

**Introduction à la cryptographie
et
Application à la sécurité informatique**

Toulouse, 24 & 25 septembre 2015

Je me présente...



- Jean-Baptiste ANGELELLI
- Ingénieur (École Centrale de Paris 2005)
- Architecte logiciel au Grand Port Maritime de Marseille

Au menu :



- Les bases de la cryptographie
- Les fonctions de hachage
- Les certificats électroniques
- Utilisation des certificats électroniques
- Les mathématiques de la cryptographie
- L'environnement de la sécurité informatique

Qu'est ce que la cryptographie ?

L'art est la science du chiffrement

Une petite mais cruciale partie de la sécurité informatique

Et vous ça vous fait penser à quoi ?



La cryptographie à la Jules César

- Essayons de chiffrer “julius caesar”

j u l i u s c a e s a r

- On utilise une table de correspondance

| | | | | | | | | | |
|---|---|---|---|---|-------|---|---|---|-------|
| A | C | E | I | J | L | R | S | U | space |
| K | Z | U | O | B | space | C | L | R | H |

- Et on remplace la lettre par la lettre qui lui correspond

b r o r l h z k u l k c

- Et on obtient “br orlhzkulkc” !



Qu'en pensez vous?



Principes de bases

- **L'algorithme**, c'est le mécanisme à exécuter pour chiffrer et déchiffrer.



- **La clé**, c'est l'élément à fournir à l'algorithme en plus du clair pour obtenir le chiffré, ou en plus du chiffré pour obtenir le clair



Le vocabulaire de la cryptographie

- Essayons de chiffrer “julius caesar”

j u l i u s c a e s a r

- On utilise une table de correspondance

| | | | | | | | | | |
|---|---|---|---|---|-------|---|---|---|-------|
| A | C | E | I | J | L | R | S | U | space |
| K | Z | U | O | B | space | C | L | R | H |

- Et on remplace la lettre par la lettre qui lui correspond

b r o r l h z k u l k c

- Et on obtient “br orlhzkulkc” !

Le clair

La clé

L'algorithme

Le chiffré



Les attaques

- Le modèle *Chiffré seul*
- Le modèle *Clair connu*
- Le modèle *Clair choisi*
- Le modèle *Chiffré choisi*

Nos amis



Alice



Bob

Sauf que...



Alice



Charlie



Bob



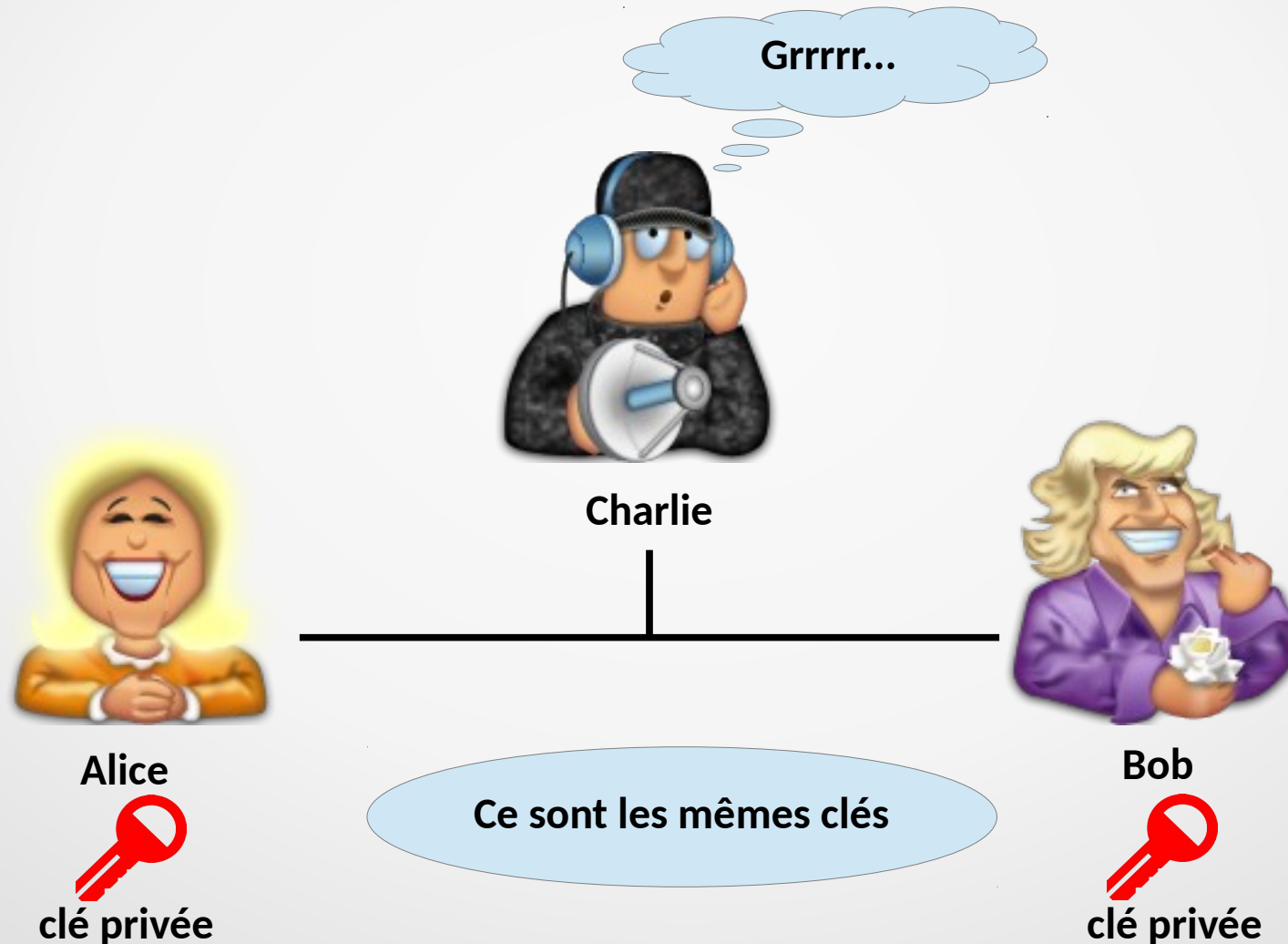
Principes de base



- En cryptographie, on part du principe que toutes les communications sont interceptées.
- Empêcher Charlie d'accéder à la ligne n'est pas l'objet de la cryptographie.

Cryptographie symétrique

- Également appelée “à clé privée”



Cryptographie symétrique



- Soit m le message *clair*
- Soit c le message *chiffré*
- Soit K la *clé*
- Soit E la fonction de chiffrement, on a :
$$c = E(K, m)$$
- Soit D la fonction de déchiffrement, on a
facilement : $m = D(K, c)$
mais difficilement : $m = D(c)$

Cryptographie symétrique

Je fais $c = E(K,m)$
et j'envoie c à Bob!



Alice



clé privée K



Charlie

Grrrrr.. J'ai c et D
mais pas K !!

Trop facile! Je reçois c
et je fais $m = D(K,c)$



Bob



clé privée K

c

c

c

Ce sont les mêmes clés

Cryptographie symétrique

Quelques algorithmes de cryptographie symétrique :

- Advanced Encryption System (AES)
- Triple Data Encryption Algorithm (3DES)
- One time pad
- Twofish



Cryptographie symétrique



Moi, je vois quelques problèmes dans ce processus..

Lesquels ??



Principes de base



Principe de Kerckhoff :

La clé est secrète, pas l'algorithmme, ni son implémentation !!

Pourquoi ??



Principes de base

L'algorithme doit être public car :

- premièrement, il n'est pas possible de le garder secret, donc pas question de baser une partie de la sécurité là-dessus.
- deuxièmement, la publication permet la revue critique : plus il y a de revues critiques, plus l'algorithme est sûr !



Principes de base

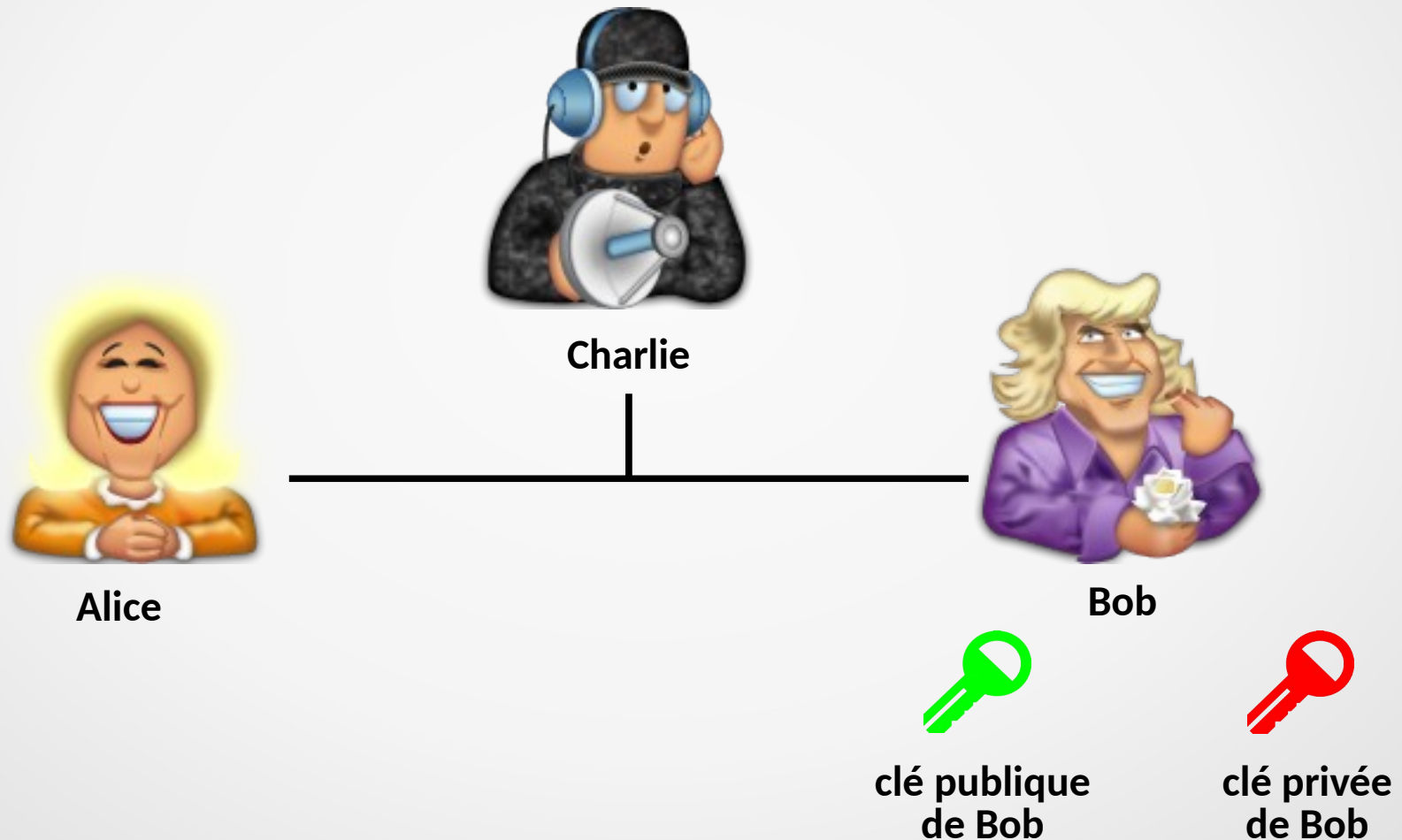


L'implémentation doit être publique car :

- premièrement, pour être sûr que le logiciel fait ce qu'il est supposé faire.
- deuxièmement, la publication permet la revue critique : plus il y a de revues critiques, plus l'implémentation est sûre !

Cryptographie asymétrique “à clé publique”

- Le plus souvent appelée cryptographie à clé publique



Cryptographie asymétrique “à clé publique”



- Les messages sont chiffrés au moyen de la clé publique.
- Les messages sont déchiffrés au moyen de la clé privée.

Cryptographie asymétrique “à clé publique”



Bob

Génial !! Je distribue ma clé publique à tout le monde! Ils l'utilisent pour chiffrer les messages qu'ils m'envoient, et je suis le seul à pouvoir les déchiffrer avec ma clé privée!



**clé publique
de Bob**



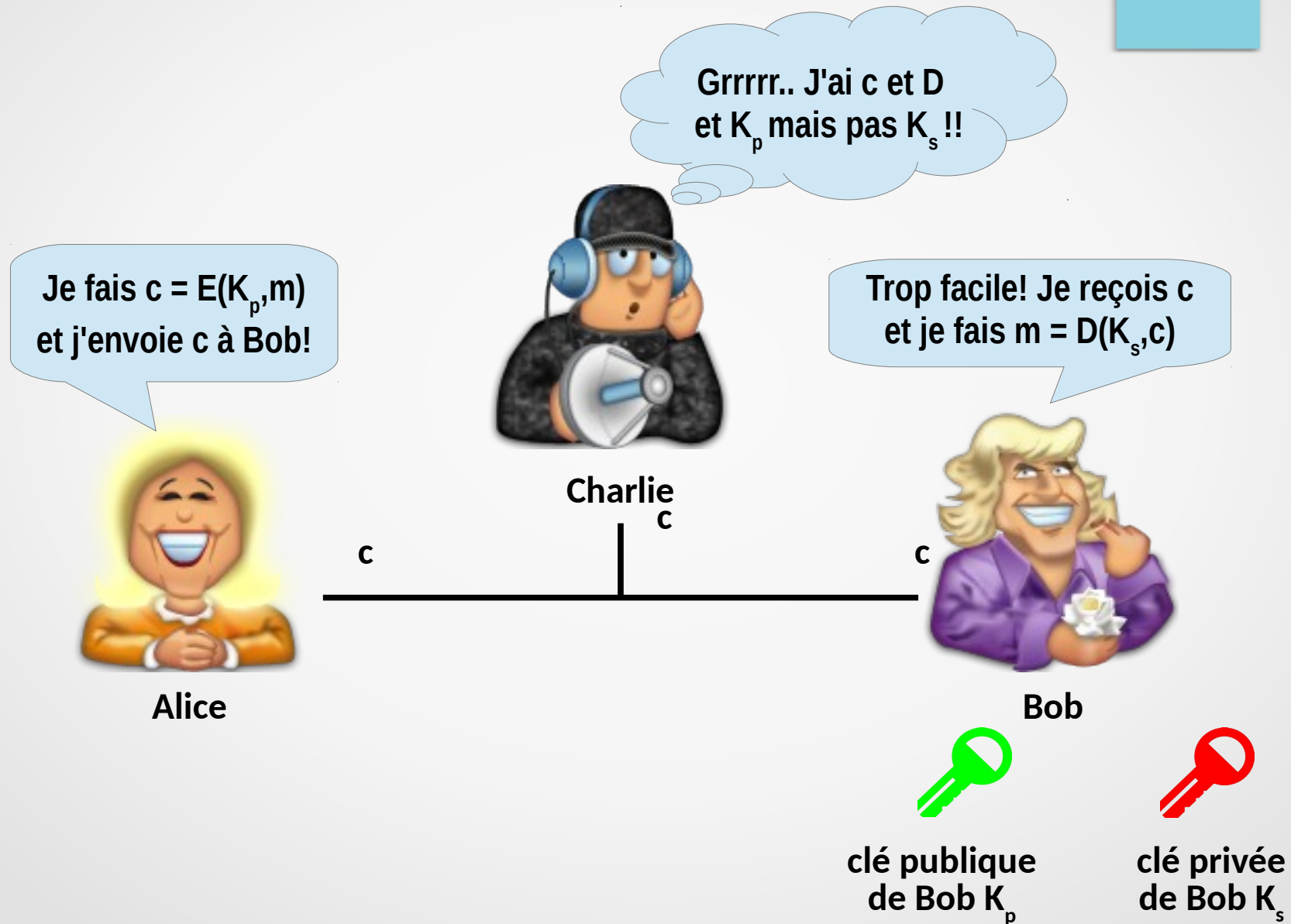
**clé privée
de Bob**

Cryptographie asymétrique “à clé publique”



- Soit m le message *clair*
- Soit c le message *chiffré*
- Soit K_p la *clé publique*
- Soit K_s la *clé privée*
- Soit E la fonction de chiffrement, on a :
$$c = E(K_p, m)$$
- Soit D la fonction de déchiffrement, on a
facilement : $m = D(K_s, c)$
mais difficilement : $m = D(K_p, c)$

Cryptographie asymétrique “à clé publique”



Cryptographie asymétrique “à clé publique”

Quelques algorithmes de cryptographie à clé publique :

- RSA Algorithm
- Digital Signature Algorithm
- Cramer – Shoup



Cryptographie asymétrique “à clé publique”



Moi, je vois quelques problèmes dans ce processus..

Lesquels ??

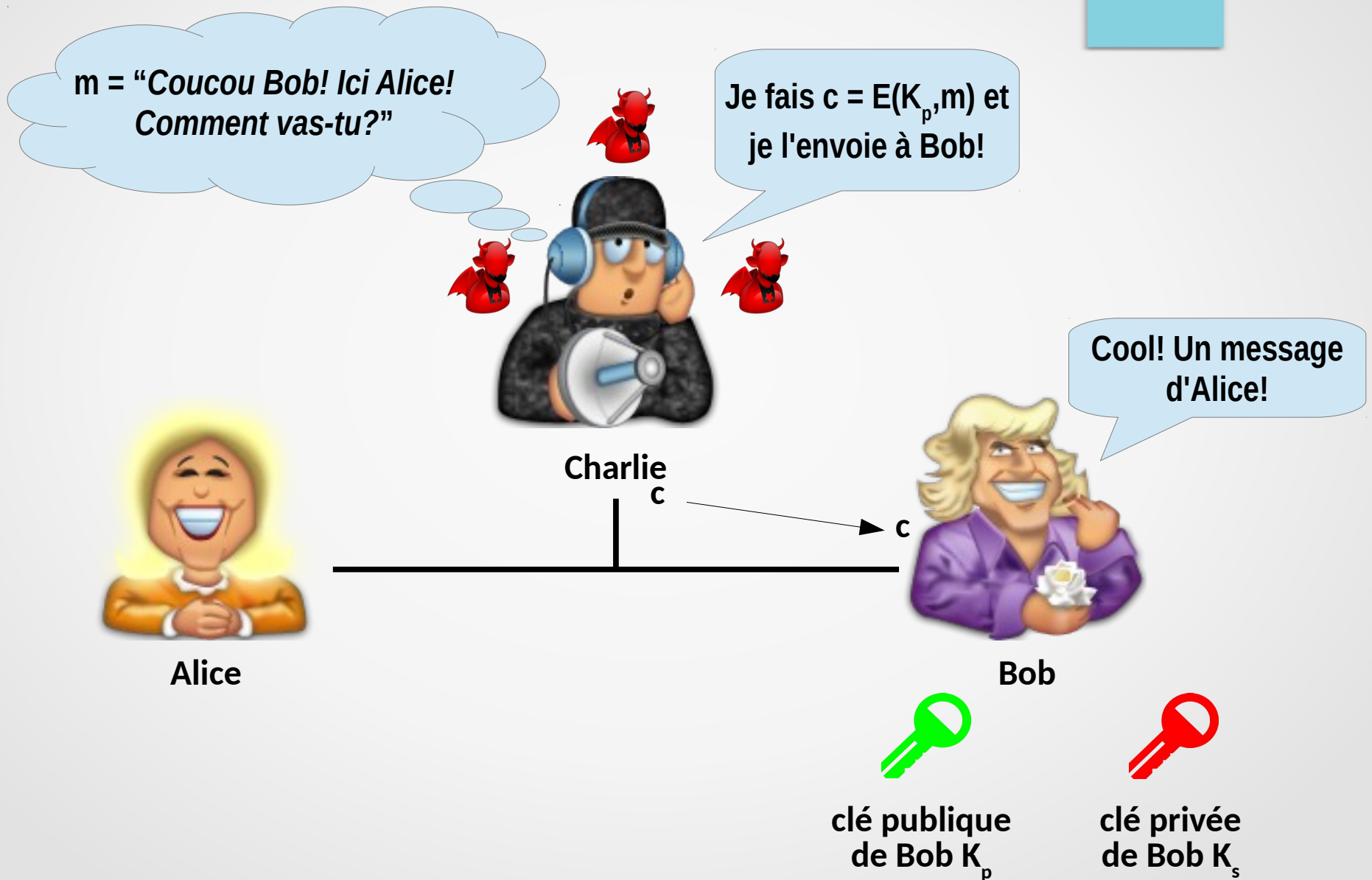


Cryptographie asymétrique : signature numérique




Puisque Bob a publié sa clé publique,
je vais lui écrire en me faisant passer
pour Alice !

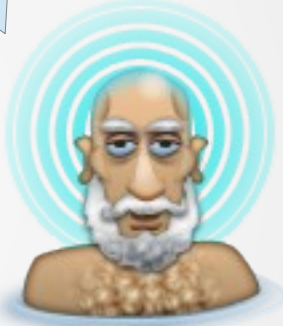
Cryptographie asymétrique : signature numérique



Cryptographie asymétrique : signature numérique



Tout le monde peut m'envoyer des messages chiffrés maintenant, comment je peux être sûr que celui-ci vient bien d'Alice ???



Demande lui de signer ses messages



Comment faire ?

Fonctions de hachage



- Une fonction de hachage prend en entrée un objet, par exemple un texte, et produit en sortie une empreinte ou condensat (ou digest, ou hash) de cet objet.
- Les objets en entrée sont de tailles variables mais les empreintes ont toutes la même longueur.

En connaissez-vous?



Exemples de fonctions de hachage : MD5

- MD5 (Message Digest 5)
- Publiée en avril 1992 (Ron Rivest)
- Produit des empreintes de 128 bits, typiquement présentées sous la forme d'une série de 32 caractères hexadécimaux



Exemples de fonctions de hachage : SHA-1

- SHA-1 (Secure Hashing Algorithm)
- Publiée en 1995 (National Security Agency)
- Produit des empreintes de 160 bits, typiquement présentées sous la forme d'une série de 40 caractères hexadécimaux



Exemples de fonctions de hachage : SHA-256

- SHA-256 (famille SHA-2)
- Publiée en 2001 (National Security Agency)
- Produit des empreintes de 256 bits, typiquement présentées sous la forme d'une série de 64 caractères hexadécimaux

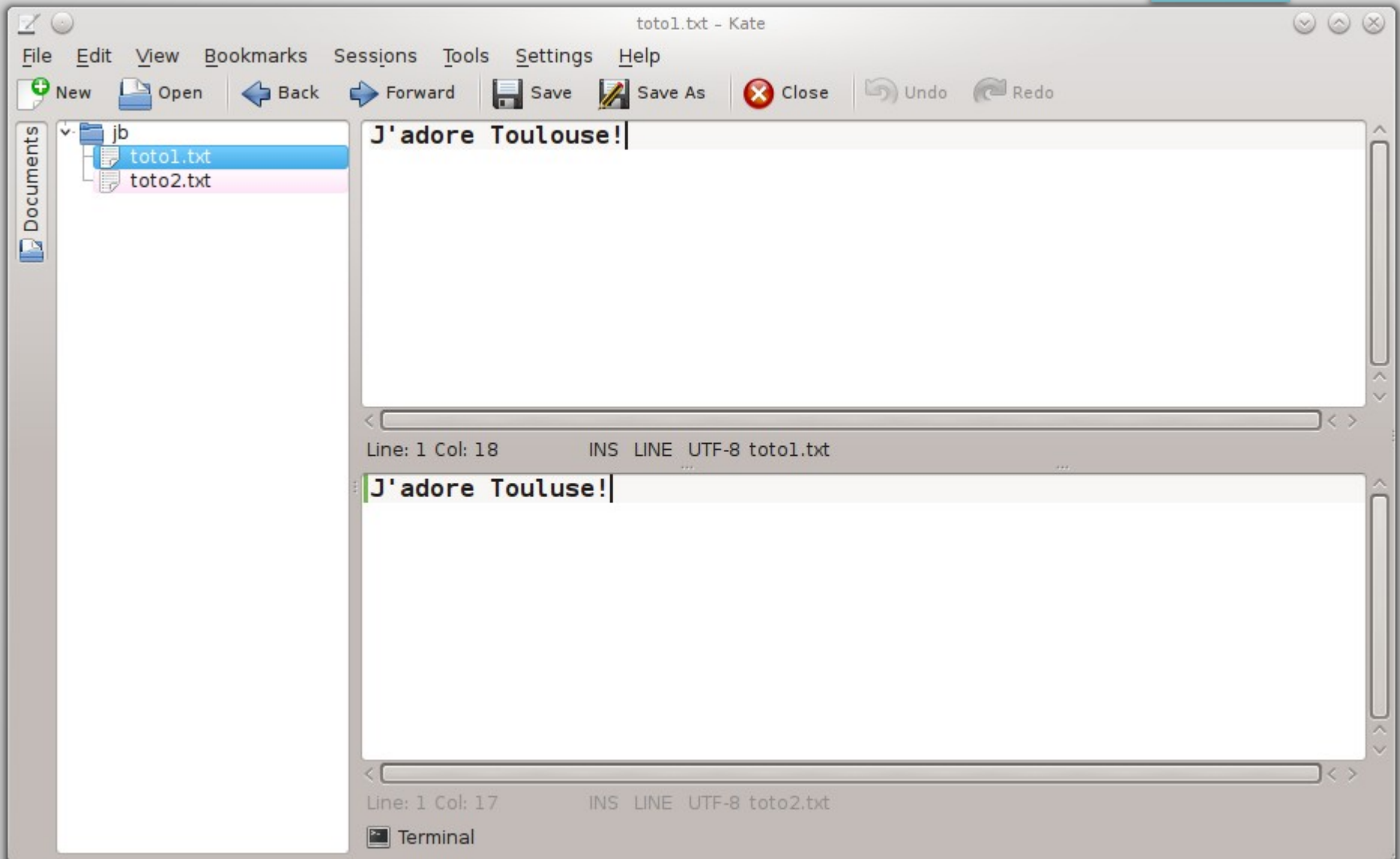


Exemples de fonctions de hachage : SHA-3

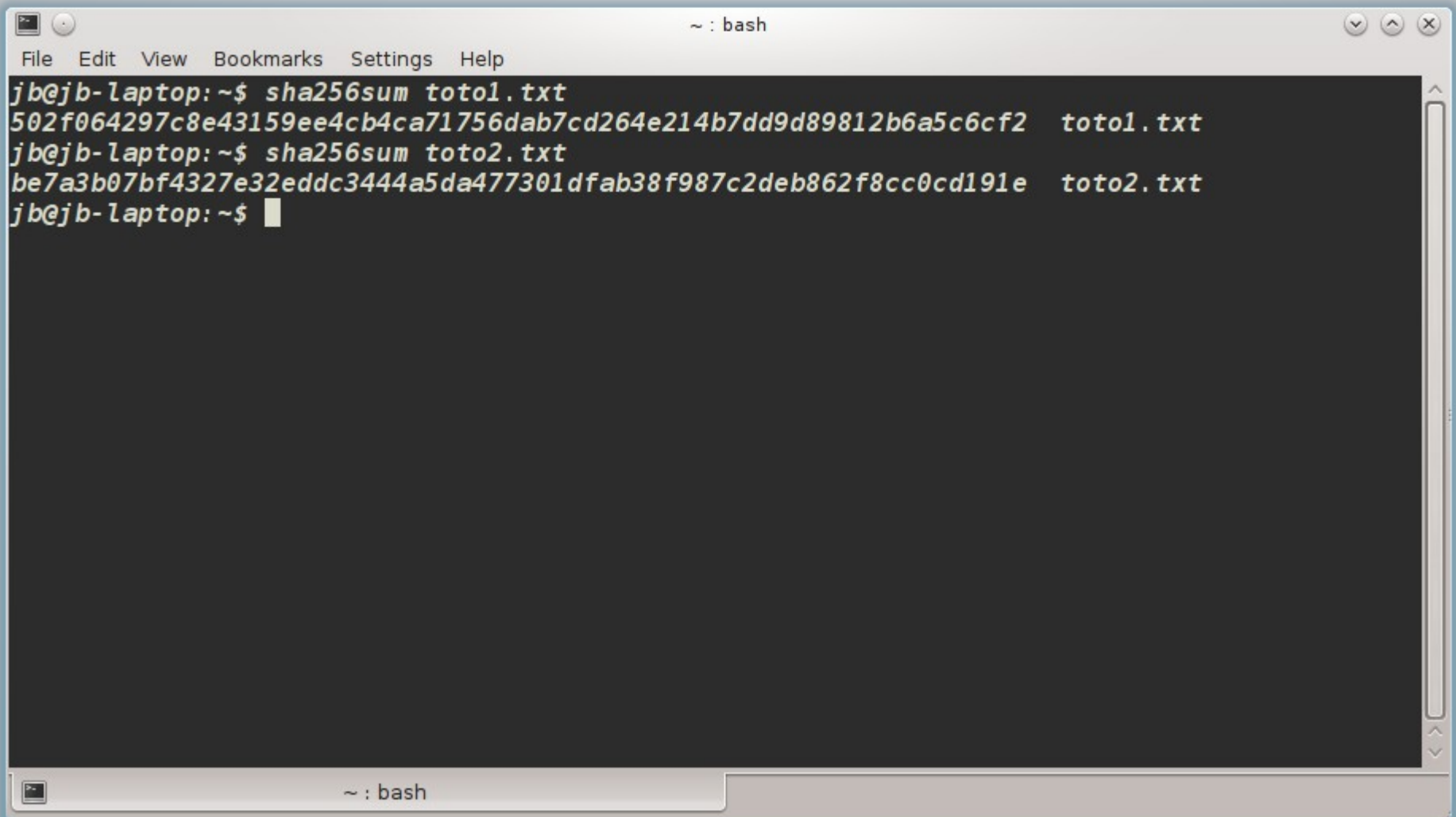


- SHA-3 (également connu comme Keccak)
- Publiée en 2012 (Bertoni, Daemen, Peeters, Van Assche)
- Vainqueur d'un concours du NIST
- Produit des empreintes de 1600 bits

Exemples de fonctions de hachage : SHA-256



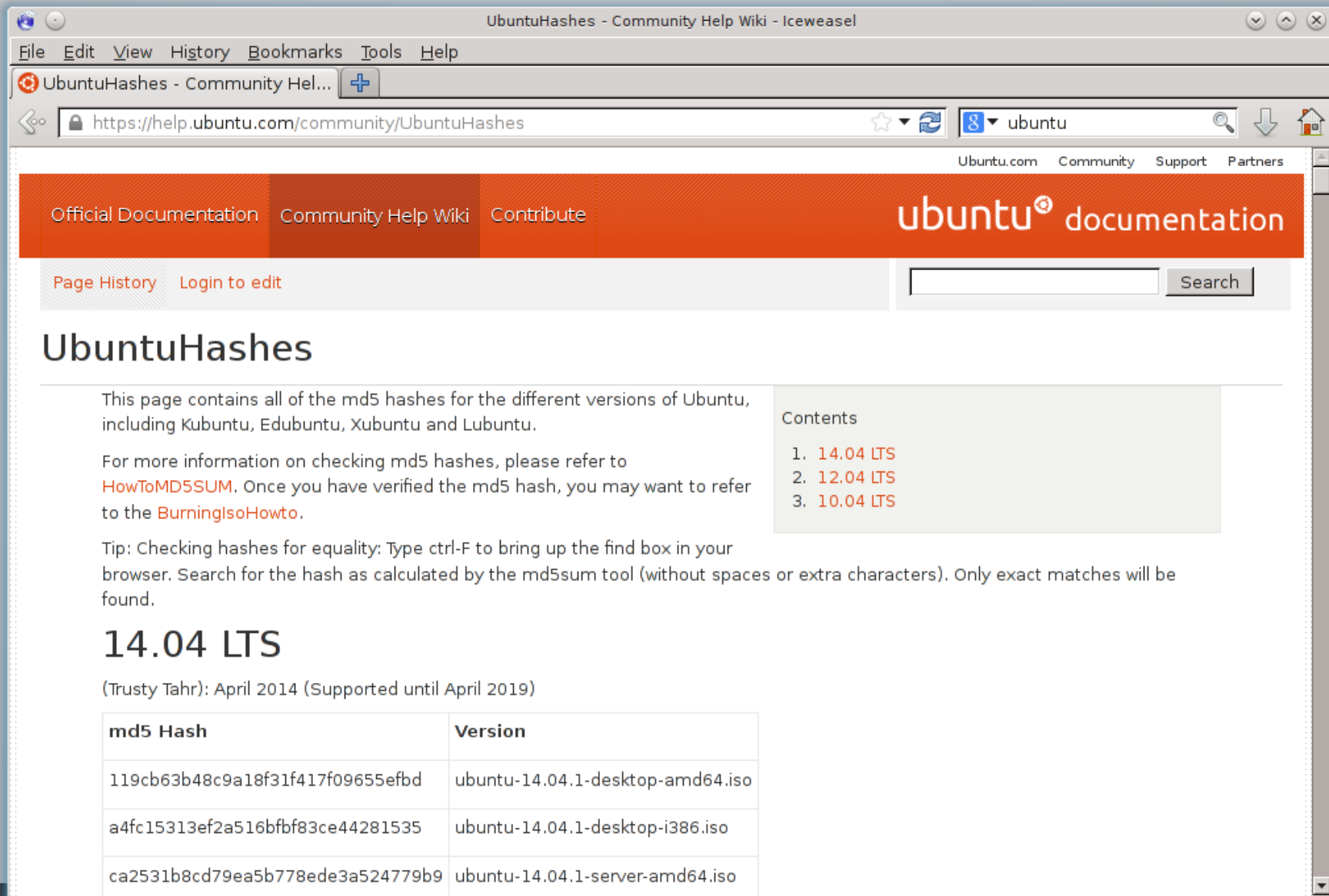
Exemples de fonctions de hachage : SHA-256



```
~ : bash
File Edit View Bookmarks Settings Help
jb@jb-laptop:~$ sha256sum toto1.txt
502f064297c8e43159ee4cb4ca71756dab7cd264e214b7dd9d89812b6a5c6cf2  toto1.txt
jb@jb-laptop:~$ sha256sum toto2.txt
be7a3b07bf4327e32eddc3444a5da477301dfab38f987c2deb862f8cc0cd191e  toto2.txt
jb@jb-laptop:~$ █
```

The image shows a terminal window with a menu bar (File, Edit, View, Bookmarks, Settings, Help) and a title bar (~ : bash). The terminal content shows two successful executions of the sha256sum command. The first command hashes 'toto1.txt' and produces a 64-character hexadecimal hash. The second command hashes 'toto2.txt' and produces a different 64-character hexadecimal hash. The prompt is followed by a cursor (█) on the third line.

Exemples de fonctions de hachage : MD-5



The screenshot shows a web browser window displaying the UbuntuHashes - Community Help Wiki page. The browser's address bar shows the URL <https://help.ubuntu.com/community/UbuntuHashes>. The page features a navigation bar with links for "Official Documentation", "Community Help Wiki", and "Contribute". The main content area is titled "UbuntuHashes" and contains the following text:

This page contains all of the md5 hashes for the different versions of Ubuntu, including Kubuntu, Edubuntu, Xubuntu and Lubuntu.

For more information on checking md5 hashes, please refer to [HowToMD5SUM](#). Once you have verified the md5 hash, you may want to refer to the [BurningIsoHowto](#).

Tip: Checking hashes for equality: Type ctrl-F to bring up the find box in your browser. Search for the hash as calculated by the md5sum tool (without spaces or extra characters). Only exact matches will be found.

14.04 LTS

(Trusty Tahr): April 2014 (Supported until April 2019)

| md5 Hash | Version |
|----------------------------------|----------------------------------|
| 119cb63b48c9a18f31f417f09655efbd | ubuntu-14.04.1-desktop-amd64.iso |
| a4fc15313ef2a516bfbf83ce44281535 | ubuntu-14.04.1-desktop-i386.iso |
| ca2531b8cd79ea5b778ede3a524779b9 | ubuntu-14.04.1-server-amd64.iso |

On the right side of the page, there is a "Contents" section with a list of links:

- [1. 14.04 LTS](#)
- [2. 12.04 LTS](#)
- [3. 10.04 LTS](#)

Fonctions de hachage



Une bonne fonction de hachage pour la cryptographie doit avoir les propriétés suivantes :

- Un changement minime sur l'objet en entrée résulte en une empreinte très différente
- Il est très difficile d'obtenir des informations sur l'objet en entrée à partir de l'empreinte
- MD-5 et SHA-1 sont considérées non fiables
- Aucune attaque connue à ce jour sur SHA-256

Cryptographie asymétrique : signature numérique



Le principe de la signature est de prouver qu'un message a été émis par une personne et n'a pas été modifié après avoir été signé.

Cryptographie asymétrique : signature numérique



Alice



clé publique
d'Alice K_p^{Alice}



clé privée
d'Alice K_s^{Alice}

- Pour signer le message Alice va :
 - générer une empreinte de son message : $h = \text{sha256}(m)$
 - chiffrer l'empreinte avec sa clé privée :
 $S = E(K_s^{Alice}, h)$
 - envoyer S à Bob en même temps que son message chiffré (ou non)

S est la signature !

Cryptographie asymétrique : signature numérique



Alice



clé publique
d'Alice K_p^{Alice}



clé privée
d'Alice K_s^{Alice}

- Pour signer le message Alice va :
 - générer une empreinte de son message : $h = \text{sha256}(m)$
 - chiffrer l'empreinte avec sa clé privée :
 $S = E(K_s^{Alice}, h)$
 - envoyer S à Bob en même temps que son message chiffré (ou non)

S est la signature !

Cryptographie asymétrique : signature numérique



Bob

- Pour vérifier la signature Bob va :
 - récupérer le message m' (en le déchiffrant si nécessaire)
 - récupérer la signature S
 - générer une empreinte de son message : $h' = \text{sha256}(m')$
 - déchiffrer la signature avec la clé publique d'Alice : $V = D(K_p^{\text{Alice}}, S)$
 - comparer h' et V . Si $h' = V$, c'est à dire $\text{sha256}(m') = D(K_p^{\text{Alice}}, E(K_s^{\text{Alice}}, \text{sha256}(m)))$, alors Bob est *en pratique* sûr que $m' = m$

Cryptographie asymétrique : signature numérique



- Il se pourrait que $m \neq m'$ et $\text{sha256}(m) = \text{sha256}(m')$ mais la probabilité est très faible.
- Avec une bonne fonction de hachage cryptographique et sans la clé privée d'Alice, il est impossible à Charlie de remplacer à la fois m par m' et S par $S' = E(K_s^{\text{Alice}}, \text{sha256}(m'))$.
- Connaissant K_p^{Alice} , Charlie peut retrouver facilement $\text{sha256}(m)$ en faisant $D(K_p^{\text{Alice}}, S)$, mais il ne peut remonter à m .

Cryptographie asymétrique : signature numérique



La clé privée étant nécessaire pour signer un message, Charlie peut toujours modifier le message, mais il ne peut pas le resigner !

Cryptographie asymétrique : signature numérique



C'est ce mécanisme qui est utilisé pour réaliser *l'authentification*

Que savez-vous de l'authentification ?



Cryptographie asymétrique : En résumé



Pour écrire un message chiffré à Bob,
Alice utilise :

la clé publique de Bob

Pour déchiffrer un message reçu d'Alice,
Bob utilise :

la clé privée de Bob



Normal, je veux que tout le monde puisse m'écrire
mais être le seul à déchiffrer mes messages !

Cryptographie asymétrique : En résumé



Pour signer un message, Alice utilise :
la clé privée d'Alice (+ hachage)

Pour vérifier une signature d'Alice, Bob utilise:
la clé publique d'Alice (+ hachage)



Normal, je signe avec quelque chose qui n'appartient qu'à moi seule,
mais tout le monde doit pouvoir vérifier ma signature !

Cryptographie asymétrique “à clé publique”



Moi, je vois toujours quelques problèmes dans ce processus..

Lesquels ??



Cryptographie asymétrique “à clé publique”

Je chiffre mes messages à Bob
avec sa clé publique K_p^{Bob} ,
mais est-ce vraiment la sienne?



Alice

Je vérifie les messages signés
d'Alice avec sa clé publique K_p^{Alice} ,
mais est-ce vraiment la sienne?



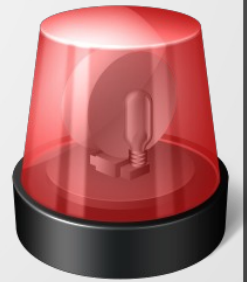
Bob

Cryptographie asymétrique “à clé publique”



Je vais envoyer ma clé publique K_p^{Charlie} à Alice et Bob en leur faisant croire à chacun que c'est la clé publique de l'autre. Je pourrais donc déchiffrer leur messages et leur envoyer des faux messages signés avec ma clé privée K_s^{Charlie} !

C'est l'attaque de l'homme du milieu!!!



Les certificats électroniques



Les certificats électroniques permettent d'associer de façon fiable une **clé publique** à une **entité** (personne, entreprise, serveur...)



Les certificats électroniques

Un certificat électronique est un fichier signé qui contient essentiellement :

- des informations sur l'entité
- la clé publique de l'entité

plus :

- l'identifiant de l'émetteur
- une date de validité
- un numéro de série

et surtout :

- la signature de l'émetteur



Les certificats électroniques



Alice

Voici mon certificat !

Nom : Alice

Clé publique : 24E1F2893A

Emis par : SuperSign Corp.

Valide jusqu'au : 23/09/2024

N° de série : 56882442

Signature :



SuperSign Corp.

Les certificats électroniques



Une minute! J'ai un certificat certifiant la clé publique d'Alice, signé par la clé privée de SuperSign Corp.

Je vais vérifier cette signature avec la clé publique de SuperSign Corp. Mais le problème recommence!

Qui me garantit que j'ai bien la clé publique de SuperSign Corp. ?
Il me faut le certificat de SuperSign Corp !!!

Les certificats électroniques

Et voilà le certificat
de SuperSign Corp. !



Nom : SuperSign Corp.

Clé publique : 962BF283

Emis par : SuperSign Corp.

Valide jusqu'au : 23/09/2024

N° de série : 52598214

Signature :



SuperSign Corp.

Les certificats électroniques

- On obtient un chaîne de certificats !
- Le certificat de SuperSign Corp. est auto-signé !

Nom : Alice

Clé publique :
24E1F2893A

Emis par : SuperSign Corp.

Valide jusqu'au :
23/09/2024

N° de série : 568



Signature :

SuperSign Corp.

Nom : SuperSign Corp.

Clé publique : 962BF283

Emis par : SuperSign
Corp.

Valide jusqu'au :
23/09/2024

N° de série : 525



Signature :

SuperSign Corp.

Les certificats électroniques



**SuperSign Corp. auto-signé ??
Ca veut dire que je suis supposé lui faire confiance ?
Sur parole ?**

Les certificats électroniques



SuperSign Corp. auto-signé ??
Ca veut dire que je suis supposé lui faire confiance ?
Sur parole ?

Enchanté ! Je travaille pour SuperSign Corp.
Nous sommes une autorité de certification.
Nous sommes connus et fiables,
tellement fiables que nous figurons
par défaut dans tous les navigateurs web!



SuperSign Corp.

Les certificats électroniques : en résumé



Un certificat électronique certifie une clé publique, il est signé par (la clé privée de) l'émetteur, typiquement une autorité de certification de confiance.

Les certificats électroniques : en résumé



Une clé publique est sans aucune utilité si elle n'est pas associée de façon sûre à son propriétaire par un certificat.

En pratique, on ne rencontre jamais une clé publique sans son certificat.

Les certificats électroniques : en résumé

Les clés publiques sont tellement fortement associées au certificats qu'on dit souvent :



- “Chiffré avec le certificat de Bob” pour dire “Chiffré avec la clé publique qui se trouve dans le certificat de Bob”.
- “Déchiffré avec le certificat de Bob” pour dire “Déchiffré avec la clé privée associée à la clé publique qui se trouve dans le certificat de Bob”.

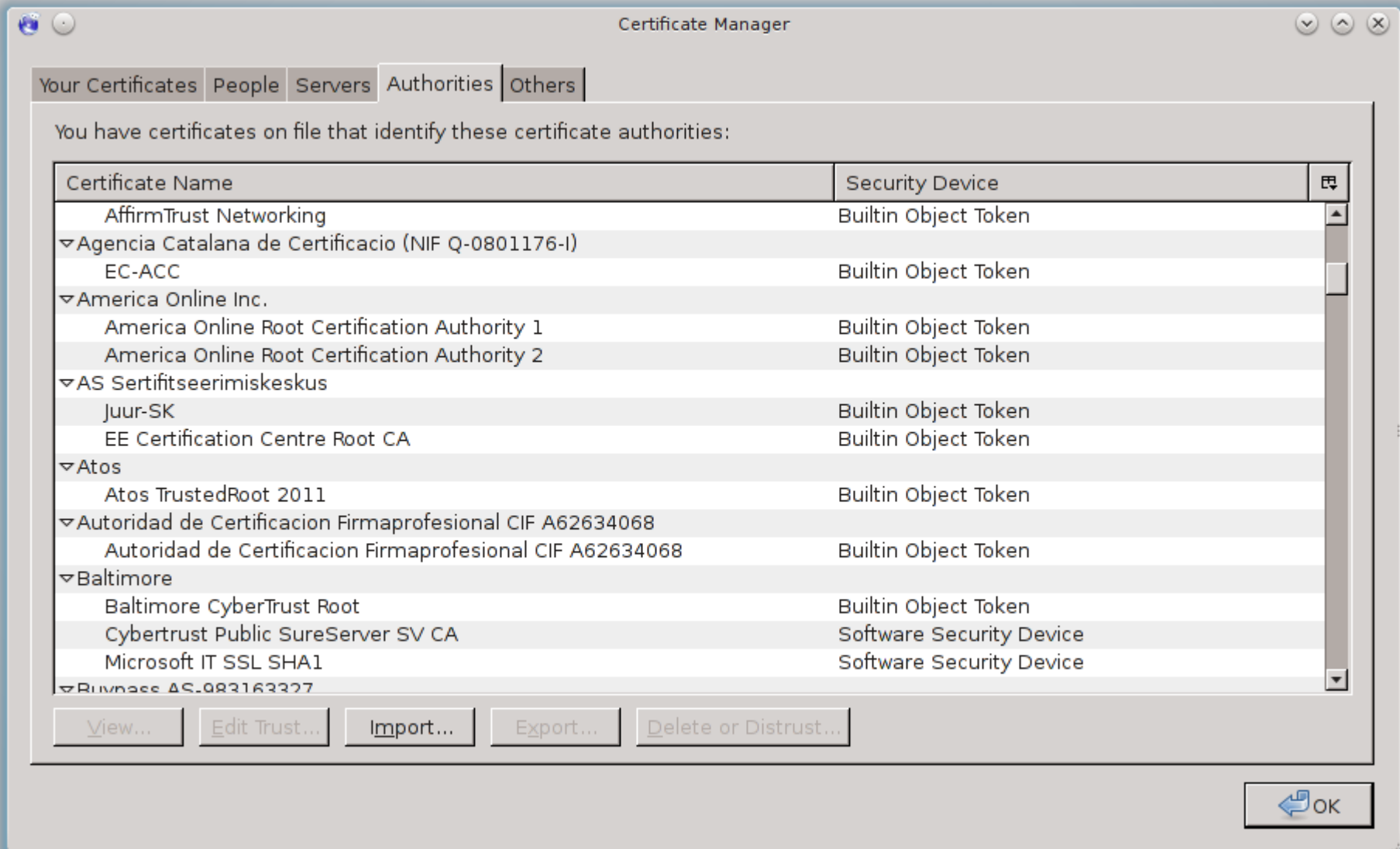
Les certificats électroniques : en résumé

Les clés publiques sont tellement fortement associées au certificats qu'on dit souvent :



- “*Signé avec le certificat d'Alice*” pour dire “*Signé avec la clé privée associée à la clé publique qui se trouve dans le certificat d'Alice*”.
- “*Signature vérifiée avec le certificat d'Alice*” pour dire “*Signature vérifiée avec la clé publique qui se trouve dans le certificat d'Alice*”.

Les certificats électroniques



Les certificats électroniques

En pratique, l'autorité de certification sépare souvent la fonction racine (*root*) de la fonction signante (*signing*)

Nom : Alice

Clé publique : 24E1F2893A

Emis par : SuperSign Signing CA

Valide jusqu'au : 23/09/2024

N° de série : 56882442

Signature :



*SuperSign Signing
CA*

Nom : SuperSign Signing CA

Clé publique : 962BF283

Emis par : SuperSign Root CA

Valide jusqu'au : 23/09/2024

N° de série : 52598214

Signature :



SuperSign Root CA

Nom : SuperSign Root CA

Clé publique : 962BUF43

Emis par : SuperSign Root CA

Valide jusqu'au : 23/09/2024

N° de série : 52598814

Signature :



SuperSign Root CA

Création d'une autorité de certification

- Nous utilisons l'outil OpenSSL
- <https://www.openssl.org/>

A screenshot of a web browser displaying the OpenSSL website. The browser's address bar shows the URL <https://www.openssl.org/>. The page features the OpenSSL logo at the top, followed by navigation links such as "Sponsor OpenSSL", "Purchase a Support Contract", "Contract a Team Member", and "Our Sp...". A sidebar on the left contains a list of menu items: "Title", "FAQ", "About", "News", "Documents", "Source", "Support", "Related", and "Security". The main content area has a heading "Welcome to the OpenSSL Project" and a paragraph of introductory text. Below this, there is a section titled "Date" and "Newsflash" with a sub-heading "06-Aug-2014: Security Advisory: nine security fixes".

OpenSSL: The Open Source toolkit for SSL/TLS - Icew

File Edit View History Bookmarks Tools Help

OpenSSL: The Open Source tool... +

https://www.openssl.org

OpenSSL

Cryptography and SSL/TLS Toolkit

Sponsor OpenSSL | Purchase a Support Contract | Contract a Team Member | Our Sp...

Title

FAQ

About

News

Documents

Source

Support

Related

Security

Welcome to the OpenSSL Project

The OpenSSL Project is a collaborative effort to develop a robust, commercial-implementing the **Secure Sockets Layer (SSL v2/v3)** and **Transport Layer S** general purpose cryptography library. The project is managed by a worldwide communicate, plan, and develop the OpenSSL toolkit and its related document

OpenSSL is based on the excellent SSLeay library developed by Eric Young an an Apache-style licence, which basically means that you are free to get and us subject to some simple license conditions.

| Date | Newsflash |
|--------------|--|
| 06-Aug-2014: | Security Advisory: nine security fixes |

Création d'une autorité de certification



- La première étape consiste à :
 - Générer la clé privée de l'autorité racine
 - Générer la clé publique de l'autorité racine
 - Générer la *Certificate Signing Request* de l'autorité racine

Création d'une autorité de certification



```
rootCA.conf - Kate
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
  rootCA
    rootCA.conf
distinguished_name = ca_dn >> # DN section
req_extensions     = ca_reqext >> # Desired extensions

[ ca_dn ]
0.domainComponent = "home.lan"
1.domainComponent = "supersign"
organizationName  = "SuperSign Corp."
commonName        = "SuperSign Root Certificate Authority"

[ ca_reqext ]
keyUsage           = critical,keyCertSign,cRLSign
basicConstraints   = critical,CA:true
subjectKeyIdentifier = hash

# The remainder of the configuration file is used by the openssl ca command.
# The CA section defines the locations of CA assets, as well as the policies
# applying to the CA.

[ ca ]
default_ca         = root_ca >> # The default CA

[ root_ca ]
certificate         = $dir/$ca/$ca.crt >> # The CA certificate
private_key         = $dir/$ca/private/$ca.key >> # CA private key
new_certs_dir       = $dir/$ca/certs >> # Certificate directory
serial             = $dir/$ca/db/crt_serial >> # Serial number

Line: 1 Col: 1      INS LINE UTF-8 rootCA.conf
Terminal
```

Création d'une autorité de certification

- Cela se fait en une seule étape avec OpenSSL



```
~/superSignCA/rootCA : bash
File Edit View Bookmarks Settings Help
jbb@jbb-laptop: ~/superSignCA/rootCA$ openssl req -new -config rootCA.conf -out rootCA.csr
-keyout private/rootCA.key
Generating a 2048 bit RSA private key
.....+++
...+++
writing new private key to 'private/rootCA.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
jbb@jbb-laptop: ~/superSignCA/rootCA$
```


Création d'une autorité de certification

- On obtient ainsi la CSR rootCA.csr ainsi que la clé privée rootCA.key (2048 bits)



```
rootCA.key - Kate
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
private
  rootCA.key
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQI6f0eui/oAJwCaggA
MBQGCCqGSIb3DQMHBAiQVJn8l1E9QSCBMij9WP4Lw+NAY1bKKL++nEvvvjAAu2A
SsoV4pt3xPmtGIfw9GZymcWbcNwZXTdM2EJ6PH7TltLDqsnwKyI3RftvCs6hWrIM
kWetqlrHwleCNOQL3mW8+dPSeMERHu/t0QjgSB+EpTxfvlgTbPmkdaj1rZMb+PLg
SH03o6EjaILVsvINSKEPxJ42g6t/wFvCPKqlmNqcelr3cZkYCqPUjykvb/Xzy2SK
pjdVYwZhmZEfphbjtFxovtiDB23uypKCG+pK/o97RMQE95SNpmiD4n+CvTAUDngq
u6XbX1WYcF22Fo+CSZR0STZTwXaf4TCmlnIyvjP5d8tLVULiHtcGasXVKNWID3yb
6otwR9rqM67LpgxLntkzxnEIIIfC1YptdAgooULg/qMYMzzZ8z02JSHE/KIb/F0Vc
+VHJRD1YUglwjgcYmuCeaYRW0WtPAGR93k1/Y3IYKs3hW5d07zumPxxbNU5yQrd2
ZK6ql5NbZvN8UmdCLAbTR+cj0u2N+1ESVrTAIW5HjKKenPTjlc0UqaRas2bk3Bhg
zI0mAVfFhRtJnkA4Un+0umW7Dy50TFT2PEbYPAHISV93qWn+b/fgwuZn2ndKtxkb
AmQiyVrxnyxD0L239br14zdNVy0Aapbvq6eQZFa/H0C7c3STfr9h50aFKGNVlqTw
M91vv2sKr0ul2shHvV4u0Jd8vrNa5jhr5dvlal7LG2LApFtWqy4VhmjPeouFTBrLT
lr3zCADC2LIYXT3GMt6nycb+aUQy9dkgUghMXCAAD9Cv1yiICUxC8mxSCRliHqst
T42MHxDPCvX7Z2RVL2yN5eq9u3w8GQBqpj9/qSDfEXKkiUUxwFklU82FNqag5KYY
tCK593o4U96o15cc8ExHoMyZZ7ZeAzDzdnZTGaSKKSKLxPqRDUUp8FrGMdbN07g1
3sE3bplAL6qT4jx9BfjSm/sn1kgDdHpNsQnw8B0USgRo0dn5FY89nbbYRpZ5D6dF
YfA+nGvWMu7KUchr3t9LugYYkKZMcZ/2QP5MWu+iv1ZsfJRYSFic0ni4PgpKdFI
Y150HSJ0eC42UfDgPb5tQTFWDD2ef8/uoTperM/QE0amljinTbFnDC930kB0zu3
NbtE7kVAxl2a+drRMuM//RF7qSshqHLXimy09xwo2JWu/NIDr9t73TJ0ZTvGjr8U
iEVunYWUX2ytYTWY/9xdh73ha+kC0BDC1rnlBKatHdcPFA4NXLNWCEM7giN33DPf
+hVTacf+cAwqNkTHs0WS+dcndwFqYGMke8yWLNgzThyGCvv0qsjaqL8rIX3vZiRS
Ng8K6f4Z16dIy/FKHB5Z55FHvJl79yxM0wsMZQ+QaGYGGemEeTl7Z7qH22q0rwQ0
0D/+6I3VU67P8k6drMnpk8PeY44599lxHbkqVBWLwQ0n9BvnbCQ9V3a+FCu2xNn
np3h27bsZcihwj9eYVKii83J54l1LnpM3Xcd1xxMlYxhmeQQS8IY/T6VUNHqLrrh
hHzZeIJPya8K45UtuzHF2bHh4EaocTgNFbfj/Pi8IDGkjH/0C18TlZq3WouM0raK
m2PauLgCyoM4vyx5BFeId4+J+rp09Qe+LD8pj6leCVDRJonZXmS3rArt6nELXqM
QtE=
-----END ENCRYPTED PRIVATE KEY-----
Line: 1 Col: 1      INS LINE UTF-8 rootCA.key
Terminal
```

Création d'une autorité de certification

- Maintenant nous allons signer la CSR au moyen de la clé privée rootCA.key



```
~/superSignCA/rootCA : bash
File Edit View Bookmarks Settings Help
jbb@jbb-laptop: ~/superSignCA/rootCA$ openssl ca -selfsign -config rootCA.conf -in rootCA.csr
-out rootCA.crt -extensions root_ca_ext
Using configuration from rootCA.conf
Enter pass phrase for /home/jbb/superSignCA/rootCA/private/rootCA.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 1 (0x1)
  Validity
    Not Before: Sep 15 14:28:55 2014 GMT
    Not After : Sep 14 14:28:55 2024 GMT
  Subject:
    domainComponent           = home.lan
    domainComponent           = supersign
    organizationName          = SuperSign Corp.
    commonName                 = SuperSign Root Certificate Authority
  X509v3 extensions:
    X509v3 Key Usage: critical
      Certificate Sign, CRL Sign
    X509v3 Basic Constraints: critical
```

Création d'une autorité de certification



- Et on obtient notre certificat d'autorité racine auto-signé!

Création d'une autorité de certification

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1 (0x1)

Signature Algorithm: sha1WithRSAEncryption

Issuer: DC=home.lan, DC=supersign, O=SuperSign Corp., CN=SuperSign Root Certificate Authority

Validity

Not Before: Sep 15 14:28:55 2014 GMT

Not After : Sep 14 14:28:55 2024 GMT

Subject: DC=home.lan, DC=supersign, O=SuperSign Corp., CN=SuperSign Root Certificate Authority

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

00:c6:a3:30:1c:d8:c3:f5:5a:39:07:e9:82:45:b1:
ae:e2:e9:51:fd:38:b0:08:f1:0c:ed:b4:be:f9:
c0:bf:f1:23:33:78:78:19:ee:d2:60:6a:8b:a7:
ac:ef:6a:a7:46:9b:cc:1f:86:be:83:01:7f:
8e:fb:74:ac:67:ea:c8:e3:4b:88:f2:e3:8:
74:6b:27:2f:49:da:c6:5a:25:4d:3f:07:8:
17:83:7b:67:53:ac:92:60:80:87:b4:20:22:c5:eb:
0e:22:35:c7:4d:4b:04:54:25:39:fd:1b:8e:42:91:
0b:ca:46:8c:a3:28:16:58:e1:12:9f:e6:eb:3c:d3:
19:e5:1f:91:f1:59:22:22:15:88:05:fa:6b:cc:18:
f0:87:ee:4e:09:05:1a:1e:0c:26:72:14:00:ef:9b:
0e:92:fe:1a:dd:72:e3:02:dc:21:18:7f:4c:e1:1f:
ff:8a:50:96:a4:45:c7:54:f8:1c:56:78:31:83:a5:
92:48:54:35:cc:98:82:10:d8:73:c4:a9:e3:6d:1e:
d2:f0:60:c9:96:db:46:f5:c3:2b:d8:68:cd:4a:47:
d9:ed:34:33:36:04:38:f9:28:01:1b:4c:f0:95:a5:
01:0c:4c:34:18:a2:64:b1:03:62:b5:25:f4:dd:2f:
f1:db

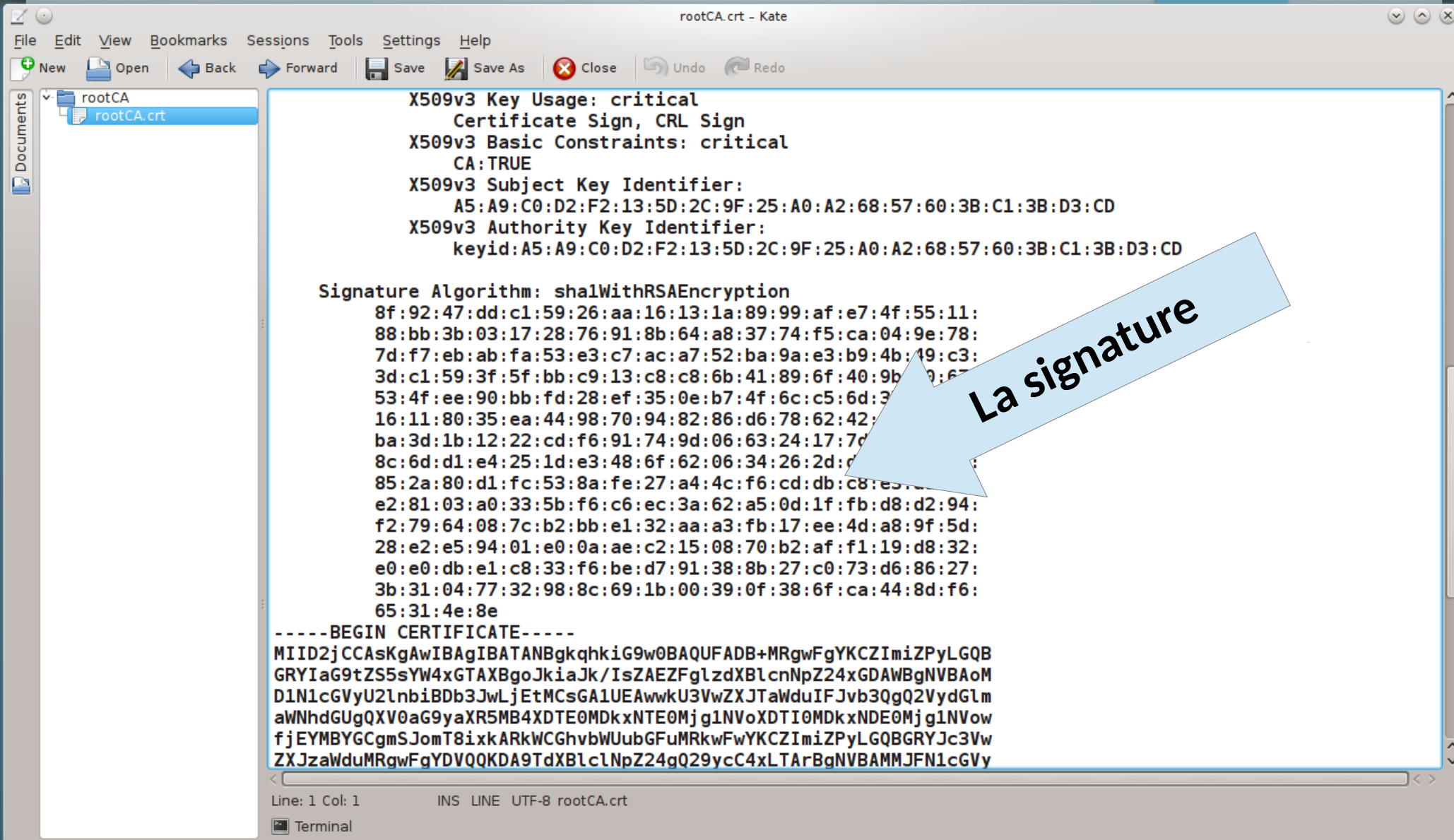
La clé publique

Line: 1 Col: 1

INS LINE UTF-8 rootCA.crt

Terminal

Création d'une autorité de certification



```
rootCA.crt - Kate
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
rootCA
rootCA.crt

X509v3 Key Usage: critical
Certificate Sign, CRL Sign
X509v3 Basic Constraints: critical
CA:TRUE
X509v3 Subject Key Identifier:
A5:A9:C0:D2:F2:13:5D:2C:9F:25:A0:A2:68:57:60:3B:C1:3B:D3:CD
X509v3 Authority Key Identifier:
keyid:A5:A9:C0:D2:F2:13:5D:2C:9F:25:A0:A2:68:57:60:3B:C1:3B:D3:CD

Signature Algorithm: sha1WithRSAEncryption
8f:92:47:dd:c1:59:26:aa:16:13:1a:89:99:af:e7:4f:55:11:
88:bb:3b:03:17:28:76:91:8b:64:a8:37:74:f5:ca:04:9e:78:
7d:f7:eb:ab:fa:53:e3:c7:ac:a7:52:ba:9a:e3:b9:4b:19:c3:
3d:c1:59:3f:5f:bb:c9:13:c8:c8:6b:41:89:6f:40:9b:09:67:
53:4f:ee:90:bb:fd:28:ef:35:0e:b7:4f:6c:c5:6d:30:
16:11:80:35:ea:44:98:70:94:82:86:d6:78:62:42:
ba:3d:1b:12:22:cd:f6:91:74:9d:06:63:24:17:7d:
8c:6d:d1:e4:25:1d:e3:48:6f:62:06:34:26:2d:
85:2a:80:d1:fc:53:8a:fe:27:a4:4c:f6:cd:db:c8:e3:
e2:81:03:a0:33:5b:f6:c6:ec:3a:62:a5:0d:1f:fb:d8:d2:94:
f2:79:64:08:7c:b2:bb:e1:32:aa:a3:fb:17:ee:4d:a8:9f:5d:
28:e2:e5:94:01:e0:0a:ae:c2:15:08:70:b2:af:f1:19:d8:32:
e0:e0:db:e1:c8:33:f6:be:d7:91:38:8b:27:c0:73:d6:86:27:
3b:31:04:77:32:98:8c:69:1b:00:39:0f:38:6f:ca:44:8d:f6:
65:31:4e:8e

-----BEGIN CERTIFICATE-----
MIID2jCCAsKgAwIBAgIBATANBgkqhkiG9w0BAQUFAADB+MRgwFgYK CZImiZPyLGQB
GRYIaG9tZS5sYW4xGTAXBgoJkiaJk/IsZAEZFglzdXB1cnNpZ24xGDAWBGNVBAoM
D1NlcGVyU2lnbiBDb3JwLjEtMCSGA1UEAwkU3VwZXJTaWduIFJvb3QgQ2VydGlm
aWNhdGUgQXV0aG9yaXR5MB4XDTE0MDkxNTE0Mjg1NVVoXDTI0MDkxNDE0Mjg1NVVow
fjEYMBYGCgmSjomT8ixkArkWCghvbWUubGFuMRkwFwYK CZImiZPyLGQBGRYJc3Vw
ZXJzaWduMRgwFgYDVQKDA9TdXB1cnNpZ24xGDAwQ29ycC4xLTAhBgNVBAMMJFN1cGVy
```

Line: 1 Col: 1 INS LINE UTF-8 rootCA.crt

Terminal

Création d'une autorité de certification



- La deuxième étape consiste à :
 - Générer la clé privée de l'autorité signante
 - Générer la clé publique de l'autorité signante
 - Générer la *Certificate Signing Request* de l'autorité signante

Création d'une autorité de certification



```
signingCA.conf - Kate
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
  signingCA
  signingCA.conf
distinguished_name = ca_dn # DN section
req_extensions = ca_reqext # Desired extensions

[ ca_dn ]
0.domainComponent = "home.lan"
1.domainComponent = "supersign"
organizationName = "SuperSign Corp."
commonName = "SuperSign Signing Certificate Authority"

[ ca_reqext ]
keyUsage = critical,keyCertSign,cRLSign
basicConstraints = critical,CA:true,pathlen:0
subjectKeyIdentifier = hash

# The remainder of the configuration file is used by the openssl ca
# The CA section defines the locations of CA assets, as well as the
# applying to the CA.

[ ca ]
default_ca = signing_ca # The default CA section

[ signing_ca ]
certificate = $dir/$ca/$ca.crt # The CA cert
private_key = $dir/$ca/private/$ca.key # CA private key
new_certs_dir = $dir/$ca/certs # Certificate directory
serial = $dir/$ca/db/crt_serial # Serial number file

Line: 29 Col: 67 INS LINE UTF-8 signingCA.conf
Terminal
```

Création d'une autorité de certification

- Cela se fait en une seule étape avec OpenSSL



```
~/superSignCA/signingCA : bash
File Edit View Bookmarks Settings Help
jib@jib-laptop: ~/superSignCA/signingCA$ openssl req -new -config signingCA.conf -out signingCA.csr
-keyout private/signingCA.key
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'private/signingCA.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
jib@jib-laptop: ~/superSignCA/signingCA$
```

Creation d'une autorité de certification

- On obtient ainsi la CSR signingCA.csr ainsi que la clé privée signingCA.key (2048 bits)



```
signingCA.key - Kate
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
private
  signingCA.key
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQ0wDgQIUScNn5KG7u8CAgga
MBQGCCqGSIB3DQMHBAGgg6cgZ/ah7gASCBMhp3PZRVIqvX9j8pVzdqpywK8qlc7As
PAoGWA0TdmPpsQCBCI1FGqKb3PHfZZsDXpeh03wM082U9kEojd2JU3m0msuWUC6U
sGZA9w5DpM1fc9Xp01WjWEKiRLcPTCbU13l8NwrtJLdwkww8h09ayuVjSk8mRMO
PT18TiAdRLTTBKIBNIRUiCr7MKvCwc1F4v4yqRnXAR1YI9KCKpkxSuXLSeroQB23
4sUP2F/bWAR/wIFTLirkTz9Ug2YLmGpkJaXVy/dgdIKiQcew12FWVUL28hCnmL1y
sXCilE4JWC40ivGoMT8pcijhSj7U1k/2Ij+yfS1ClQsPSLzfs2X3grivZwuxhvDx
fsACHtdaI40FpRMUuM/xCEcT74+kzD2sAY8GVUQ547Xww02JtQT9ovYR4t3U9EH
QJfgFKRrH00ulPz355iSj0rNDjJ4QBqIwSRt1n9Lte+uEppN5p7TJ32uPijT4KpW8
Qjxcg3QxhtF+umHhD56KL5FSmLJ5pwLQCN4Zb+1BhyRj9dd8SNczjzrUadwfbt68
00BqIPckePOP0gq89mn9pqCtL+IPHKQBihnz3ooBYeYzbLfnQre2xyuUdKY8bfeP
jTs3trgZih2aVRk/E81cgQY41tGS160+o9mJcZ0KBvUBNC1z00hA19wvMcIPPQkv
WZcERayeN0oHgFA22vNwnRyH1WY66hLlLueGI+nA8TzqplSqv529rPibW9XEL0s
KYM9eFwGwP/hl8ImE0lp6mTjNmsrnaq0YyvK40xo9jNlRhnBtUBAPiFp7y42s+
Lzy0Lzt1fD0KSqEu1Yqa5PWaciSnqkZtjjUN2unHMBn3K3H8X1lIWN5hz/ZaXWJ3
q3AGQY0AbS0J8t7ai5jz0uUo1e4iJzTT40nUrtv6tdKzwV06pchJSj4oWe+HVNAs
9CXQxW120m8EU4uZCinFni52f73QysSb4JEADLY+T4gJ7EaXCa5+Urt5GEC8cL+C
A2LsxNgovzSVlowot/RtmxB2QAcugS16fENNip9p8J46DCpZf++FNW4jVXNURyk8
k1Eu5UnNBC00BcbIFSN8iYhBKgnFYIvui1ZS7IiIZgj9zscrZ9VzgfT3vGMw745m
7obUHLAyhLgELXXSSohe622PRbZyIfPwLZ1AwX8rTf2wtP8Tv/2kbHpfEjo/cRX
EaZljULFAUZ49Up49bCyEhzEwt+yYSgKag+NGP8T3xGg1lRuBeIoJ3j1PL3aXa47
GTz/PCnglcnmg2kVFao/6QBvNAWbYzT+uDImQ01NvoeRYtpkZBSnC7p1otk0DoR0
6LdUeANWRiMP0uHV+Escue0WfSeZe2l82uacvZACTM8vjviWqn2HzpWSeDVLiwuh
pbvYKZFPjH+Dhl+Zb5iotd12+Z0DPoJicBDtdNnN0p81T5zym3+FIS4lv8RF0tV5
GtRL6RtWH0RZD3zELATAm18lbT6KT/iq/H1T+VUYhXAZhHbzrutqWU0oz0Cfyv7c
pGF5ag0qh5+k0DeYGHAg6Q3DGD0/i/GSxeuFrp6040pN0x6qb0pi0xTWb9szTEvc
yepS6ljARq1kuj2L1TMrNwYUjZqw2WY7LGW31EmtEbof89Tv0IHZwQAY7N6VTLAN
dtU=
-----END ENCRYPTED PRIVATE KEY-----
Line: 1 Col: 1      INS LINE UTF-8 signingCA.key
Terminal
```


Creation d'une autorité de certification

- Maintenant nous allons signer la CSR au moyen de la clé privée rootCA.key



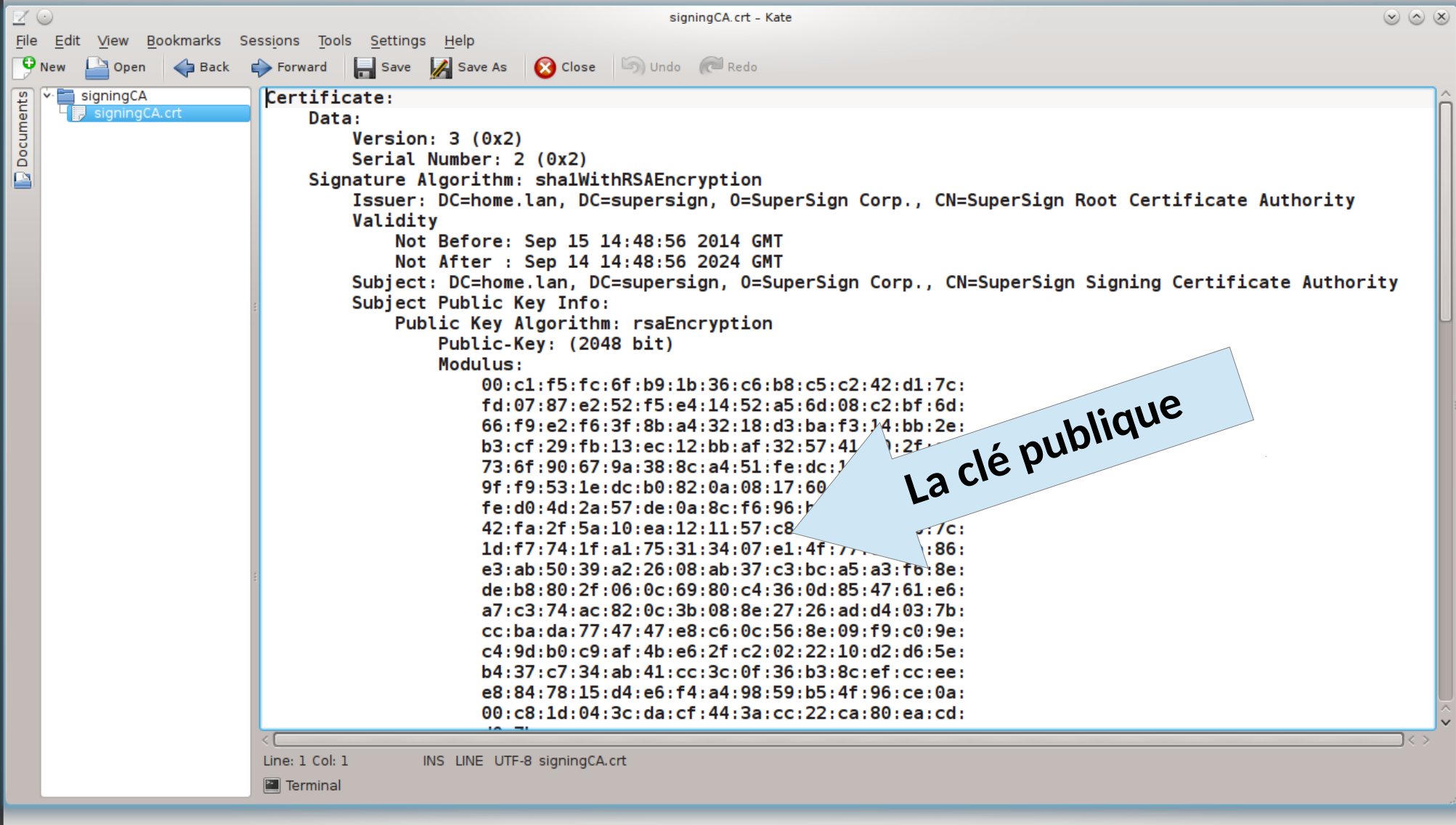
```
~/superSignCA/signingCA : bash
File Edit View Bookmarks Settings Help
jbb@jbb-laptop:~/superSignCA/signingCA$ openssl ca -config ../rootCA/rootCA.conf -in signingCA.csr
-out signingCA.crt -extensions signing_ca_ext
Using configuration from ../rootCA/rootCA.conf
Enter pass phrase for /home/jbb/superSignCA/rootCA/private/rootCA.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 2 (0x2)
  Validity
    Not Before: Sep 15 14:48:56 2014 GMT
    Not After : Sep 14 14:48:56 2024 GMT
  Subject:
    domainComponent           = home.lan
    domainComponent           = supersign
    organizationName          = SuperSign Corp.
    commonName                 = SuperSign Signing Certificate Authority
  X509v3 extensions:
    X509v3 Key Usage: critical
      Certificate Sign, CRL Sign
    X509v3 Basic Constraints: critical
      CA:TRUE, pathlen:0
~/superSignCA/signingCA : bash
```

Création d'une autorité de certification



- Et on obtient notre certificat d'autorité signante signé par l'autorité racine (c'est-à-dire par la clé privée de l'autorité racine) !

Création d'une autorité de certification



```
certificates
├── signingCA
│   └── signingCA.crt
└── ...

Certificate:
Data:
  Version: 3 (0x2)
  Serial Number: 2 (0x2)
  Signature Algorithm: sha1WithRSAEncryption
  Issuer: DC=home.lan, DC=supersign, O=SuperSign Corp., CN=SuperSign Root Certificate Authority
  Validity
    Not Before: Sep 15 14:48:56 2014 GMT
    Not After : Sep 14 14:48:56 2024 GMT
  Subject: DC=home.lan, DC=supersign, O=SuperSign Corp., CN=SuperSign Signing Certificate Authority
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
    Modulus:
      00:c1:f5:fc:6f:b9:1b:36:c6:b8:c5:c2:42:d1:7c:
      fd:07:87:e2:52:f5:e4:14:52:a5:6d:08:c2:bf:6d:
      66:f9:e2:f6:3f:8b:a4:32:18:d3:ba:f3:14:bb:2e:
      b3:cf:29:fb:13:ec:12:bb:af:32:57:41:17:2f:
      73:6f:90:67:9a:38:8c:a4:51:fe:dc:1
      9f:f9:53:1e:dc:b0:82:0a:08:17:60
      fe:d0:4d:2a:57:de:0a:8c:f6:96:h
      42:fa:2f:5a:10:ea:12:11:57:c8
      1d:f7:74:1f:a1:75:31:34:07:e1:4f:77:
      e3:ab:50:39:a2:26:08:ab:37:c3:bc:a5:a3:fb:8e:
      de:b8:80:2f:06:0c:69:80:c4:36:0d:85:47:61:e6:
      a7:c3:74:ac:82:0c:3b:08:8e:27:26:ad:d4:03:7b:
      cc:ba:da:77:47:47:e8:c6:0c:56:8e:09:f9:c0:9e:
      c4:9d:b0:c9:af:4b:e6:2f:c2:02:22:10:d2:d6:5e:
      b4:37:c7:34:ab:41:cc:3c:0f:36:b3:8c:ef:cc:ee:
      e8:84:78:15:d4:e6:f4:a4:98:59:b5:4f:96:ce:0a:
      00:c8:1d:04:3c:da:cf:44:3a:cc:22:ca:80:ea:cd:
      10:71:

Line: 1 Col: 1      INS LINE UTF-8 signingCA.crt
Terminal
```


Création d'une autorité de certification

```
signingCA.crt - Kate
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
  signingCA
  signingCA.crt
X509v3 extensions:
  X509v3 Key Usage: critical
    Certificate Sign, CRL Sign
  X509v3 Basic Constraints: critical
    CA:TRUE, pathlen:0
  X509v3 Subject Key Identifier:
    6B:05:71:1F:4A:FB:61:18:11:4B:33:33:53:2B:61:A5:D2:5F:AD:A6
  X509v3 Authority Key Identifier:
    keyid:A5:A9:C0:D2:F2:13:5D:2C:9F:25:A0:A2:68:57:60:3B:C1:3B:D3:CD

Signature Algorithm: sha1WithRSAEncryption
17:dd:be:f1:6e:8a:28:cc:3e:ed:f4:79:53:c6:0a:5a:07:00:
0c:2e:93:a6:b6:20:05:df:57:49:42:0f:d5:a3:64:4d:1b:81:
4e:9c:c9:3c:b3:15:27:2b:24:2c:17:26:95:99:07:12:2e:f2:
93:48:8c:7c:6f:6e:8c:a8:98:07:3b:29:c6:5b:bb:8:54:
c7:86:20:31:54:ad:57:45:54:a4:3a:11:8e:8d:f:
9f:60:37:ff:a1:4c:df:ff:0c:cf:44:a0:c1:ae:
fc:3b:c6:6b:ba:53:ee:e3:c5:6e:f3:4a:a6:a8:
39:e2:56:3b:76:2d:ea:a7:50:59:fd:1c:5e:2:
45:82:51:41:07:dc:d2:d6:3f:23:71:6e:0b:68:
3d:e9:a9:eb:44:19:c8:4a:da:2c:63:68:86:53:5b:cb:0:
03:5f:ef:b1:c3:98:da:e7:35:ec:71:15:a7:17:7e:14:59:3b:
9e:43:af:29:38:3d:82:65:36:f8:ed:ec:69:60:df:c5:4e:be:
4d:f2:dc:71:f7:2c:5e:3e:46:60:3b:fc:f0:9e:19:0b:7b:be:
8e:c9:31:f0:a6:97:b1:bd:57:7d:28:f6:c9:fa:fe:3e:44:98:
1c:18:75:04
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAJANBgkqhkiG9w0BAQUFADB+MRgwFgYKZCZImiZPyLGQB
GRYIaG9tZS5sYW4xGTAXBgoJKiaJk/IsZAEZFglzdXB1cnNpZ24xGDAWBgNVBAoM
D1N1cGVyU2lnbiBDb3JwLjEtMCsGA1UEAwkU3VwZXJTaWduIFJvb3QgQ2VydGlm
aWNhdGUgQXV0aG9yaXR5MB4XDTE0MDkxNTE0NDg1NlloXDTI0MDkxNDE0NDg1Nllo
-----END CERTIFICATE-----
Line: 1 Col: 1      INS LINE UTF-8 signingCA.crt
Terminal
```

La signature

Création d'une autorité de certification



- A présent nous allons :
 - Générer la clé privée de `monsiteweb.home.lan`
 - Générer la clé publique de `monsiteweb.home.lan`
 - Générer la *Certificate Signing Request* de `monsiteweb.home.lan`

Création d'une autorité de certification

- Cela se fait en une seule étape avec OpenSSL



```
~/superSignCA/server : bash
File Edit View Bookmarks Settings Help
jbb@jbb-laptop: ~/superSignCA/server$ openssl req -new -config server_tls.conf -out requests/server.csr
-keyout private/server.key
Generating a 2048 bit RSA private key
.....
.....+++
.....+++
writing new private key to 'private/server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
1. Domain Component          (eg, com)          []:home.lan
2. Domain Component          (eg, company)     []:supersign
4. Organization Name         (eg, company)     []:SuperSign Corp.
6. Common Name               (eg, full name)   []:monsiteweb.home.lan
jbb@jbb-laptop: ~/superSignCA/server$
```

Création d'une autorité de certification

- On obtient ainsi la CSR server.csr ainsi que la clé privée server.key (2048 bits)



```
-----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBAKkwggSIAgEAAoIBAQDfTD8SSJNB0PDn
Yo1Q1jSKgBM2BURtPWZ3MRsbAa8Q0uzjA0WGoIR4XEbMpQbId8VVqrB+AqqR5NgI
iw2dx0p4oiXxLYfpItb+K/hjQ2+iCkJSy4eBaUrc3wn6aJSEJ4uB5q60fK+qUUDJ
FXfhyzjHo87gmmJb/ix4xr7hGrtoew2a54/xDVb8PEshz9j0S2yL8jM4AMIWj1UH
ohMR37DBN4/ZfZRQ+y18XFZ4B7YjgU0LLi4AatSaLxuZS3+RBQ//KJag85F04yD
K6cFcWMOKXC7wKYHATbtwPntNLrab8V+dhS8lqzmST3x8562KG/I3/kNehIzrzTa
iCQmh0KPAgMBAECCggEAZVjk03aPJ3Mes9Y8PpA9/pKmpZsXW7/fbr+h7sr/2yHw
lERa3PSjy0E0fTL1b600qmITQs4Q/rCgAJikMHQ8e04U7Hbe1ciTTb9F0EBWPXRs
j8mCkhDpAoBUtgubRLPyfJ5PoaajzKihMyGETL4A410NxNEtTVV6d8K0xdRtNMiz
ZzxhJDbZsZLSFQcheTU6ZKJb4XcYM3ycI0//0YBz8DSeCaYw+Ki9SSpfdK8MMxp
QPowMSTyBHgFTZxXIIsAo0c9JEJSA27W0KBK/2JZYIyT5GwyR/qNbMXsAPKkEhdz4
pq6JdPC0yQ6kbFf5sj228i8c4e9wsfDhHnu477vcYQKbgQDz+JJt0F+yIicZW6GY
HpXTGjY37awNcXbtnSAvHHvgTeaIJ+l0dKu8bHd0CF1+MqmtaRn6nqfX0A16N45S
UCK/WwN/htTq2swa7Avz0DJ+3+o0DT27EWPw5SV29d2pGZg5/aB5XLbivA+8Xid1
tgbLKL48QL53Jn0IQp6soV0R0QKBgQDqTrxXZYf0+YJe5jZSPZ2gCbXSSUf2WbZ0
7cxXL8DWh5JISHeQnm0doJcCJtna9dZkjZmd9ja8F04wKqbjXa/6t67LVaCs3bJ7
Fc/o896Aq4NAgLXgXyTjMI0a7fcoNwmgfhfZWEiqDe7CiDu+Khm41y1ZLB8VD7Mgd
RmjvGEJmXwKBgQDaukxs51HEXqI3Nd0WkjU44hMh6U+LbeJ1/ZzRH/RfkPfrShaV
ZDxj3crIhu9rKPi+o6/K5VsNPVYX8J8EZRwB+xMTzkotLKPQrzc4Cz6S2z2qFt3
Jgw3+PixwpRL+spjk20/6sN8W6av9qnh+qP0mb0KyXD5WdvSmgewNas/QQKBgQct
pbCbrNl3yrq+DitQA+4fv1dcPIciohZAzkPUkcV+uHbHLz29ZkU5etyvcY+fYPaF
CxRAJ0slVrhRpiqM8tALJbbpY5hyxNMTuXSelq3drStG0Jz6LkkuPrQqTBYSLJ0d
reqEm236h96+HAeXRxl0cMAUnA+saX1dh0Mb34TGTQKBgQCw5grjRDNJMMKdWcGT
EsKJlUdiAk3X6PrR+l002ZsRWr0qHjuocQLP1KawPuLc7UxeNG8W5Mg4EeoMeo+7
LzUfUTcLR1v5yE22WIA3Ede7g+rYBdFxxLDv/CQYyh6D1S+aFB0gPB4oZmW8zvZ0
Wk1hUH0+GzqqMfWakQXb0vdvw==
-----END PRIVATE KEY-----
```


Création d'une autorité de certification

- Maintenant nous allons signer la CSR au moyen de la clé privée signingCA.key



```
~/superSignCA/server : bash
File Edit View Bookmarks Settings Help
jib@jib-laptop:~/superSignCA/server$ openssl ca -config ../signingCA/signingCA.conf -in requests/server.csr
-out certs/server.crt -extensions server_ext
Using configuration from ../signingCA/signingCA.conf
Enter pass phrase for /home/jib/superSignCA/signingCA/private/signingCA.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 1 (0x1)
  Validity
    Not Before: Sep 15 15:54:46 2014 GMT
    Not After : Sep 14 15:54:46 2016 GMT
  Subject:
    domainComponent           = home.lan
    domainComponent           = supersign
    organizationName          = SuperSign Corp.
    commonName                 = monsiteweb.home.lan
  X509v3 extensions:
    X509v3 Key Usage: critical
      Digital Signature, Key Encipherment
    X509v3 Basic Constraints:
      CA:FALSE
    X509v3 Extended Key Usage:
```

Création d'une autorité de certification



- Et on obtient notre certificat serveur pour `monsiteweb.home.lan` !

Création d'une autorité de certification

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1 (0x1)

Signature Algorithm: sha1WithRSAEncryption

Issuer: DC=home.lan, DC=supersign, O=SuperSign Corp., CN=SuperSign Signing Certificate Authority

Validity

Not Before: Sep 15 15:54:46 2014 GMT

Not After : Sep 14 15:54:46 2016 GMT

Subject: DC=home.lan, DC=supersign, O=SuperSign Corp., CN=monsiteweb.home.lan

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

```
00:df:4c:3f:12:48:93:41:d0:f0:e7:62:89:50:96:
34:8a:80:13:36:05:4a:ed:3d:66:77:31:1b:1b:01:
af:10:d2:ec:e3:03:45:86:a0:84:78:5f:46:
06:c8:77:c5:55:aa:b0:7e:02:aa:91:
0d:9d:c4:ea:78:a2:25:f1:2d:87:
f8:63:43:6f:a2:0a:42:52:cb:87:
09:fa:68:94:84:27:8b:81:e6:ac:
47:49:15:77:e1:cb:38:c7:a3:ce:e0:9a:0:5b:fe:
2c:78:c6:be:e1:1a:bb:68:7b:0d:9a:e7:8f:f1:0d:
56:fc:3c:4b:21:cf:d8:ce:4b:6c:8b:f2:33:38:00:
c2:16:8f:55:07:a2:13:11:df:b0:c1:37:8f:d9:7d:
94:50:fb:2d:7c:5c:56:78:07:b6:23:81:4d:25:2e:
2e:00:6a:d4:9a:2f:1b:99:4b:7f:91:05:0f:ff:fc:
a2:5a:83:ce:45:3b:8c:83:2b:a7:05:71:63:0e:29:
70:bb:c0:a6:07:01:36:ed:c0:f9:ed:34:ba:da:6f:
```

La clé publique

Line: 1 Col: 1

INS LINE UTF-8 server.crt

Terminal

Création d'une autorité de certification



The image shows a text editor window titled "server.crt - Kate". The left sidebar displays a file tree with folders "server", "certs", "private", and "signingCA". The "certs" folder is expanded, showing "server.crt" selected. The main text area contains the following content:

```
X509v3 Extended Key Usage:  
    TLS Web Server Authentication, TLS Web Client Authentication  
X509v3 Subject Key Identifier:  
    EF:9C:08:9C:37:C2:A6:74:6C:0B:D2:9C:B8:71:72:5E:6C:6B:80:32  
X509v3 Authority Key Identifier:  
    keyid:6B:05:71:1F:4A:FB:61:18:11:4B:33:33:53:2B:61:A5:D2:5F:AD:A6  
  
Signature Algorithm: sha1WithRSAEncryption  
a6:91:2f:3c:8c:11:19:fb:dc:db:be:5c:a8:00:27:57:0c:d5:  
bb:f5:5e:c2:b6:32:2f:37:0c:83:16:0c:62:84:bb:6b:96:0b:  
b1:af:76:97:b9:02:3d:cd:fe:81:f0:b7:cc:f8:de:73:e4:e5:  
9c:f5:b8:67:f1:0a:05:60:9a:ce:18:a1:93:01:3f:15:e8:d1:  
7a:c9:f0:ce:80:df:65:4e:dc:fd:02:b0:aa:e9:22:dc:df:c3:  
47:75:ac:22:9a:ff:d7:c0:ca:03:37:b3:b6:59:74:37:7f:37:  
c4:01:3d:b3:94:bf:51:52:72:74:c0:25:9b:ae:cc:f9:99:  
d6:27:f6:f6:5b:8a:5e:b4:c7:82:9d:26:cf:ae:3f:99:  
2d:10:11:fc:51:08:f0:20:62:78:f7:f1:91:13:73:  
51:c8:53:a1:c4:b1:84:dc:90:08:3b:04:8f:38:e9:  
0c:a5:60:aa:f8:b2:a7:aa:b3:68:12:46:53:d0:00:  
59:d3:ad:3e:17:99:97:10:b4:57:dc:1f:3f:88:81:10:  
7c:40:7a:6a:04:c7:53:36:28:67:3e:38:d4:53:a5:6b:93:6f:  
f1:6a:f8:a3:0f:33:66:aa:59:95:90:a2:49:aa:54:9f:5b:c7:  
67:15:8e:ae  
  
-----BEGIN CERTIFICATE-----  
MIID5jCCAs6gAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTEYMBYGCgMSJomT8ixk  
ARkWCGhvbWUubGFuMRkwFwYKcZImiZPyLQBGGRYJc3VwZXJzaWduMRgwFgYDVQQK  
DA9TdXB1c1NpZ24gQ29ycC4xMDAuBgNVBAMMJ1N1cGVyU2lnbiBTaWduaW5nIEN1  
cnRpZmlyYXRlIEF1dGhvcml0eTAeFw0xNDA5MTUxNTU0NDZaFw0xNjA5MTQxNTU0  
NDZaMG0xGDAWBgoJkiaJk/IsZAEZFghob211LmxbhjEZMbcGCgMSJomT8ixkARKW
```

A blue arrow points to the signature block, with the text "La signature" written inside it.

Line: 7 Col: 17 INS LINE UTF-8 server.crt

Terminal

Création d'une autorité de certification



- Et pour finir nous allons :
 - Générer la clé privée d'Alice
 - Générer la clé publique d'Alice
 - Générer la *Certificate Signing Request* d'Alice

Création d'une autorité de certification

- Cela se fait en une seule étape avec OpenSSL



```
~/superSignCA/user : bash
File Edit View Bookmarks Settings Help
jbb@jbb-laptop: ~/superSignCA/user$ openssl req -new -config user.conf -out requests/alice.csr
-keyout private/alice.key
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'private/alice.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
1. Domain Component          (eg, home.lan)          []:home.lan
2. Domain Component          (eg, excelsior)        []:supersign
4. Organization Name         (eg, Excelsior)        []:SuperSign Corp.
6. Common Name                (eg, John Smith)      []:Alice
7. Email Address              (eg, john.smith@foobar.com) []:alice@machintruc.com

~/superSignCA/user : bash
```

Création d'une autorité de certification

- On obtient ainsi la CSR `alice.csr` ainsi que la clé privée `alice.key` (2048 bits)



```
alice.key - Kate
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
private
  alice.key
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBABGkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQIjfdEN0rRv/ICAggA
MBQGCCqGSIb3DQMHBACeJoi40Ga+QSCBMgtpq5zJUL0/EImYSyIEACqB1sHeMhp
SnJ+eWgYBHGYavENZ9L0N6tGfxdt+G7s3DkAgSVC4R+8QQXDGxaeuwWe3MfbZWM4
XEwyE1WIMf6NtQ+sL6zYMDfU0z0CNzjnP8HmRty0tU/W9kSHf+AAzs55gKjAjaQP
LSga0Qe6IYCv18cKxTBEvtITbC0cEU9Uutv+sctleSF9pKRL2PI0Jf0Ro3z14cRW
u5vCv/bwErrorAtC0xokEFiBy5Cx4a0p7ZPTV/8UdmHaE9G6b600Wr2mjXqyMpo5PQ
at9J20J7Zdmi2xxfgFLimxZxevNV8MS02ByRm0YnmiEInHJzPobbmIXhPPu+WKu1
TCac0Wdnnmxi0PzPqEihYnAlmIl3hZ3JpT0a41VtrZI7MwMkA1Mq0VfmEn4+w0Tn
4V9Ce0wW0++HTxjB23EgxRzeSU8hm7yoDxHcmft6ud1Psf+c+04GHScrUeVt1Tt+3
0HwEQ0i0m0Gi3+d+vG41AVh0EMRtNkKTxSFx9DWmHQ1R+Q5y8IjLW1vVb8B2j7x6
wLuTKFUd3WgjqZtbZsk4K0Ld6FQgJSc90LbnSXbsK/lf97b28bUiEKZ50a78pxzg
QV9uGSiPmZM6tJs2K75gxMaM5++IJUUGfTx7iZsAqzA+kakw9ZKVTtTtPX8wMsNL
pc0WaZqn5kHG46jAzLCSg57urmipJVWVIhSDNcBlVT1fG9I7L1tP0dku0yayBEn4
HQ/qRKi9e5mHtmYaRzpp1+t9Q5vz0y2/QxfP+hj70HuN0ovmheK25NTzoZS1oyo/
wtW8EkxAnNZ6p/vJMq48K4ZcC4kb+bT2NX0qeDuFcmjubGl+c+uTLZWRah09vf4hF
nh1L2V2p5u80GMDLHmwTW+TkXpYZ8YcSe7F0xpVqjVJ07SLsm0V5duQHwPf13Lun
w0UkJyidF640fhkTRPt4nwQd5VU5sZT2yry88b2HDk/L/W/F82rSCR6h5iD/4/U3
bpkBAiCu26oAcwUQ27J/bCndHbM3JX9HmRIR1V20/6dZ+HETAew/UQFXiG280TrE
BvvVbLucgw0fQzVSym0y/k89ajzVqinj2Bkoa26EMI3FzCfcqZLibr+n0LUC11+
Qp9GccRuTgx0Ziylk3rJY+c+rSHlxKetCdEdMk4MnVYoCLksCbf07lhywDaGD6x
c6w2JgzG930jU12j3nJtxLyFpAcRl2mdYmLR4JkexwycPjCd8sjinTmfpVT4zr7x
7NxTAo/0IY0s+RY0h2vTJ/lj008pRqi9GZsHckjGXfUNSA1FtiPcmFr9hMMSCpsE
B7r/Pa5MSKkuVQs8xZ5Wr1kCFEpghtGorfqNo3l+pf9ADFcpJ1m8Gu2RHeLlvpt
hyK6z7CDFjqXptCamGx3RCMu9wzYkekCneqXX9uEubYw5lYp0z7gmdm+KqXnZ1IB
1yMrhLmuN0GxAjiEh7T9xHlBaf9U77CK0CvFrnk0XwHyqGrYf6pdofNiKEY06Mgm
MCKJGM5CU0Vz0VXUngZ03dyRF10ZT52h0cRLzqcfV/4s5yVrnjtVD3rx9InIPh10
hJ/R4seNaEMzpey7WCIPBS/MIqzwd4dizA2VvjEx1aEBVgcFAR8rWSorfS9YEvM
0Dw=
-----END ENCRYPTED PRIVATE KEY-----
Line: 1 Col: 1      INS LINE UTF-8 alice.key
Terminal
```

Création d'une autorité de certification

- Maintenant nous allons signer la CSR au moyen de la clé privée signingCA.key



```
~/superSignCA/user : bash
File Edit View Bookmarks Settings Help
jbb@jb-laptop:~/superSignCA/user$ openssl ca -config ../signingCA/signingCA.conf -in requests/alice.csr
-out certs/alice.crt -extensions email_ext
Using configuration from ../signingCA/signingCA.conf
Enter pass phrase for /home/jb/superSignCA/signingCA/private/signingCA.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 3 (0x3)
  Validity
    Not Before: Sep 15 16:23:08 2014 GMT
    Not After : Sep 14 16:23:08 2016 GMT
  Subject:
    domainComponent           = home.lan
    domainComponent           = supersign
    organizationName          = SuperSign Corp.
    commonName                 = Alice
  X509v3 extensions:
    X509v3 Key Usage: critical
      Digital Signature, Key Encipherment
    X509v3 Basic Constraints:
      CA:FALSE
```

Création d'une autorité de certification



- Et on obtient le certificat de Alice!

Création d'une autorité de certification

```
alice.crt - Kate
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
  certs
    alice.crt
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number: 3 (0x3)
Signature Algorithm: sha1WithRSAEncryption
Issuer: DC=home.lan, DC=supersign, O=SuperSign Corp., CN=SuperSign Signing Certificate Authority
Validity
  Not Before: Sep 15 16:23:08 2014 GMT
  Not After : Sep 14 16:23:08 2016 GMT
Subject: DC=home.lan, DC=supersign, O=SuperSign Corp., CN=Alice
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  Public-Key: (2048 bit)
  Modulus:
    00:c8:a9:b0:10:ee:3c:f1:40:88:f1:cc:62:8d:da:
    30:ed:4c:73:95:0d:81:bf:0e:05:b0:c7:96:3d:07:
    8d:68:29:a5:af:ec:fb:80:d1:48:5a:3c:ca:6d:60:
    e5:16:9e:ea:cf:7c:0d:60:cd:83:5d:0f:6b:97:58:
    c3:be:42:dc:94:02:ed:80:8e:1b:9f:1e:b3:b2:f7:
    dc:b3:10:05:b4:7c:ae:af:65:ee:24:
    3d:66:75:db:f8:d7:6c:5d:af:fe:5
    fa:b3:ed:73:8e:db:04:cd:43:bd:
    7c:18:65:e5:48:82:8e:83:51:f
    72:f6:c7:95:6e:bf:a4:fe:44:b8:5b:
    c2:d0:b2:d9:66:64:b5:10:36:03:09:2f:1b:00:47:
    59:47:55:7b:e7:02:12:e0:82:6b:6c:da:6f:5a:59:
    24:42:68:2e:d6:23:47:c6:b3:a3:97:9e:e1:53:70:
    a9:a6:e6:e8:d0:c0:2c:06:9d:a4:08:6a:05:0d:28:
    03:bc:45:18:ea:d1:8d:16:f6:0e:74:24:fb:55:1f:
    6f:6a:24:30:ca:e4:76:a7:2f:cf:72:de:e0:d2:73:
    85:30:66:f2:05:c5:59:0f:66:58:81:05:f5:8e:5b:
    88.59
Line: 1 Col: 1      INS LINE UTF-8 alice.crt
Terminal
```

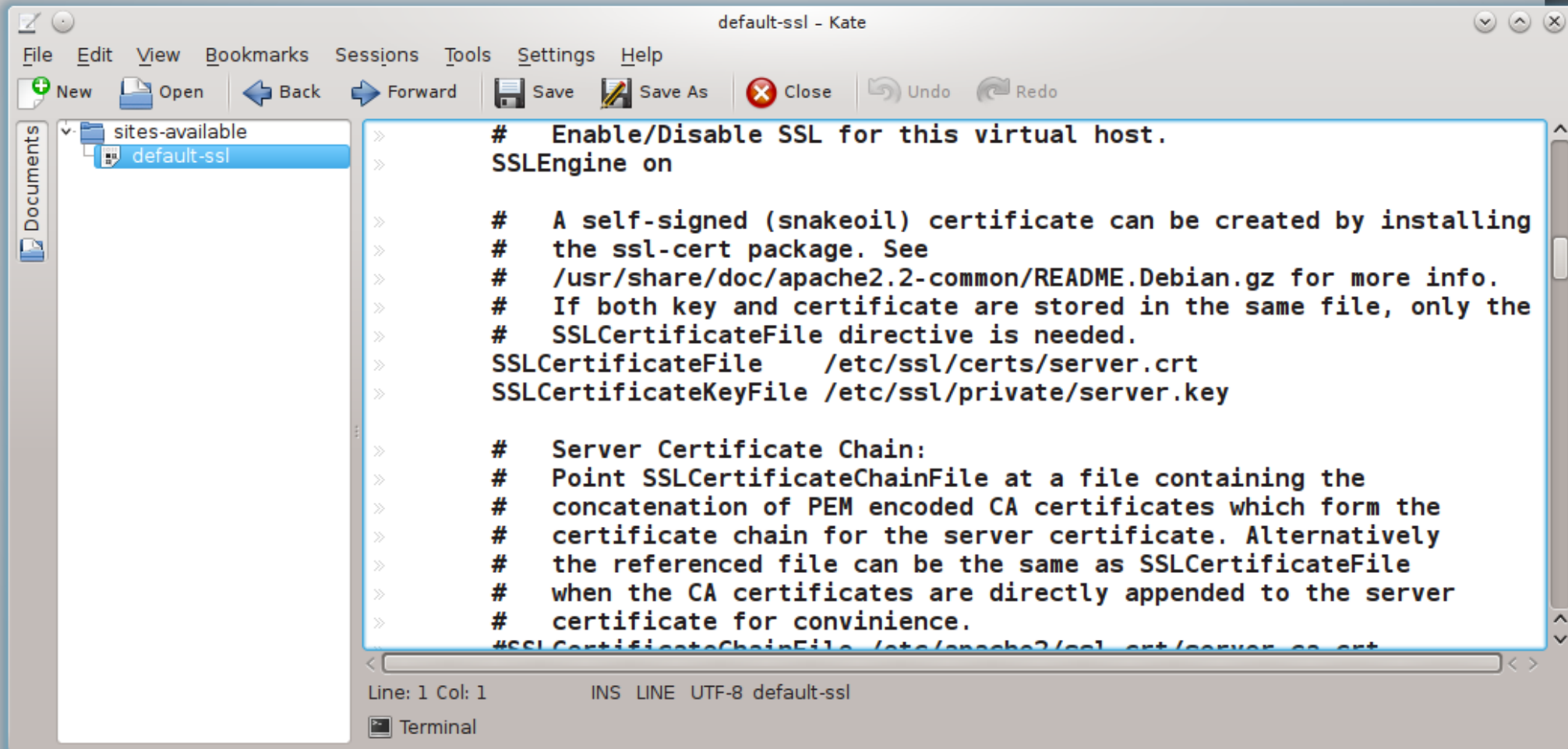

Création d'une autorité de certification

- Dans la pratique, les certificats “utilisateurs” sont souvent associés à la clé privée dans un “bundle” ou enveloppe. La norme la plus utilisée est PKCS 12.



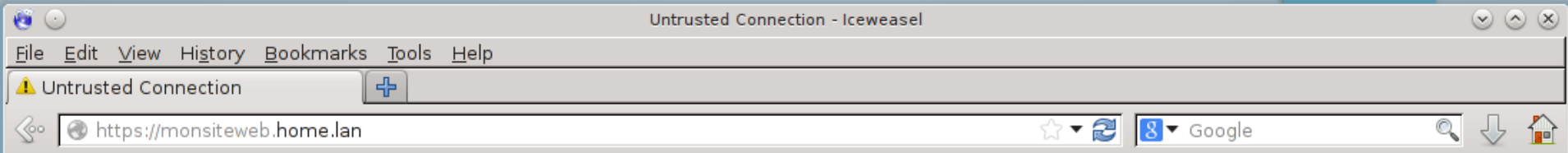
```
~/superSignCA/user : bash
File Edit View Bookmarks Settings Help
jb@jb-laptop: ~/superSignCA/user$ openssl pkcs12 -export -name "Alice" -inkey private/alice.key
-in certs/alice.crt -out bundles/alice.p12
Enter pass phrase for private/alice.key:
Enter Export Password:
Verifying - Enter Export Password:
jb@jb-laptop: ~/superSignCA/user$
```


Déploiement du certificat serveur



```
default-ssl - Kate
File Edit View Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
sites-available
default-ssl
>> # Enable/Disable SSL for this virtual host.
>> SSLEngine on
>>
>> # A self-signed (snakeoil) certificate can be created by installing
>> # the ssl-cert package. See
>> # /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
>> # If both key and certificate are stored in the same file, only the
>> # SSLCertificateFile directive is needed.
>> SSLCertificateFile /etc/ssl/certs/server.crt
>> SSLCertificateKeyFile /etc/ssl/private/server.key
>>
>> # Server Certificate Chain:
>> # Point SSLCertificateChainFile at a file containing the
>> # concatenation of PEM encoded CA certificates which form the
>> # certificate chain for the server certificate. Alternatively
>> # the referenced file can be the same as SSLCertificateFile
>> # when the CA certificates are directly appended to the server
>> # certificate for convenience.
>> #SSLCertificateChainFile /etc/apache2/ssl_cert/server_ca.crt
Line: 1 Col: 1 INS LINE UTF-8 default-ssl
Terminal
```

Déploiement du certificat serveur



This Connection is Untrusted

You have asked Iceweasel to connect securely to **monsiteweb.home.lan**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

[Get me out of here!](#)

- ▶ **Technical Details**
- ▶ **I Understand the Risks**

Déploiement du certificat serveur

The image shows a sequence of windows in the Iceweasel browser. The background window is the 'Iceweasel Start Page'. Overlaid on it is the 'Iceweasel Preferences' window, with the 'Certificates' tab selected. The 'Certificates' tab shows options for handling server requests and a 'View Certificates' button. In the foreground is the 'Certificate Manager' window, which displays a list of certificate authorities. A 'Downloading Certificate' dialog box is open, asking for trust in a new CA named 'SuperSign Root Certificate Authority'. The dialog has three checked options: 'Trust this CA to identify websites', 'Trust this CA to identify email users', and 'Trust this CA to identify software developers'. There are 'View', 'Cancel', and 'OK' buttons.

Iceweasel Start Page - Iceweasel

File Edit View History Bookmarks Tools Help

Iceweasel Start Page

Iceweasel Preferences

General Tabs Content Applications Privacy Security Sync Advanced

General Network Update Certificates

When a server requests a certificate:

Select one automatically

View Certificates View...

Help

Certificate Manager

Your Certificates People Servers Authorities Others

You have certificates on file that identify these certificate authorities:

| Certificate Name | Security Device |
|--|-----------------|
| ▷ Dhimyotis | |
| ▷ DigiCert Inc | |
| ▷ Digital Signature Trust | |
| ▷ Digital Signature Trust Co. | |
| ▷ Disig a.s. | |
| ▷ E-Tuğra EBG Bilişim Teknolojileri ve Hizmetleri A.Ş. | |
| ▷ EBG Bilişim Teknolojileri ve Hizmetleri A.Ş. | |
| ▷ EDICOM | |
| ▷ Elektronik Bilgi Guvenligi A.Ş. | |
| ▷ Entrust, Inc. | |
| ▷ Entrust.net | |
| ▷ Equifax | |
| ▷ Equifax Secure Inc. | |
| ▽ Excelsior | |
| Excelsior Root CA | |
| Excelsior Signing CA | |
| ▷ Generalitat Valenciana | |
| ▷ GeoTrust Inc. | |

View... Edit Trust...

Downloading Certificate

You have been asked to trust a new Certificate Authority (CA).

Do you want to trust "SuperSign Root Certificate Authority" for the following purposes?

Trust this CA to identify websites.

Trust this CA to identify email users.

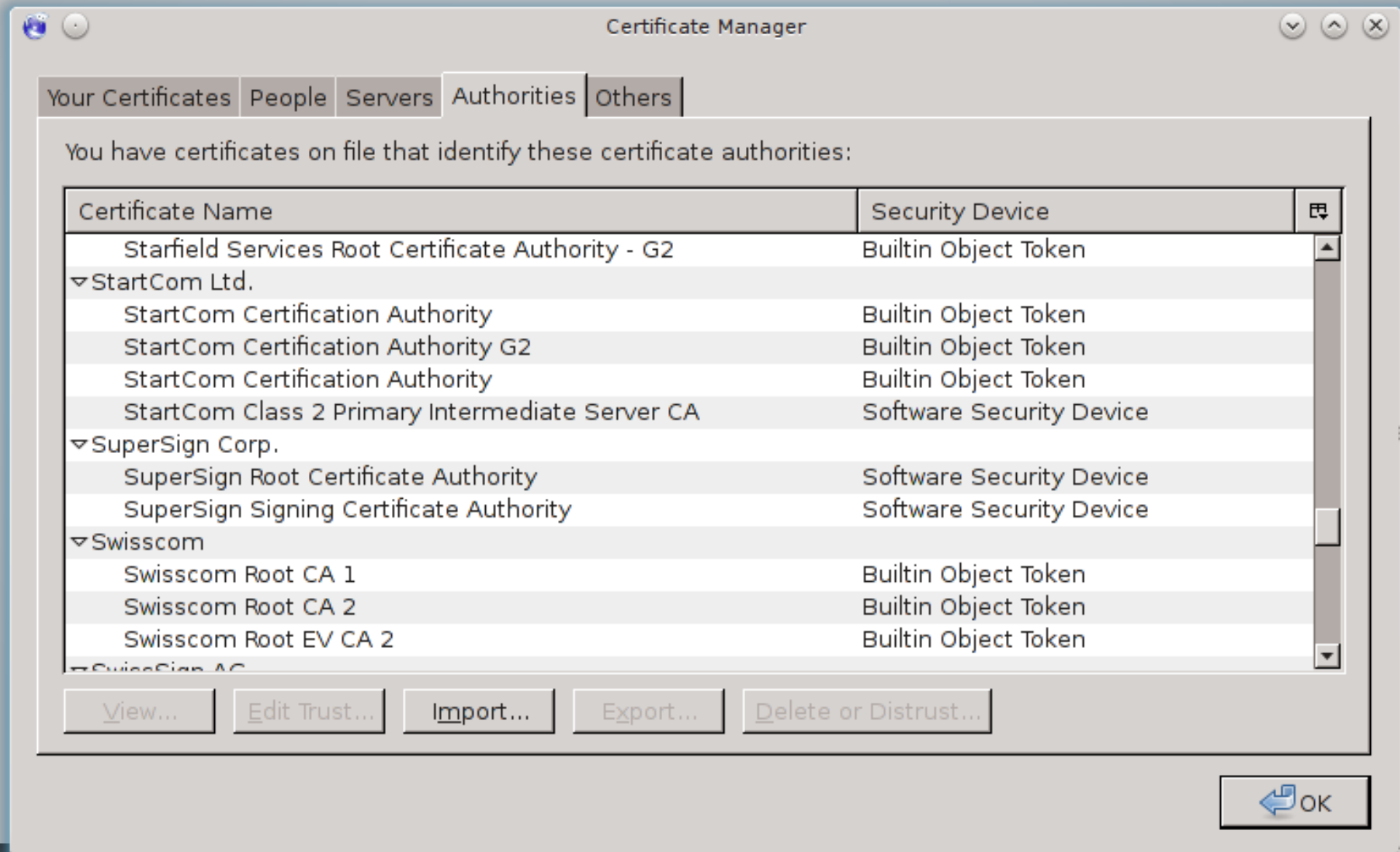
Trust this CA to identify software developers.

Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).

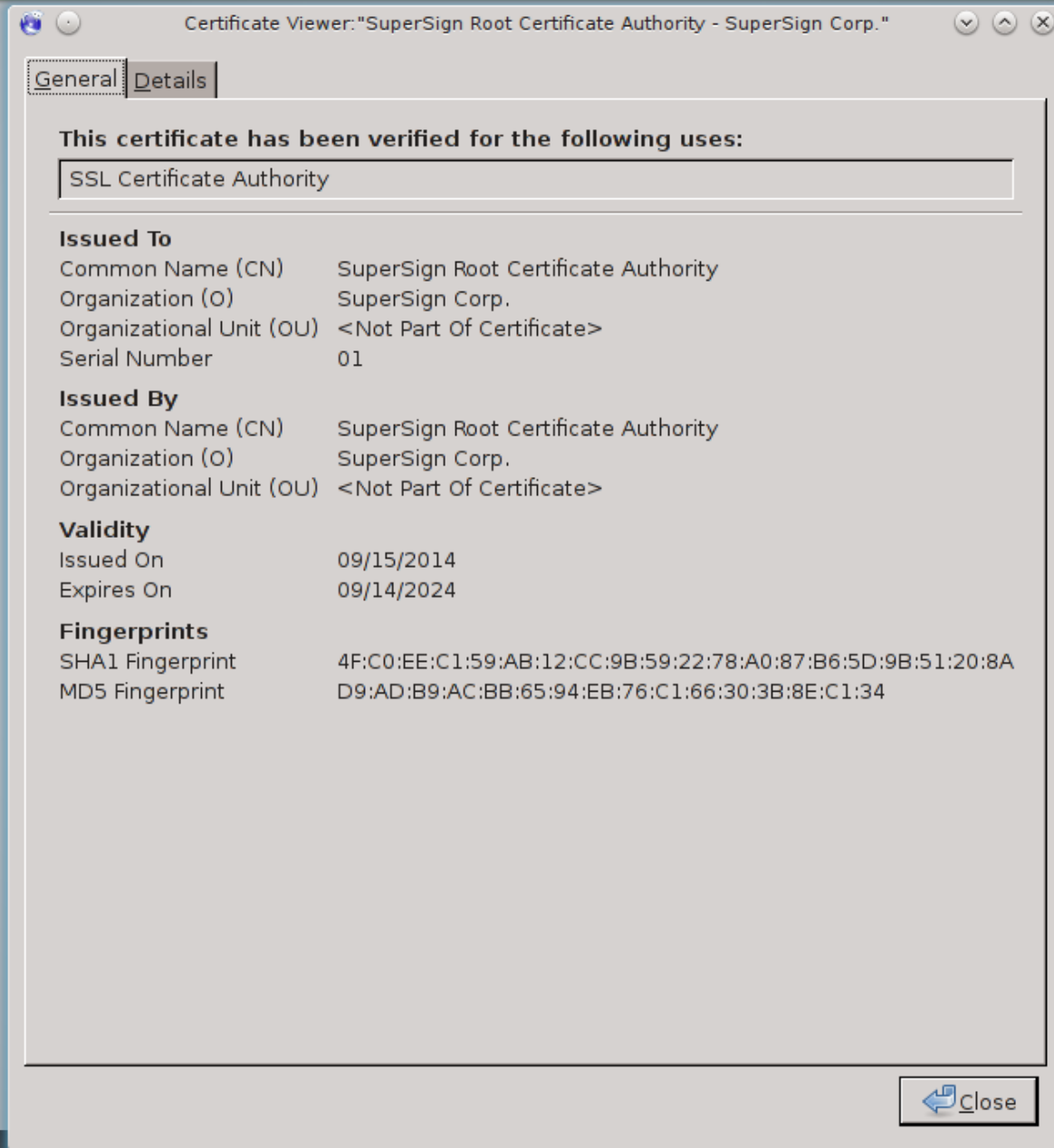
View Examine CA certificate

Cancel OK

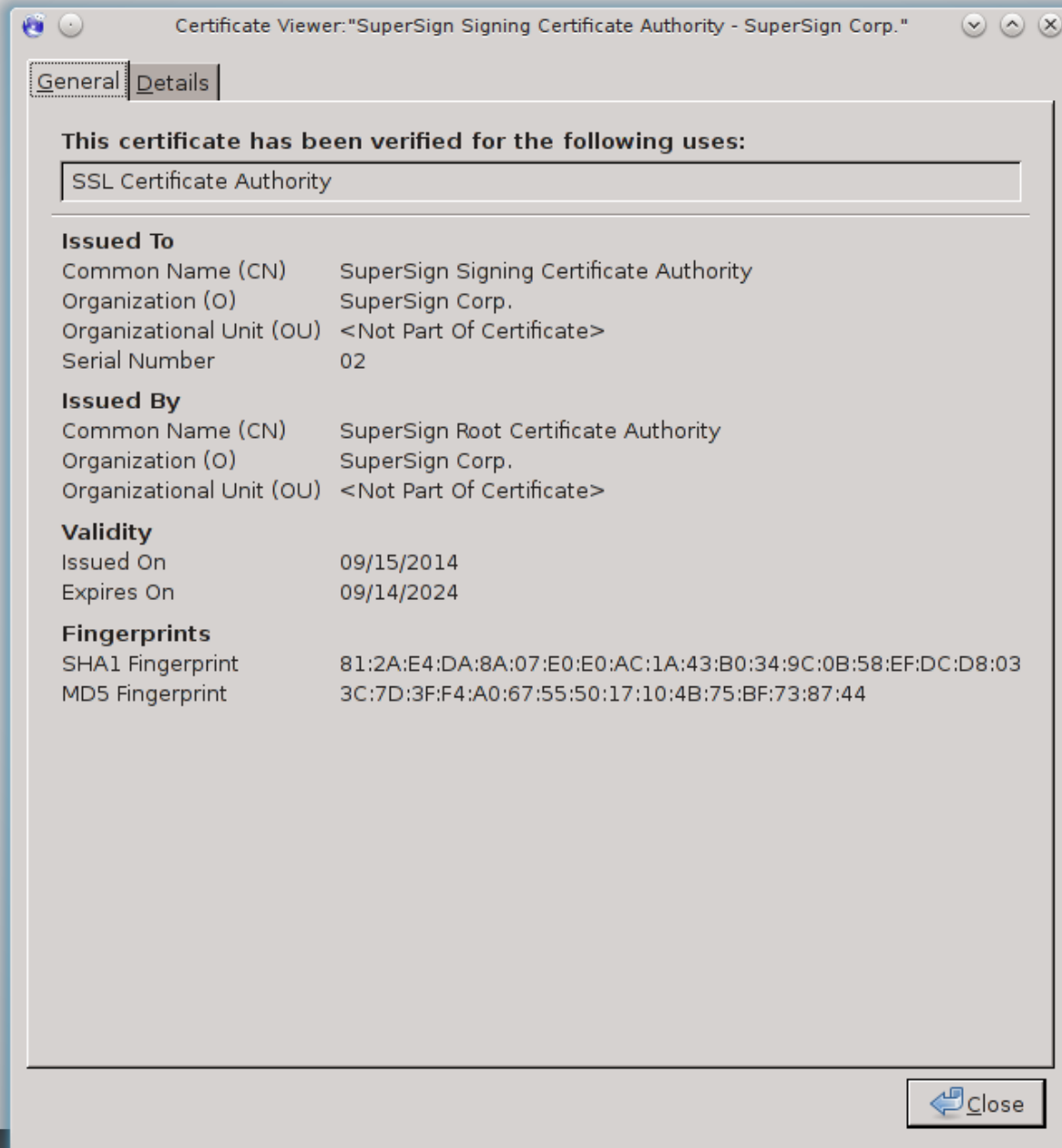
Déploiement du certificat serveur



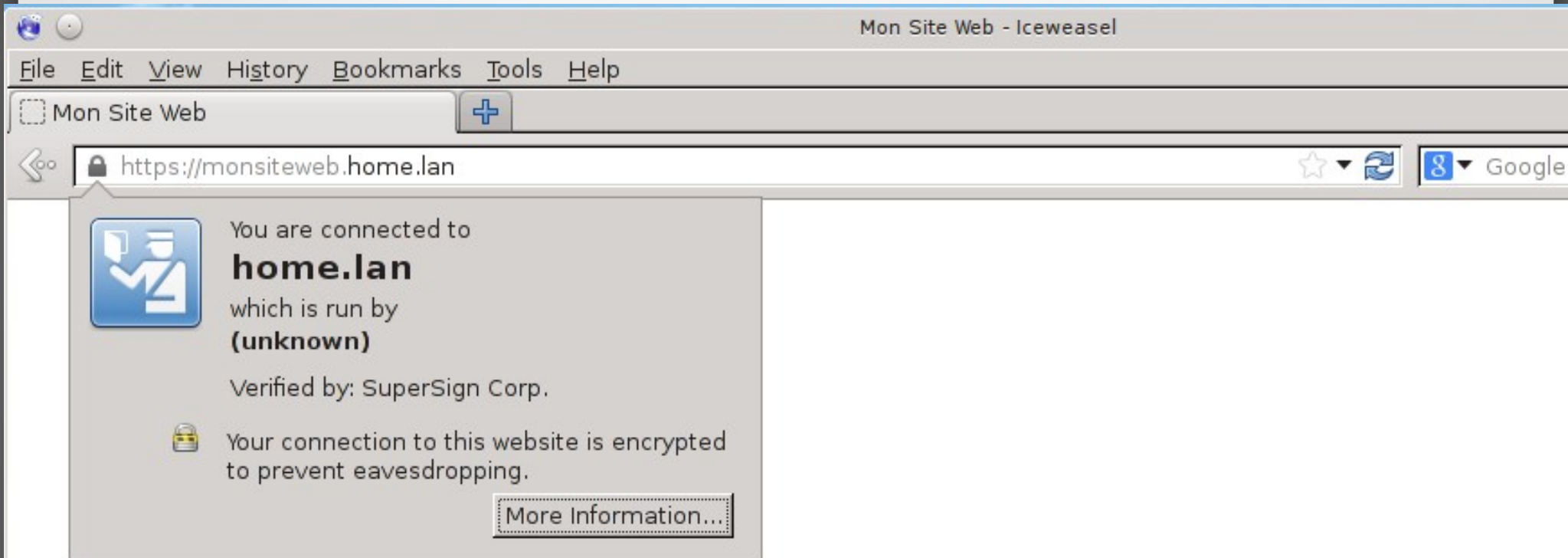
Les certificats électroniques



Déploiement du certificat serveur



Déploiement du certificat serveur



Bienvenue sur Mon Site Web!

Déploiement du certificat serveur

The image shows a web browser window titled "Mon Site Web - Iceweasel" with the address bar displaying "https://monsiteweb.home.lan". A "Page Info" dialog box is open, showing the "Security" tab. The dialog box contains the following information:

Website Identity

- Website: **monsiteweb.home.lan**
- Owner: **This website does not supply ownership information.**
- Verified by: **SuperSign Corp.**

[View Certificate](#)

Privacy & History

- Have I visited this website prior to today? **No**
- Is this website storing information (cookies) on my computer? **No** [View Cookies](#)
- Have I saved any passwords for this website? **No** [View Saved Passwords](#)

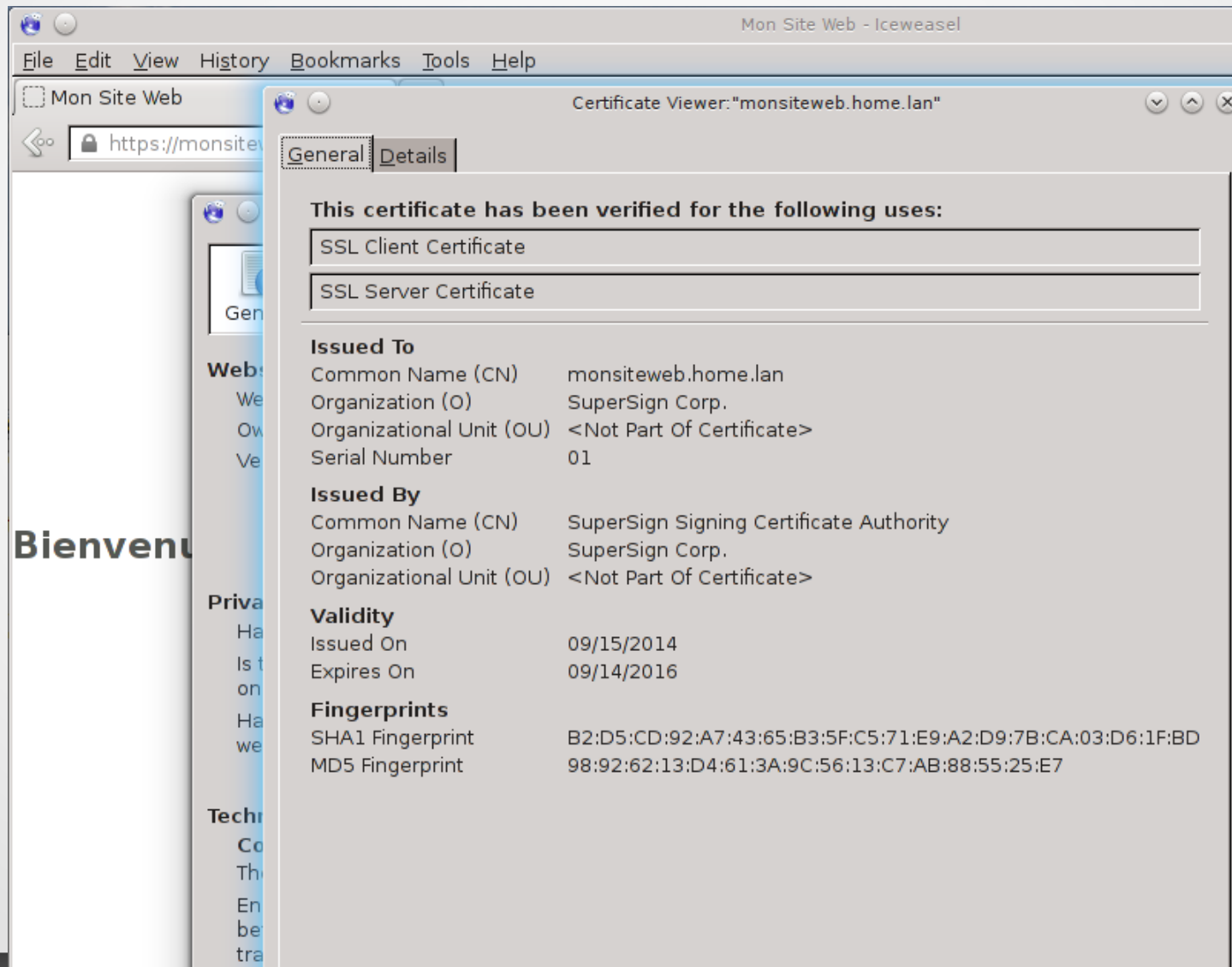
Technical Details

Connection Encrypted: High-grade Encryption (AES-128, 128 bit keys)

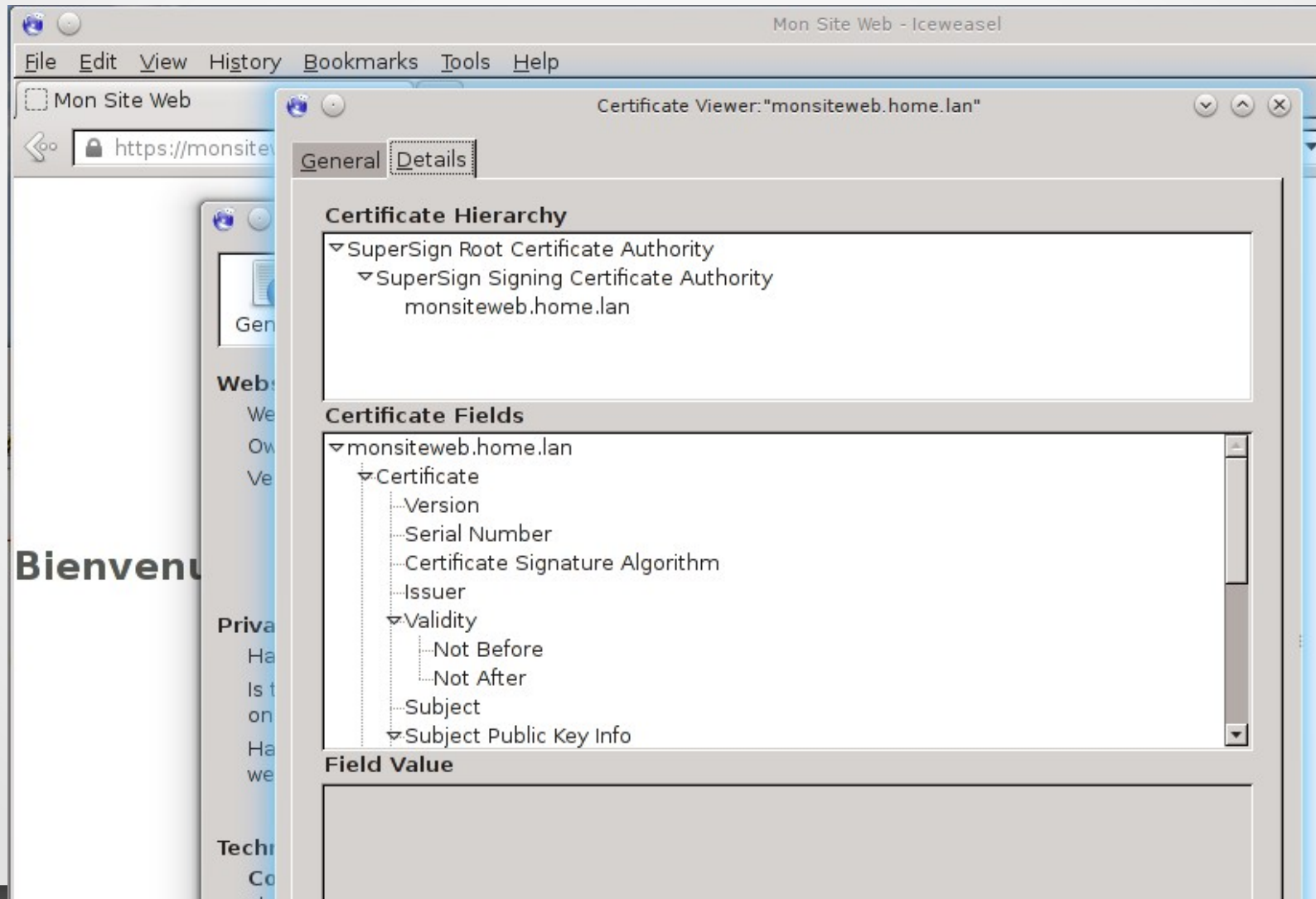
The page you are viewing was encrypted before being transmitted over the Internet. Encryption makes it very difficult for unauthorized people to view information traveling between computers. It is therefore very unlikely that anyone read this page as it traveled across the network.

Bienvenu

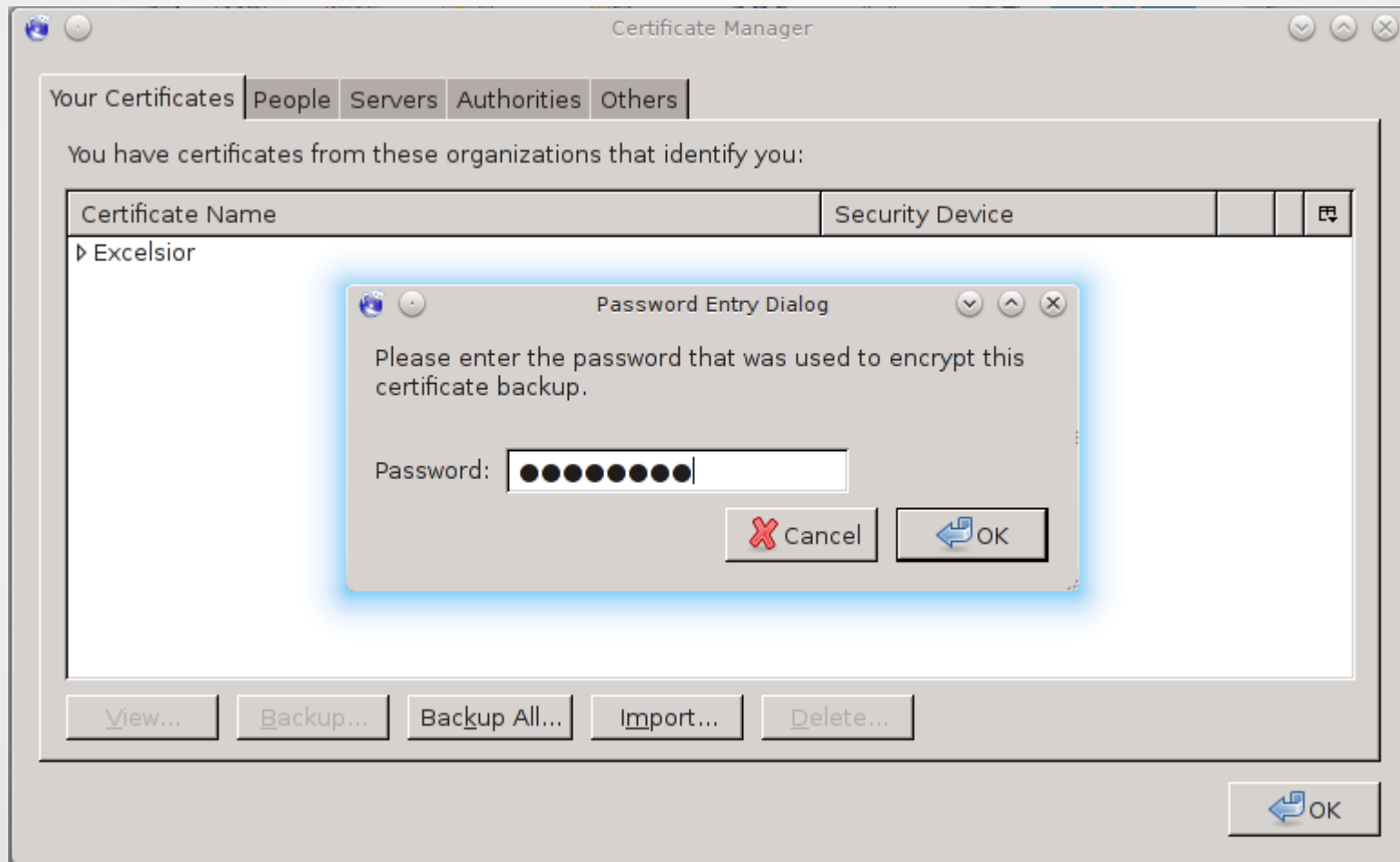
Déploiement du certificat serveur



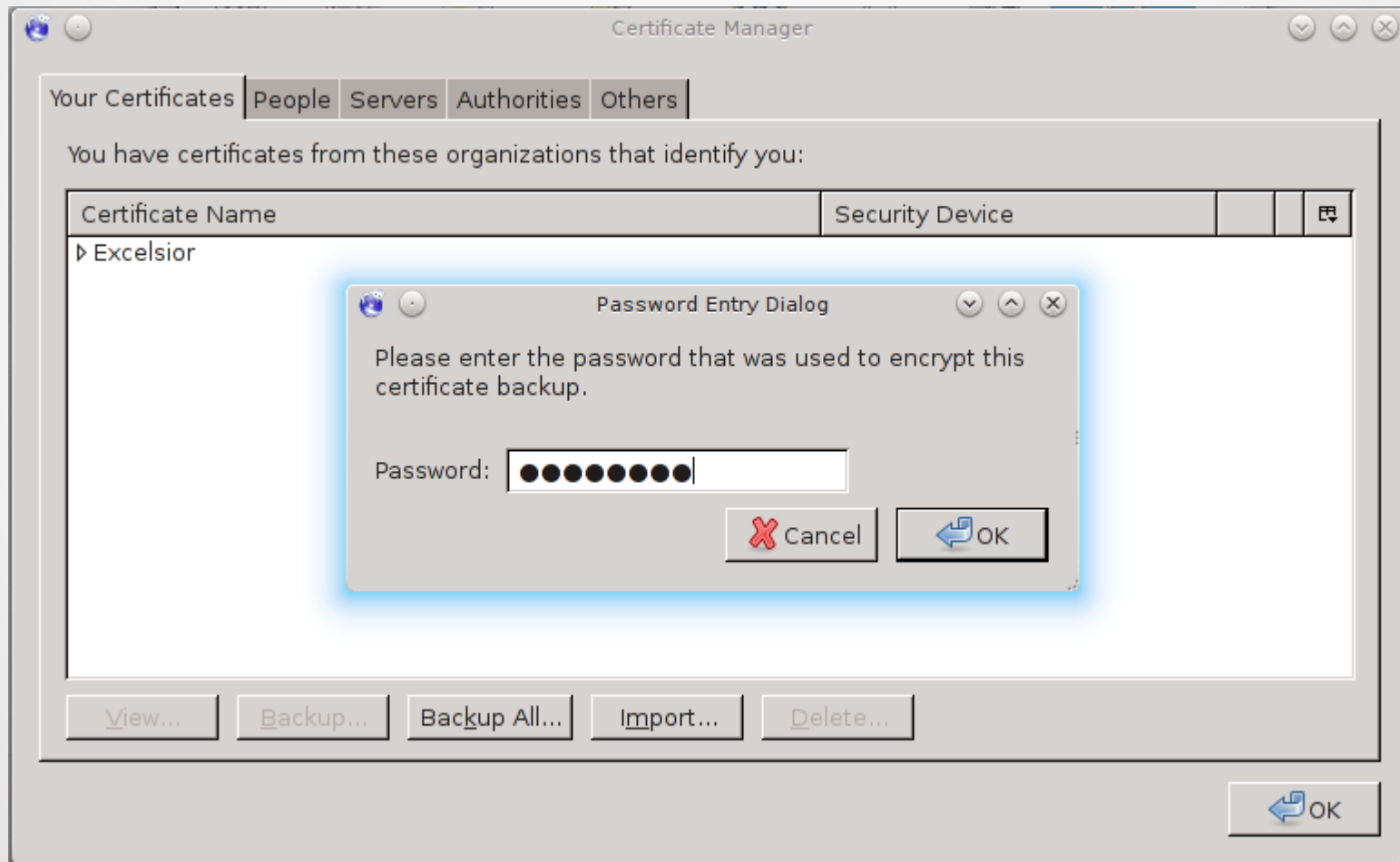
Déploiement du certificat serveur



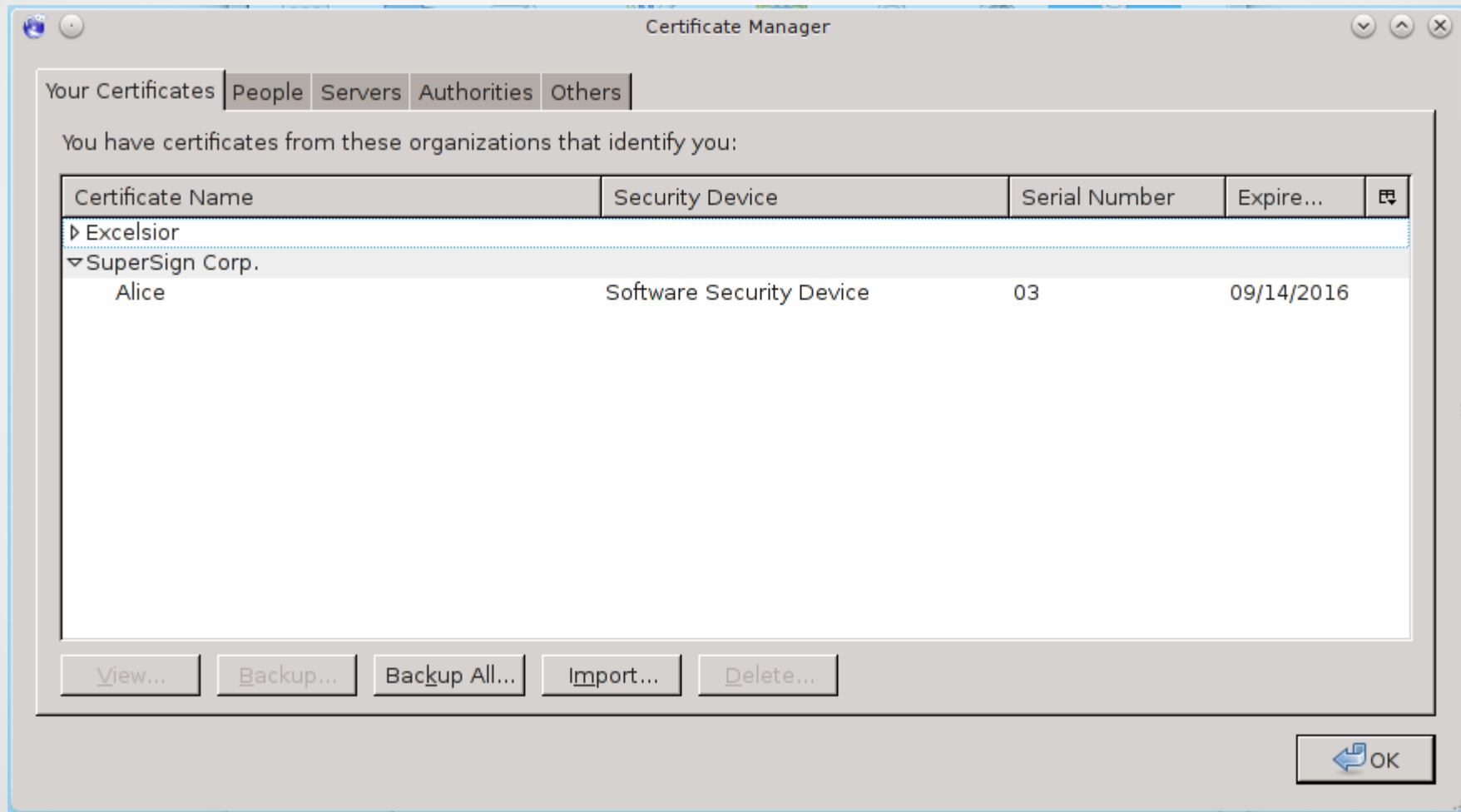
Déploiement du certificat d'Alice



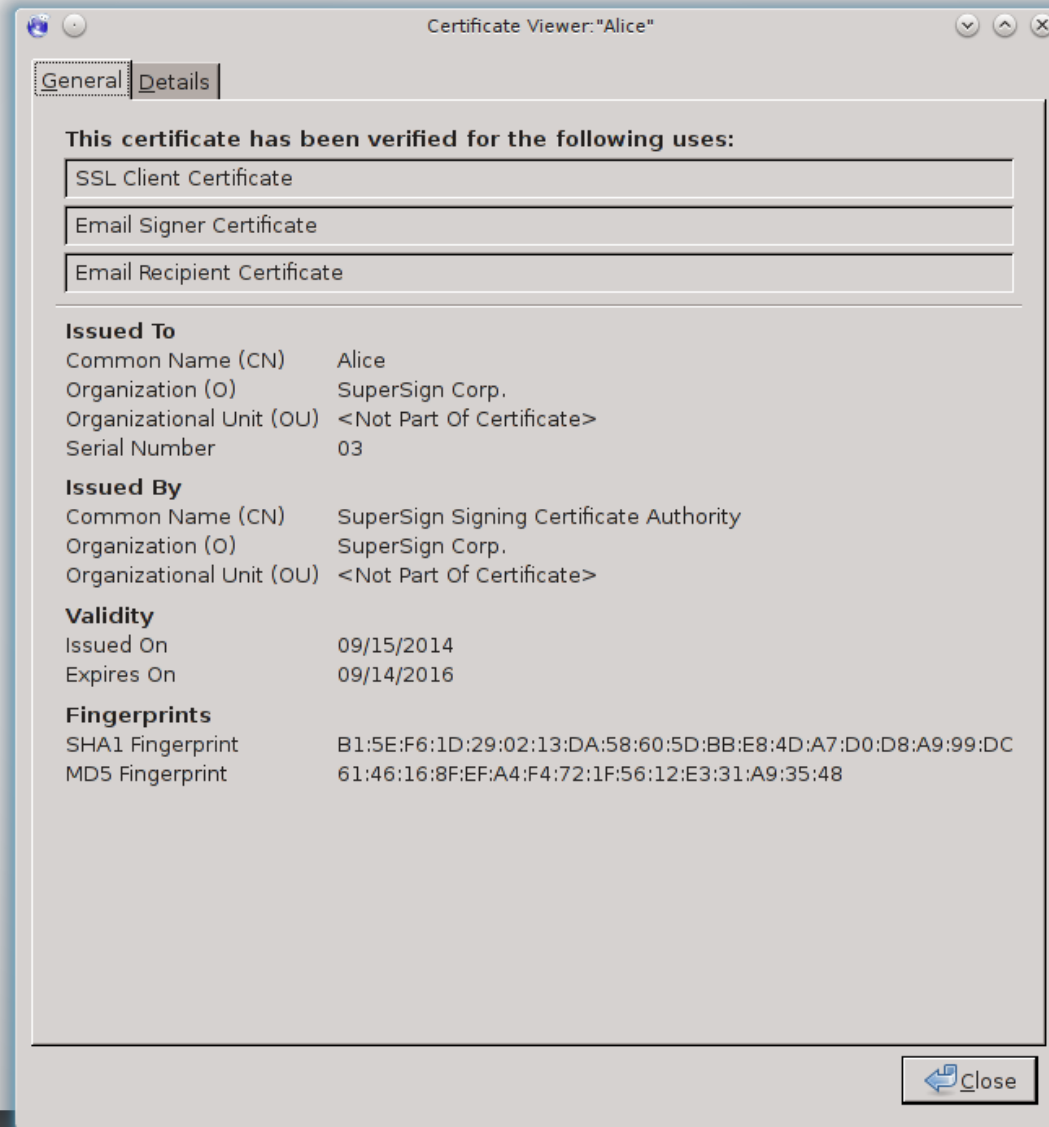
Déploiement du certificat d'Alice



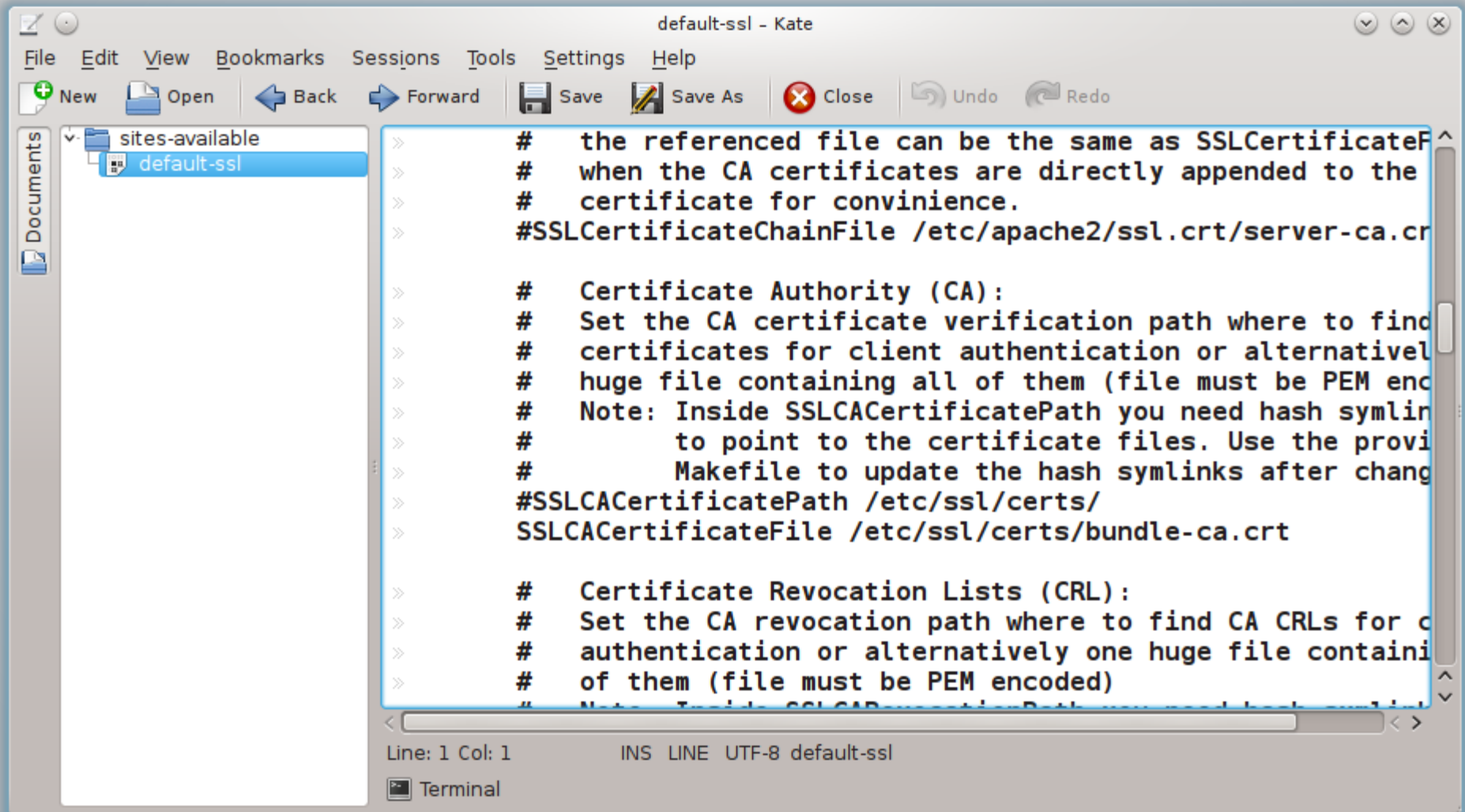
Déploiement du certificat d'Alice



Déploiement du certificat d'Alice



Déploiement du certificat d'autorité



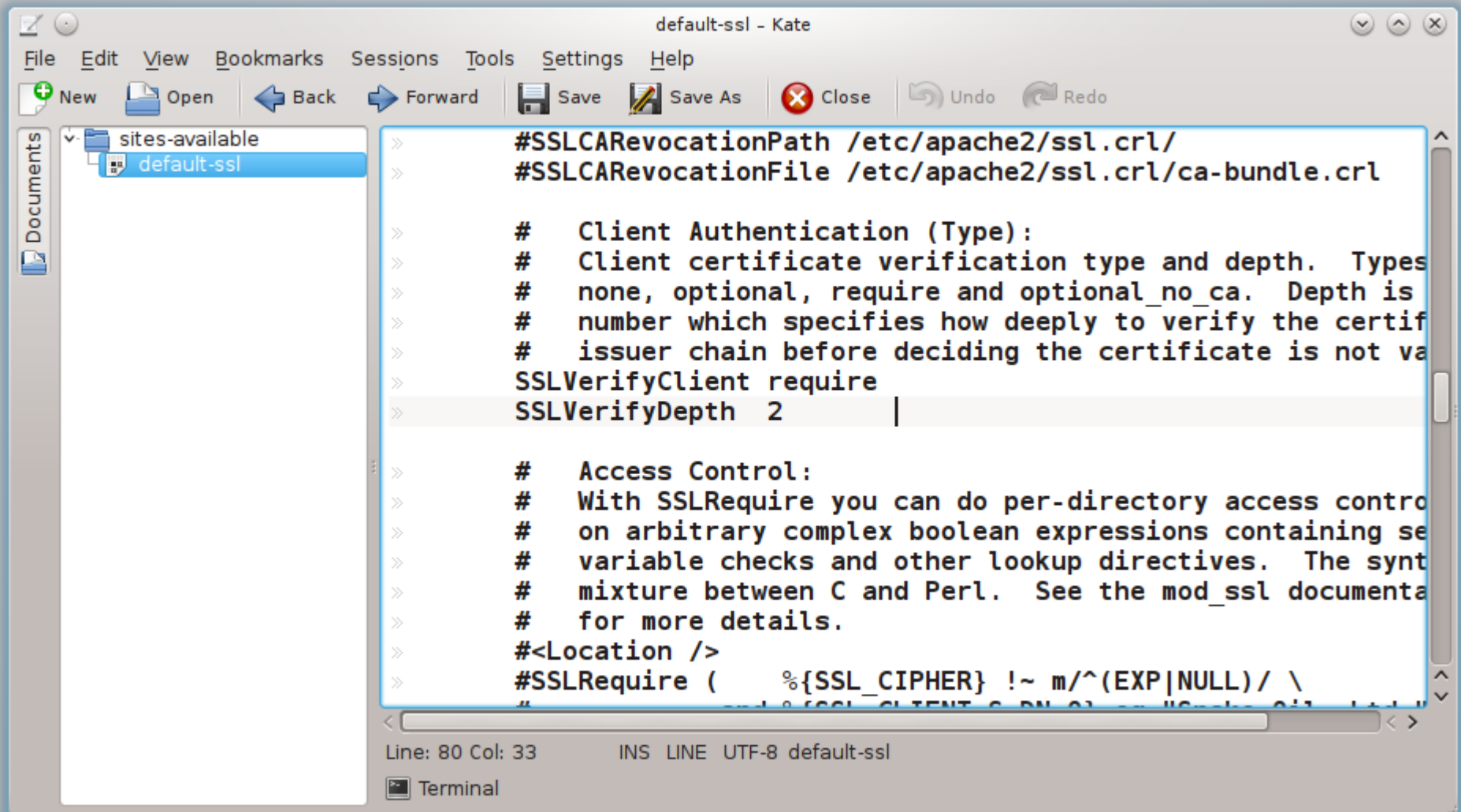
The image shows a text editor window titled "default-ssl - Kate". The window contains a configuration file with several commented-out lines. The left sidebar shows a file explorer with "sites-available" and "default-ssl" folders. The status bar at the bottom indicates "Line: 1 Col: 1" and "INS LINE UTF-8 default-ssl".

```
>> # the referenced file can be the same as SSLCertificateF
>> # when the CA certificates are directly appended to the
>> # certificate for convenience.
>> #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.cr

>> # Certificate Authority (CA):
>> # Set the CA certificate verification path where to find
>> # certificates for client authentication or alternativel
>> # huge file containing all of them (file must be PEM enc
>> # Note: Inside SSLCACertificatePath you need hash symlin
>> #         to point to the certificate files. Use the provi
>> #         Makefile to update the hash symlinks after chang
>> #SSLCACertificatePath /etc/ssl/certs/
>> SSLCACertificateFile /etc/ssl/certs/bundle-ca.crt

>> # Certificate Revocation Lists (CRL):
>> # Set the CA revocation path where to find CA CRLs for c
>> # authentication or alternatively one huge file containi
>> # of them (file must be PEM encoded)
>> # Note: Inside SSLCARevocationPath you need hash symlin
```

Déploiement du certificat d'autorité

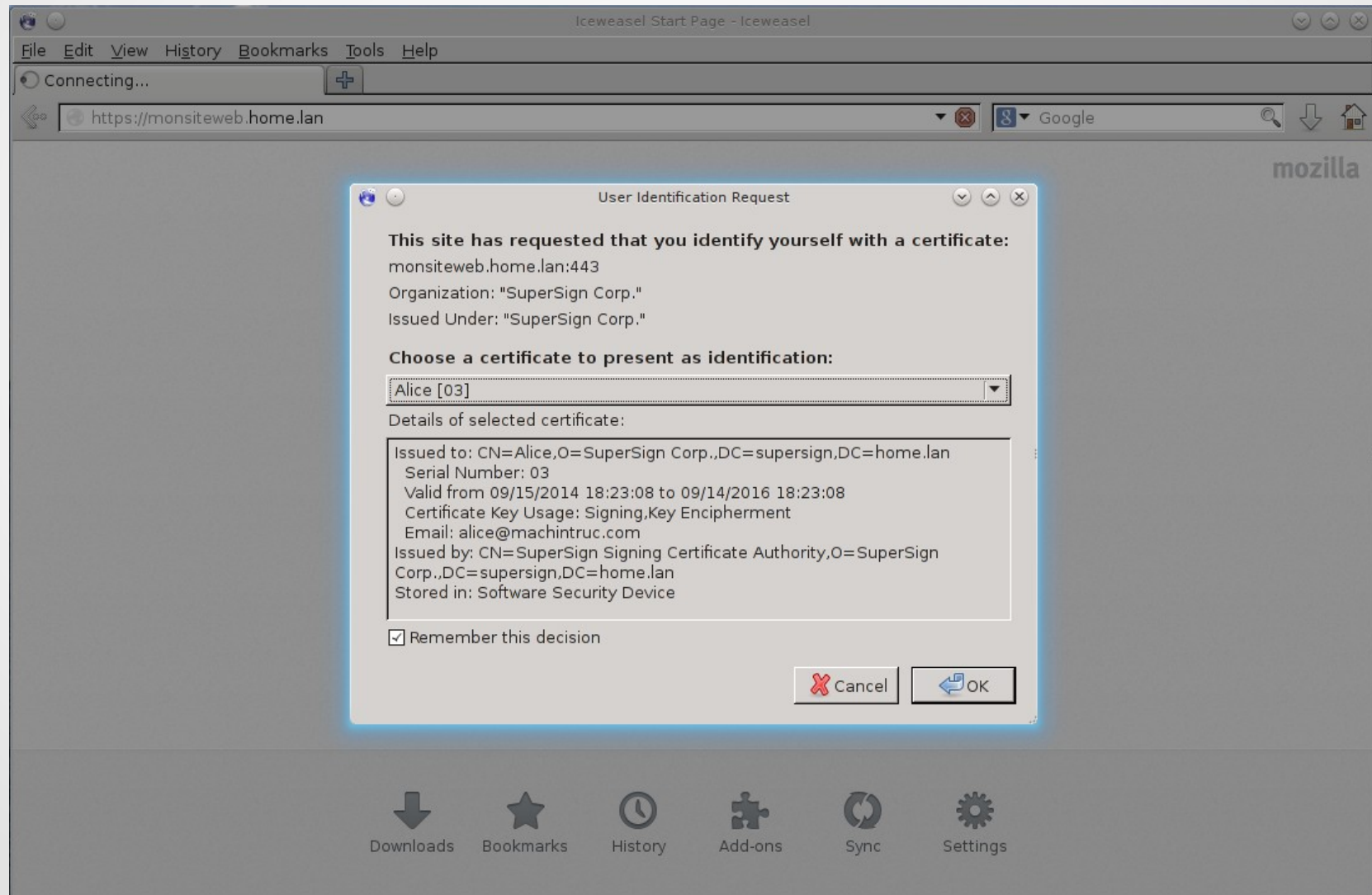


The image shows a text editor window titled "default-ssl - Kate". The left sidebar shows a file tree with "sites-available" and "default-ssl". The main editor area contains the following configuration:

```
>> #SSLCARevocationPath /etc/apache2/ssl.crl/  
>> #SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl  
  
>> # Client Authentication (Type):  
>> # Client certificate verification type and depth. Types  
>> # none, optional, require and optional_no_ca. Depth is  
>> # number which specifies how deeply to verify the certif  
>> # issuer chain before deciding the certificate is not va  
>> SSLVerifyClient require  
>> SSLVerifyDepth 2 |  
  
>> # Access Control:  
>> # With SSLRequire you can do per-directory access contro  
>> # on arbitrary complex boolean expressions containing se  
>> # variable checks and other lookup directives. The synt  
>> # mixture between C and Perl. See the mod_ssl documenta  
>> # for more details.  
>> #<Location />  
>> #SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
```

At the bottom, the status bar shows "Line: 80 Col: 33" and "INS LINE UTF-8 default-ssl". A "Terminal" icon is visible in the bottom left corner.

Déploiement du certificat d'autorité



Les certificats électroniques

American Express® Premier Rewards Gold Card - Iceweasel

File Edit View History Bookmarks Tools Help


American Express® Premier Re...

American Express Company (US) https://www304.americanexpress.com/credit-card/premier-rewards-gold/25330 american express

You are connected to **americanexpress.com** which is run by **American Express Company** Phoenix Arizona, US Verified by: VeriSign, Inc. Your connection to this website is encrypted to prevent eavesdropping. [More Information...](#)

ards Why American Express Respond to Your Mail Offer Your Special Card Offers Small Business Cards

oints
your new Card in



American Express® Premier Rewards Gold Card

Get a decision in as little as 60 seconds.

[Apply Now](#)

GOOD FOR
Earn points for the things you buy and use points for the things you want.†
†Benefit Terms


NO INTEREST CHARGES
No interest charges because you pay your balance in full each month.‡
‡Benefit Terms Rates and Fees

ANNUAL FEE
\$0 intro annual fee for the first year, then \$175.†
†Offer Terms Rates and Fees

HIGHLIGHTS BENEFITS COMPARE


Highlights of This Card

3X Points




Airfare
Get 3X points for flights booked directly

2X Points



Gas Stations & Supermarkets
Get 2X points at US gas stations and US

1X Points



Other Purchases
Get 1X points on other purchases.

[Chat With Us](#)

Les certificats électroniques

The screenshot shows a web browser window displaying the American Express Premier Rewards Gold Card page. The browser's address bar shows the URL: <https://www304.americanexpress.com/credit-card/premier-rewards-gold/25330>. The page features a navigation menu with options like "Get Started", "View All Cards", and "Compare Cards". A large banner at the top reads "Earn 25,000 Points" with subtext "after you spend \$1,000 on your first 3 months".

Overlaid on the page is a "Certificate Viewer" window titled "Certificate Viewer: 'www304.americanexpress.com'". The window has two tabs: "General" and "Details". The "General" tab is active, showing the following information:

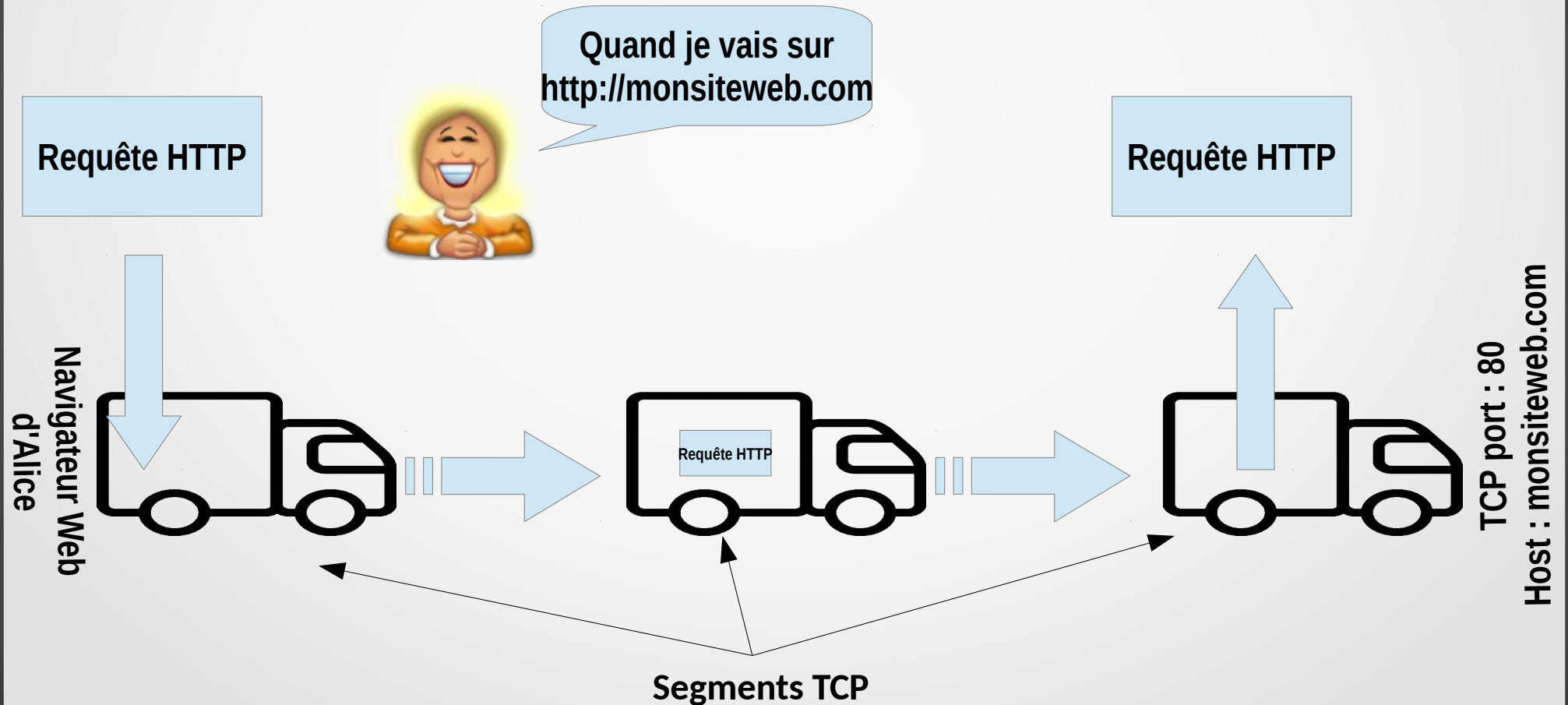
- Certificate Hierarchy:**
 - VeriSign Class 3 Public Primary Certification Authority - G5
 - VeriSign Class 3 Extended Validation SSL SGC CA
 - www304.americanexpress.com

- Certificate Fields:**
- www304.americanexpress.com
 - Certificate
 - Version
 - Serial Number
 - Certificate Signature Algorithm
 - Issuer
 - Validity
 - Not Before
 - Not After
 - Subject
 - Subject Public Key Info
- Field Value:** (Empty text area)
- Export...** button
- Close** button

Below the certificate viewer, a security information panel is visible, showing a lock icon and the text "Security". It displays the following information:

- Address: www304.americanexpress.com/credit-card/premier-rewards-g...
- Organization: American Express Company, Inc.
- Buttons: [View Certificate](#), [View Cookies](#), [View Saved Passwords](#)
- Security status: "High-grade Encryption (RC4, 128 bit keys)"
- Text: "Information was encrypted before being transmitted over the Internet. It is therefore very difficult for unauthorized people to view information traveling over the Internet. It is therefore very unlikely that anyone read this page as it traveled over the Internet."
- Section: **Other Purchases**
- Text: "Get 1X points on other purchases."

La suite Internet ultra-simplifiée



Secure Socket Layer (SSL/TLS)

- Le nom actuel est Transport Layer Security mais l'ancien nom persiste
- C'est un ensemble de protocoles qui appliquent la cryptographie à clé publique
- On le retrouve partout ! (web, e-mail, messagerie instantannée, VoIP, ...)



C'est moi qui mets en oeuvre tous les principes
vus précédemment!

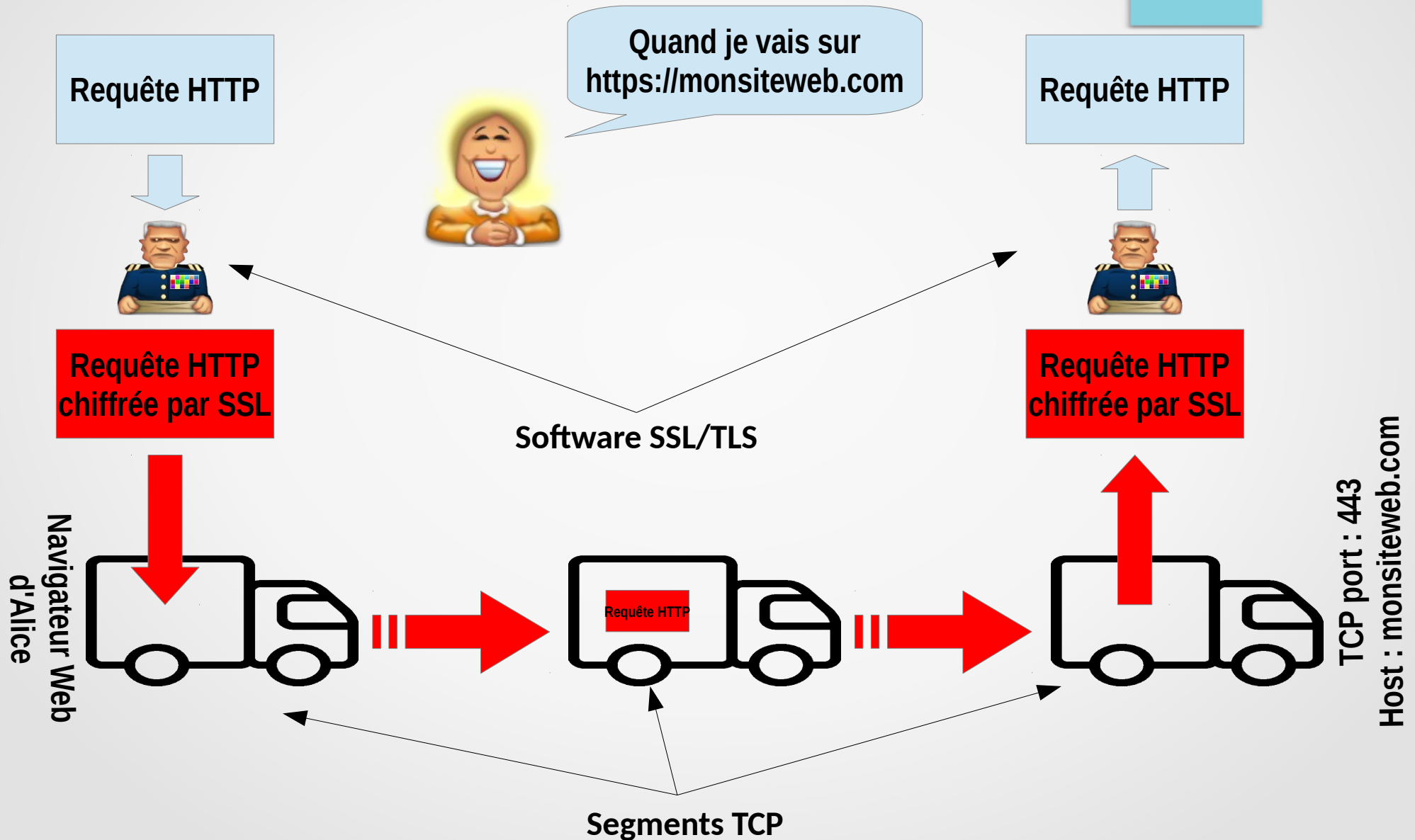
Secure Socket Layer (SSL/TLS)

- Assure l'authentification des deux parties : *handshake*
- Procède à un échange de clé symétriques temporaires
- Les messages sont ensuite chiffrés, signés et échangés par cryptographie symétrique

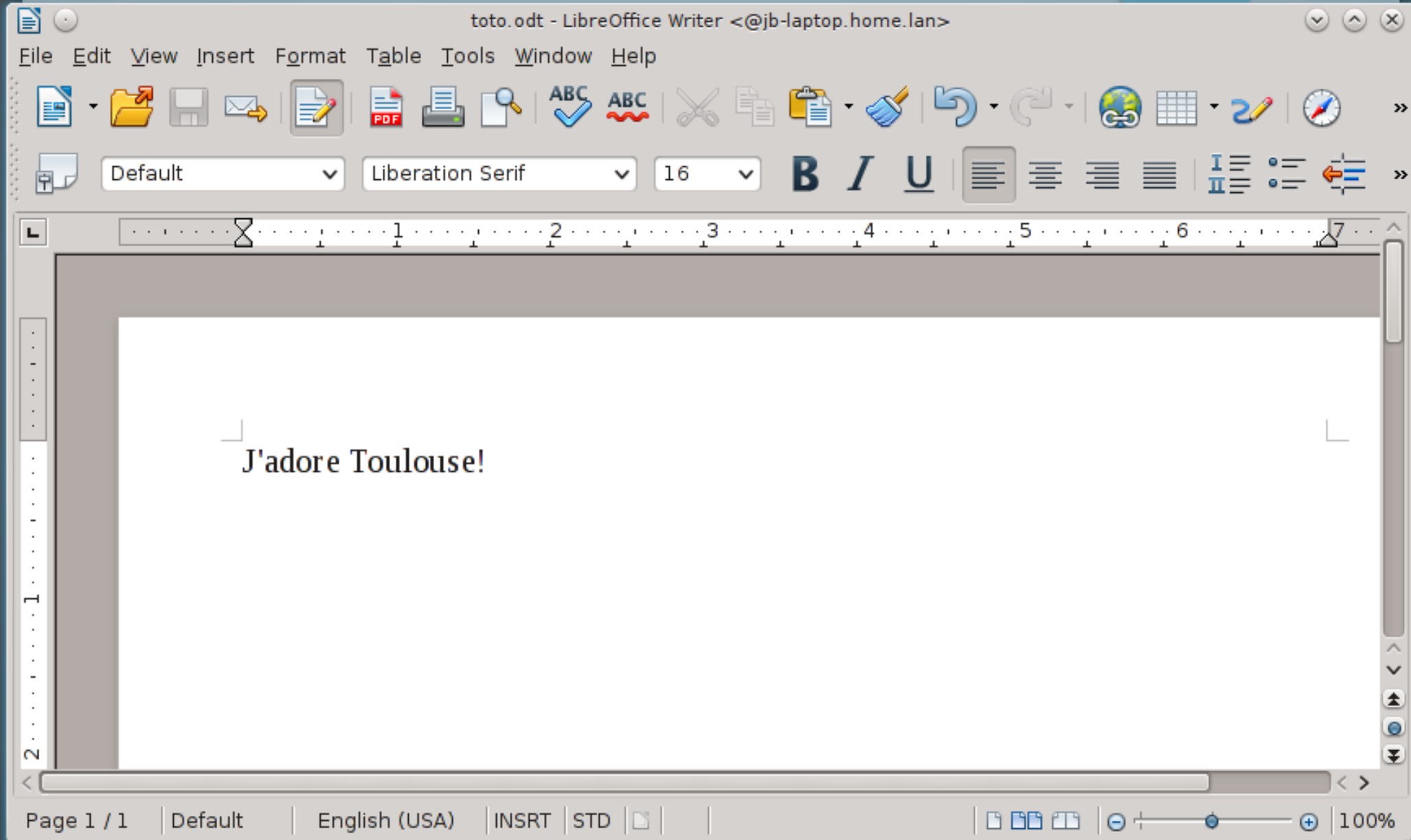


A vos ordres !

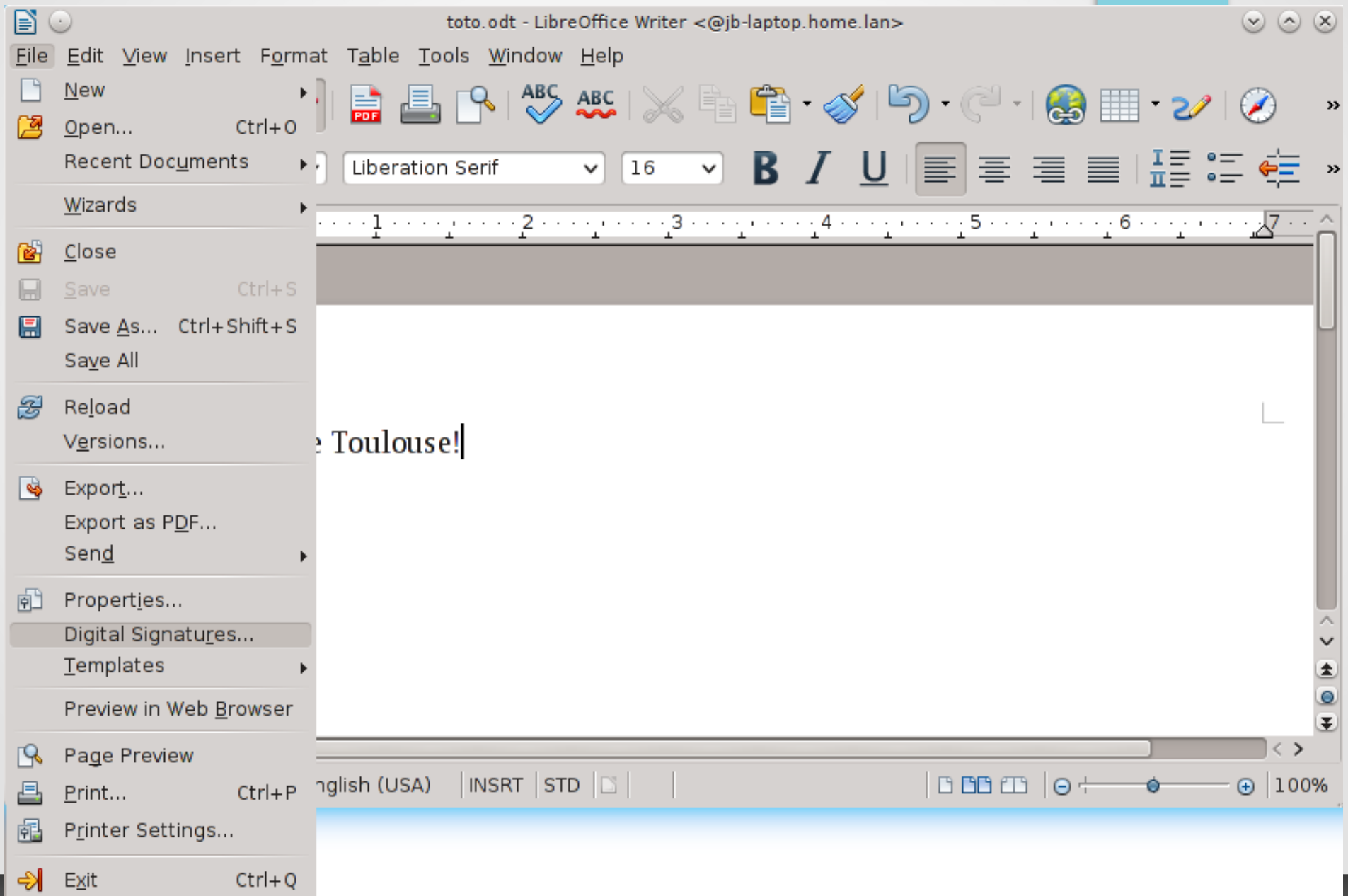
Le protocole HTTPS



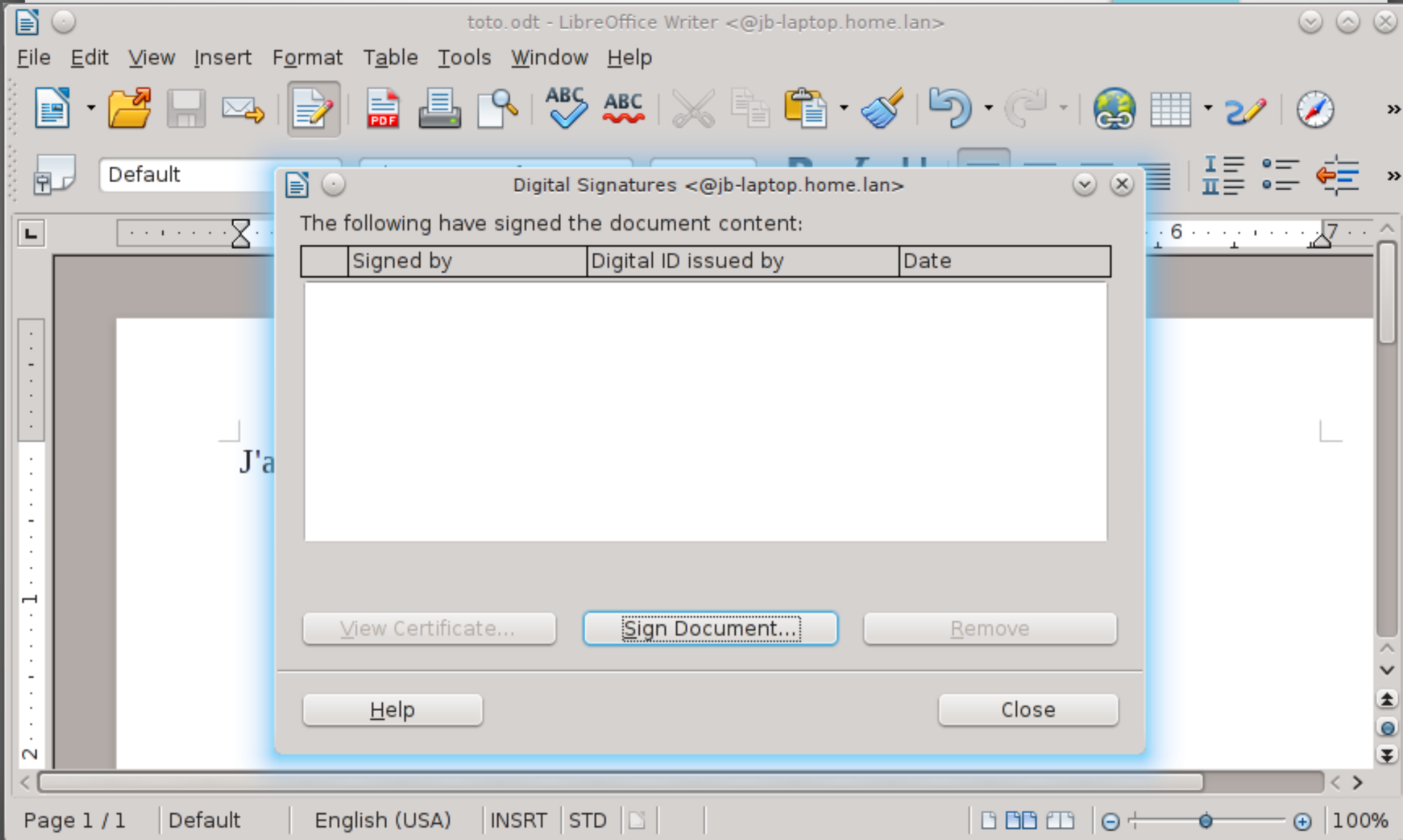
Signer un document avec un utilitaire intégré



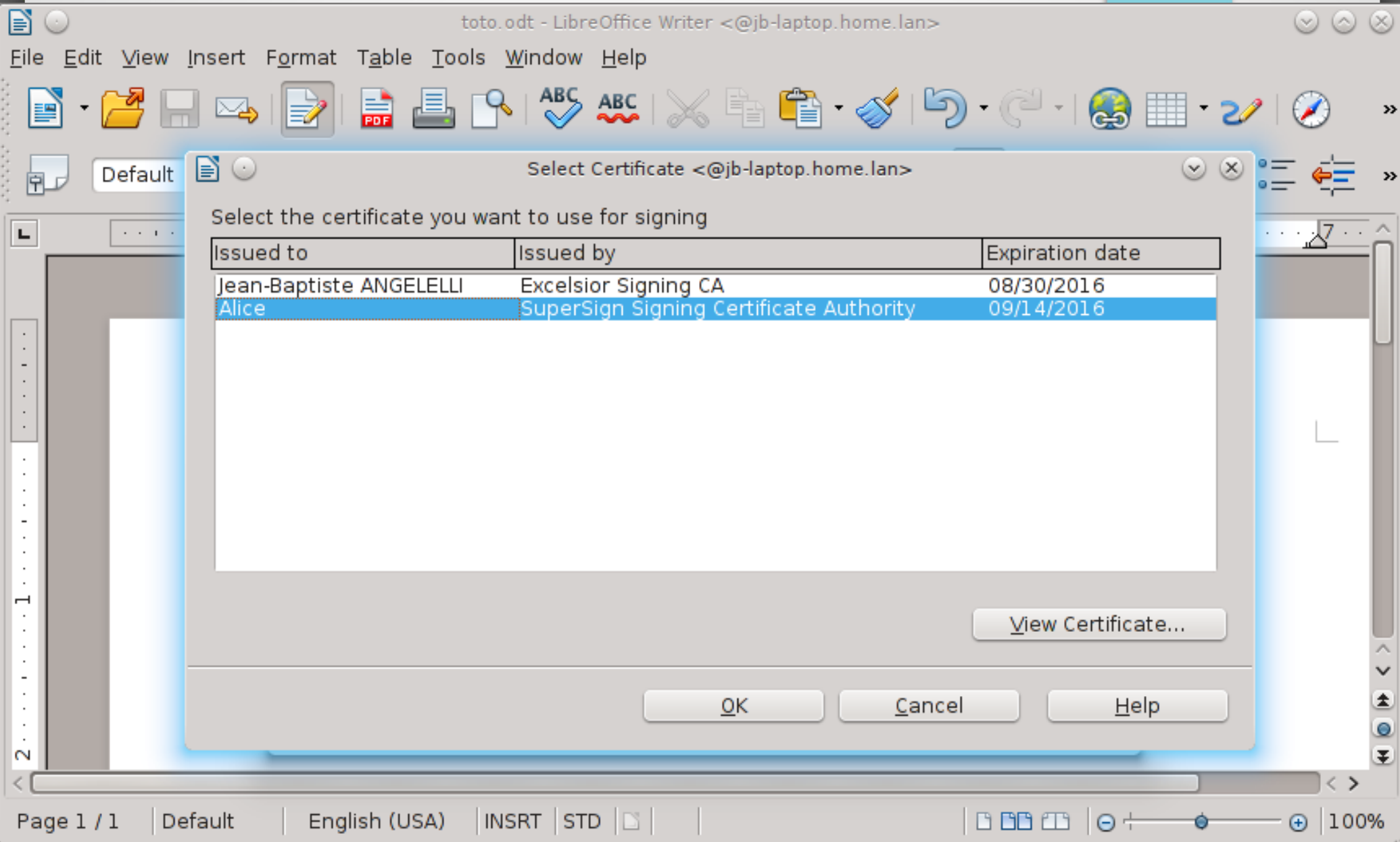
Signer un document avec un utilitaire intégré



Signer un document avec un utilitaire intégré



Signer un document avec un utilitaire intégré



Signer un document avec un utilitaire intégré

The image shows a screenshot of the LibreOffice Writer application window. The title bar reads "toto.odt - LibreOffice Writer <@jb-laptop.home.lan>". The menu bar includes "File", "Edit", "View", "Insert", "Format", "Table", "Tools", "Window", and "Help". The toolbar contains various icons for file operations, editing, and formatting. A dialog box titled "Digital Signatures <@jb-laptop.home.lan>" is open in the foreground. The dialog box contains the following text and elements:

The following have signed the document content:

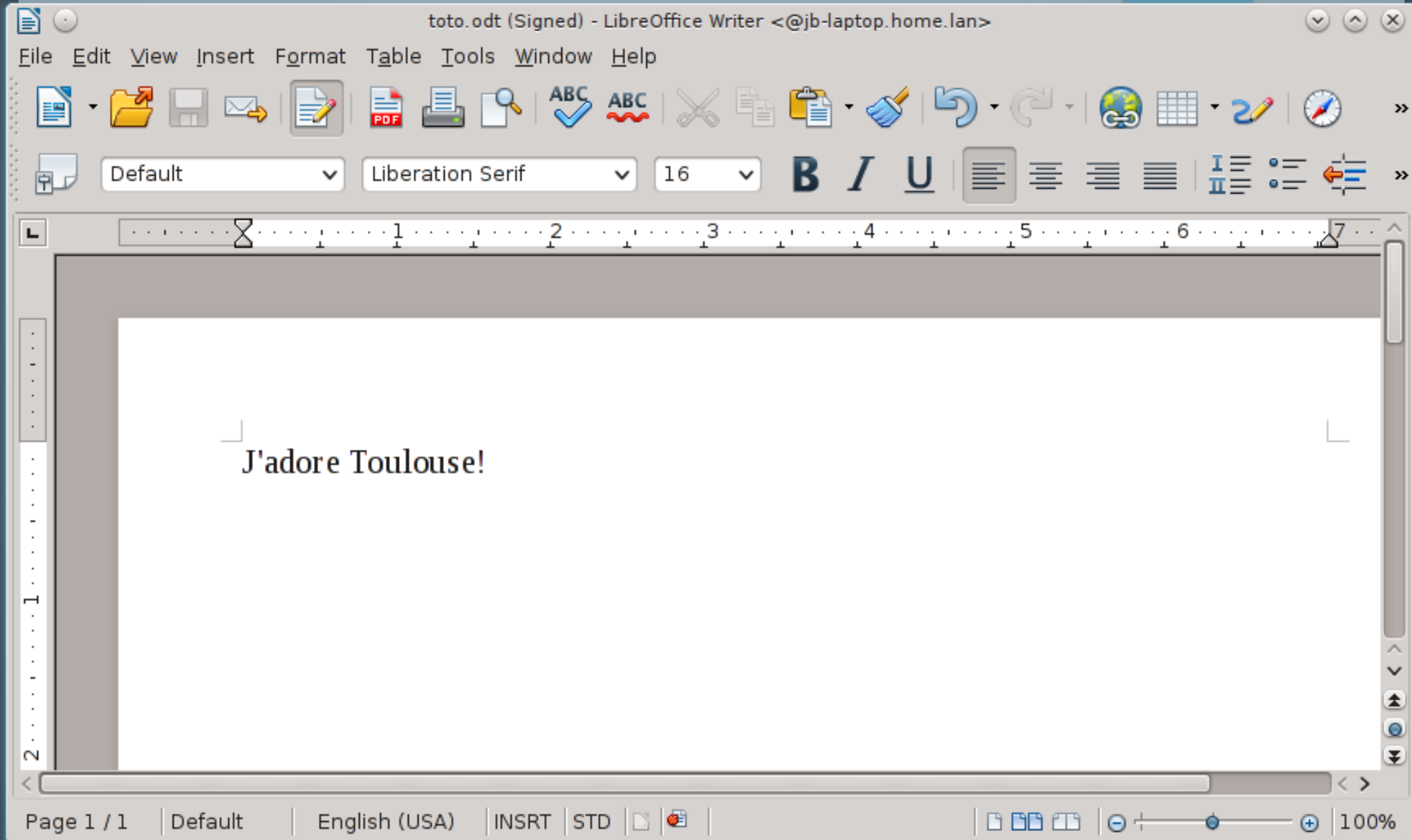
| Signed by | Digital ID issued by | Date |
|-----------|-------------------------------|---------------------|
| Alice | SuperSign Signing Certificate | 09/16/2014 20:08:15 |

The signatures in this document are valid

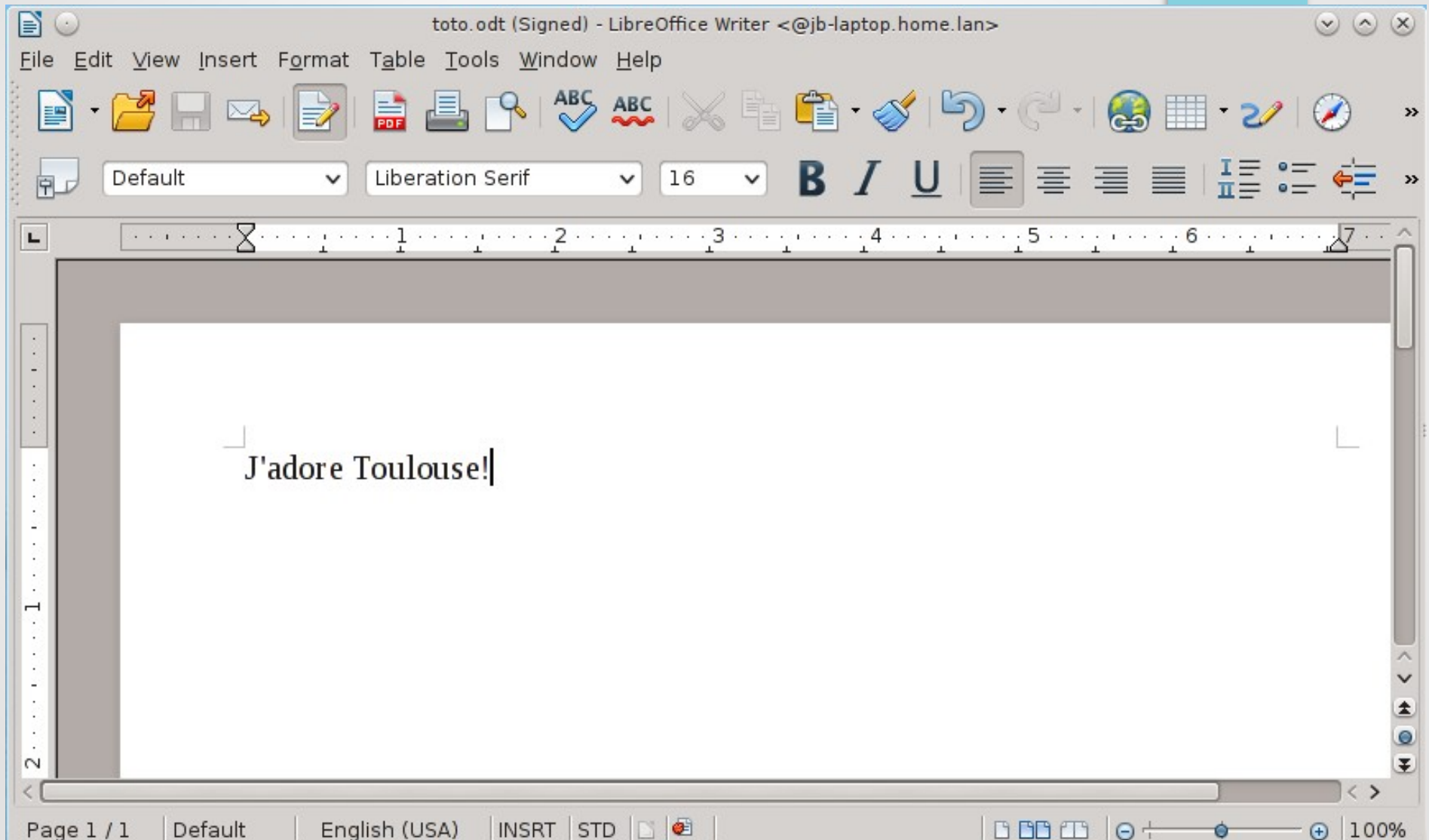
Buttons: View Certificate..., Sign Document..., Remove, Help, Close

Page 1 / 1 | Default | English (USA) | INSRT | STD | 100%

Signer un document avec un utilitaire intégré



Signer un document avec un utilitaire intégré



Digital Signature: The document signature is OK.

Signer un document avec un utilitaire intégré

The image shows a LibreOffice Writer window titled "toto.odt (Signed) - LibreOffice Writer <@jb-laptop.home.lan>". The interface includes a menu bar (File, Edit, View, Insert, Format, Table, Tools, Window, Help) and a toolbar with various icons. A "Digital Signatures" dialog box is open, displaying the text "The following have signed the document content:" and a table with the following content:

| Signed by | Dig |
|-----------|-----|
| Alice | Su |

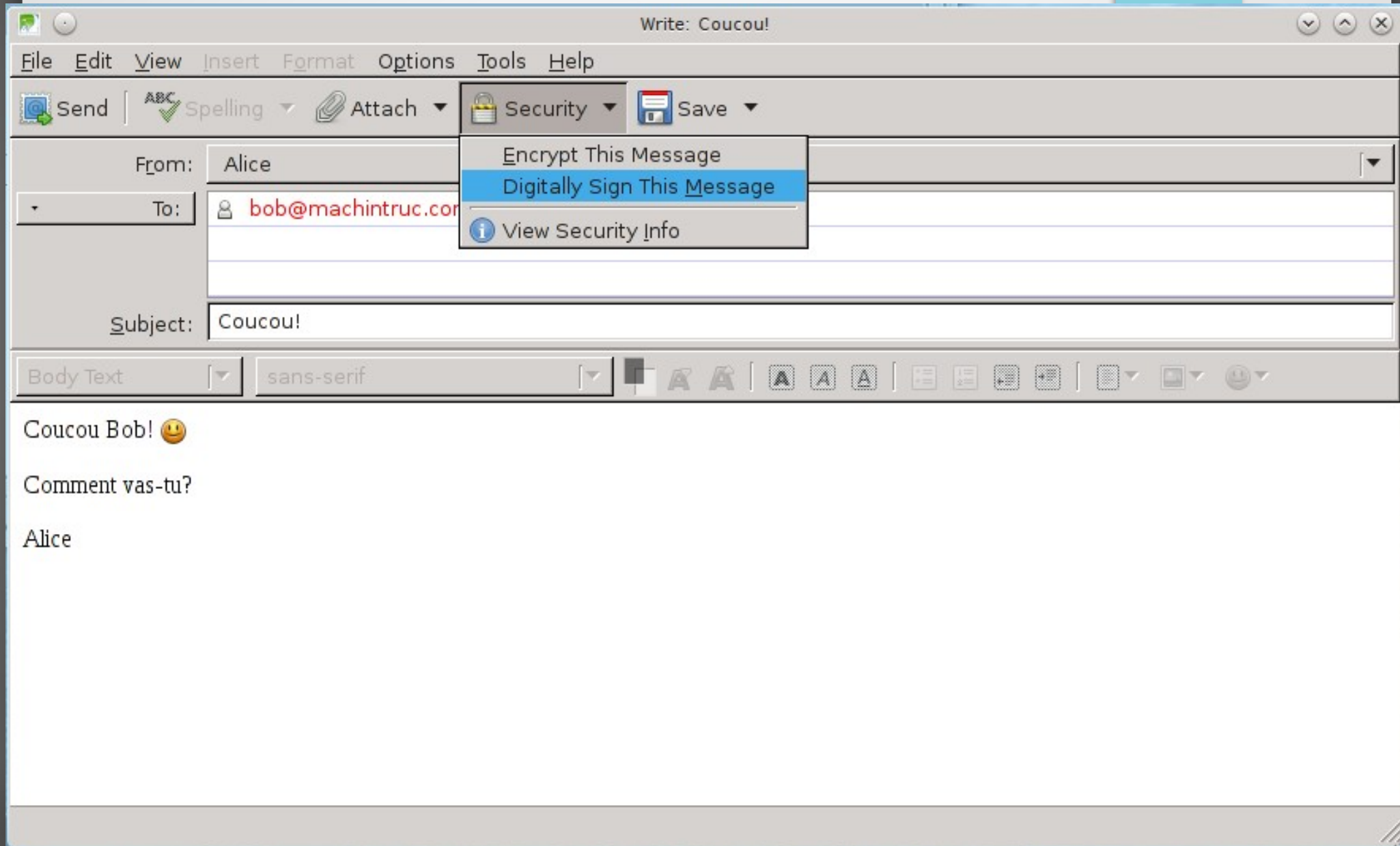
Below the table, there is a section titled "The signatures in this document" and a "View Certificate..." button. A "View Certificate" dialog box is also open, showing the "Certification Path" tab. The path is:

- SuperSign Root Certificate Authority
 - SuperSign Signing Certificate Authority
 - Alice

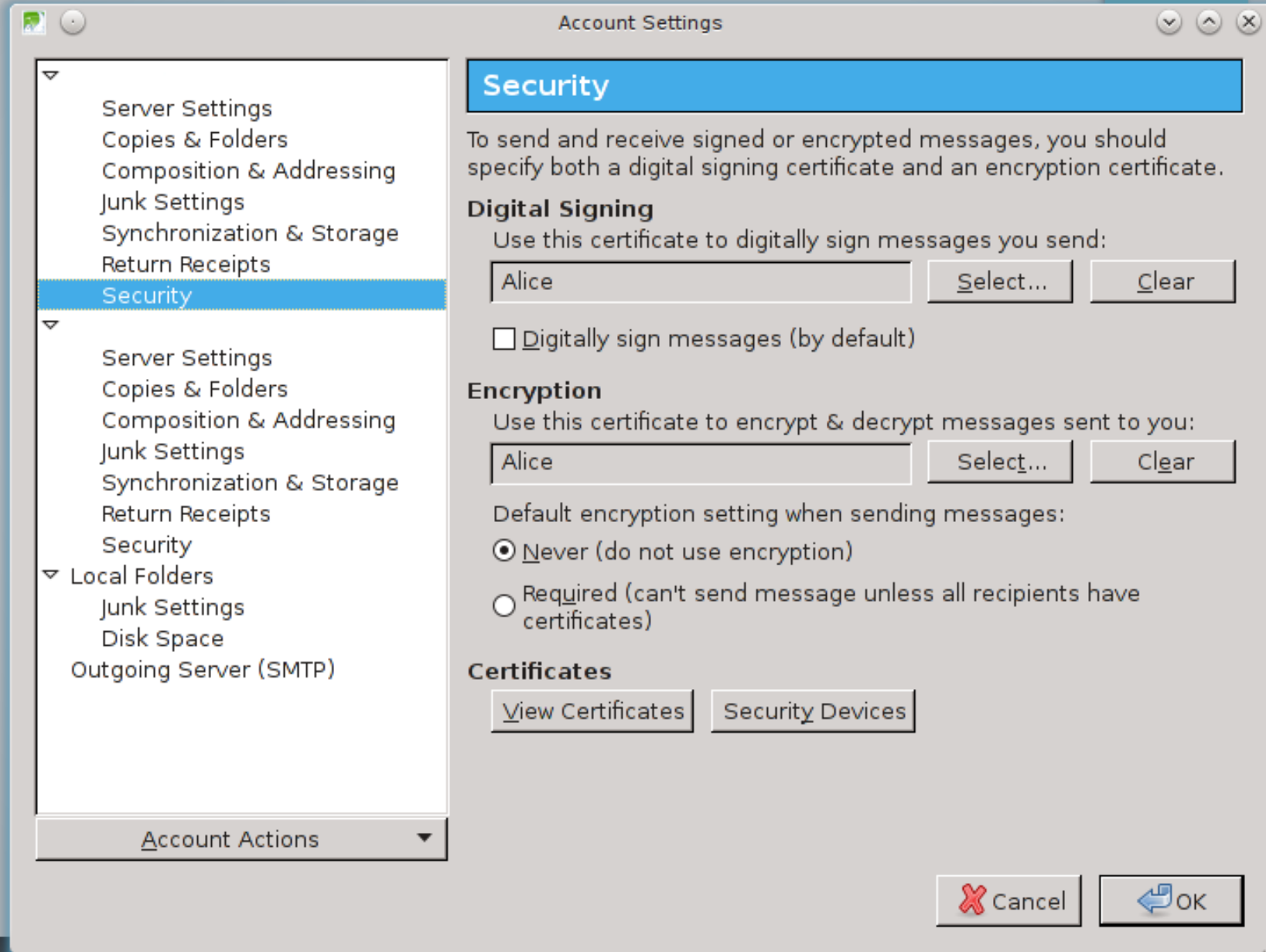
The "Certification status" section shows "The certificate is OK." and a "View Certificate..." button. The "View Certificate" dialog has "OK" and "Help" buttons at the bottom.

At the bottom of the LibreOffice window, the status bar shows "Page 1 / 1", "Default", "English (USA)", "INSRT", and "STD".

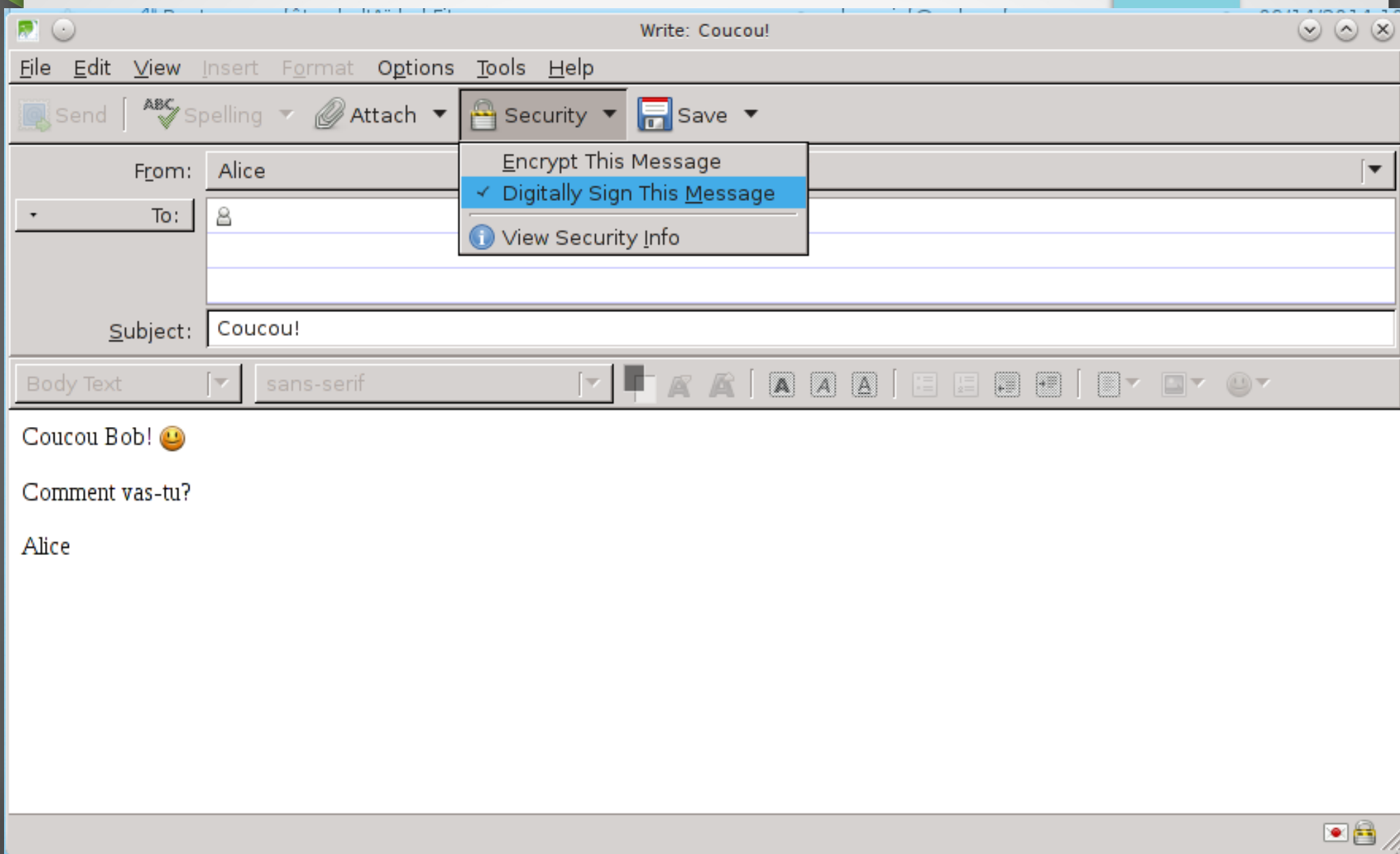
La signature d'e-mail



La signature d'e-mail



La signature d'e-mail



La signature d'e-mail

The screenshot displays the Icedove Mail/News interface. The window title is "Coucou! - Inbox -" and the application name is "- Icedove Mail/News". The interface includes a toolbar with icons for "Get Mail", "Write", "Chat", "Address Book", "Tag", and "Quick Filter". A search bar contains the text "Search... <Ctrl+K>". The email header shows "From Alice", "Subject Coucou!", and "To Bob". Action buttons for "Reply", "Forward", "Archive", "Junk", and "Delete" are visible. The email content reads: "Coucou Bob! 😊", "Comment vas-tu?", and "Alice". The time "09:41 PM" and "Other Actions" are also shown.

Coucou! - Inbox - - Icedove Mail/News

Coucou! - Inbox

Get Mail Write Chat Address Book Tag Quick Filter Search... <Ctrl+K>

From Alice
Subject **Coucou!**
To Bob

Reply Forward Archive Junk Delete

09:41 PM
Other Actions

Coucou Bob! 😊

Comment vas-tu?

Alice

La signature d'e-mail

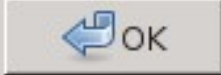
Message Security

Message Is Signed
This message includes a valid digital signature. The message has not been altered since it was sent.

Signed by: Alice
Email address: alice@machintruc.com
Certificate issued by: SuperSign Signing Certificate Authority

[View Signature Certificate](#)

Message Not Encrypted
This message was not encrypted before it was sent. Information sent over the Internet without encryption can be seen by other people while in transit.

 OK

From Me <jb.a...

Subject **Coucou!**

To Me <jb_a...

Coucou Bob! 😊

Comment vas-tu?

Alice

Coucou! - Inbox - jb_angelelli@yahoo.ca - Icedove Mail/News

Inbox - jb_angelelli@yah...

Coucou! - Inbox - jb_...

Get Mail

Write

Chat

Address Book

Tag

Quick Filter

Search <Ctrl+K>

Junk

Delete



09:41 PM

Other Actions



Les mathématiques de la cryptographie

- Factorisons en nombres premiers $15 =$

Les mathématiques de la cryptographie

- Factorisons en nombres premiers $15 = 3 * 5$

Les mathématiques de la cryptographie

- Factorisons en nombres premiers $15 = 3 * 5$
- Factorisons en nombres premiers $143 =$

Les mathématiques de la cryptographie

- Factorisons en nombres premiers $15 = 3 * 5$
- Factorisons en nombres premiers $143 = 11 * 13$

Les mathématiques de la cryptographie

- Factorisons en nombres premiers $15 = 3 * 5$
- Factorisons en nombres premiers $143 = 11 * 13$
- Factorisons en nombres premiers $1517 =$

Les mathématiques de la cryptographie

- Factorisons en nombres premiers $15 = 3 * 5$
- Factorisons en nombres premiers $143 = 11 * 13$
- Factorisons en nombres premiers $1517 = 37 * 41$

Les mathématiques de la cryptographie

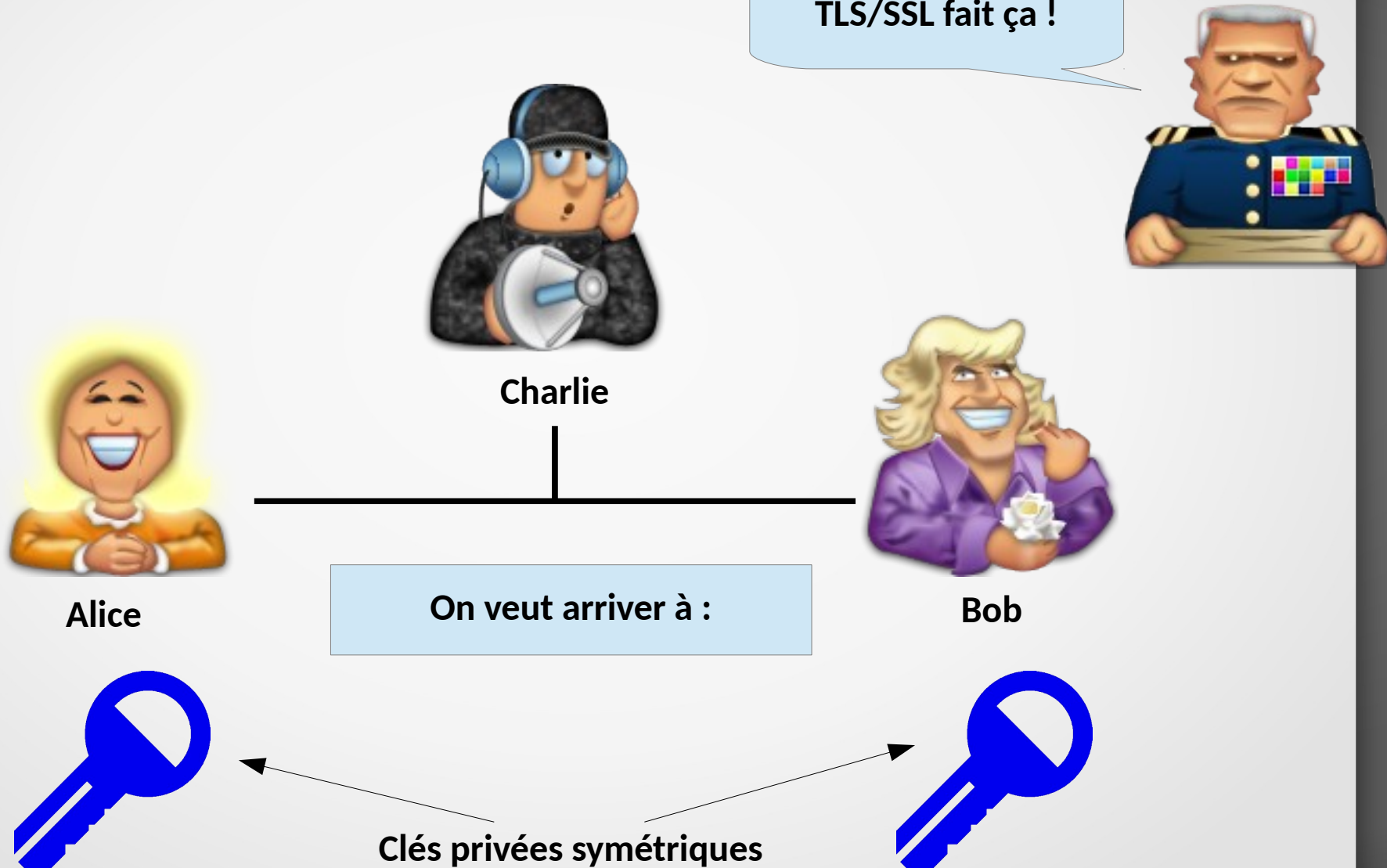
- Factorisons en nombres premiers $15 = 3 * 5$
- Factorisons en nombres premiers $143 = 11 * 13$
- Factorisons en nombres premiers $1517 = 37 * 41$

L'idée est qu'il est très difficile de factoriser des grands nombres.

En fait, c'est un problème NP-difficile, on ne connaît pas de méthode polynomiale.

L'échange de clés de Diffie-Hellmann

TLS/SSL fait ça !



L'échange de clés de Diffie-Hellmann

- Alice et Bob choisissent un nombre premier p et un nombre g (appelé base).
- Alice choisit un nombre au hasard a et fait $A = g^a \pmod{p}$, elle l'envoie à Bob.
- Bob choisit un nombre au hasard b et fait $B = g^b \pmod{p}$, il l'envoie à Alice.
- Bob fait A^b et il obtient $K (= (g^a)^b = g^{ab} \pmod{p})$
- Alice fait B^a et elle obtient K aussi !
($= (g^b)^a = g^{ab} \pmod{p}$)

L'algorithme RSA



- Algorithme à clé publique
- Rivest, Shamir et Adleman
- Publié pour la première fois en 1977
- Inventé secrètement par Clifford Cocks en 1973 pour le compte du GCHQ

L'algorithme RSA

- Soit p, q deux (grands) nombres premiers
- $n = pq$, puis $\varphi(n) = (p-1)(q-1)$
- Soit e tel que $1 < e < \varphi(n)$ avec $\text{pgcd}(e, \varphi(n)) = 1$
- Soit $d = e^{-1} \pmod{\varphi(n)}$

n et e forment la clé publique

d est la clé privée

L'algorithme RSA

- Soit m l'entier clair
- Pour chiffrer, on fait $c = m^e \pmod{n}$
- Pour déchiffrer, on fait $m = c^d \pmod{n}$

- La force de l'algorithme vient du fait qu'il est très difficile de retrouver p et q à partir de n !

n et e forment la clé publique

d est la clé privée

L'algorithme RSA

- Soit m l'entier clair
- Pour chiffrer, on fait $c = m^e \pmod{n}$
- Pour déchiffrer, on fait $m = c^d \pmod{n}$

- La force de l'algorithme vient du fait qu'il est très difficile de retrouver p et q à partir de n !



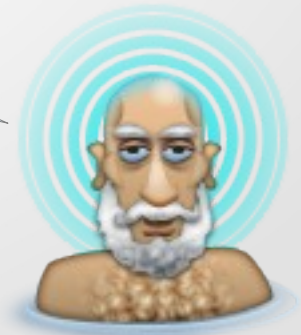
Attention aux ordinateurs
quantiques

Chiffrement par blocs



Bon ok ! Je sais chiffrer un nombre..
Maintenant comment je fais pour chiffrer un document entier?

Il te faut un chiffrement par blocs...



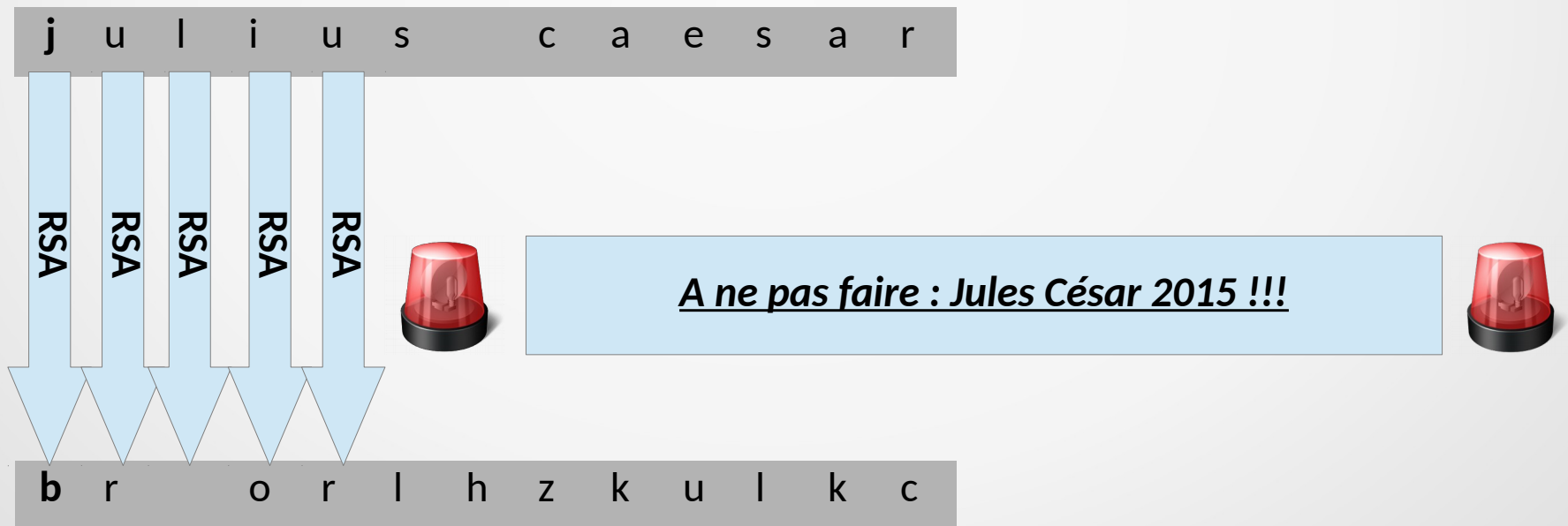
Chiffrement par blocs



- Une des deux grandes catégories de chiffrement (l'autre est le chiffrement par flot)
- Découpe l'objet à chiffrer en blocs de taille fixée (par exemple 128 bits)
- Un chiffrement par blocs est tout simplement une fonction bijective qui à un bloc de 128 bits clair associe un bloc de 128 bits chiffré

Chiffrement par blocs : attention aux utilisations trop naïves !

Blocs trop petits...



Chiffrement par blocs : attention aux utilisations trop naïves

Electronic Codebook (ECB) : Blocs chiffrés indépendamment les uns des autres.

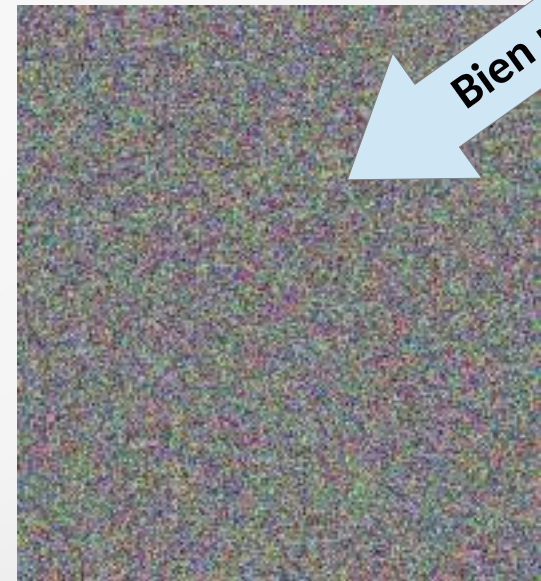
Chain block cipher (CBC) : On combine (par XOR) le clair avec le chiffré du bloc précédent avant chiffrement



Larry Ewing
lewing@isc.tamu.edu
The Gimp



Avec ECB



Avec CBC

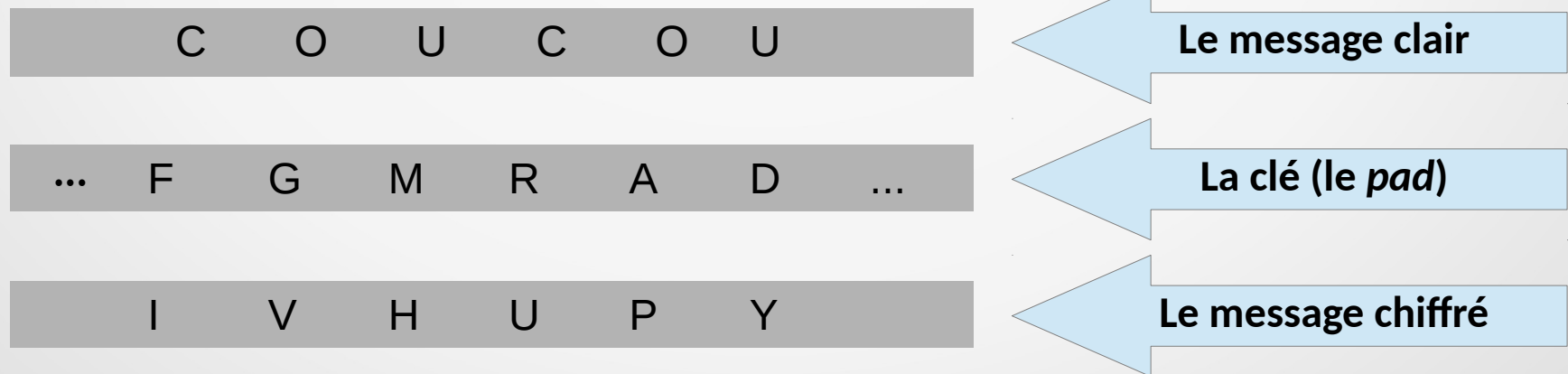
Pas terrible

Bien mieux

Le One Time Pad

On obtient le chiffré en additionnant le clair et la clé (modulo quelque chose)

Dans cet exemple, on additionne les lettres modulo 26



Le One Time Pad

- Si la clé est une suite complètement aléatoire, alors le système est parfaitement (mathématiquement) sûr (démonstré par Claude Shannon).
- L'idée est que si la clé est inconnue mais rigoureusement aléatoire, alors pour un chiffré donné, tous les clairs possibles sont équi-probables.
- Le système a une importante utilisation historique au XX^{ème} siècle.
- La génération d'une clé aléatoire est une importante limitation en pratique.

La Commission Nationale Informatique et Libertés

- Commission de l'état
- S'assurer que les systèmes informatiques ne portent pas atteinte aux droits de l'homme, aux libertés individuelles et publiques, à la vie privée
- Droit d'information, droit d'accès, droit de rectification ou de radiation, droit d'opposition, droit d'accès indirect

Remerciements pour les icônes

- www.iconarchive.com
- <http://www.afterglow.ie/icons.html>



- <http://www.aesigncontest.com/>
- <http://icons8.com/>



- <http://www.aha-soft.com/>
- <http://www.psdgraphics.com>



Et voilà! Questions?

Merci de votre attention !

jean-baptiste.angelelli@centraliens.net