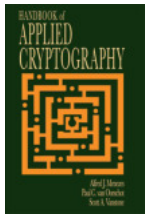
Architectures PKI

Sébastien VARRETTE

Université du Luxembourg - Laboratoire LACS, LUXEMBOURG
CNRS/INPG/INRIA/UJF - Laboratoire LIG-IMAG
Sebastien.Varrette@imag.fr
http://www-id.imag.fr/~svarrett/

Cours "Cryptographie & Securité Réseau" Master Info
Université de Yaoundé

Quelques références bibliographiques. . . (avant que j'oublie)



Menezes A. J., Vanstone S. A. and Oorschot P. C. V., *Handbook of Applied Cryptography*, Computer Sciences Applied Mathematics Engineering, CRC Press, Inc., 1st edition, 1996,

http://www.cacr.math.uwaterloo.ca/hac/

Schneier B., *"Cryptographie Appliquée"*, Vuibert, Wiley and International Thomson Publishing, NY, 2nd edition, 1997.

http://www.schneier.com/book-applied.html

Stinson D.R, *Cryptography : Theory and Practice*, Chapman & Hall/CRC Press, 2nd edition, 2002.

http://www.cacr.math.uwaterloo.ca/~dstinson/CTAP2/CTAP2.html

Ebrahimi T., Leprévost F. and Warusfeld Ed., *Cryptographie et Securité des systèmes et réseaux*, Hermes/Lavoisier,

http://www-id.imag.fr/~svarrett/book_secu_mult.html

Plan

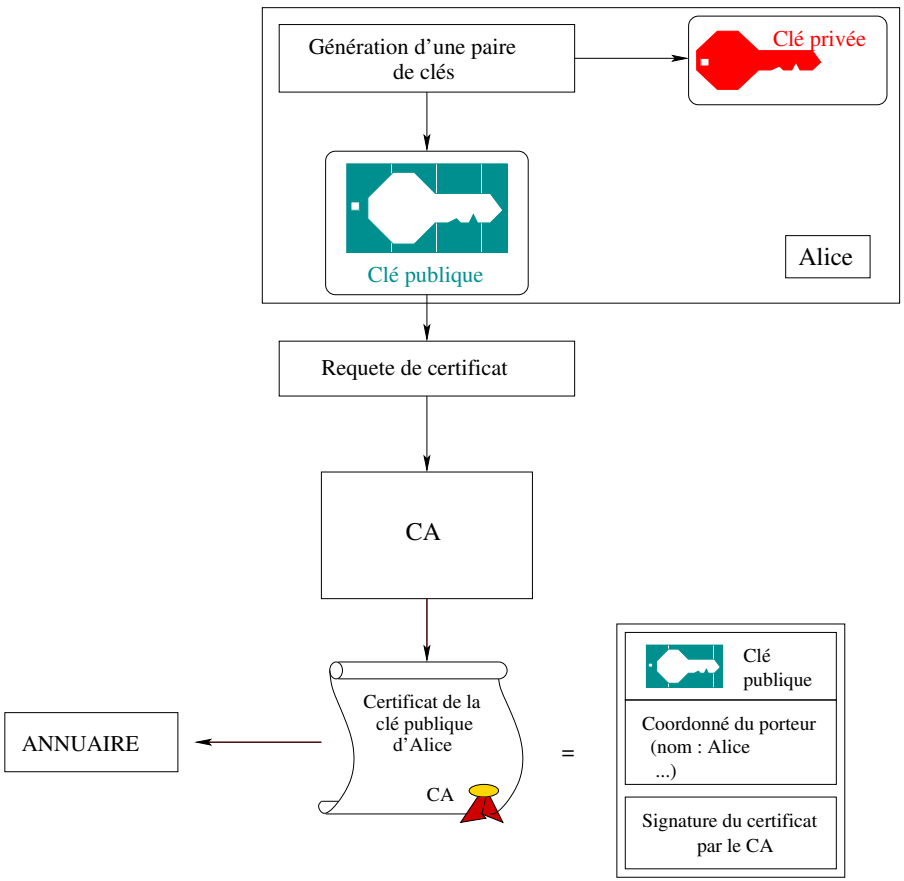
Motivations

- Comment récupérer et être sûr d'une clé publique ?
  - En effet, Oscar peut se faire passer pour le destinataire !
  - Il faut un moyen de **prouver** la correspondance entre une clé publique et une personne !
- Plus généralement : comment gérer des authentifiants dans un environnement distribué/réseau ?

Principe général

- PKI = Public Key Infrastructure
- Ensemble d'infrastructures permettant de réaliser effectivement des échanges sécurisés.
- PKI ne distribue pas des clés mais des **certificats** !
  - Un certificat contient une clé publique
  - Il contient aussi des données d'identité
    - Pour une personne : état civil, adresse, mail...
    - Pour un serveur : nom de domaine, adresse IP, mail de l'administrateur etc...
  - Un certificat est validé par un *tiers de confiance*
    - On parle d'autorité de certification = *CA*
- La PKI assure la gestion des certificats
  - création/distribution...

Création d'un certificat



- Alice génère ses clés K_e et K_d
  - K_e : clé publique
  - K_d : clé privée
- Elle émet une requête au CA pour un certificat de K_e
- CA valide la clé, authentifie Alice et génère un certificat
  - le certificat est signé par le CA
  - Cette signature certifie l'origine du certificat & son intégrité.
- Le certificat est publié dans un annuaire publique

Vérifier l'authenticité du tiers de confiance

- Chaque CA possède lui-même un certificat
  - La clé privée associée permet de signer les certificats émis par le CA
  - Ce certificat est signé par un autre CA etc...

  $\Longrightarrow$ **Chaîne de certificat**
- Le dernier certificat de la chaîne est signé par lui-même
  - On parle de certificat *auto-signé* ou certificat *racine*

Definition (PKI)

Ensemble de technologies, organisations, procédure et pratiques qui supporte l'implémentation et l'exploitation de certificats basés sur la cryptographie à clé publique.

Normes PKI

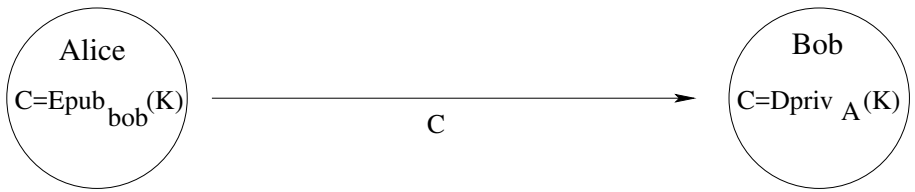
- Il existe plusieurs normes pour les PKI
  - la plupart sont en cours d'évolution
- Deux types d'infrastructures :
  1. *architectures hiérarchiques*
     - reposent sur différents CA, qui sont distincts des utilisateurs.
     - Ex : PKIX (Public Key Infrastructure X.509)
  2. *architectures non-hiérarchiques*
     - chaque utilisateur est son propre CA
     - initialement conçues pour la messagerie comme PGP et le P2P sécurisés
     - confiance mutuelle entre les utilisateurs
     - Ex : SPKI (*Simple Public Key Infrastructure*, *Spooky*), SDSII (*Simple Distributed Security Infrastucture* ou *Sudsy*)

Fonctions d'une PKI

- émettre des certificats à des entités préalablement authentifiées ;
- révoquer des certificats, les maintenir ;
- établir, publier et respecter des pratiques de certification pour établir un espace de confiance ;
- rendre les certificats publics par le biais de services d'annuaires ;
- éventuellement, gérer les clés et fournir des services d'archivage.

Gestion des clés

1. *Gestion des clefs* (Key management).
   - création, distribution, stockage, utilisation
   - recouvrement, l'archivage et destruction
   - Repose sur des règles à respecter impérativement :
     1. *Les clefs secrètes doivent l'être et le rester*
     2. *Les clefs secrètes doivent exister seulement* en clair à l'intérieur d'un module résistant.
     3. *Limiter le déploiement des clefs*
     4. *Séparer les clefs* par utilisation
     5. *Synchroniser les clefs*
     6. *Journal d'événements*
2. *Mise en commun des clefs* (cf DH)
3. *Transport des clefs*



Acteurs d'une PKI

- *Détenteur d'un certificat*
  - entité qui possède une clé privée
  - le certificat numérique contient la clé publique associée.
  - Plusieurs type de certificat : client, serveur, VPN etc...
- *Utilisateur d'un certificat*
  - récupère le certificat
  - utilise la clé publique dans sa transaction avec le détenteur.
- L' *Autorité de Certification* (CA[1])
  - Ens. de ressources défini par son nom et sa clé publique qui :
    - génère des certificats ;
    - émet et maintient les informations sur les CRL[2]
    - publie les certificats non encore expirés ;
    - maintient les archives des certificats expirés/révoqués.
  - **Entité juridique et morale d'une PKI**

[1] Certification Authority
[2] Certification Revocation List

Acteurs d'une PKI (2)

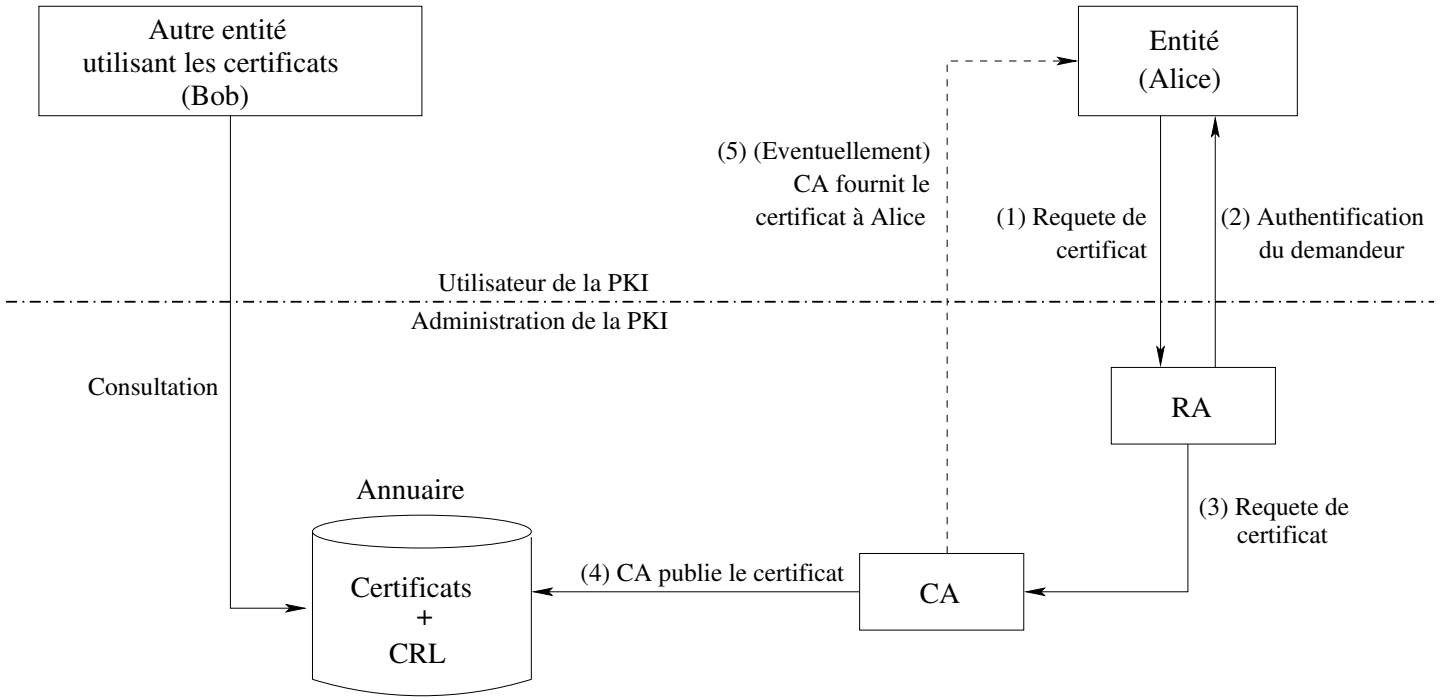
- *Autorité d'enregistrement* (RA[3])
  - Intermédiaire entre le détenteur de la clé et le CA.
  - Vérifie les requêtes des utilisateurs
  - Transmet les requêtes au CA
    - niveau de vérification dépend de la politique de sécurité
  - Chaque CA a une liste de RA accrédités.
  - Un RA est connu d'un CA par son nom et sa clé publique.
  - CA vérifie les informations du RA par le biais de sa signature
- *Emetteur de CRL*

[3] Registration Authority

Acteurs d'une PKI (3)

- *Dépôt* ou *Annuaire* (Repository)
  - Distribue les certificats et les CRL.
  - Accepte les certificats et les CRL d'autres CA et les rend disponibles aux utilisateurs.
  - Connu par son adresse et son protocole d'accès.
- *Archive*
  - stockage sur le long terme des informations pour le compte d'un CA.
  - permet de régler les litiges
    - en sachant quel certificat était valable à telle époque.

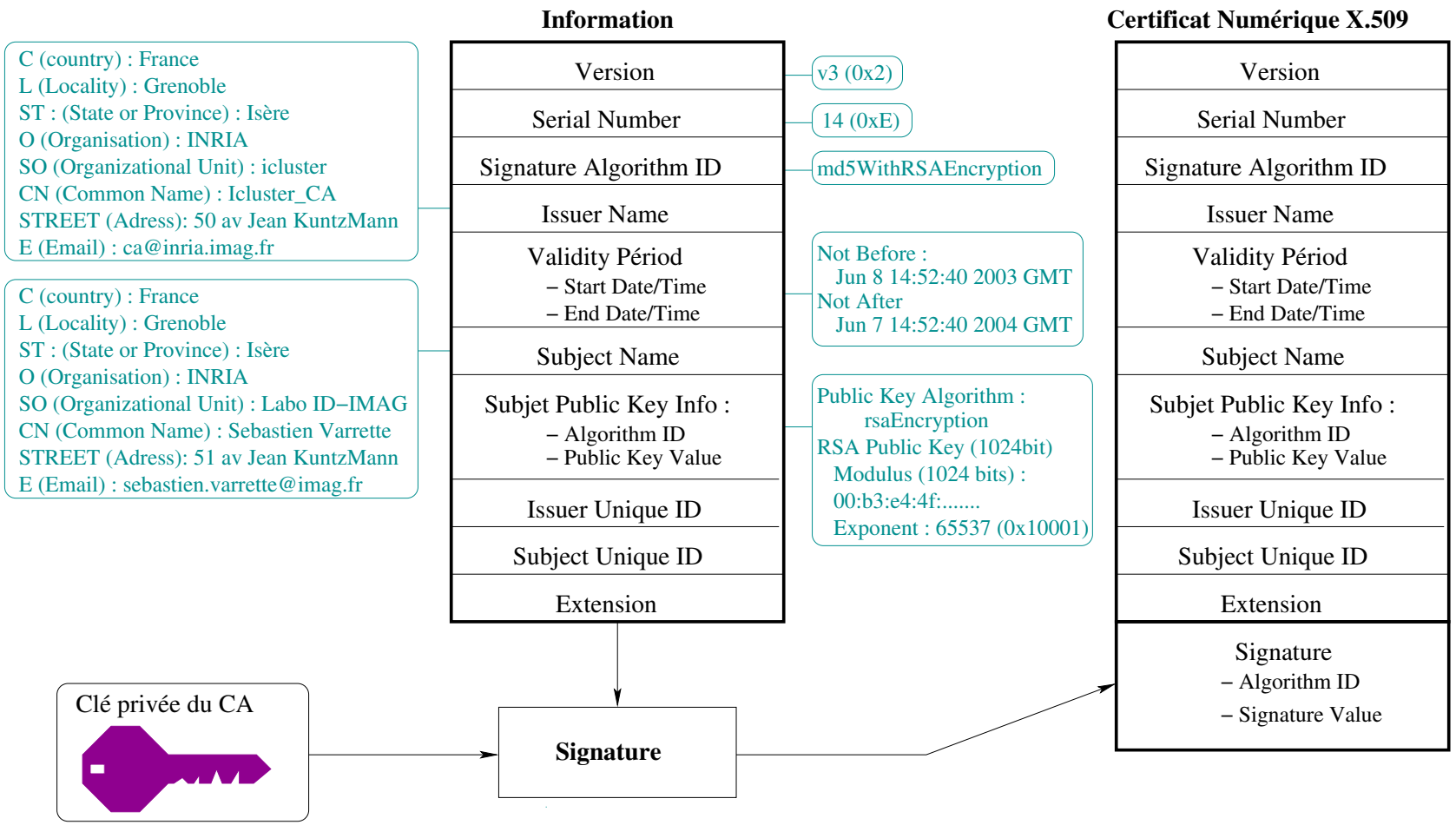
Emission d'un certificat



PGP, un premier exemple de certificat

- Format de certificat PGP a été défini par Phil Zimmermann.
- Contient les informations suivantes :
  - Num. de la version de PGP (identifie l'algo utilisé pour créer la clef)
  - la clef publique et l'algorithme associé
    - RSA, DH (Diffie-Hellman) ou DSA (Digital Signature Algorithm).
  - la signature digitale du CA
  - identité du porteur : nom, numéro ID, photographie, etc.
  - la période de validité du certificat
  - l'algorithme symétrique (à clef privée) préféré.
- Particularité : un certificat peut contenir plusieurs signatures.
  - plusieurs personnes à titre de CA peuvent certifier l'association "clef/identification"

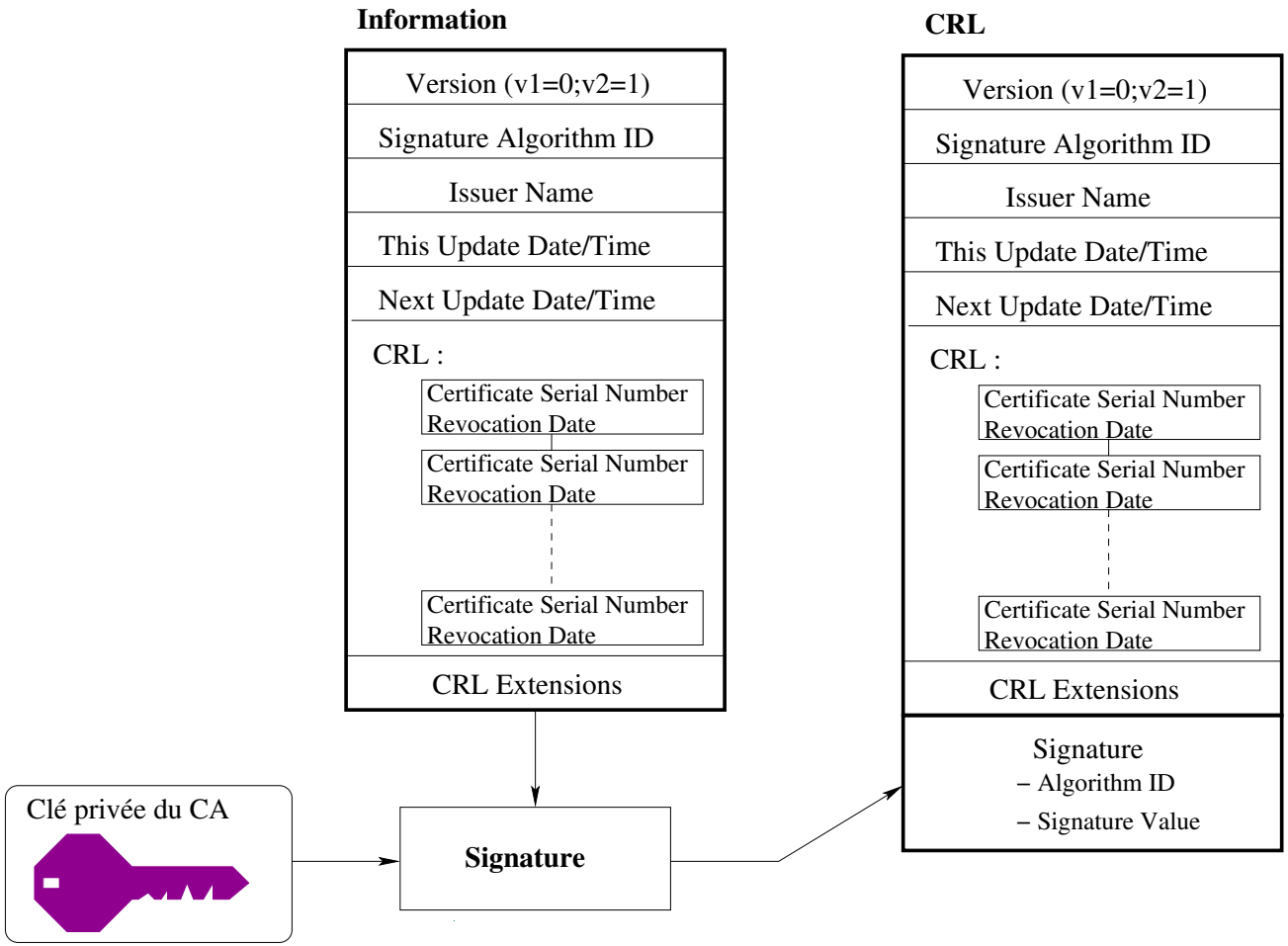
Le certificat X.509



Signification des champs d'un certificat X.509

Version :	Indique à quelle version de X.509 correspond ce certificat.
Serial number :	Numéro de série du certificat (propre à chaque CA).
Signature Algo ID :	Identifiant du type de signature utilisée.
Issuer Name :	Distinguished Name(DN) du CA qui émet le certificat
Validity period :	Période de validité.
Subject Name :	Distinguished Name (DN) du détenteur de la clé publique
Subject pub. key info :	Informations sur la clef publique de ce certificat.
Issuer Unique ID :	Identifiant unique de l'émetteur de ce certificat
Subject Unique ID :	Identifiant unique du détenteur de la clé publique
Extensions :	Extensions génériques optionnelles.
Signature :	Signature numérique du CA sur les champs précédents

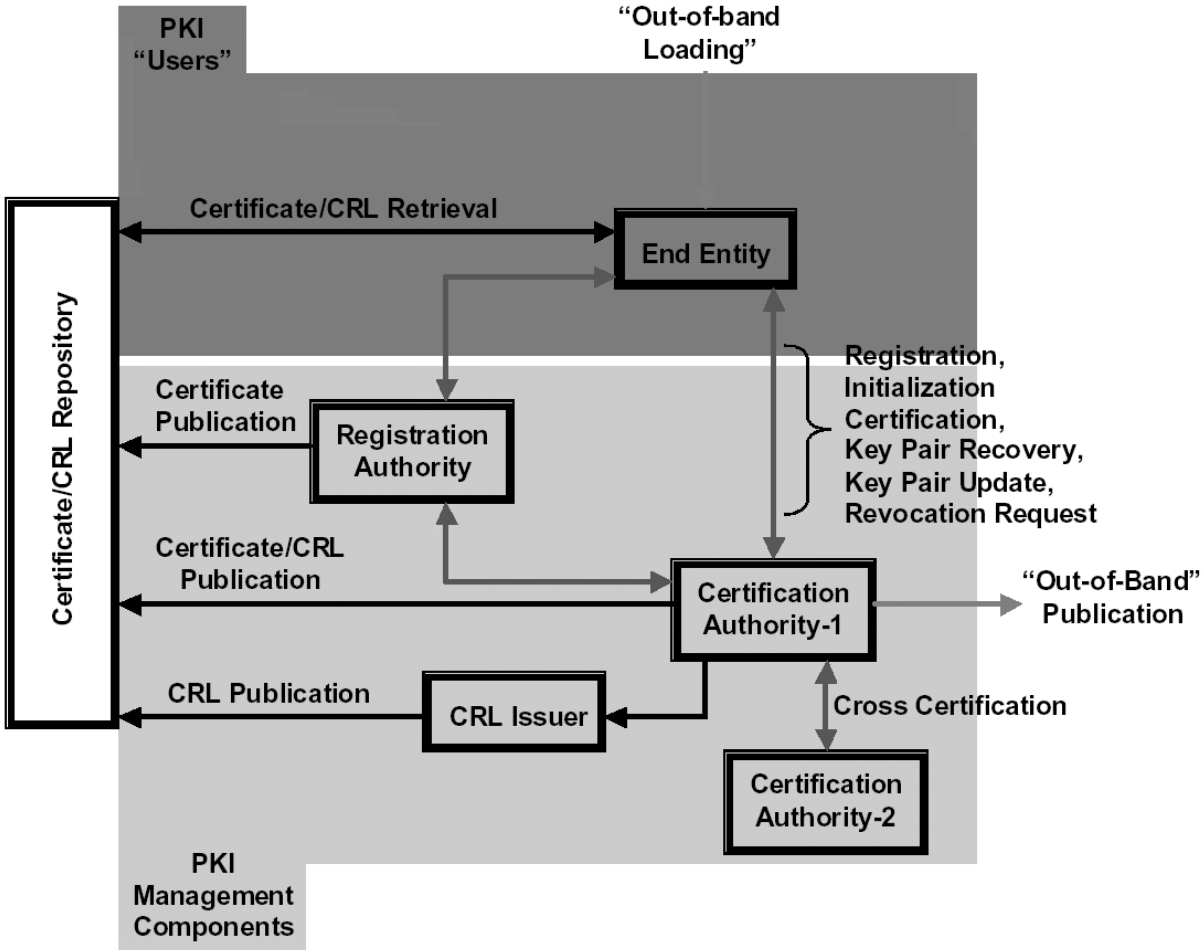
Génération et contenu d'une CRL X.509



Quelques identificateurs d'algorithmes et numéros associés

Alg.	Hash.	OID	Identificateur
3DES-CBC		1.2.840.113549.3.7	DES-EDE3-CBC
RSA		1.2.840.113549.1.1.1	RSAEncryption
RSA	MD5	1.2.840.113549.1.1.4	md5withRSAEncryption
RSA	SHA-1	1.2.840.113549.1.1.5	sha1withRSAEncryption
DSA		1.2.840.10040.4.1	id-dsa
DSA	SHA-1	1.2.840.10040.4.3	id-dsawithSha1
DSA	SHA-1.320	1.3.14.3.2	id-dsawithSha1.320
ECDSA	SHA-1	1.2.840.10045.1	ecdsawithSha1

Le modèle PKIX

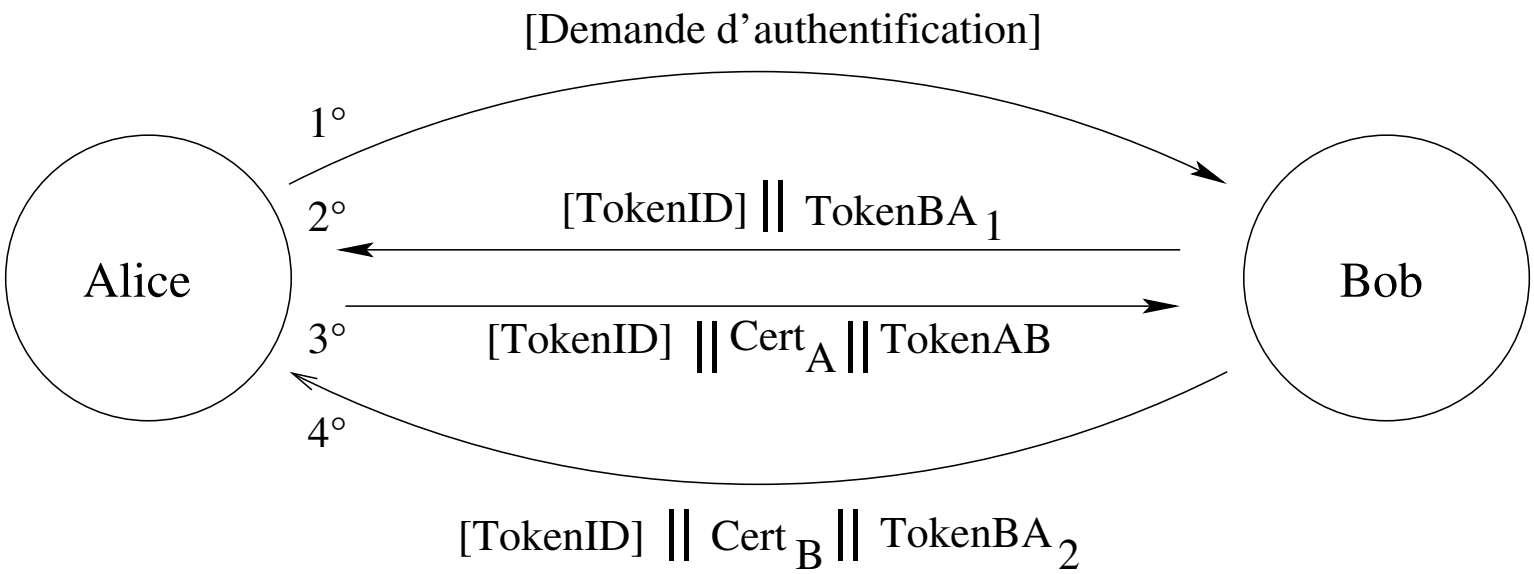


Authentification d'entités à partir de certificats

- Fait l'objet d'une norme : FIPS-196
- Notations utilisées dans FIPS-196 :

A	Nom qui identifie Alice
B	Nom qui identifie Bob
$Sign_x(M)$	La signature du message M avec la clé privée de X
R_x	Un challenge aléatoire produit par X
$Cert_X$	Le certificat de X
$X \cdot Y$	La concaténation de X et de Y
$TokenID$	Identificateur de Token. Il précise les informations identifiant le token, le type de protocole... qui faciliteront le traitement du Token.
$TokenXY$	Un token envoyé de X vers Y
$TokenXY_i$	Le i-ème token envoyé de X vers Y
$[Z]$	Précise que le champs Z est optionnel

Authentification d'entités à partir de certificats (2)



Authentification d'entités à partir de certificats (3)

- Alice envoie à Bob une demande d'authentification
- Bob génère R_b et envoit à Alice :

$$TokenBA_1 = R_b$$

- Alice génère R_a et envoit avec son certificat :

$$TokenAB = R_a \cdot R_b \cdot B \cdot Sign_a(R_a \cdot R_b \cdot B)$$

- Bob vérifie le certificat d'Alice.
  - il peut ensuite vérifier les informations de $TokenAB$
  - Après vérification, Alice est authentifiée auprès de Bob.
- Bob envoit avec son certificat :

$$TokenBA_2 = R_b \cdot R_a \cdot A \cdot Sign_b(R_b \cdot R_a \cdot A)$$

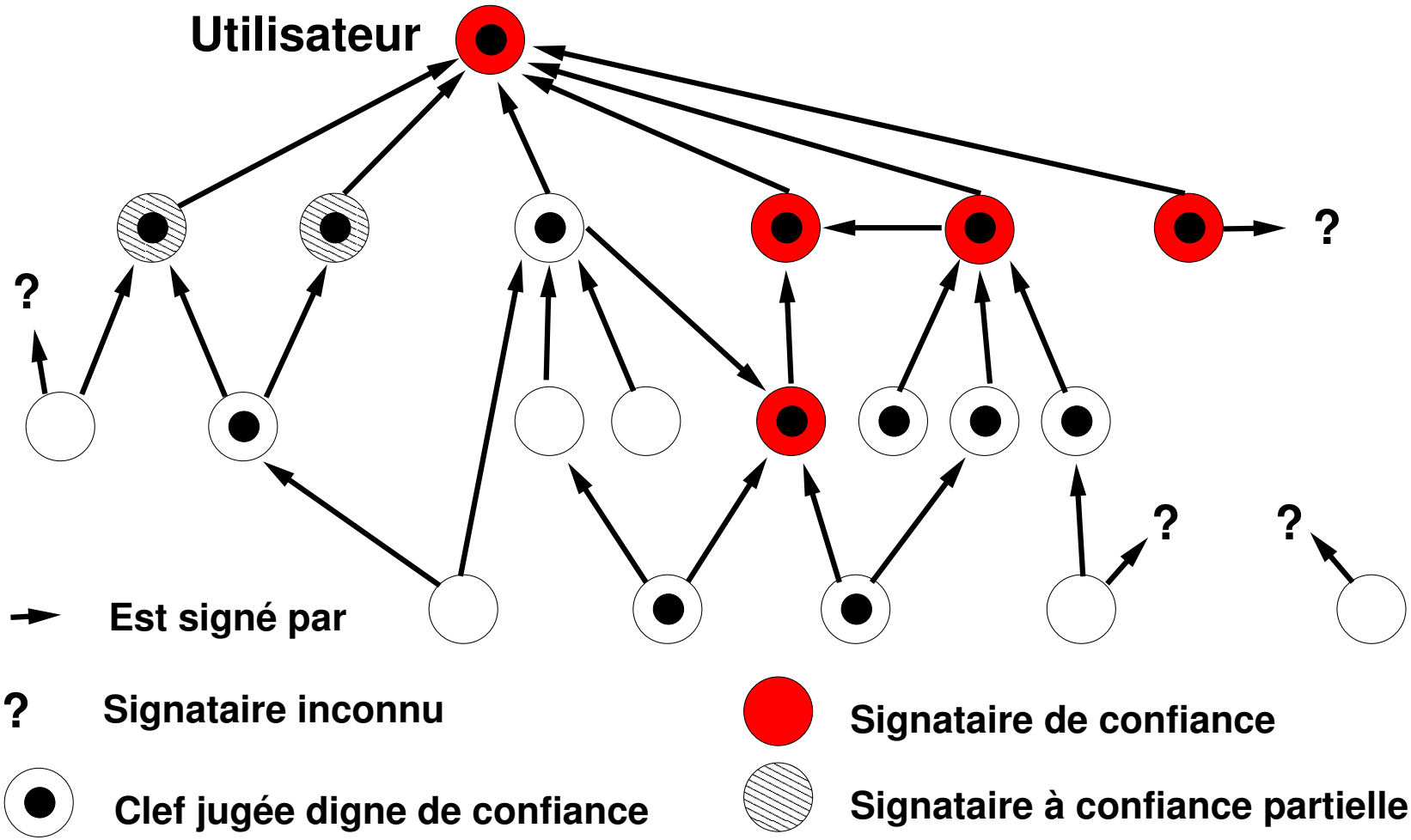
- Alice procède de la même manière pour authentifier Bob.

A la fin, il y a authentification mutuelle.

Application pratique : OpenSSL

- Création d'un répertoire pour les certificats
  `mkdir certificates ; cd certificates`
- Creation du certificat pour le CA :
  - Utiliser le script `CA.sh`
    `/usr/lib/ssl/misc/CA.sh -newca`
- Création du certificat serveur/personne :
  `openssl req -new -nodes -keyout newreq.pem -out newreq.pem -days 365`
- Signature du certificat du serveur par le CA
  `/usr/lib/ssl/misc/CA.sh -sign`

Modèle de confiance PGP



Notion de politique de sécurité

- Définit la stratégie globale et répondre aux menaces.
- En particulier, il est fondamental de décrire :
  - Qui est responsable de quoi (mise en œuvre, exécution, audit, tests, etc.).
  - Quelle est la politique de sécurité de base du réseau d'ordinateurs.
  - Pourquoi chacun doit faire ce qu'il fait.