indique à OpenSSL d'utiliser la clé privée stockée dans le fichier key.pem pour générer la CSR. Il est important que la CSR soit générée à partir de la même clé privée que celle qui sera utilisée pour le certificat signé.

- -out cert.csr indique à OpenSSL d'enregistrer la CSR générée dans un fichier nommé cert.csr. Vous pouvez choisir un autre nom de fichier ou un autre emplacement pour la CSR.

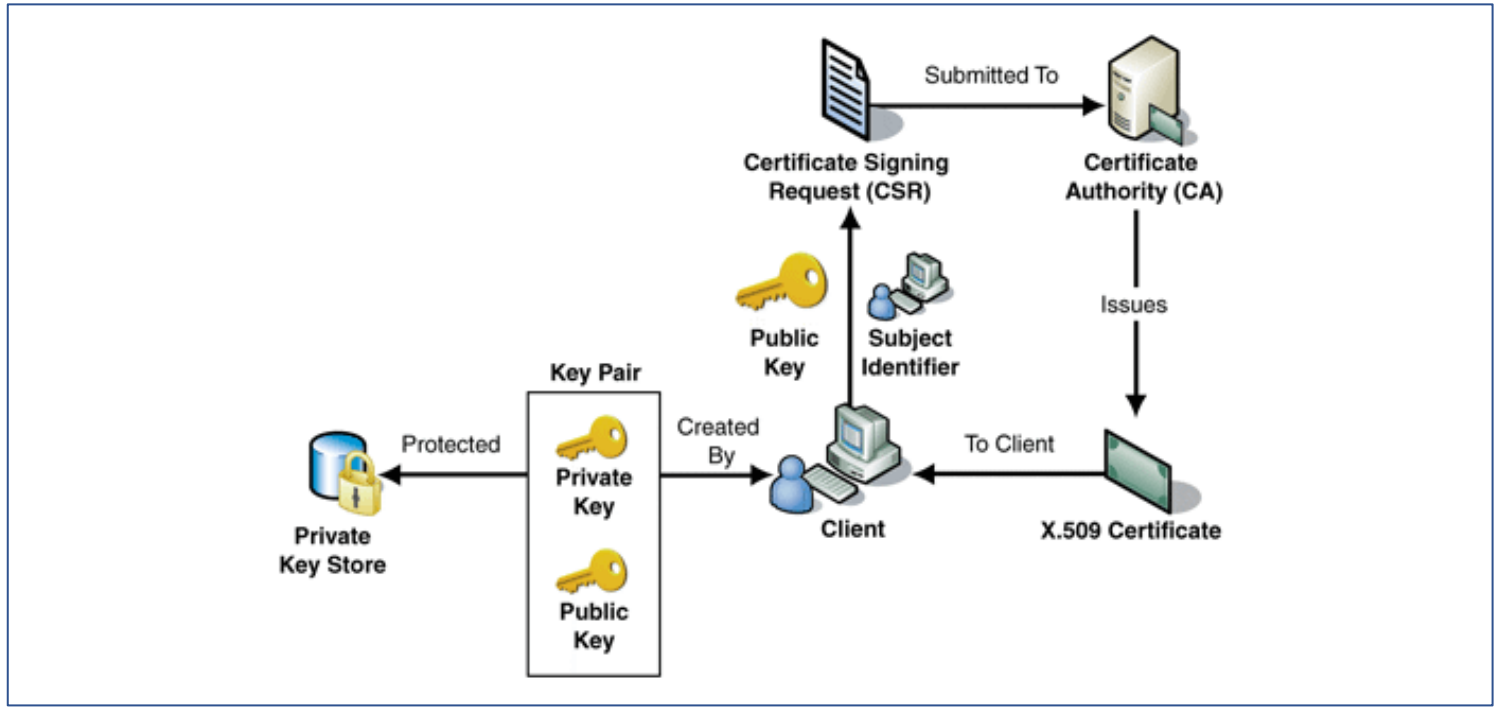
En résumé, cette commande OpenSSL génère une requête de signature de certificat (CSR) à partir de la clé privée RSA stockée dans le fichier key.pem et enregistre la CSR dans un fichier nommé cert.csr, qui peut être utilisé pour obtenir un certificat signé par une autorité de certification.

CSR : CERTIFICATE SIGNING REQUEST

Le CSR contient les informations suivantes :

- Les informations d'identification de l'entité qui demande le certificat, telles que le nom commun (CN), l'organisation (O), le département (OU), etc.
- La clé publique à inclure dans le certificat.
- Le nom de l'algorithme de hachage utilisé pour signer le CSR.

Une fois que vous avez généré un CSR, vous pouvez le soumettre à une autorité de certification (CA) pour obtenir un certificat signé. La CA vérifiera les informations dans le CSR et émettra un certificat signé numériquement contenant la clé publique et les informations d'identification de l'entité.



```
openssl x509 -req -in cert.csr -signkey key.pem -out cert.pem
```

- x509 est une commande OpenSSL pour manipuler des certificats X.509.
- -req indique à OpenSSL que la source d'entrée est une CSR (Certificate Signing Request) plutôt qu'un certificat.
- -in cert.csr spécifie le nom du fichier d'entrée contenant la CSR à signer.
- -signkey key.pem indique à OpenSSL d'utiliser la clé privée stockée dans le fichier key.pem pour signer la CSR.
- -out cert.pem indique à OpenSSL d'enregistrer le certificat signé dans un fichier nommé cert.pem.

En résumé, cette commande utilise OpenSSL pour signer une CSR avec une clé privée et enregistrer le certificat signé dans un fichier cert.pem. Ce certificat auto-signé peut être utilisé pour des tests locaux ou des développements, mais ne doit pas être utilisé pour un déploiement en production.

INTEGRATION D'UN CERTIFICAT AUTO-SIGNÉ SUR NODEJS

```
const express = require('express');
const https = require('https');
const fs = require('fs');

const app = express();

const options = {
    key: fs.readFileSync('key.pem'),
    cert: fs.readFileSync('cert.pem')
};

app.get('/', (req, res) => {
    res.send('Hello, world!');
});

https.createServer(options, app).listen(3000, () => {
    console.log('Server started on port 3000');
});
```