Authentification et contrôle d'accés

UNIVERSITE DE RENNES 1

Plan

- Les problèmes de sécurité des protocoles d'authentification et de contrôle d'accès
- Un exemple de protocole d'authentification et de contrôle d'accès : Kerberos
- Autres services d'authentification
- Les certificats d'authentification X.509
  - Liste de révocation

Service d'authentification

- Service d'authentification
  - À base de serveur d'authentification
    - Un tiers … de confiance
    - Par ex. Kerberos, AAA, Radius, Diameter
  - Sans
    - par ex. SSL

Les problèmes de sécurité

- Les services de sécurité nécessaire 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

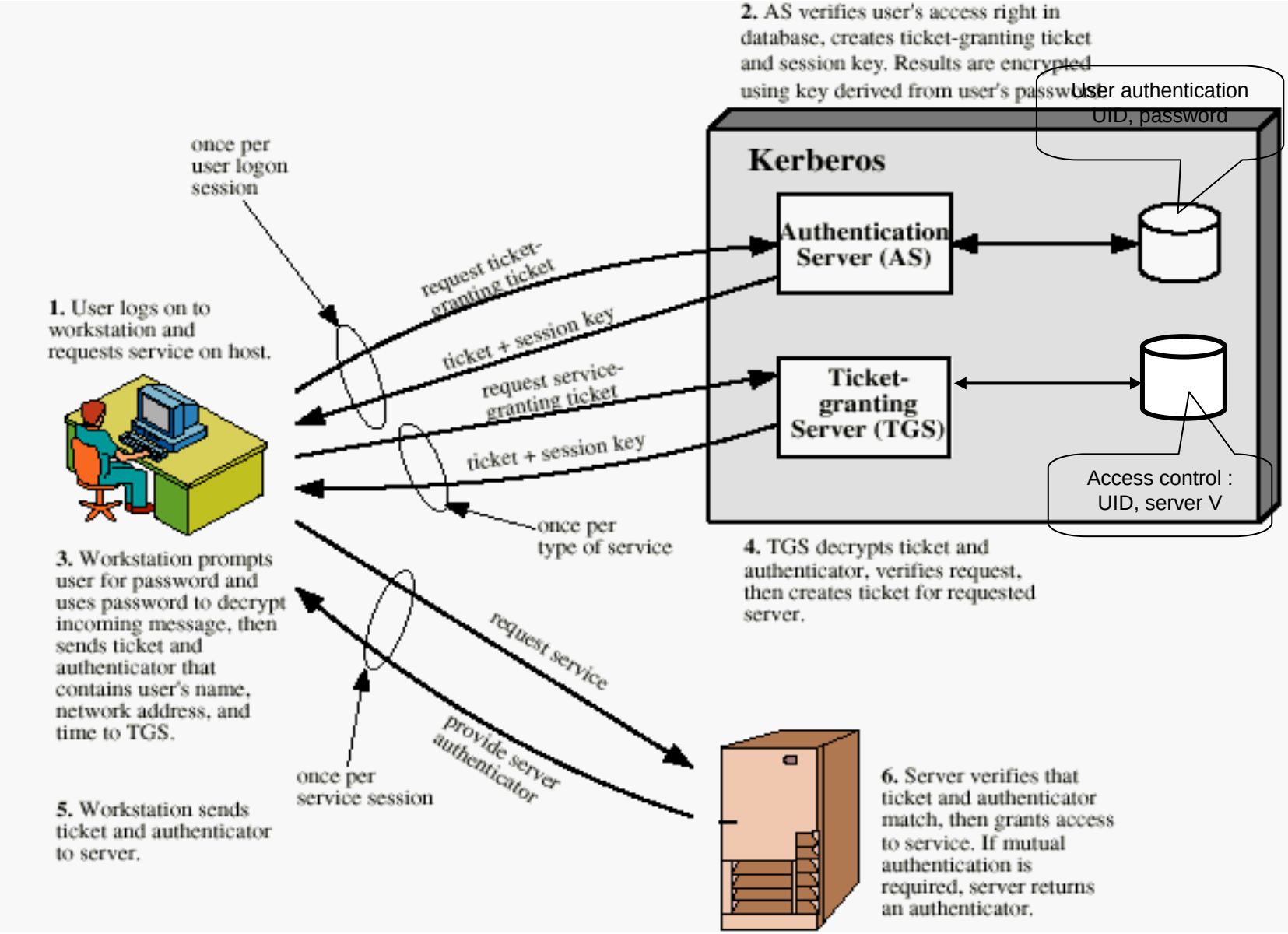
KERBEROS

- But
  - Contrôle de l'accès aux services d'un serveur
- Trois risques existent :
  - 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.

KERBEROS

- Un serveur d'authentification centralisé
  - Authentifie les clients vis-à-vis des serveurs et vice versa.
- Utilise une technique de chiffrement conventionnel (pas d'utilisation de clef publique)
- Deux versions: v4 et v5
  - La version 4 utilise DES et est mono domaine

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 → AS: $ID_c \,||\, P_c \,||\, ID_v$

(2) AS → C: Ticket

(3) C → V: $ID_c \,||\, Ticket$

$Ticket = E_{Kv}[ID_c \,||\, AD_c \,||\, ID_v]$

Une clé partagée Kv 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 → AS: IDc || ID_{tgs}

(2) AS → C: $E_{Kc}[Ticket_{tgs}]$

$Ticket_{tgs} = E_{Ktgs}$ [IDc||ADc||ID_{tgs}||TS_1||TTL_1]

(3) C → TGS: IDc||IDv||$Ticket_{tgs}$

(4) TGS → C: $Ticket_v$

$Ticket_v = E_{Kv}$ [IDc||ADc||IDv||TS_2||TTL_2]

(5) C → V: IDc||$Ticket_v$

- Une clé partagée Kc entre chaque C et AS, issue du mot de passe f(Pc)=Kc
- Une clé partagée Ktgs entre chaque TGS et AS
- Une clé partagée Kv entre chaque serveur V et son TGS.

Dialogue d'authentification

- Problèmes:
  - Un adversaire peut voler un ticket et s'en servir
    - Un durée de vie est associé au ticket "ticket-granting"
      - Trop courte → demande répétée du mot de passe
      - Trop longue → plus d'opportunité pour une attaque par rejeu
  - Un adversaire peut voler un ticket et s'en servir avant qu'il n'expire !

Dialogue d'authentification

Authentication Service Exhange: To obtain Ticket-Granting Ticket

(1) C → AS: IDc || IDtgs ||TS1

(2) AS → C: $E_{Kc}[K_{c,tgs}|| IDtgs || TS_2 || TTL_2 || Ticket_{tgs}]$

$Ticket_{tgs} = E_{Ktgs}[K_{c,tgs}|| IDc||ADc||IDtgs||TS_2 || TTL_2 || Ticket_{tgs}]$

Ticket-Granting Service Echange: To obtain Service-Granting Ticket

(3) C → TGS: $IDv ||Ticket_{tgs} ||Authenticator_c$

(4) TGS → C: $E_{Kc}[K_{c,v}|| IDv || TS_4 || Ticket_v]$

$Ticket_v = E_{Kv}[K_{c,v}|| IDc||ADc||IDv||TS_4 || TTL_4 || Ticket_{tgs}]$

$Authenticatorc = E_{Kc}[K_{c,tgs}|| IDc || ADc || TS_3]$

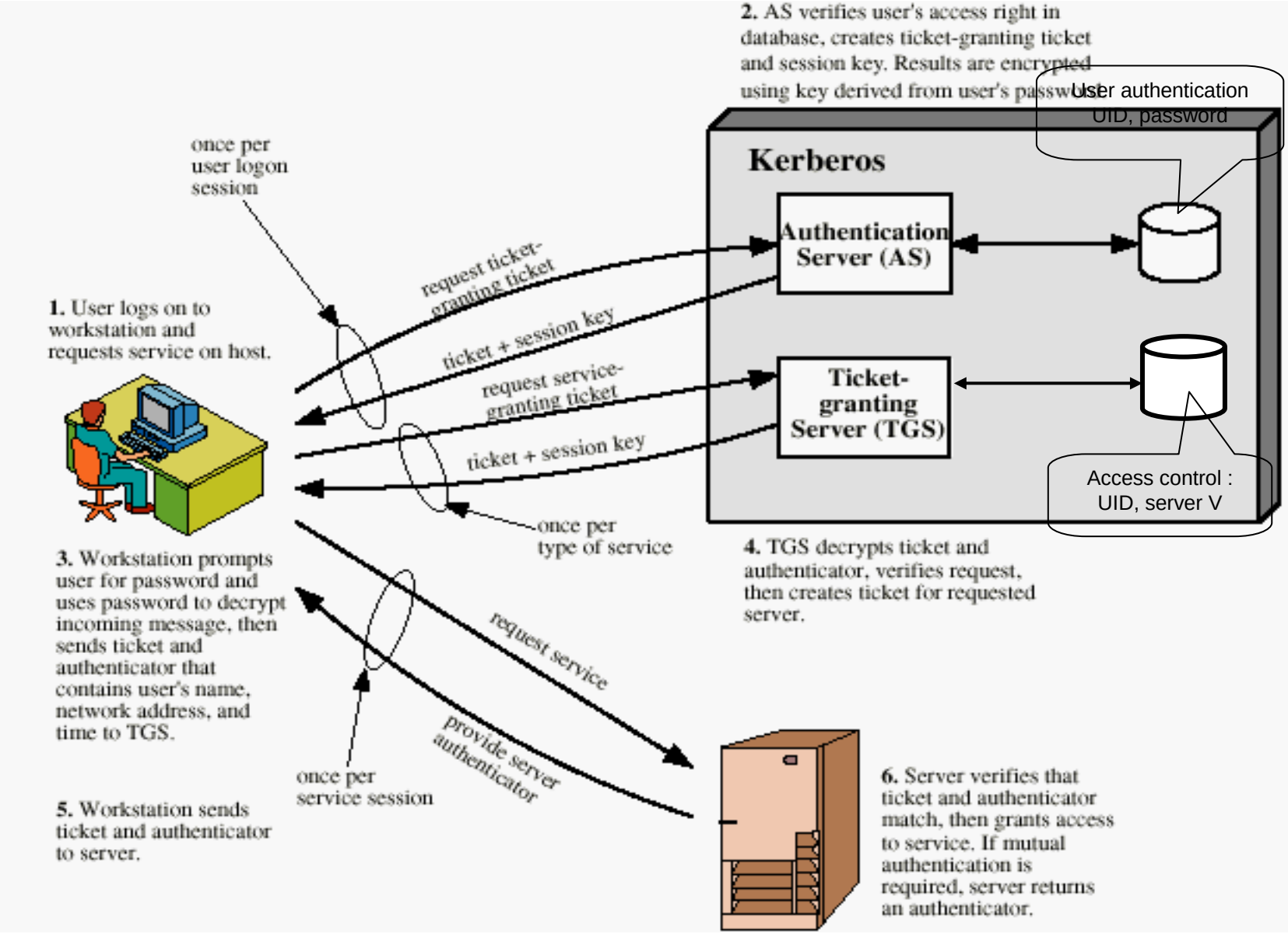
Client/Server Authentication Exhange: To obtain Service

(5) C → V: $Ticket_v$ || Authenticator'c

(6) V → C: EKc,v[TS5 +1]

$Authenticator'c = E_{Kc,v}[IDc || ADc || TS_5]$

Fonctionnement de Kerberos



Requêtes entre domaines Kerberos

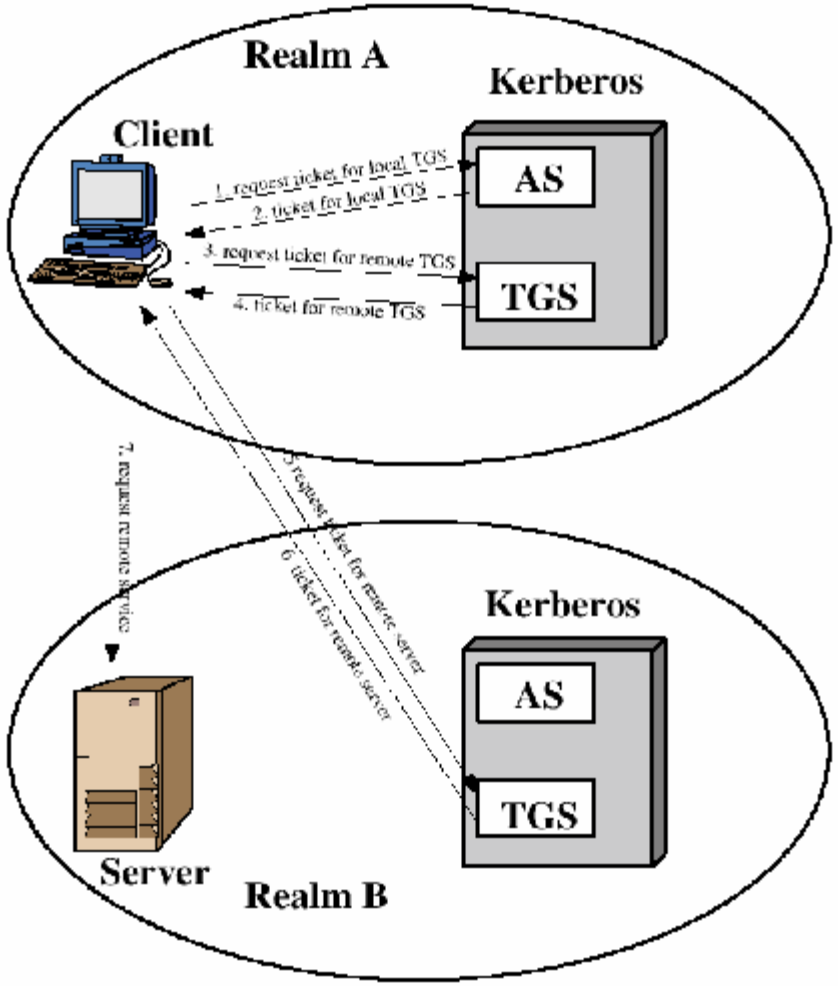


Figure 4.2 Request for Service in Another Realm

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 => V1=> V2)
- L'authentification entre domaines
  - $N^2/2$ authentification

Technique de chiffrement de Kerberos

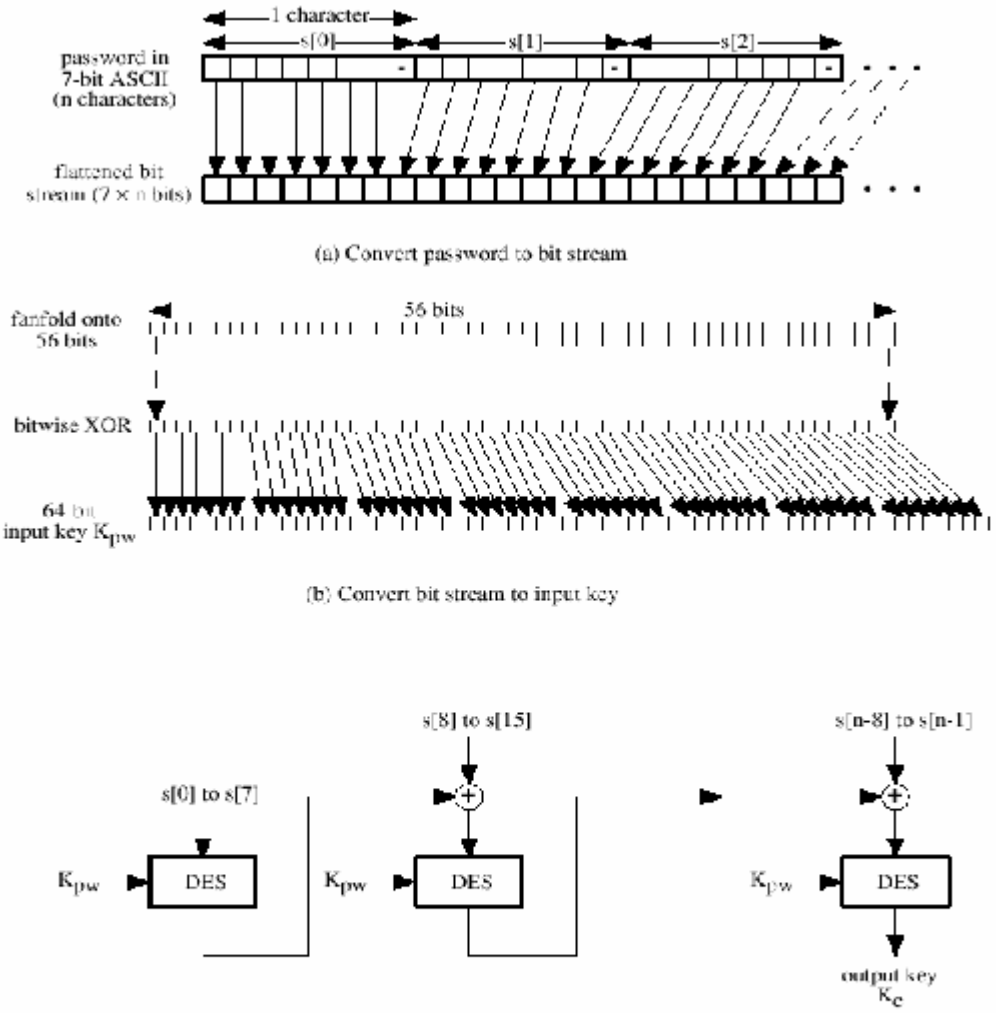


Figure 4.6 Generation of Encryption Key from Password

Le mode PCBC de DES

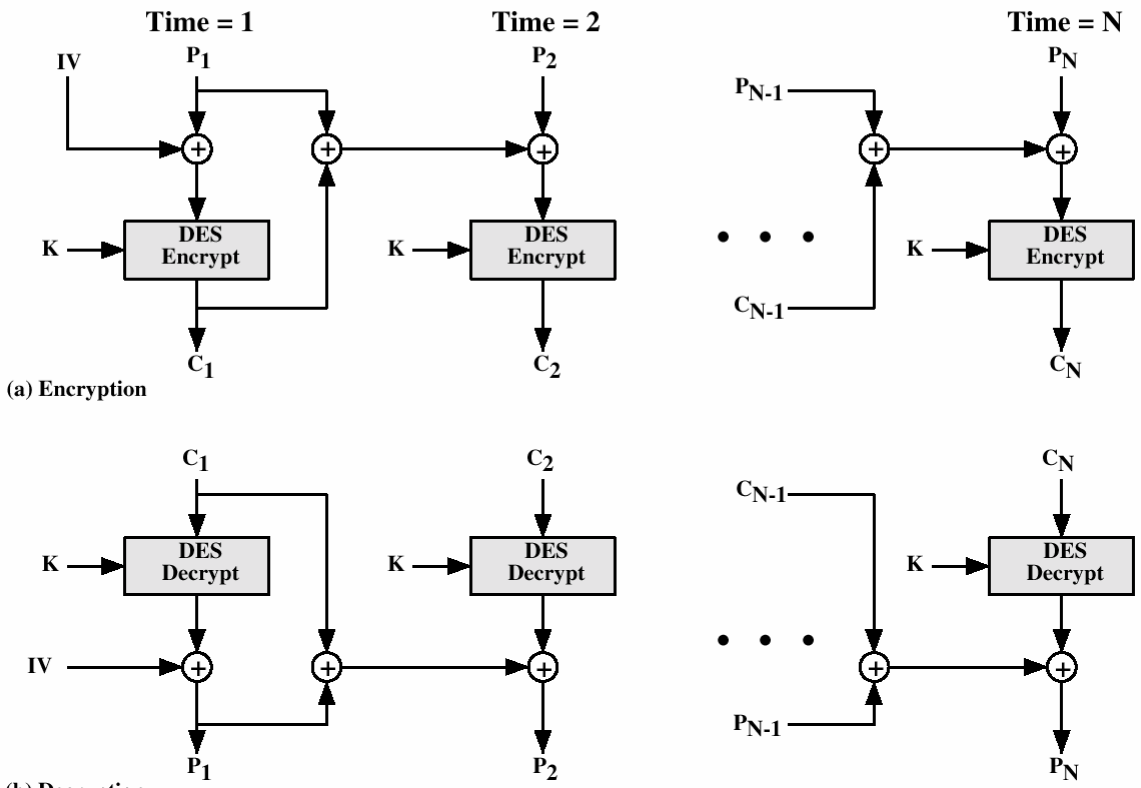


Figure 4.7 Propagating Cipher Block Chaining (PCBC) Mode

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

Qq systèmes d'authentification

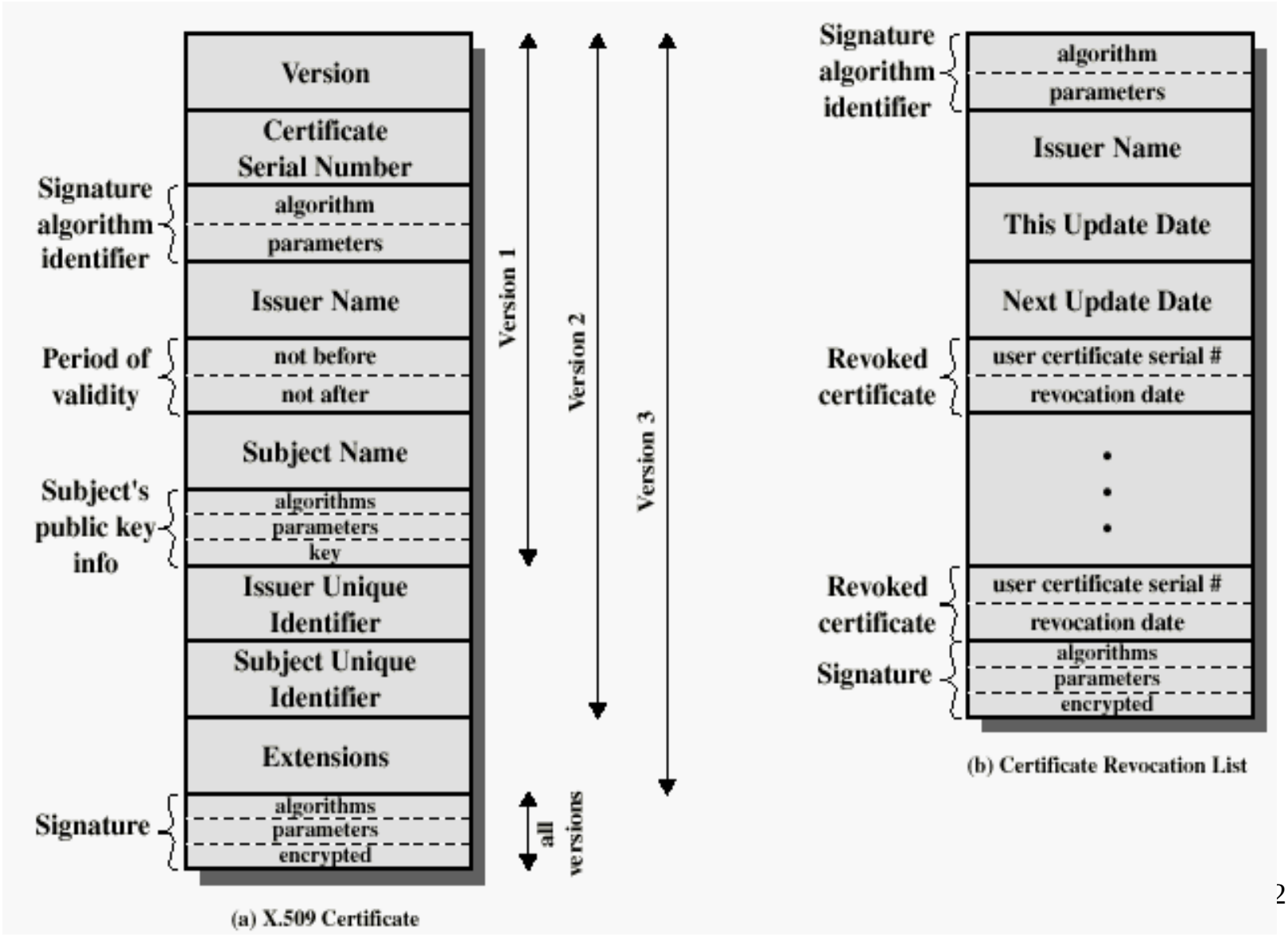
- ***AAA*** **:**
  - "Authentication, Authorization and Accounting"
  - RFC 2903 (2000)

- RADIUS
  - "Remote Authentication Dial In User Service"
  - Rfc 2865 (June 2000), m-à-j. par rfc 3373 (2003), en 2001 pour IPv6, etc.

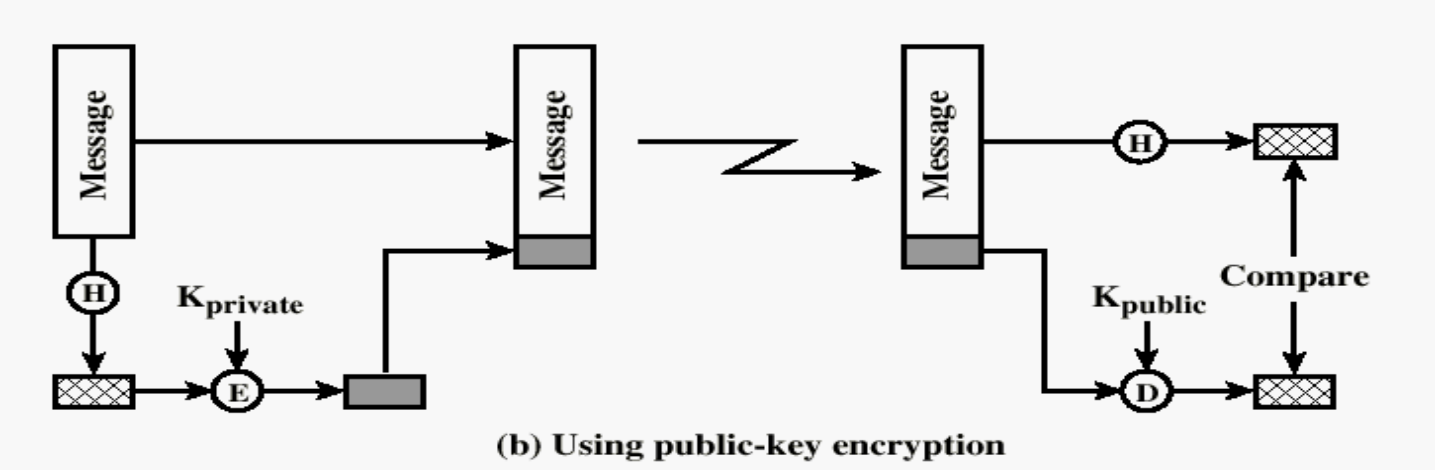
Le service d'authentification de X.509

- 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 definissait les certificats nécessaire 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 1998, V2 en 1993, V3 en 1995
    - Une certaine compatibilité ascendante

Le format X.509



Signature numérique à base de chiffrement asymétrique



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 gardée secrète
    - Intégrité des certificats

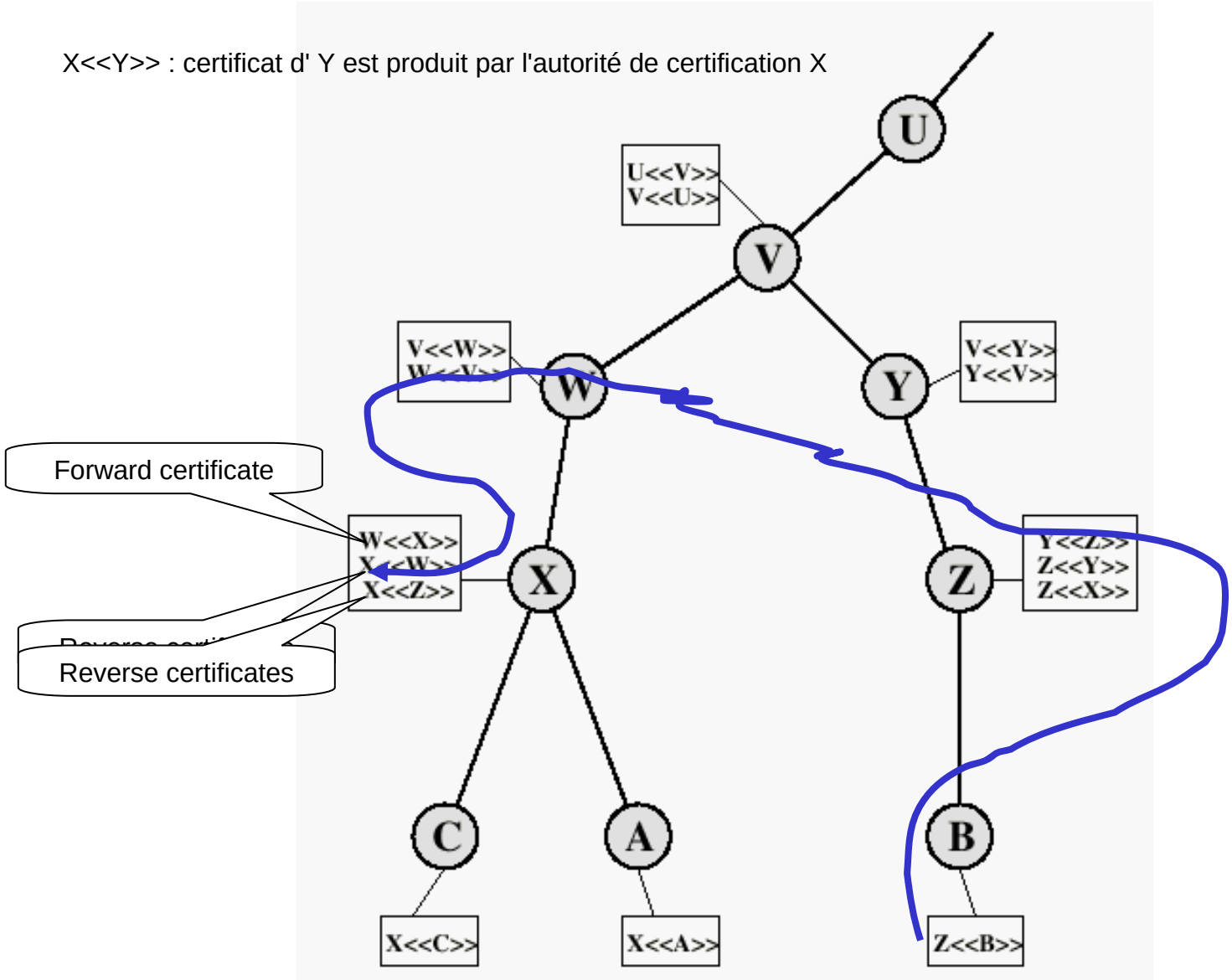
Certificat

- Notation usuel pour un certificat
  - Compatible X.509 v1

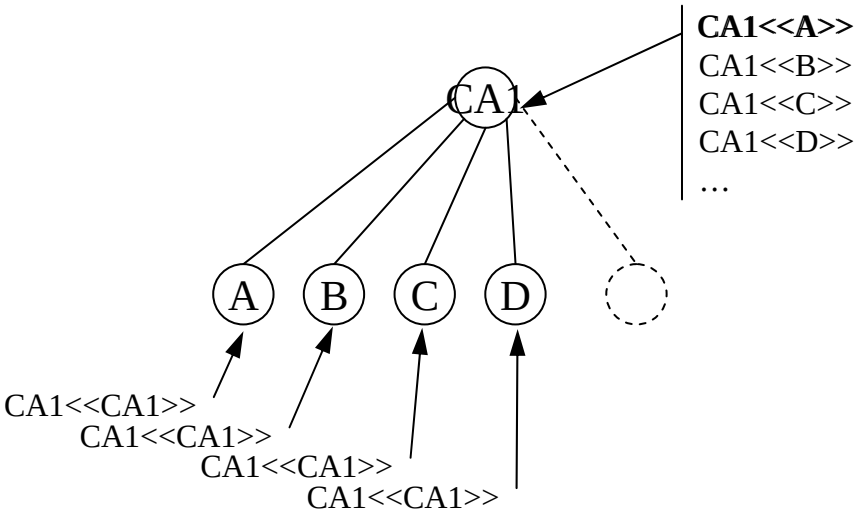
$CA\langle\langle A\rangle\rangle = CA_{KR\text{-}CA}\{$Version, Serial Number, Algorithm Identifier, CA, Timestamps, A, $KU_A\}$

Hiérarchie des certificats X.509

X<<Y>> : certificat d' Y est produit par l'autorité de certification X



Procédé d'authentification : mono CA



Hypothèses:

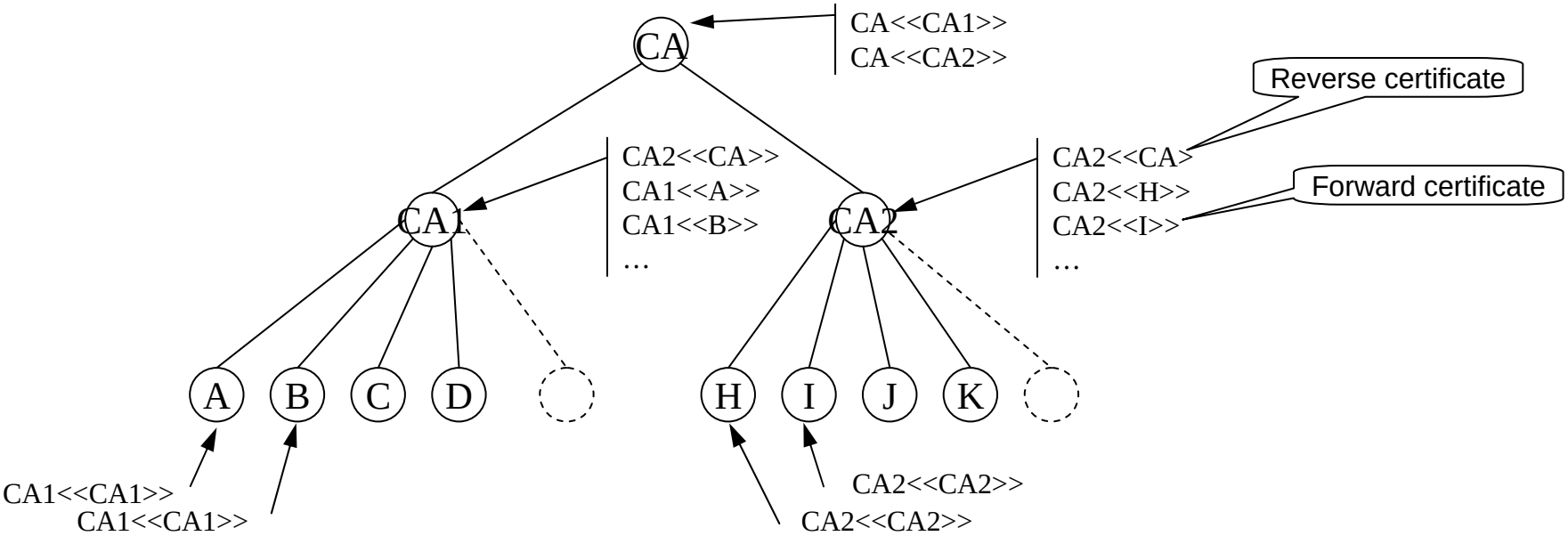
• L'autorité de certification (CA1) peut produire un certificat pour n'importe lequel de ses clients

• Chaque client fait confiance à l'autorité de certification

A envoie un message à B (A => B), le message contient un 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 clef publique de A. B obtient auprès de son autorité de certification CA1 le certificat de A : CA1<<A>>.

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<<CA1>>CA1<<A>>

Procédé d'authentification : multi *CA*



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

A envoie un message à H (A => H), le message contient un 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 CAl : CA1<<A>>.

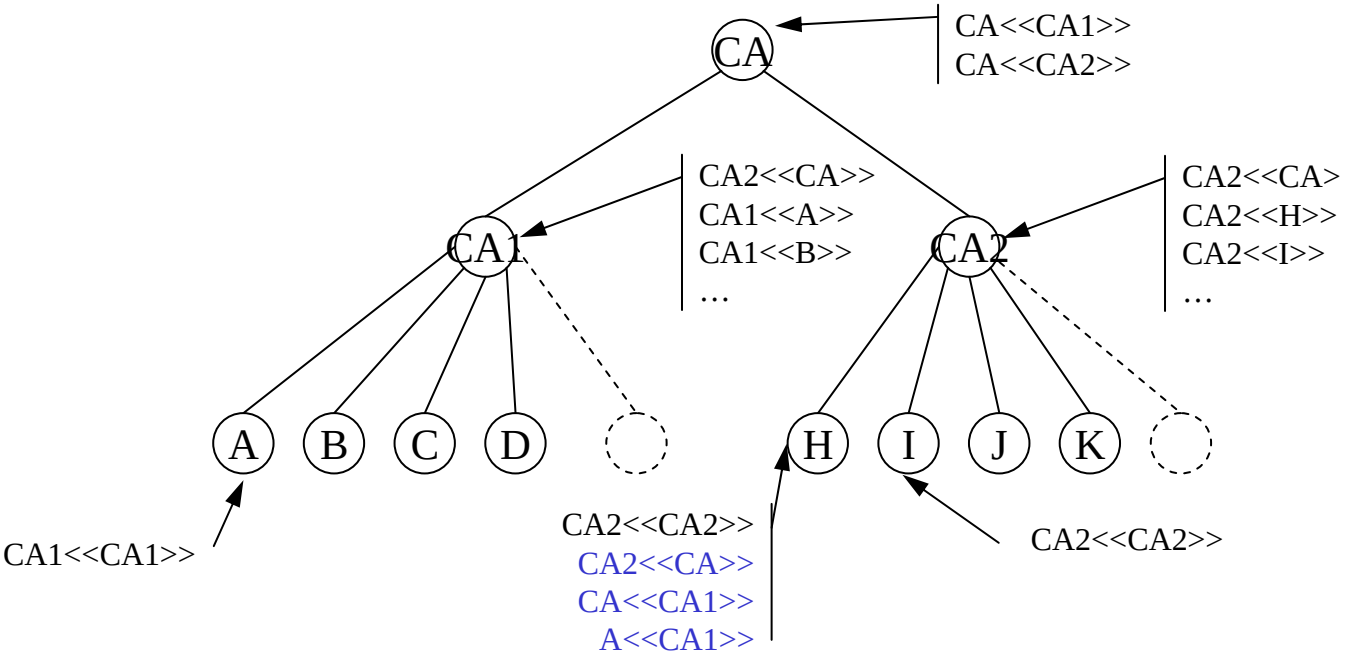
H peut vérifier l'authenticité de ce certificat car il fait confiance à CA2 (il a un certificat auto-signé de CA2). H construit une chaîne de confiance qui aboutit à A: CA2<<CA2>>CA2<<CA>>CA<<CA1>>CA1<<A>>

Hypothèses:

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

• 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.

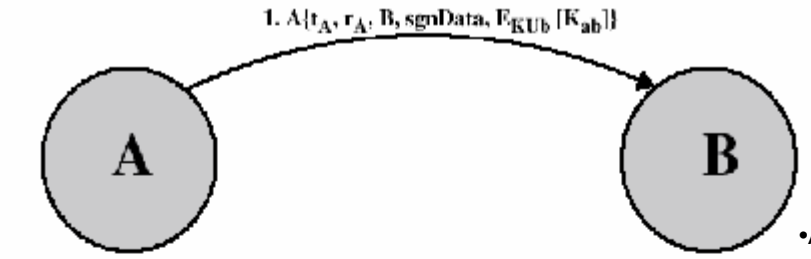
Révocation des certificats

- 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"

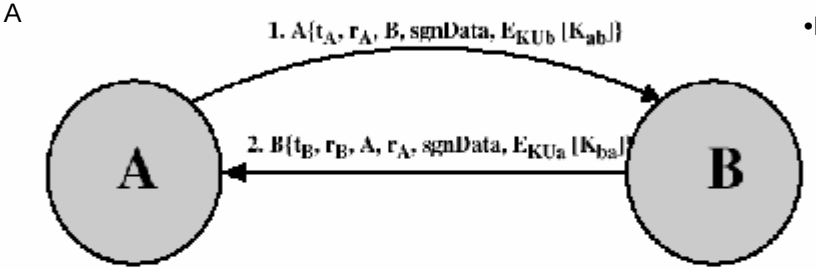
Protocoles d'authentification

t_A : timestamp (begin and end times)
r_A : nonce ("number once")
K_{ab} : session key
A{X} : message X + its digital signature by A



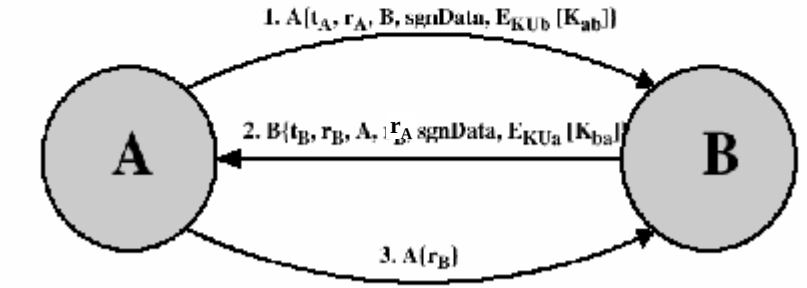
(a) One-way authentication

- Authentification of A
- Message was generated by A
- Message intented for B
- Integrity and originality of message



(b) Two-way authentication

- Idem +
- Reply message was generated by B
- Authentification of B



(c) Three-way authentication

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/gost/info/kerberos/