

# Authentification et autorisation d'accès

Bernard Cousin



UNIVERSITE DE RENNES 1

Sécurité des réseaux informatiques

1

## Plan

- Les services d'authentification et d'autorisation d'accès
- Un exemple de protocole d'authentification et d'autorisation d'accès : Kerberos
- Autres services d'authentification et d'autorisation d'accès
- Les certificats d'authentification X.509
  - Liste de révocation

Sécurité des réseaux informatiques

2

## Service d'authentification

- Service d'autorisation d'accès
  - À un réseau
    - Par ex. Radius, TACACS (Terminal Access Controller Access-Control System)
  - À un service distant
    - Par ex. Kerberos
- Nécessite un service d'authentification
  - À base de serveur d'authentification
    - Un tiers ... de confiance
- Par ex. Kerberos, Radius, Diameter (AAA)

## Les services de sécurité de l'autorisation d'accès

- Les services de sécurité nécessaires à la distribution des clefs :
  - Confidentialité et opportunité (" timeliness ")
- La confidentialité :
  - Chiffrement des informations d'identification et de la clef de session
  - Nécessite l'utilisation d'une connexion préalablement sécurisée qui utilise des clefs partagées ou publiques
- La justesse/opportunité
  - Contre les attaques de type rejeu
  - Fournit par une numérotation, horodatage ou un processus de type "challenge/response "

# KERBEROS



In Greek mythology, a many headed dog, the guardian of the entrance of Hades

# KERBEROS

- But
  - Contrôle de l'accès aux services d'un serveur
    - Les clients licites accèdent aux services
    - Les clients illicites n'accèdent pas aux services
      - Un client qui devient illicite ne doit plus pouvoir plus accéder aux services
- Plusieurs risques existent. Par ex.:
  - Un client prétend être un autre.
  - Le client modifie l'adresse IP d'une station.
  - Le client capture des messages et utilise une attaque par rejeu.
  - Etc.

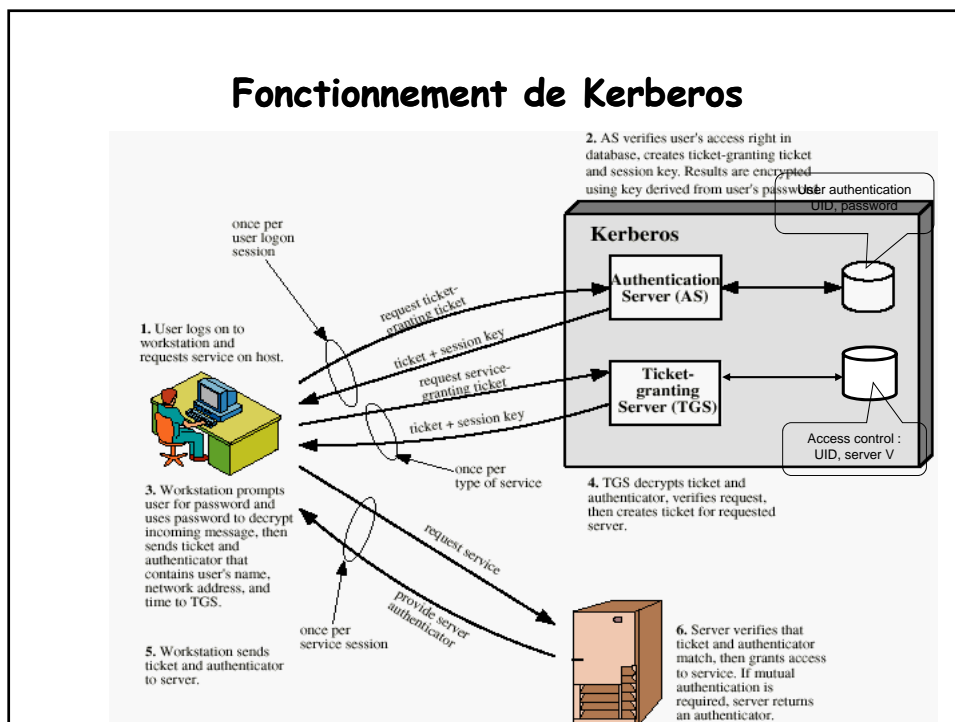
# KERBEROS

- Un serveur d'autorisation centralisé
  - Authentifie les clients vis-à-vis des serveurs et vice versa.
  - L' *Authentification* est effectuée une seule fois
  - L' *Autorisation d'accès* peut être demandée pour plusieurs services
- Utilise une technique de chiffrement symétrique
  - Peu coûteux (pas d'utilisation de clef publique)
- Deux versions: v4 et v5
  - La version 4 utilise DES et est mono domaine

Sécurité des réseaux informatiques

7

## Fonctionnement de Kerberos



## Kerberos Version 4

- Les variables :
  - $C$  = client
  - $AS$  = authentication server
  - $V$  = server
  - $ID_c$  = identifier of user on  $C$
  - $ID_v$  = identifier of  $V$
  - $P_c$  = password of user on  $C$
  - $AD_c$  = network address of  $C$
  - $K_v$  = secret encryption key shared by  $AS$  and  $V$
  - $TS$  = timestamp,  $TTL$  = lifetime
  - $||$  = concatenation

## Un dialogue simple d'authentification

(1)  $C \rightarrow AS$ :  $ID_c || P_c || ID_v$   
(2)  $AS \rightarrow C$ : Ticket  
(3)  $C \rightarrow V$ :  $ID_c || Ticket$   
Ticket =  $E_{K_v}[ID_c || AD_c || ID_v]$

Une clé partagée  $K_v$  entre chaque serveur  $V$  et  $AS$ .

- Le mot de passe est en clair
- La durée de vie du ticket est infinie
- Surcharge du serveur d'authentification

## Deuxième dialogue d'authentification

(1)  $C \rightarrow AS$ :  $ID_c || ID_{tgs}$   
(2)  $AS \rightarrow C$ :  $E_{K_c}[Ticket_{tgs}]$   
 $Ticket_{tgs} = E_{K_{tgs}}[ID_c || AD_c || ID_{tgs} || TS_1 || TTL_1]$   
(3)  $C \rightarrow TGS$ :  $ID_c || ID_v || Ticket_{tgs}$   
(4)  $TGS \rightarrow C$ :  $Ticket_v$   
 $Ticket_v = E_{K_v}[ID_c || AD_c || ID_v || TS_2 || TTL_2]$   
(5)  $C \rightarrow V$ :  $ID_c || Ticket_v$

- Une clé partagée  $K_c$  entre chaque  $C$  et  $AS$ , issue du mot de passe  $f(P_c) = K_c$
- Une clé partagée  $K_{tgs}$  entre chaque  $TGS$  et  $AS$
- Une clé partagée  $K_v$  entre chaque serveur  $V$  et son  $TGS$ .

Sécurité des réseaux informatiques

11

## Dialogue d'authentification

- Problèmes:
    - Un adversaire peut voler un ticket et s'en servir
      - Une durée de vie est associée au ticket "ticket-granting"
        - Trop courte  $\rightarrow$  demande répétée du mot de passe
        - Trop longue  $\rightarrow$  plus d'opportunité pour une attaque par rejeu
    - Un adversaire peut voler un ticket et s'en servir avant qu'il n'expire !
- $\Rightarrow$  On va donc
- $\Rightarrow$  Utiliser des "TimeStamps"
  - $\Rightarrow$  Distribuer les clefs

Sécurité des réseaux informatiques

12

## Dialogue d'authentification

### Authentication Service Exchange: To obtain Ticket-Granting Ticket

- (1)  $C \rightarrow AS:$   $ID_c \parallel ID_{TGS} \parallel TS_1$   
 (2)  $AS \rightarrow C:$   $E_{K_c} [K_{c,tgs} \parallel ID_{TGS} \parallel TS_2 \parallel TTL_2 \parallel Ticket_{TGS}]$   
 $Ticket_{TGS} = E_{K_{TGS}} [K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{TGS} \parallel TS_2 \parallel TTL_2 \parallel Ticket_{TGS}]$

### Ticket-Granting Service Exchange: To obtain Service-Granting Ticket

- (3)  $C \rightarrow TGS:$   $ID_v \parallel Ticket_{TGS} \parallel Authenticator_c$   
 (4)  $TGS \rightarrow C:$   $E_{K_c} [K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v]$   
 $Ticket_v = E_{K_v} [K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel TTL_4 \parallel Ticket_{TGS}]$   
 $Authenticator_c = E_{K_c} [K_{c,tgs} \parallel ID_c \parallel AD_c \parallel TS_3]$

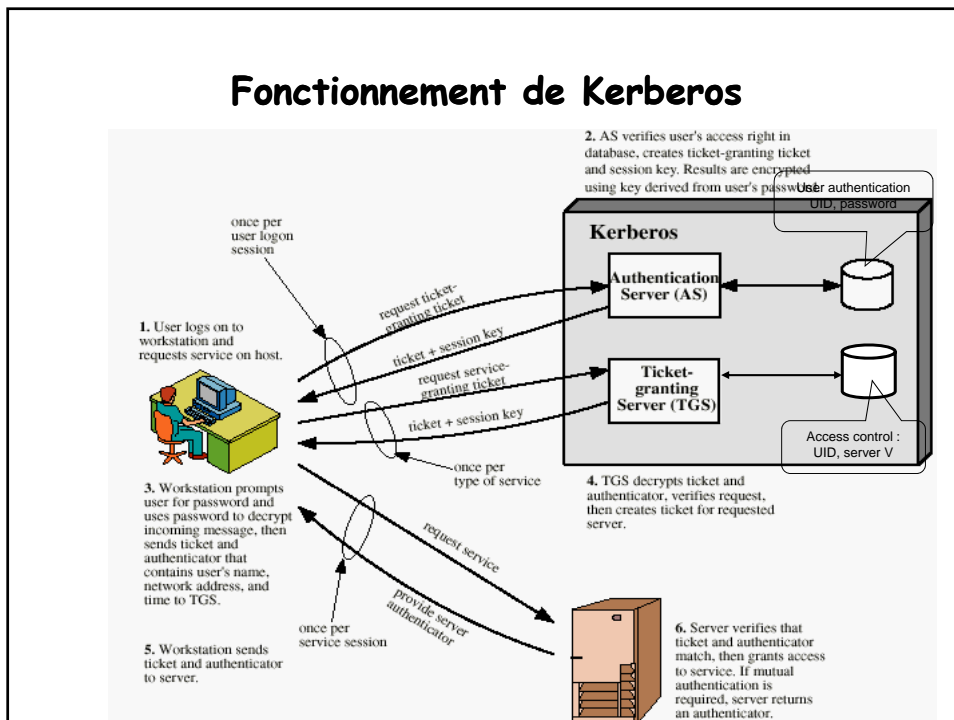
### Client/Server Authentication Exchange: To obtain Service

- (5)  $C \rightarrow V:$   $Ticket_v \parallel Authenticator'_c$   
 (6)  $V \rightarrow C:$   $E_{K_{c,v}} [TS_5 + 1]$   
 $Authenticator'_c = E_{K_{c,v}} [ID_c \parallel AD_c \parallel TS_5]$

Securite des reseaux informatiques

13

## Fonctionnement de Kerberos



## Requêtes entre domaines Kerberos

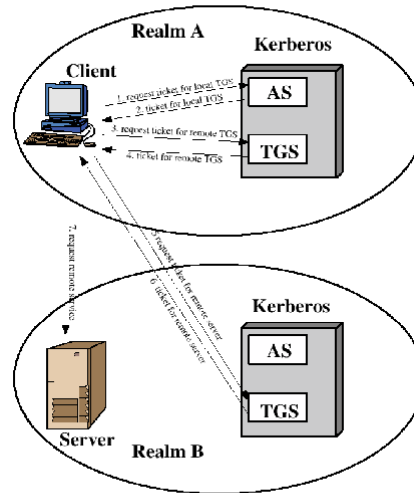


Figure 4.2 Request for Service in Another Realm

15

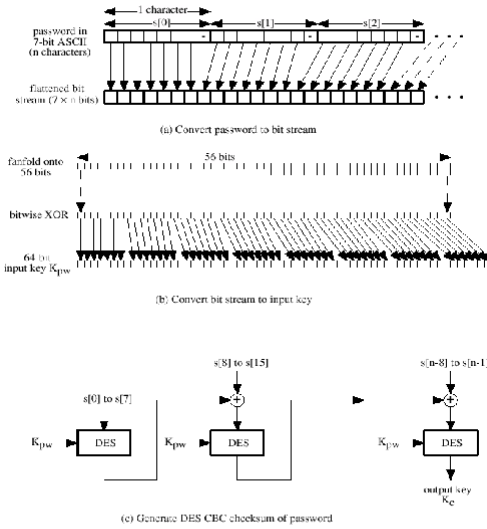
## Différence entre les versions V4 et V5 de Kerberos

- Dépendance vis-à-vis du système de chiffrement
  - V4 utilise DES
- Dépendance vis-à-vis du protocole Internet
- L'ordre des octets dans les messages
  - V5 : ASN1 + BER
- "Ticket lifetime"
  - V5 : "explicit start and end times"
- La propagation des crédits
  - transitivité :  $C \Rightarrow V1 \Rightarrow V2$ )
- L'authentification entre domaines
  - $N^2/2$  authentification

## Technique de chiffrement de Kerberos

Produire une clef à partir du mot de passe :

- La taille du mot de passe n'est pas adaptée
  - ni à la longueur de la clef,
  - ni à celle des blocs utilisés par l'algo de chiffrement



Séct

Figure 4.6 Generation of Encryption Key from Password

## Le mode PCBC de DES

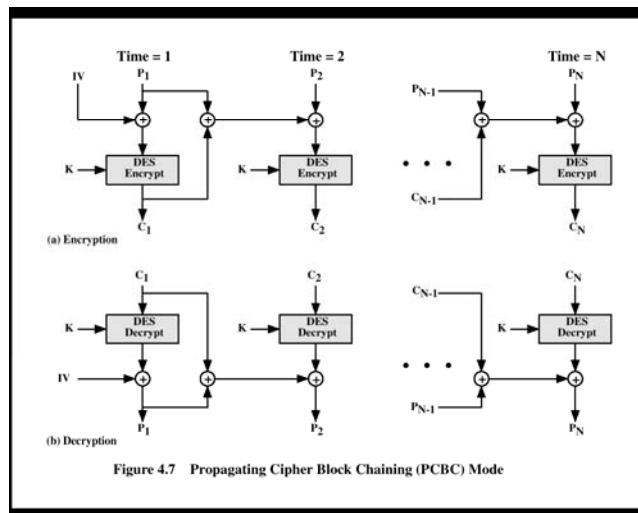


Figure 4.7 Propagating Cipher Block Chaining (PCBC) Mode

## Utilisation de Kerberos

- **Utiliser la dernière version de Kerberos :**
  - v5 : permet l'authentification entre domaines
  - Kerberos v5 :
    - RFC 1510 (septembre 1993), rfc 4120 (juillet 2005)
- **Pour utiliser Kerberos:**
  - un KDC ("Key Distribution Center") dans votre réseau
  - Des applications adaptées et utilisant Kerberos dans tout vos systèmes
- Kerberos est disponible
  - Par ex. Mac OS X, Windows 2000
  - OpenLDAP, OpenSSH, etc.

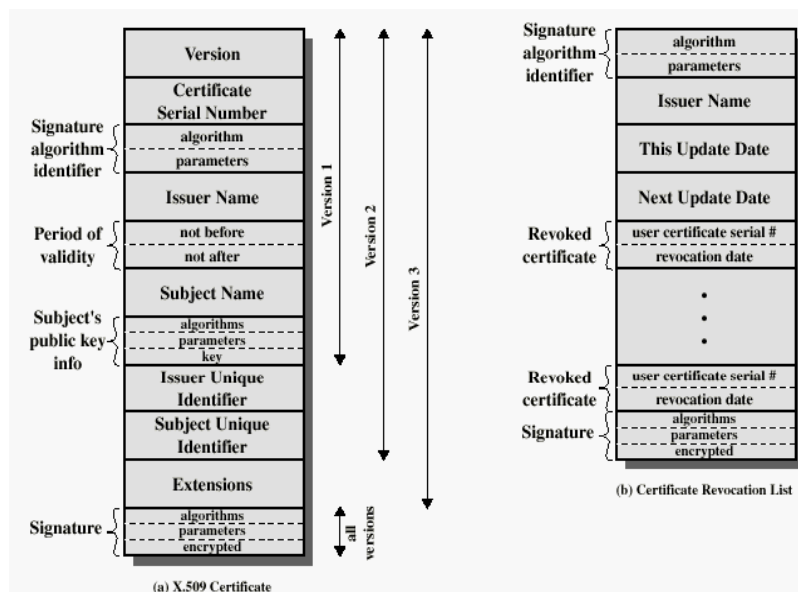
## Qq systèmes d'authentification

- **AAA :**
  - "Authentication, Authorization and Accounting"
  - RFC 2903 (2000)
  - Diameter...
- **RADIUS :**
  - "Remote Authentication Dial In User Service"
  - Rfc 2865 (June 2000), m-à-j. par rfc 3373 (2003), en 2001 pour IPv6, etc.

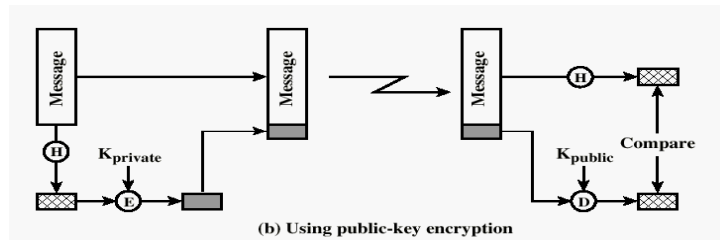
## Le service d'authentification de X.509

- X500 :
  - Le service de l'OSI pour l'annuaire (equiv. à DNS) directory Service (eq. DNS)
    - Un ensemble distribué de serveurs qui maintiennent une base de données des noms et de leur attribut (adresse IP)
  - Cette base de données peut servir de stockage "repository" pour les clefs publiques
- X.509 :
  - Historiquement, X.509 définissait les certificats nécessaires au service d'authentification de X.500
  - Chaque certificat contient la clef publique d'un utilisateur. Il est signé par la clef privée d'une autorité de certification (CA).
  - Utilisé par S/MIME, IPsec, SSL/TLS, SET, etc.
  - L'utilisation de RSA est initialement prévue.
  - Les versions de X.509 :
    - V1 en 1988, V2 en 1993, V3 en 1995
    - Une certaine compatibilité ascendante
    - X.509v3 repris et adapté par l'IETF (rfc 2459 - 1999)
    - X.509v3 repris et corrigé par l'IETF (rfc 5280 - 2008)

## Le format X.509



# Signature numérique à base de chiffrement asymétrique



Certificate:  
 Data:  
 Version: 1 (0x0)  
 Serial Number: 7829 (0x1e95)  
 Signature Algorithm: md5WithRSAEncryption  
 Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,  
 OU=Certification Services Division,  
 CN=Thawte Server CA/emailAddress=server-certs@thawte.com  
 Validity  
 Not Before: Jul 9 16:04:02 1998 GMT  
 Not After : Jul 9 16:04:02 1999 GMT  
 Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,  
 OU=FreeSoft, CN=www.freソフト.org/emailAddress=baccala@freソフト.org  
 Subject Public Key Info:  
 Public Key Algorithm: rsaEncryption  
 RSA Public Key: (1024 bit)  
 Modulus (1024 bit):  
 00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:  
 33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:  
 66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:  
 70:33:52:14:c9:ee:4f:91:51:70:39:de:53:85:17:  
 16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:  
 c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:  
 8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:  
 d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:  
 e8:35:1c:9e:27:52:7e:41:8f  
 Exponent: 65537 (0x10001)  
 Signature Algorithm: md5WithRSAEncryption  
 93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:  
 92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:  
 ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:  
 d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:  
 0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:  
 5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:  
 8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:  
 68:9f

## Exemple de certificat

## L'obtention d'un certificat

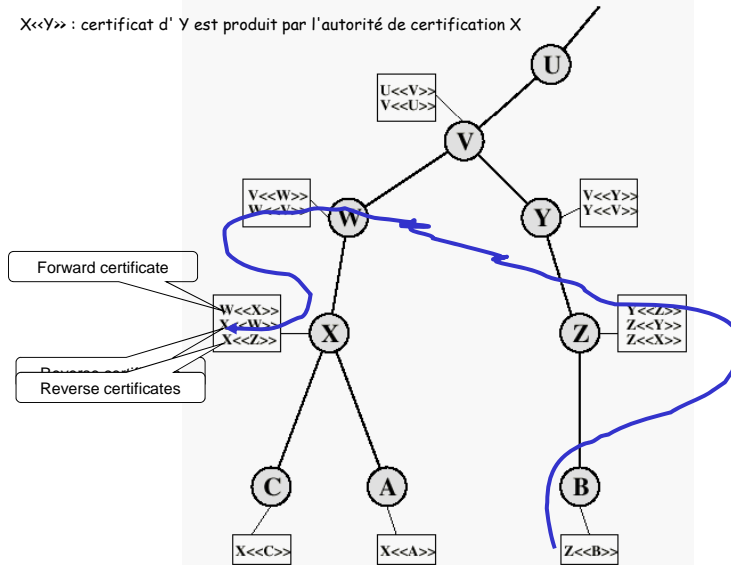
- Les propriétés d'un certificat généré par un CA :
  - N'importe quel client ayant la clef publique du CA peut obtenir la clef publique d'un client qui a été certifiée.
    - Cette association est sûre
      - le niveau de sûreté est celui accordé au CA
    - L'association est authentique
  - Personne hormis le CA peut modifier la certificat sans être détecté :
    - La clef privée de CA est gardée secrète
    - Intégrité des certificats

## Certificat

- Notation usuel pour un certificat
  - Compatible X.509 v1
$$CA\langle\langle A \rangle\rangle = CA_{KR-CA}\{Version, Serial\ Number, Algorithm\ Identifier, CA, Timestamps, A, KU_A\}$$

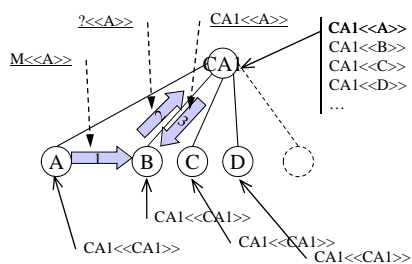
## Hiérarchie des certificats X.509

$X\langle\langle Y \rangle\rangle$  : certificat d' Y est produit par l'autorité de certification X



27

## Procédé d'authentification : mono CA



Hypothèses:

- L'autorité de certification (CA1) peut produire un certificat pour n'importe lequel de ses clients
- Chaque client fait confiance à l'autorité de certification

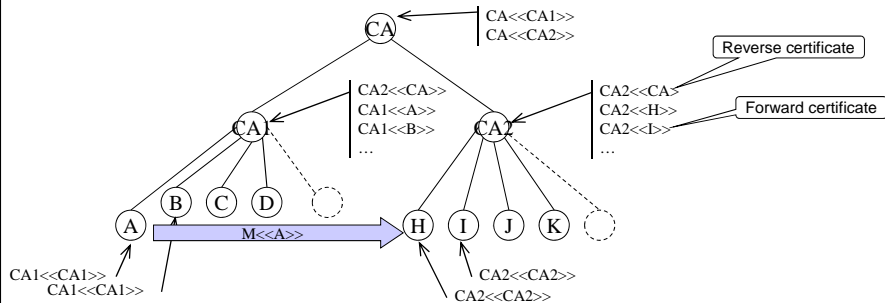
A envoie un message à B ( $A \Rightarrow B$ ), le message contient une signature numérique produite par A, B veut vérifier l'authenticité (l'intégrité) du message donc il lui faut un certificat associant l'identité de A et la clé publique de A. B obtient auprès de son autorité de certification CA1 le certificat de A :  $CA1\langle\langle A \rangle\rangle$ .

B peut vérifier l'authenticité de ce certificat car il fait confiance à CA1 (il a un certificat auto-signé de CA1). B construit (le début d') une chaîne de confiance qui aboutit à A :  $CA1\langle\langle CA1 \rangle\rangle CA1\langle\langle A \rangle\rangle$

Sécurité des réseaux informatiques

28

## Procédé d'authentification : multi CA



Pour gérer un nombre important de clients, on propose une hiérarchie d'autorité de certification  $CA(CA1, CA2, \text{etc.})$ .

A envoie un message à H ( $A \Rightarrow H$ ), le message contient une signature numérique produite par A, H veut vérifier l'authenticité (l'intégrité) du message donc il lui faut un certificat associant l'identité de A et la clef publique de A. H obtient le certificat de A auprès de son autorité de certification  $CA1 : CA1\langle\langle A \rangle\rangle$ .

H peut vérifier l'authenticité de ce certificat car il fait confiance à  $CA2$  (il a un certificat auto-signé de  $CA2$ ). H construit une chaîne de confiance qui aboutit à A:  $CA2\langle\langle CA2 \rangle\rangle CA2\langle\langle CA \rangle\rangle CA\langle\langle CA1 \rangle\rangle CA1\langle\langle A \rangle\rangle$

Hypothèses:

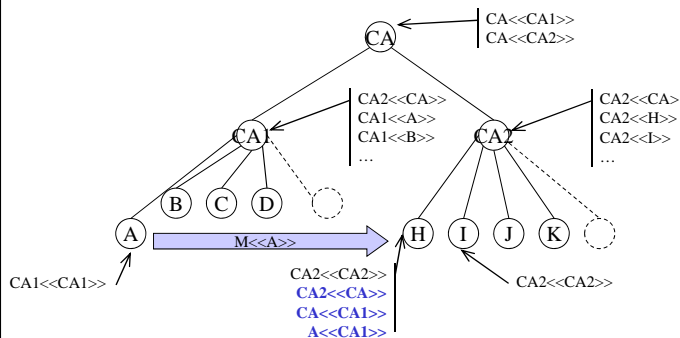
- L'autorité de certification ( $CA_x$ ) peut produire un certificat pour n'importe lequel de ses clients (ou  $CA$  de niveau inf.).

Sécurité des réseaux informatiques

29

- Chaque client (ou  $CA$ ) fait confiance à l'autorité de certification de niveau supérieur.

## Procédé d'authentification : graphe



Les entités peuvent mémoriser les certificats : cela accélère les prochaines vérifications.

Les relations de certifications forment alors un graphe quelconque.

Sécurité des réseaux informatiques

30

## Révocation des certificats

- Les raisons de révoquer un certificat :
  - Lorsque la clef secrète est potentiellement compromise.
  - Lorsque le client n'est plus certifié par le CA (Fin de contrat).
  - Changement de CA.
  - Le CA est potentiellement compromis.
- CRL :
  - "Certificat Revocation List"

## Protocoles d'authentification

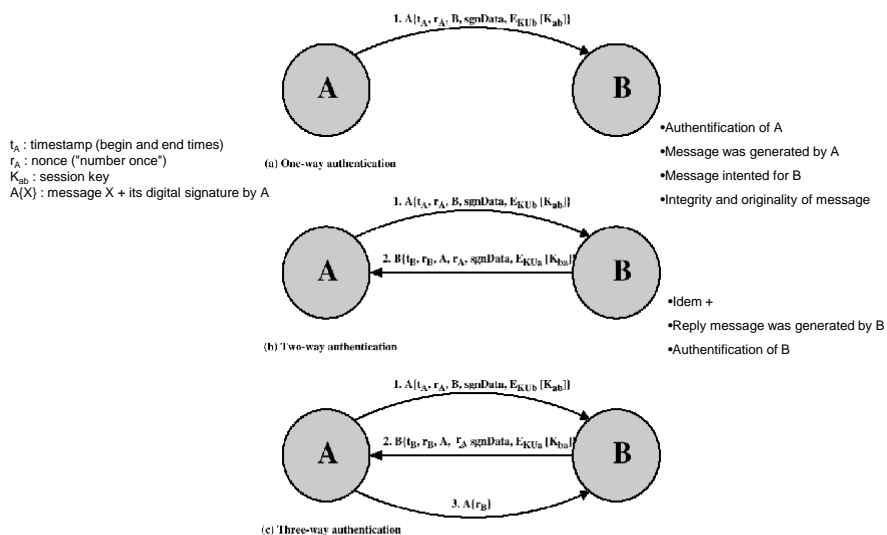


Figure 4.5 X.509 Strong Authentication Procedures

## Bibliographie et sites web

- Bryant, W. "Designing an Authentication System: A Dialogue in Four Scenes".  
<http://web.mit.edu/kerberos/www/dialogue.html>
- Kohl, J.; Neuman, B. "The Evolution of the Kerberos Authentication Service"  
<http://web.mit.edu/kerberos/www/papers.html>
- <http://www.isi.edu/qost/info/kerberos/>
- D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List Profile", IETF standard rfc 5280, May 2008