Cours sécurité informatique

Authentification

Plan

- Authentification, autorisation, identification
- Éléments ou facteurs d'authentification
- Types d'authentification
- Protocoles d'authentifications
  - Protocole de Needham-shroeder
  - Kerberos
  - Authentification SSL
  - Singne Sign On (SSO)
  - One time Password (OTP)
  - Authentification à seuil

User authentification, identification, autorisation

- **Authentification:**
  - Vérifier (s'assurer) de l'identité d'une entité (utilisateur, machine, processus…), afin de l'autoriser à accéder à des ressources (systèmes, BD, services, applications, …) selon ses droits d'accès (rôle)
  - ==> nécessaire pour le contrôle d'accès, l'autorisation, la coptabilté, les responsabilités...
- **Identification**
  - vérifier l’identité d’une **personne** en utilisant l'information qu'il introduit au système d'authentification.
- **Autorisation**
  - C'est la possession des droits nécessaires pour agir sur le système en exécutant une liste d’opérations (recherche, suppression, insertion, mise à jour,…).

Éléments ou facteurs d'authentification

- **Ce que l'entité connaît :**Mot de passe, code PIN, phrase secrète...
- **Ce que l'entité détient:** Carte magnétique, Carte à puce, Token (physique)...
- **Ce que l'entité est:** éléments biométrique: Empreinte digitale, empreinte rétinienne...
- **Ce que l'entité sait faire ou fait:** signature manuscrite, reconnaissance de la voix, un type de calcul connu de lui seul, un comportement...
- ==> peuvent être combinés

Types d'authentification

- **Authentification simple:**
  - Repose sur **un seul** élément d'authentification
  - Exemple: login/password
- **Authentification forte:**
  - Repose sur deux éléments ou plus
  - Exemple : l'utilisateur insère sa carte à puce et spécifie son code PIN

Protocoles d'authentification

- Utilisé par des intervenants pour prouver leurs identités les uns aux autres et , le plus souvent, échanger des clés de session
- Peuvent être “one way” ou mutuels
- Problème de:
  - Confidentialité de la clé de session échangée
  - Attaques de rejeu: ré-envoyer des anciens messages légitimes

    ==> **Solution 1**: utiliser des numéro de séquence (peu pratique)

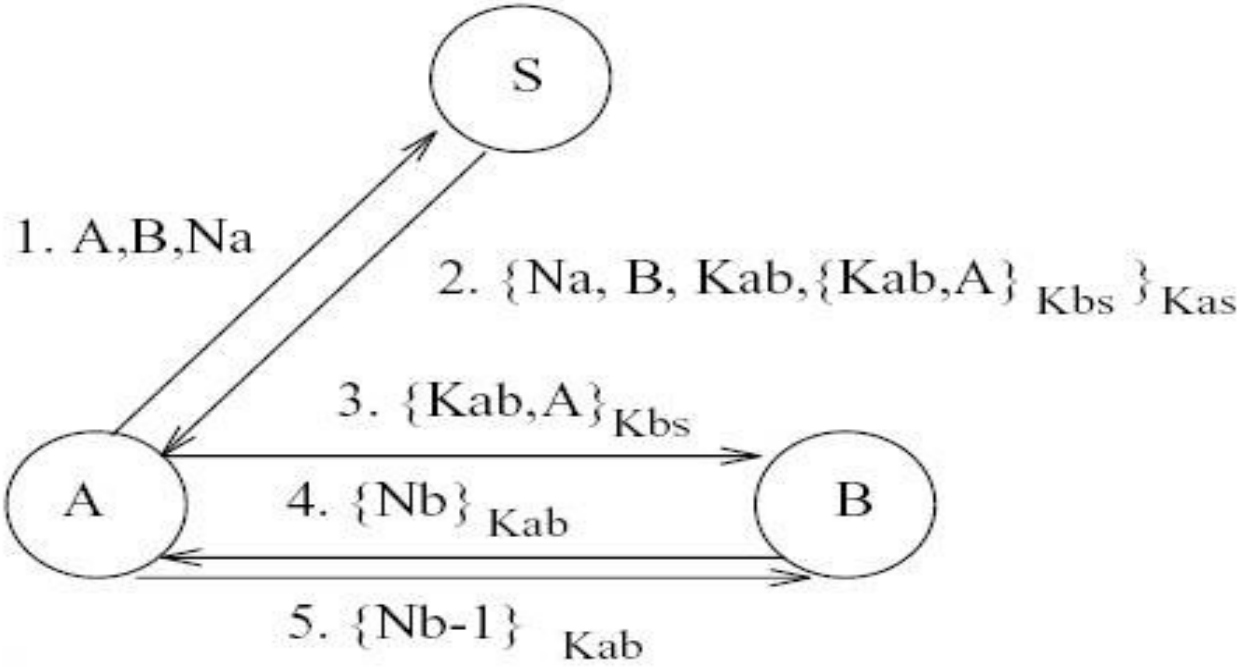
    ==> **Solution 2**: utiliser des timestamps (nécessite des horloges synchronisées)

    ==> **Solution 3**: challenge/response (utiliser des nonces uniques)

    ==> ...

Needham-Schroeder Authentication Protocol

- Se base sur KDC (Key distribution center) de confiance
- Chaque entité maintient une clé symétrique avec le KDC (Kas, Kbs)
- Le KDC génère une clé symétrique Kab pour deux entités

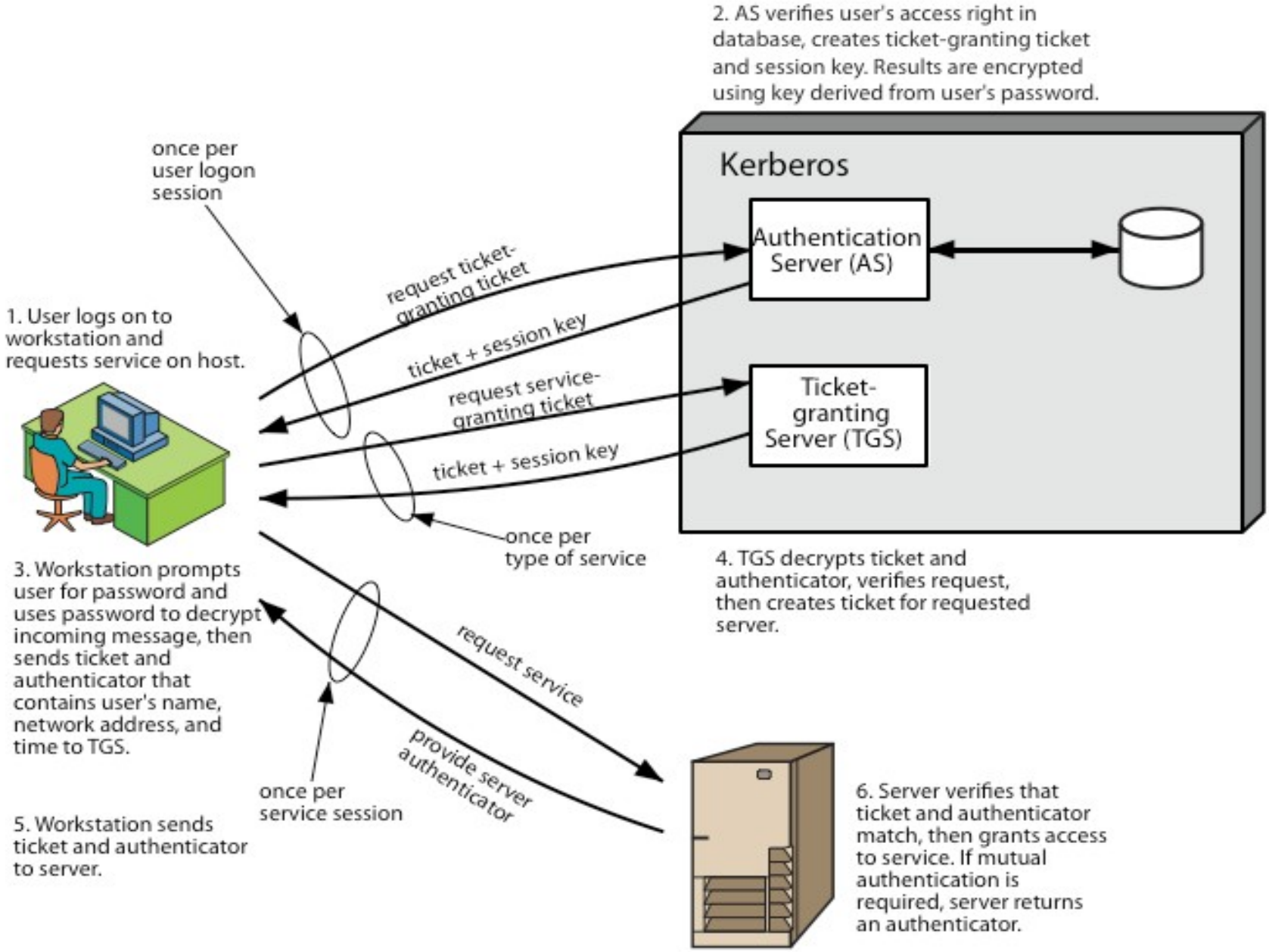


- **Vulnerable à l'attaque de rejeu** si une ancienne clé Kab est découverte (message 3)
- Solution 1: timestamps aux étapes 2 et 3 (Denning 81)
- Solution 2: utiliser un autre nonce (Neuman 93)

Kerberos authentication protocol

- Authentification centralisé basé sur un tier de confince
- Permet aux utilisateurs d'accéders à des services sur le réseau
- Version utilisées: 4 et 5
- Implémentation basé sur le protocole Needham-shroeder

Kerberos authentication protocol V4 overview



Kerberos authentication protocol V4 overview

(1) $C \rightarrow AS \quad ID_c \| ID_{tgs} \| TS_1$

(2) $AS \rightarrow C \quad E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) $C \rightarrow TGS \quad ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) $TGS \rightarrow C \quad E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) $C \rightarrow V \quad Ticket_v \| Authenticator_c$

(6) $V \rightarrow C \quad E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$$

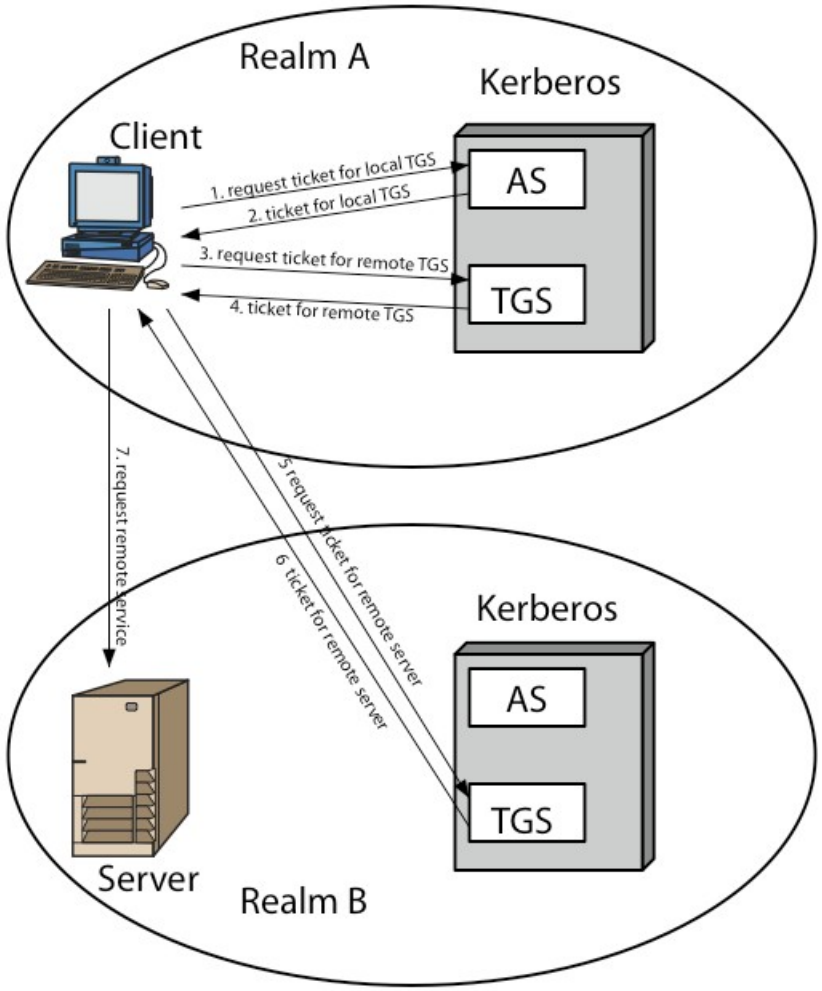
(c) Client/Server Authentication Exchange to obtain service

Kerberos authentication protocol V5

- Développé en 1990 ==> Internet standard RFC 1510
- Améliore la version 4
  - Lacunes générales:
    - Encryption algorithm, ticket lifetime, network protocol, byte order, authentication forwarding, inter-realm authentication
  - Insuffusances techniques
    - Double encryption, non std mode of use, sessions keys, password attacks

Kerberos realms

- Un environnement kerberos englobe:
  - Un serveur kerberos krb_serv
  - Des utilisateurs enregistré avec krb_serv (clés de type Kc)
  - Serveurs d'applications enregistrés avec le krb_serv (clé Kv)
- ==> cela constitue un realm (domaine administratif)
- Cas de multiple realms ==> les serveurs kerberos partages des clés.



Kerberos authentication protocol V5

(1) C → AS Options ‖ ID_c ‖ $Realm_c$ ‖ ID_{tgs} ‖ $Times$ ‖ $Nonce_1$

(2) AS → C $Realm_c$ ‖ ID_C ‖ $Ticket_{tgs}$ ‖ $E(K_c, [K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) C → TGS Options ‖ ID_v ‖ $Times$ ‖ ‖ $Nonce_2$ ‖ $Ticket_{tgs}$ ‖ $Authenticator_c$

(4) TGS → C $Realm_c$ ‖ ID_C ‖ $Ticket_v$ ‖ $E(K_{c,tgs}, [K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v])$

$Ticket_{tgs} = E(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

$Ticket_v = E(K_v, [Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$

$Authenticator_c = E(K_{c,tgs}, [ID_C \| Realm_c \| TS_1])$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) C → V Options ‖ $Ticket_v$ ‖ $Authenticator_c$

(6) V → C $E_{K_{C,V}}$ [TS_2 ‖ Subkey ‖ Seq#]

$Ticket_v = E(K_v, [Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$

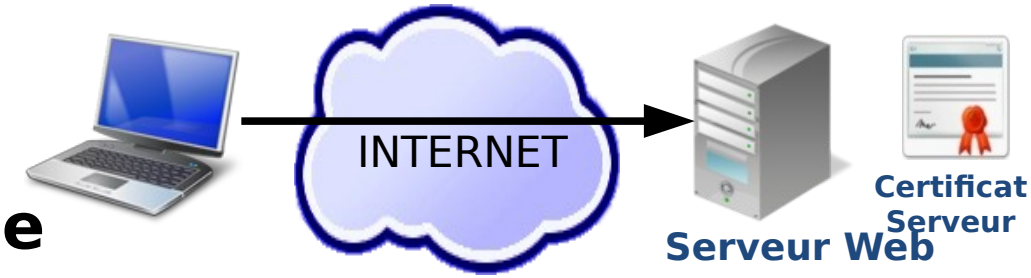
$Authenticator_c = E(K_{c,v}, [ID_C \| Realm_c \| TS_2 \| Subkey \| Seq\#])$

(c) Client/Server Authentication Exchange to obtain service

Remote user authentification

- Public key encryption for session key distribution
  - Suppose que les 2 intervenants possèdent les **vraies** clés publiques l'un de l'autre
- Denning protocol using timestamps
  - Utile un serveur central qui donne des cetificats des clés publiques
  - Nécessitent la synchronisation
  -
- Woo and Lam protocol utilisant des nonces

Authentification SSL



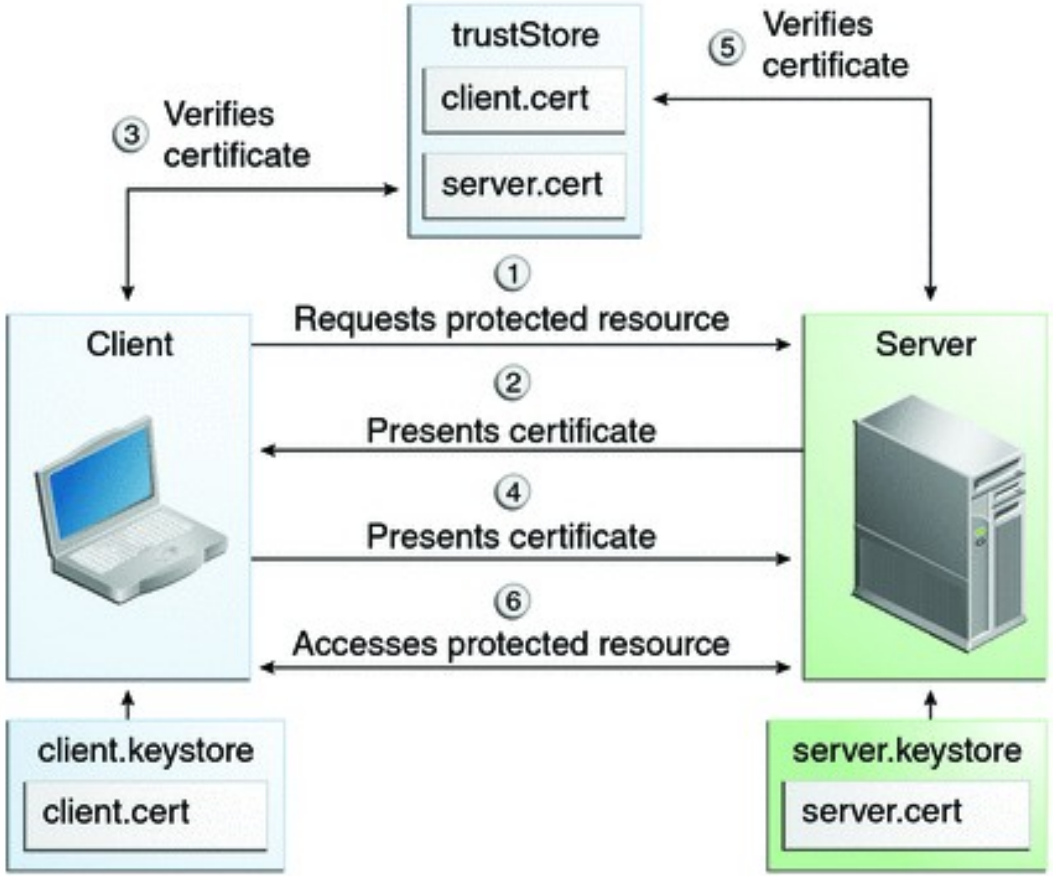
- **Authentification SSL simple**
  - L'internaute sera capable d'identifier le serveur grâce à son **certificat électronique** (avant de faire des opérations)

```
No.  Time          Source          Destination     Protoc  Length  Info
 34  0.774472000   192.168.1.4     193.95.13.57    TCP         66  35164 > https [ACK] Seq=206 Ack=2049 Win=34944 Len=0 TSval=665628
 35  0.778457000   193.95.13.57    192.168.1.4     TLSv1.2   1454  Certificate

    Certificate Length: 1197
  ▼Certificate (id-at-commonName=*.google.com.tn,id-at-organizationName=Google Inc,id-at-localityName=Mountain View,id-at-stateO
   ▼signedCertificate
     version: v3 (2)
     serialNumber: -2074359676
    ▶signature (sha256WithRSAEncryption)
    ▼issuer: rdnSequence (0)
     ▶rdnSequence: 3 items (id-at-commonName=Google Internet Authority G2,id-at-organizationName=Google Inc,id-at-countryName=US
    ▼validity
     ▶notBefore: utcTime (0)
     ▶notAfter: utcTime (0)
    ▼subject: rdnSequence (0)
     ▶rdnSequence: 5 items (id-at-commonName=*.google.com.tn,id-at-organizationName=Google Inc,id-at-localityName=Mountain View,
    ▼subjectPublicKeyInfo
     ▼algorithm (rsaEncryption)
       Algorithm Id: 1.2.840.113549.1.1.1 (rsaEncryption)
      Padding: 0
      subjectPublicKey: 3082010a0282010100c33c6270c99b7c4d176bd755935d52...
    ▶extensions: 8 items
   ▶algorithmIdentifier (sha256WithRSAEncryption)
    Padding: 0
    encrypted: 77a11396ed091e8838f62c8ffd7b641bfe2222dc6d93185f...
   Certificate Length: 1012
```

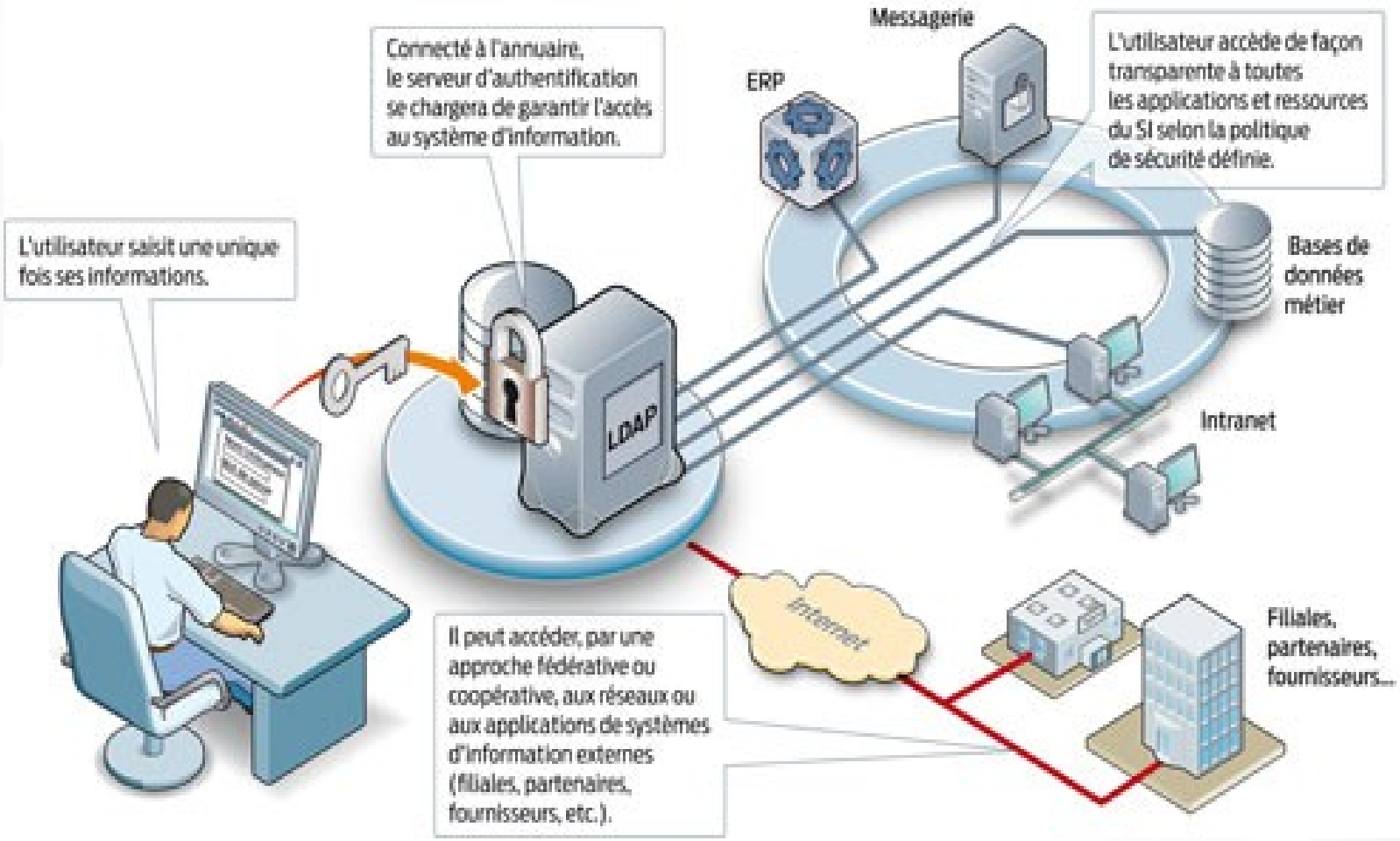
Authentification SSL

- **Authentification SSL mutuelle**
  - L'internaute identifier le serveur et le serveur identifie le client grâce aux **certificats électroniques**



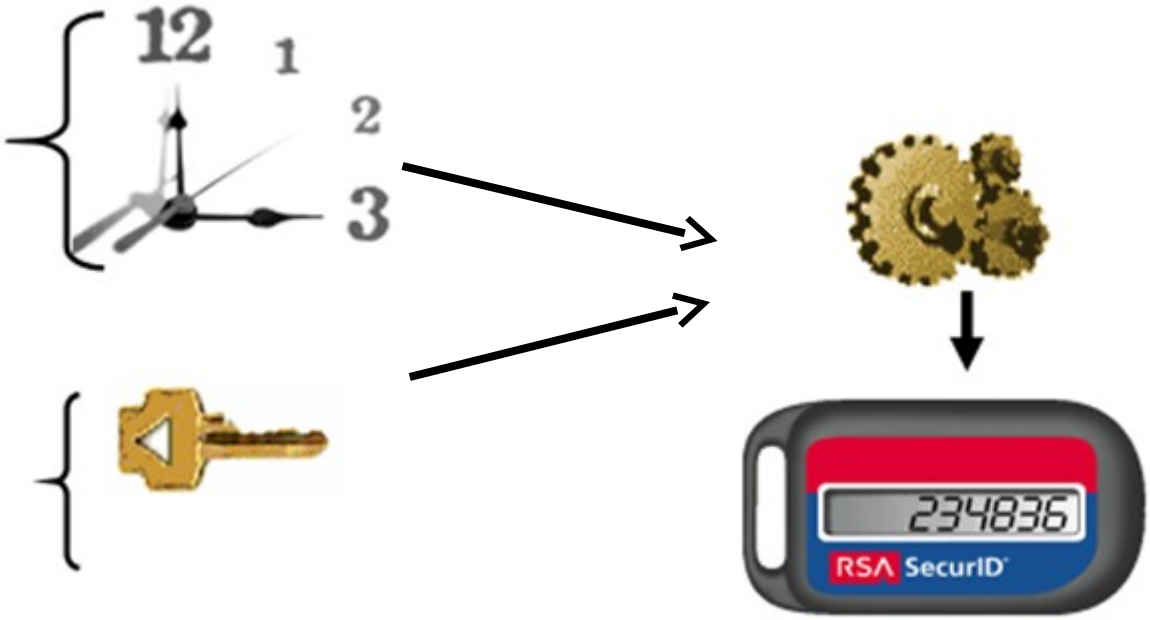
Authentification SSO (Single Sign On)

- Méthode où lutilisateur s'authentifie une seule fois pour accéder à plusieurs applications ou services.



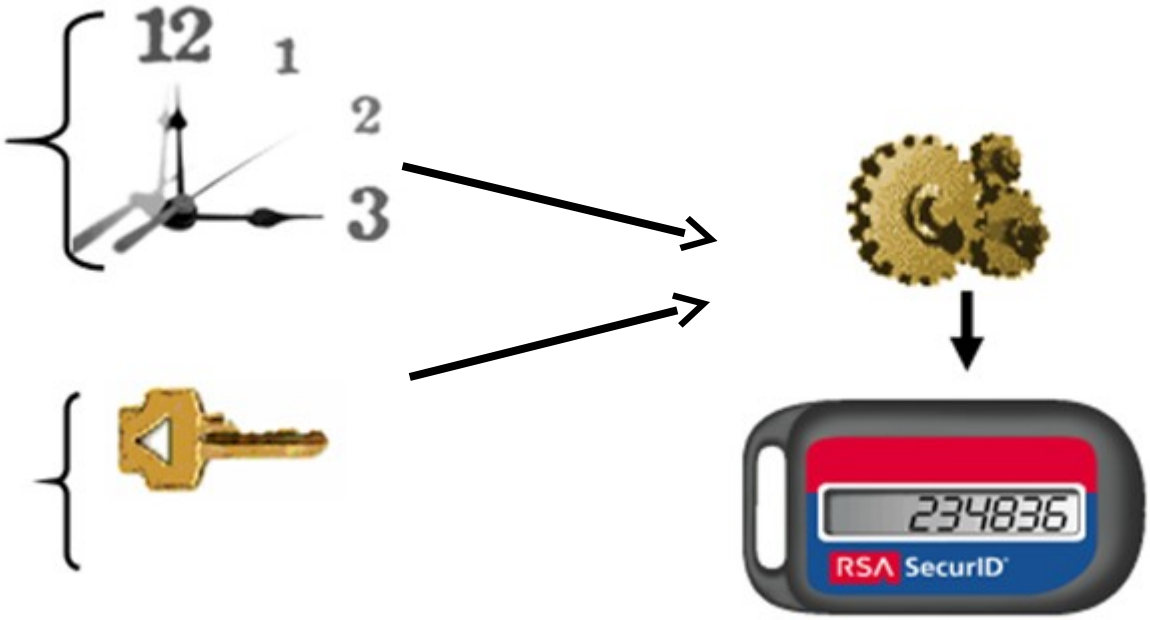
Authentification OTP (One Time Password)

- Authentification avec un **mot de passe à usage unique** (un mot de passe par session)
- Basée sur l'utilisation d'un secret partagé entre l'authentificateur et le serveur d'authentification ==> permet de generer les mêmes OTP chez les deux systèmes
- Un OTP est généré à partir du secret partagé et du temps de la transaction=> n'est valable que pour la transaction ou la session.



Authentification OTP (One Time Password)

- Authentification avec un **mot de passe à usage unique** (un mot de passe par session)
- Basée sur l'utilisation d'un secret partagé entre l'authentificateur et le serveur d'authentification ==> permet de generer les mêmes OTP chez les deux systèmes
- Un OTP est généré à partir du secret partagé et du temps de la transaction=> n'est valable que pour la transaction ou la session.

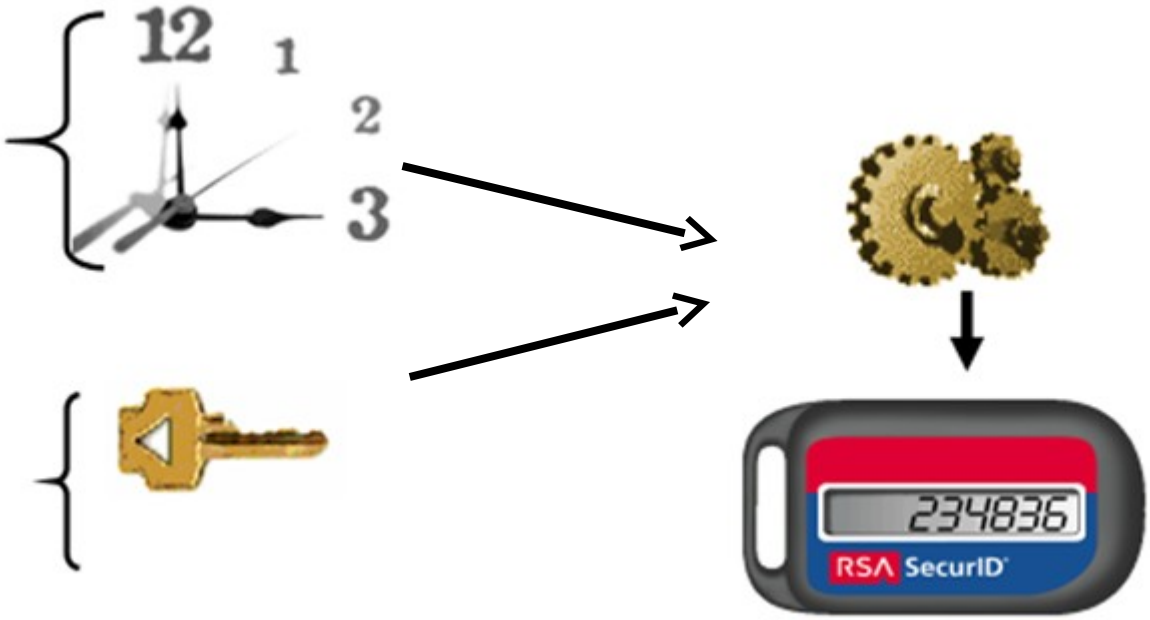


Authentification à seuil (K parmi N)

- Baseé sur le partage de secret.
- Cas de certaines tâches qui se font obligatoirement en la présence d’un nombre minimal K de personnes parmi N personnes autorisées.

Authentification OTP (One Time Password)

- Authentification avec un **mot de passe à usage unique** (un mot de passe par session)
- Basée sur l'utilisation d'un secret partagé entre l'authentificateur et le serveur d'authentification ==> permet de generer les mêmes OTP chez les deux systèmes
- Un OTP est généré à partir du secret partagé et du temps de la transaction=> n'est valable que pour la transaction ou la session.



Authentification SSO: approches

- **Approche centralisée:** Authentification unique est basée sur une base de données ou annuaire globale et centralisée de tous les utilisateurs commune à tous les services. La gestion de la politique de sécurité est centralisée;

Approche fédérative: Chaque service gère une partie des données d'un utilisateur (l'utilisateur peut donc disposer de plusieurs comptes), mais partage les informations dont il dispose sur l'utilisateur avec les services partenaires. Cette approche permet une gestion décentralisée des utilisateurs et une gestion décentralisée de la politique de sécurité;

Approche coopérative: Cette approche part du principe que chaque utilisateur dépend d'une des entités partenaires. Ainsi, lorsqu'il cherche à accéder à un service du réseau, l'utilisateur est authentifié par le partenaire dont il dépend. Comme dans l'approche fédérative, cependant, chaque service du réseau gère indépendamment sa propre politique de sécurité.

Identity management

- ..

Je teste mes connaissances…

- …