Module Trc-8
Sécurité dans les réseaux

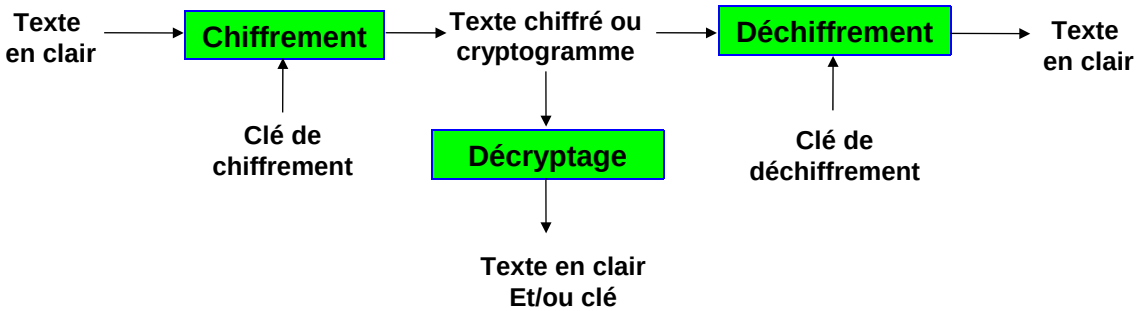
- Introduction à la cryptographie
  - TP1: **Cryptage avec openssL**
- Protocole SSL/TLS (Layer 4)
  - TP2: **Mise en œuvre du protocole HTTPS**
- Sécurité des réseaux – Les menaces
- Les réseaux privés virtuels (VPN)
  - TP3: **Mise en place d'un client/serveur VPN**
- Le protocole IPSec (IP Security – Layer 3)
  - TP4: **Configuration et mise en œuvre de IPSec**

Introduction à la cryptographie

- Terminologie
- Mécanismes et services de sécurité
- Confidentialité et algorithmes de chiffrement
- Fonctions de hachage, scellement et signature
- Authentification et échange de clefs de session
- Certificats et PKI
- Liens

Terminologie

- Cryptologie = cryptographie + cryptanalyse :
  - la cryptographie qui est le chiffrement ou cryptage de messages en clair et le déchiffrement ou décryptage de messages codés, connaissant la clé.
  - la cryptanalyse qui est l'art de décrypter des messages codés sans connaître la clé ("code breaking").



Terminologie - Cryptographie

Un système cryptographique ou *crypto-systèmes* est composé d'un algorithme de cryptage (chiffrement) et d'un algorithme de décryptage (déchiffrement).

Types de crypto-systèmes :

• Systèmes à usage restreint: les algorithmes de chiffrement et de déchiffrement sont secrets. La sécurité repose sur leur confidentialité.

• Systèmes à usage général: la confidentialité ne repose pas sur l'algorithme, mais sur une clé. Tout le monde peut utiliser le système.

Les crypto-systèmes modernes sont des systèmes à clé.

$$E_K(m) = c,\ D_{K'}(c) = m$$

Services de sécurité

But de la cryptographie moderne : fournir un certain nombre de services de sécurité :

• Confidentialité: le message crypté doit rester secret. Ne peut être décrypté par un tiers.

- Est-ce qu'un autre nous écoute ?

• Authentification: assurance de l'authenticité de l'origine.

- Est-ce bien lui ?

• Intégrité: assurance que le message n'a pas été modifié durant la transmission.

- Le contenu est-il intact ?

• Non répudiation: l'expéditeur ne peut pas nier, ultérieurement, avoir envoyé le message.

- Correspondant de mauvaise Foi

Les mécanismes

Moyens mis en œuvre : mécanismes de sécurité construits au moyen d'outils cryptographiques (fonctions, algorithmes, générateurs aléatoires, protocoles…) mettant en œuvre les services précédents

- Chiffrement: assure la confidentialité des données.
- Scellement: assure l'intégrité des données.
- Signature numérique: authentifie l'émetteur des données.
- Protocoles d'authentification mutuelle avec échange de clés: sécurise l'échange des clés.

Confidentialité et algorithmes de chiffrement

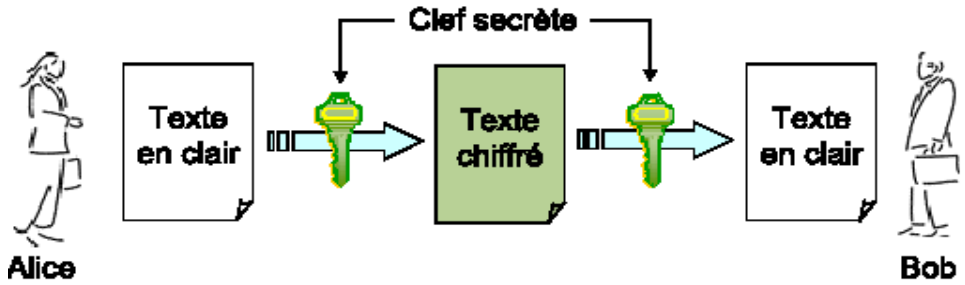
La confidentialité est historiquement le premier problème posé à la cryptographie. Il se résout par la notion de chiffrement.

Deux classes d'algorithmes:

- Algorithmes *symétriques* ou *à clé secrète*
  - Plus rapides donc préférés pour le chiffrement de données.
- Algorithmes *asymétriques* ou *à clé publique*.
  - Bien plus lents que les algorithmes à clé secrète.
  - Utilisés pour l'échange de clés secrètes (**clés de session**)

Chiffrement symétrique - Principe

Clé de chiffrement = clé de déchiffrement, elle doit rester secrète aux tiers communiquant.



(Source : G. Labouret – © 1999-2001, Hervé Schauer Consultants)

Chiffrement symétrique – Algorithmes (1)

Deux catégories :

• Algorithmes de chiffrement en continu (***stream ciphers***): travaillent bit à bit ou octet à octet.

- exemple : RC4 (taille de la clé variable, 128 bits en pratique)

• Algorithmes de chiffrement par blocs (***block ciphers***): opèrent sur le texte clair par blocs (généralement, 64 bits).

- DES (clé de 56 bits codée sur 64 bits)
- 3DES : application de 3 DES successivement avec 3 clés indépendantes.
- AES (Advanced Encryption Standard) : longueur de clé variable (128,192, 256)

Chiffrement par blocs - Principe

L'idée générale est la suivante:

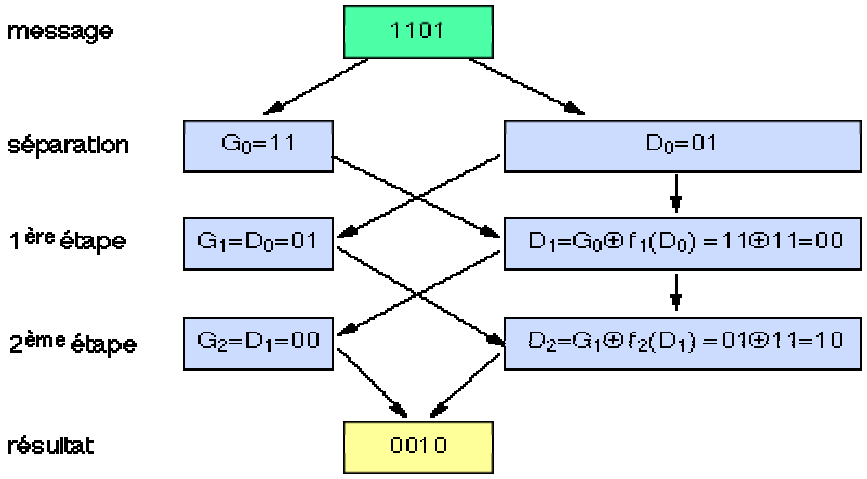
- Remplacer les caractères par un code binaire (code ASCII en base 2).

3. Découper cette chaîne en blocs de longueur de 64 bits.
4. Chiffrer un bloc en l'additionnant bit par bit à une clef.
5. Déplacer certains bits du bloc.
6. Recommencer un certain nombre de fois l'opération 3. On appelle cela une ronde.
7. Passer au bloc suivant et retourner au point 3 jusqu'à ce que tout le message soit chiffré.

Réseaux de Feistel

Exemple : chiffrer à deux rondes un message constitué de quatre bits

entrée	f_1	sortie	entrée	f_2	sortie
00	→	01	00	→	11
01	→	11	01	→	00
10	→	10	10	→	00
11	→	01	11	→	01



Chiffrement symétrique – Algorithmes (2)

• Cryptage par substitution: chaque lettre est remplacée par une autre lettre (permutation circulaire σ de l'alphabet).

- le texte $m = a_1 \ldots a_n$ donne $c = \sigma(a_1) \ldots \sigma(a_n)$
- code de César d'ordre 3 : $\sigma(a) = a + 3 \bmod 26$
- CESAR → FHVDU

• Cryptage par transposition : consiste à modifier, selon une loi prédéfinie, l'ordre des lettres mais ne les masquent pas.

- ex. transposition par colonnes : soit $n = pq$. On écrit le texte en p lignes de longueur q, puis on lit les colonnes, dans l'ordre donné par la clé.
- texte en clair : JE NE DORS PAS EN COURS RT; n=20, p=4, q=5
- clé : TUNIS (45213) ; texte chiffré : EPCRNSNSDAOTJOSUERER

Chiffrement symétrique – Algorithmes (3)

• Code de Vigénère: On choisit une clé $k = k_1 \cdot \cdot \cdot k_p$. On répète la clé jusqu'à obtenir la longueur du message $m = a_1 \cdot \cdot \cdot a_n$. Le message crypté est : $c_i = k_i + a_i \bmod 26$

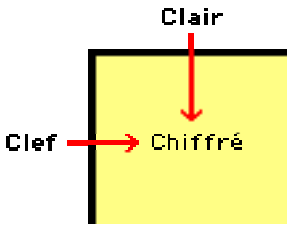
- utilise le principe du carré de Vigénère

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H

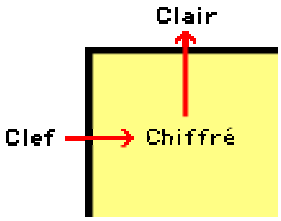
…

Utilisation du carré vigénère

- Vigénère **(chiffré = clair + clef)**
  - La lettre de la clef est dans la colonne la plus à gauche
  - la lettre du message clair est dans la ligne tout en haut
  - La lettre chiffrée est à l'intersection de la ligne de la lettre clef et de la colonne de la lettre claire.



Pour chiffrer



Pour déchiffrer

Clair HELLOWORLD
Clef ECSECSECSE
chiffré LGDPQOSTDH

Exemple

Le chiffre de Vigénère

- La clef définit un décalage pour chaque lettre du message (A: décalage de 0 cran, B: 1 cran, C: 2 crans, ..., Z: 25 crans).
  - Exemple : chiffrer le texte "CHIFFRE DE VIGENERE" avec la clef "BACHELIER".

Clair	C	H	I	F	F	R	E	D	E	V	I	G	E	N	E	R	E
Clef	B	A	C	H	E	L	I	E	R	B	A	C	H	E	L	I	E
Décalage	1	0	2	7	4	11	8	4	17	1	0	2	7	4	11	8	4
Chiffré	D	H	K	M	J	C	M	H	V	W	I	I	L	R	P	Z	I

- La grande force du chiffre de Vigénère est que la même lettre sera chiffrée de différentes manières.
  - Par exemple le E du texte clair ci-dessus a été chiffré successivement M V L P I.

Chiffrement symétrique – avantages et inconvénients

- Inconvénients :
  - la découverte de la clé secrète donne accès à l'information.
  - Problème de la distribution des clefs
    - ***Transport sécurisé*** de la clef de chiffrement ? (Problématique de l'échange de clé)
      - le secret doit être transmis d'où risques d'interception
      - Établissement préalable d'un canal sûr pour transmettre la clé
    - Nombre de clés échangées (en n^2)
- Avantages :
  - demande relativement peu de puissance.
  - Rapidité, et facilité de mise en oeuvre sur des circuits "bon marché ".

Chiffrement symétrique – Principal usage

- Service de confidentialité :
  - Problème de la distribution des clés secrètes.
- Peut assurer un service d'authentification :
  - Sans service de non répudiation
    - L'utilisateur n'est pas le seul à pouvoir produire la signature!
- Exemple de la monétique :
  - Clé secrète dépend du porteur (carte à puce)
  - Chiffrement (DES 56 bits) des paramètres suivants :
    - Montant de la transaction, Identifiant du commerçant
  - Authentification au niveau des ATM (par la piste magnétique)

Chiffrement asymétrique (1)

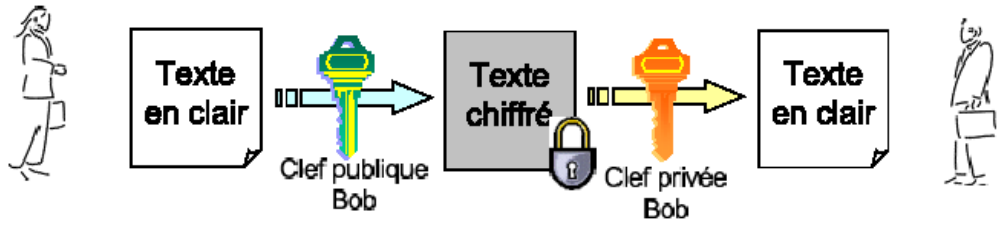
- Clé de chiffrement et de déchiffrement distinctes:
  - une clé publique (e, *clé de chiffrement*) connue de tous, et une clé privée (d, *clé de déchiffrement*) qui n'est connue que de l'un des correspondants.
  - sa sécurité dépend de la difficulté à résoudre certains problèmes mathématiques :
    - utilisation d'une fonction à sens unique : $f(x) = x^e [n]$
    - d non déductible modulo la connaissance de **e** et de **n**.
  - algorithmes très lents pour une utilisation intensive (chiffrement de données)
    - utilisés seulement pour l'échange de clé (chiffrer une clé de session), la signature.
  - algorithme RSA (Rivest, Shamir & Adelman), 1977

Chiffrement asymétrique (2)

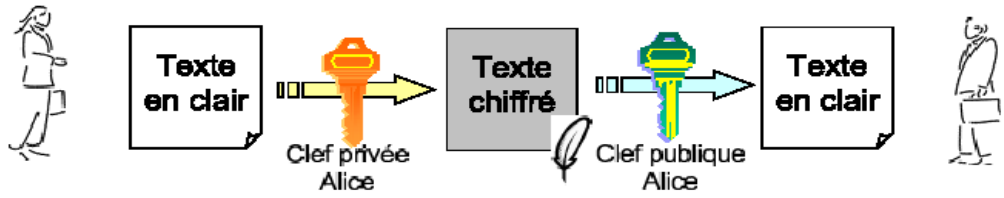
- Chiffrement → confidentialité
  - clé publique utilisée pour le chiffrement; tout le monde peut chiffrer un message, que seul le propriétaire de la clé privée pourra déchiffrer.
- Signature → authentification
  - L'authentification de l'émetteur peut être obtenue en chiffrant avec la clé privée et en le déchiffrant avec la clé publique;
  - Il ne garantit que l'origine (détenteur de la clé privée) mais pas la confidentialité des données (n'importe qui pourra déchiffrer).

Chiffrement asymétrique (3)

• Chiffrement : clé publique utilisée pour le chiffrement, seul le détenteur de la clé privée peut déchiffrer



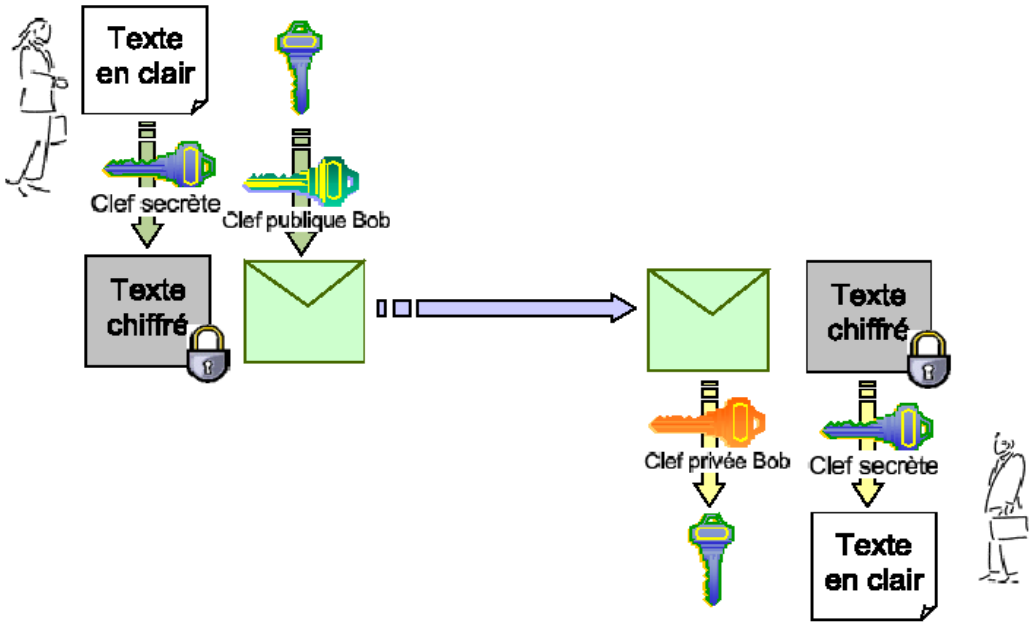
• Signature : clé privée utilisée pour le chiffrement, seul son détenteur peut chiffrer mais tout le monde peut déchiffrer (et donc vérifier la signature)



(Source : G. Labouret – © 1999-2001, Hervé Schauer Consultants)

Chiffrement asymétrique (4)

• Transport de la clé de session



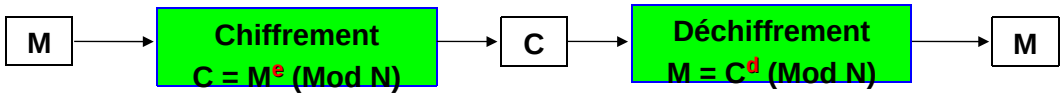
(Source : G. Labouret – © 1999-2001, Hervé Schauer Consultants)

Algorithmes à clef publique

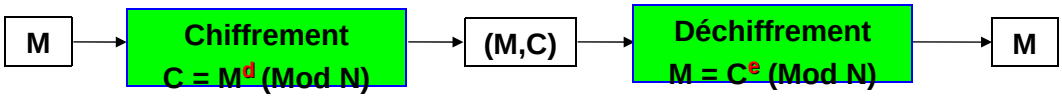
- Algorithme RSA (Rivest, Shamir et Adelman, 1977)
  - Fondé sur la difficulté de la factorisation des grands nombres
  - Sécurité dite "calculatoire": système inconditionnellement sûr
- Schéma
  - On choisit p, q deux « grands » nombres premiers
  - On calcule **n=p.q** et **n'=(p-1)(q-1)**
  - On choisit un entier **d** premier avec **n'** (pgcd(d,**n'**)=1) et **d < n'**
  - On choisit un entier **e** tel que **e.d = k(p-1)(q-1)+1;** c'est l'algorithme d'Euclide étendu qui permet de calculer **e**.
  - le couple d'entier **(n, e)** représente **la clé publique**
  - L'entier **d** représente la **clé privée**

RSA
(Rivest, Shamir et Adelman)

Chiffrement RSA



Signature RSA



- Tout le monde peut vérifier la signature par possession de la clé publique
- Service de non répudiation

Validité d'un chiffrage (1)

• **Pour les clés secrètes**, la référence est la recherche exhaustive (dépend de la longueur de la clé, 2^{n-1} opérations) :

- Complexité en temps : $2^{128} = 10^{40}$ opérations.
- Moyens : Un million de stations travaillant à un milliard d'opérations par seconde (10^{15}).
- Temps requis : Avec 10^{15} opérations par seconde, il faut 10^{25} secondes, c'est-à-dire de l'ordre d'un milliard de milliards d'années.

Validité d'un chiffrage (2)

• **Pour les clés publiques**, l'attaquant doit résoudre le problème mathématique sur lequel repose l'algorithme :

- Le système RSA repose sur le fait qu'à l'heure actuelle il est pratiquement impossible de retrouver dans un temps raisonnable **p** et **q** à partir de **n** si celui-ci est très grand (> 100 chiffres) ;
- **Le problème mathématique repose sur la difficulté de factoriser des grands nombres premiers ;**
- De plus il est impossible de retrouver (en un temps raisonnable) **d** à partir de **e**.
  - Pour faire ce calcul on a besoin de connaître n'=(p-1)(q-1) ce qui est très difficile si on ne connaît ni **p** ni **q**.
  - Les entiers **p** et **q** ne sont jamais transmis, ce qui empêche leur piratage.

Validité d'un chiffrage (3)

- Exemple : **factoriser n (RSALabs propose de gagner $200.000).**

n=25195908475657893494027183240048398571429282126204032027771378360436
62020707595556264018525880784406918290641249515082189298559149176184502
80848912007284499268739280728777673597141834727026189637501497182469116
50776133798590957000973304597488084284017974291006424586918171951187461
21515172654632282216869987549182422433637259085141865462043576798423387
18477444792073993423658482382428119816381501067481045166037730605620161
96762561338441436038339044149526344321901146575444541784240209246165157
23350778707749817125772467962926386356373289912154831438167899885040445
36402352738195137863656439121201039712282212072035 7

- **Risque de prendre plusieurs décennies/siècles**

Validité d'un chiffrage (4)

Pour RSA

En 1999, le record de factorisation est obtenu pour un nombre de **512 bits** (155 chiffres):

- **Trois cents ordinateurs** (divers stations de travail et PCs) ont travaillé à ce résultat cumulant un total de calcul évalué à 8 000 Mips
  - ➢1 Mips = un million d'instructions par seconde.

- Le déroulement des opérations s'est étalé sur une période de trois mois et demi.

Chiffrement asymétrique – Principaux usages

- Service de confidentialité :
  - Chiffrement avec la clé publique du destinataire.
  - Faible volume de données
- Signature numérique :
  - Chiffrement avec la clé privée du signataire
- Transport (RSA) de clés secrètes:
  - Résolution de la problématique de l'échange de clé secrète

Fonctions de hachage, signature et scellement

Mécanismes fournissant les services d'intégrité, d'authentification de l'origine des données et de non-répudiation de la source.

Intégrité et authentification

- Service souhaité: pouvoir s'assurer que le message:
  - émane bien de l'expéditeur annoncé (authentification de l'origine),
  - n'a pas été modifié pendant le transfert (intégrité)

- Un seul et même mécanisme:
  - Nécessité de s'assurer en plus que seul l'expéditeur est capable de calculer l'empreinte.
  - Sans mécanisme garantissant l'intégrité des données authentifiées, un intrus peut modifier le message puis recalculer l'empreinte, et faire accepter comme authentifiées des données qu'il a choisies.

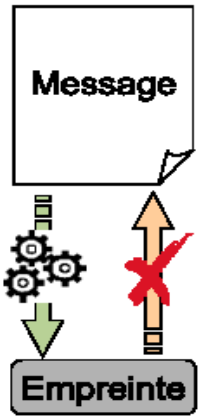
- **Authenticité** = authentification + intégrité

- Deux mécanismes : scellement et signature

Fonction de hachage (1)

- Fonction de hachage :
  - convertit une chaine de longueur quelconque en une chaine de taille inférieure et généralement fixe = **digest** (empreinte ou sceau)

- Critères de sécurité :
  - à sens unique : facile à calculer mais difficile à inverser pour reproduire l'original.
  - sans collisions : il est difficile de trouver deux messages ayant la même empreinte.



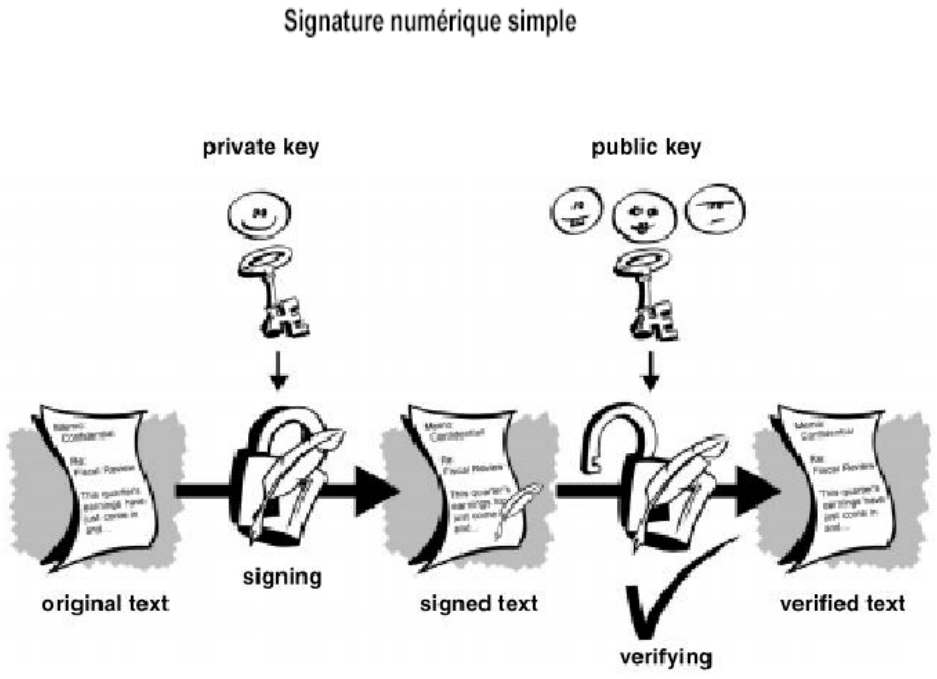
Fonction de hachage (2)

Principaux algorithmes :

- MD5 (Message Digest 5)
  - http://www.ietf.org/rfc/rfc1312.txt
  - empreinte de 128 bits
  - **openssl md5 -c message.txt**
- SHA-1 (Secure Hash Algorithm, 1994, NIST)
  - http://www.itl.nist.gov/fipspubs/fip180-1.htm
  - empreinte de 160 bits
  - **openssl sha1 -c message.txt**

➢ SHA-2 (2000) agrandit la taille de l'empreinte

Signature numérique



(Source : Une Introduction à la Cryptographie, NAI)

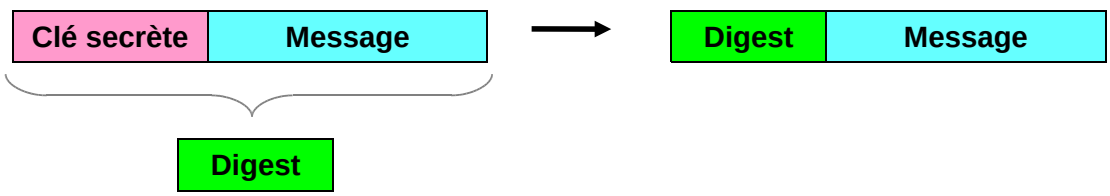
Signature numérique d'un message

- La combinaison d'un système de cryptographie avec une fonction de hachage permet à la fois :
  - de garantir l'intégrité du message;
  - Son authentification (**MAC**, **Message Authentication Code**)

- Deux types de signature, selon que le système de cryptographie est symétrique ou asymétrique :
  - Signature numérique symétrique (**scellement**);
  - Signature numérique asymétrique.

Signature numérique symétrique (1)

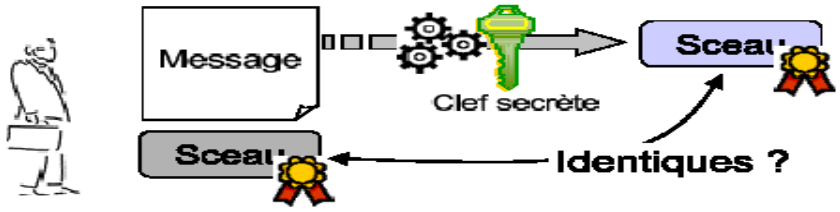
- L'émetteur calcule le digest sur la concaténation de la clé secrète et du message.
- Le destinataire procède de même :
  - Si digest trouvé est identique à celui reçu, alors :
    - d'une part le message n'a pas été altéré,
    - d'autre part qu'il a bien été émis par l'autre détenteur de la clé partagé.



Signature numérique symétrique (2)

- Scellement :

- Vérification :



(Source : G. Labouret – © 1999-2001, Hervé Schauer Consultants)

Signature numérique symétrique (3)

- Mécanisme qui fournit les services suivants :
  - Authentification de l'origine des données
  - Intégrité
  - ne fournit pas la non-répudiation

- Sceau ou code d'authentification de message (MAC)
  - fonction de hachage à sens unique à clé secrète :
    - dépend à la fois des données et de la clé
    - n'est calculable que par les personnes connaissant la clé

- Algorithmes :
  - Keyed-MAC (Keyed-MD5, Keyed-SHA1) :
    - valeurs pour MAC du type H(secret, message), H(message, secret), H(secret, message, secret)
  - HMAC (HMAC-MD5, HMAC-SHA-1) :

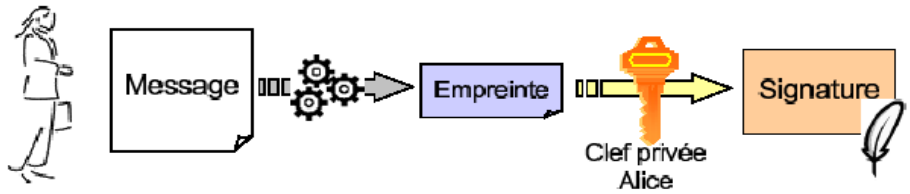
Signature asymétrique (1)

- Le digest est calculé sur le message puis chiffré à l'aide de la clé privée de l'émetteur, le résultat est joint au message envoyé

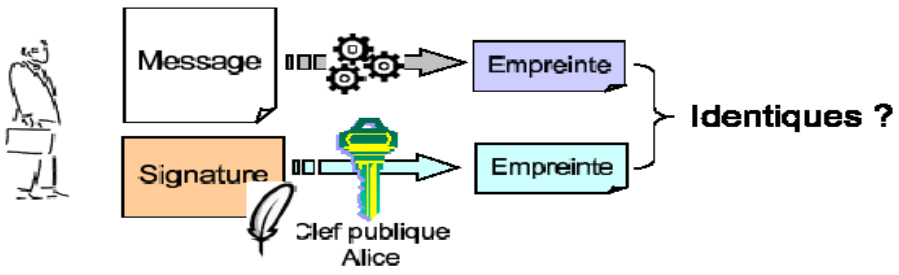
- Le destinataire calcule le digest sur le message, déchiffre le digest reçu à l'aide de la clé publique de l'émetteur :
  - Si digest trouvé est identique à celui déchiffré, alors :
    - d'une part le message n'a pas été altéré,
    - d'autre part l'émetteur est identifié, c'est le possesseur de la clé publique.

Signature asymétrique (2)

- Signature :

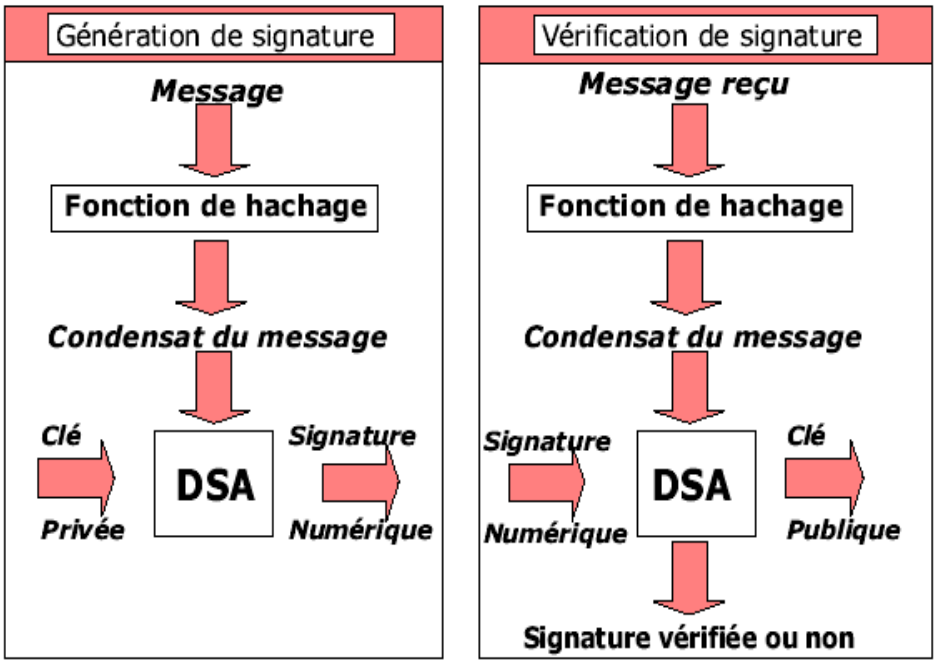


- Vérification :



(Source : G. Labouret – © 1999-2001, Hervé Schauer Consultants)

Signature asymétrique (3)



(Source : **La cryptographie moderne, transparents de Michel Van Caneghem**)

Signature asymétrique (4)

- Mécanisme qui fournit les services suivants :
  - Authentification de l'origine des données
  - Intégrité
  - Non-répudiation de la source

- Algorithmes
  - RSA :
    - **MD5 + RSA**
    - **SHA-1 + RSA**
  - DSS : Standard proposé par le NIST pour les signatures numériques en utilisant DSA (**DSA**: **Digital Signature Algorithm, algorithme de signature à clé publique**).

Fonction de hachage

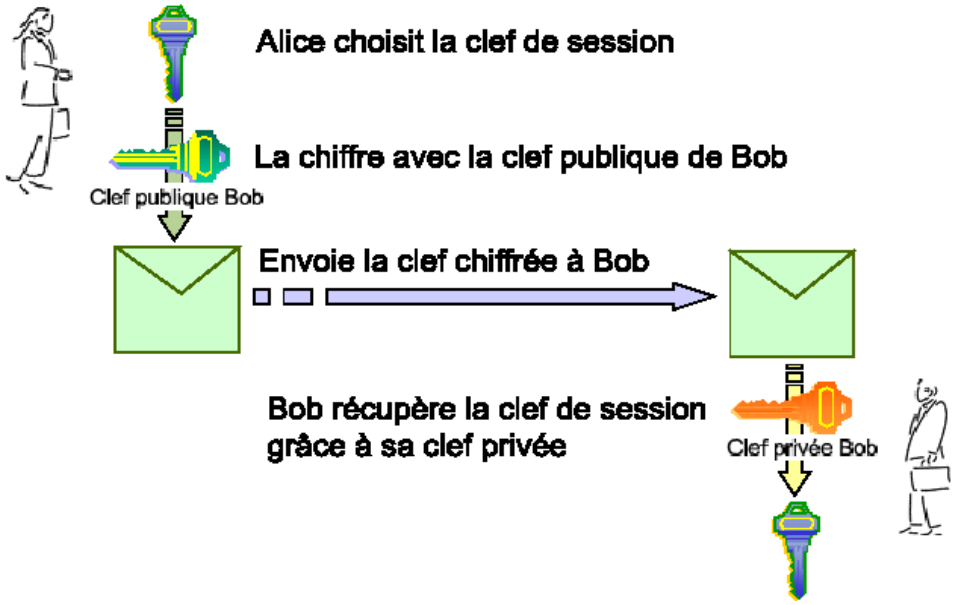
Principaux usages

- Signature numérique
  - Norme de fait : RSA
  - Mécanisme de chiffrement par une clef privée RSA d'une empreinte MD5 ou SHA-1
  - Service de non-répudiation
- Scellement
  - Génération d'un sceau, ou code d'authentification de message (MAC)
  - fonction de hachage à sens unique indexée par une clef secrète
  - Service d'authenticité des données
- IPSec utilise les deux systèmes :
  - Les correspondants s'authentifient d'abord par signature numérique asymétrique;
  - Les échanges suivants sont authentifiés par signature symétrique.

Problématique de l'échange de clés

- Echange de clés entre différents entités (cas symétrique)
  - prévoir un canal sécurisé à cet effet
- Relation entre échange de clés et authentification mutuelle :
  - L'échange de clés doit être authentifié pour éviter les attaques
  - Protocole d'authentification mutuelle avec échange de clés
    - fournit authentification mutuelle et un échange de clés authentifiés tout-en-un
- Deux protocoles d'échange de clés :
  - mode transport : exemple transport RSA (utilisé par SSL/TLS)
  - mode construction : exemple du protocole Diffie-Hellman

Transport de la clé de session



(Source : G. Labouret – © 1999-2001, Hervé Schauer Consultants)

Génération de la clé de session

- Permettre à deux tiers de générer un secret partagé (clé de session), utilisable pour le chiffrement , sans la faire transiter sur le réseau : **Principe du protocole Diffie-Hellman**

- Utilisation d'un schéma asymétrique (à clé publique) :
  - échanger de valeurs publiques;
  - générer un secret partagé à partir des valeurs publiques;
  - utiliser le secret généré pour dériver une ou plusieurs clés.

- Sa sécurité dépend de la difficulté de calculer des logarithmes discrets ($Y = g^x \bmod p$) :
  - retrouver x lorsque Y est donné est un problème difficile

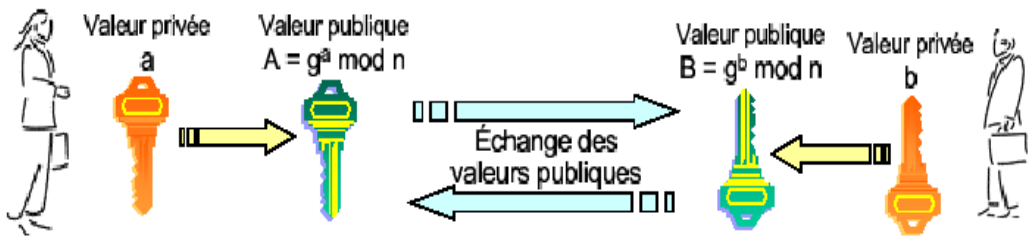
- Utilisé dans IPSec pour sécuriser l'échange de clés

Diffie-Hellman - Principe

- Alice et Bob se mettent d'accord sur un grand entier n tel que $(n-1)/2$ soit premier et sur un entier g primitif par rapport à n. Ces deux entiers sont publics.
- Alice choisit de manière aléatoire un grand nombre entier a, qu'elle garde secret, et calcule sa valeur publique, $A = g^a \bmod n$. Bob fait de même et génère b et $B = g^b \bmod n$.
- Alice envoie A à Bob ; Bob envoie B à Alice.
- Alice calcule $K_{AB} = B^a \bmod n$; Bob calcule $K_{BA} = A^b \bmod n$. $K_{AB} = K_{BA} = g^{ab} \bmod n$ est le secret partagé par Alice et Bob.
- Une personne qui écoute la communication connaît g, n, A et B, ce qui ne lui permet pas de calculer K_{AB} : il lui faudrait pour cela calculer le logarithme de A ou B pour retrouver a ou b.

Diffie-Hellman - Illustration

- Échange de valeurs publiques



- Permettant de génération du secret partagé

- Un espion ne peut reconstituer le secret partagé à partir des valeurs publiques

Diffie-Hellman - Propriétés

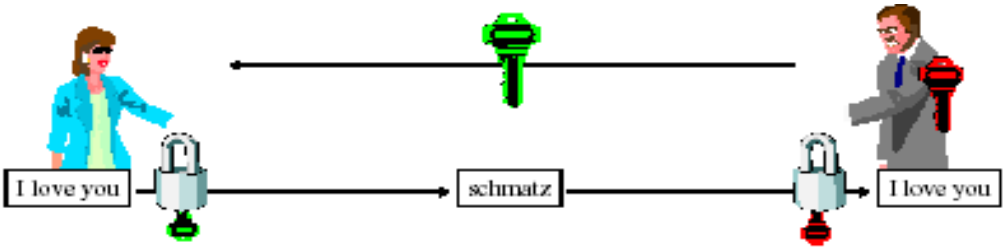
- Sensible à l'attaque de l'intercepteur :
  - l'attaquant, Ingrid, envoie sa valeur publique à la place d'Alice et Bob et partage ainsi un secret avec chaque tiers.
  - solution = authentifier les valeurs publiques utilisées :
    - authentifier les valeurs publiques à l'aide de certificats ;
    - authentifier les valeurs publiques après les avoir échangées, en les signant par exemple;

Certificats (1)

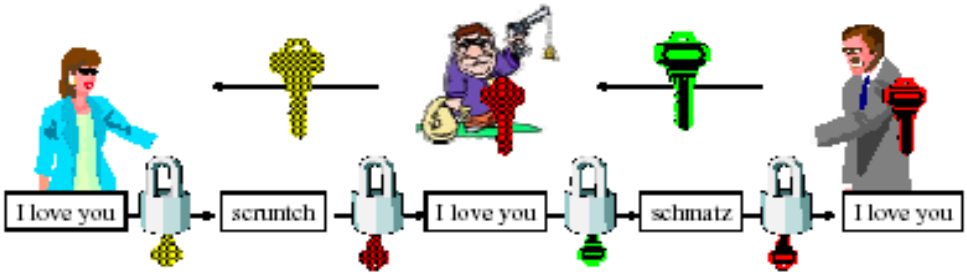
- Nécessite de pouvoir gérer des listes importantes de clés publiques dans les crypto-systèmes à clés publiques à grande échelle
- Avec les crypto-systèmes à clés publiques, la distribution de clés publiques est sujette à l'attaque d'un intermédiaire (Man in the Middle).
- S'assurer que la clé publique que vous vous apprêtez à utiliser pour chiffrer un message appartient bien au destinataire désiré (problème d'usurpation d'identité)
- Idée de départ : simple annuaire des clés publiques.

Certificats (2)

• Voilà ce qui devrait se passer lorsque Alice veut envoyer un message à Bob:



• Voilà ce qui peut se produire :(Man In the Middle)



(Source : G. Labouret – © 1999-2001, Hervé Schauer Consultants)

Certificats (3)

• Problèmes à résoudre :
  - Établir la réelle appartenance d'une clé publique à son propriétaire supposé : **authentifier la distribution des clés**.
  - Stocker les clés de façon sure : **protection en intégrité**.

• Solution = certificats et hiérarchies de certification :
  - Un certificat est un document qui sert à prouver qu'une clé appartient bien à qui de droit.
  - Le certificat est signé par un tiers de confiance (CA) dont on connaît la clé publique (**notez la récursion**)

Certificats (4)

- Un certificat contient au moins les informations suivantes:
  - Identité (Nom et adresse e-mail de la personne)
  - Clé publique
  - Date d'expiration
  - Signature du certificat

Certificats (5)

- Authentification :
  - Le certificat est signé avec la clé privée de l'autorité d'authentification.
  - L'objet de la signature numérique sur un certificat est de garantir que les informations de certification ont été contrôlées par une autre personne ou organisme.
  - La signature numérique ne garantit pas l'authenticité du certificat complet, elle garantit seulement que les informations d'identité ainsi signées correspondent bien à la clé publique à laquelle elles sont attachées.
- Intégrité : Toute modification du certificat sera détectée.

Certificats (6)

- Caractéristiques : doit verifier les propriétés suivantes :
  - Être propre à l'entité pour laquelle il a été créé;
  - Être infalsifiable;
  - indiquer l'usage de la clé publique qu'il contient (chiffrement, signature…)
- Composition d'un certificat : norme X509v3 [RFC 2459] :
  - Tbs Certificate (To be Signed certificate) ;
  - Signature Algorithm :
    - algorithme de signature utilisé pour la signature du certificat par CA
  - Signature value
    - valeur de cette signature

Certificats (7)

- Tbs certificate contient les champs suivants :
  - **Version :** indique à quelle version de X509 correspond ce certificat
  - **Numéro de série :** Numéro de série du certificat
  - **Algorithme de signature:** identifiant le type de signature utilisée
  - **Émetteur :** Distinguished Name (DN) de l'autorité de certification qui a émis ce certificat.
  - **Valide à partir de:** la date de début de validité de certificat
  - **Valide jusqu'à :** la date de fin de validité de certificat
  - **Objet:** Distinguished Name (DN) de détenteur de la clef publique
  - **Clé publique :** infos sur la clef publique de ce certificat (algorithme utilisé et valeur de la clé publique)
  - **Contraintes de base :** extensions génériques optionnelles
  - **Utilisation de la clé :** l'objet d'utilisation de la clé

Exemple de Certificat

Certificate:
Data: ← **tBSCertificate**
Version: 1 (0x0)
Serial Number:
32:50:33:cf:50:d1:56:f3:5c:81:ad:65:5c:4f:c8:25
Signature Algorithm: md5WithRSAEncryption
Issuer: C=US, O=VeriSign, Inc., OU=Class 1 Public Primary Certification Authority
Validity
Not Before: Jan 29 00:00:00 1996 GMT
Not After : Jan 7 23:59:59 2020 GMT
Subject: C=US, O=VeriSign, Inc., OU=Class 1 Public Primary Certification Authority
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:e5:19:bf:6d:a3:56:61:2d:99:48:71:f6:67:de:
[...]
2a:2f:31:aa:ee:a3:67:da:db
Exponent: 65537 (0x10001)
Signature Algorithm: md2WithRSAEncryption ← **signatureAlgorithm**
4b:44:66:60:68:64:e4:98:1b:f3:b0:72:e6:95:89:7c:dd:7b:
[...] ← **signatureValue**
f8:45

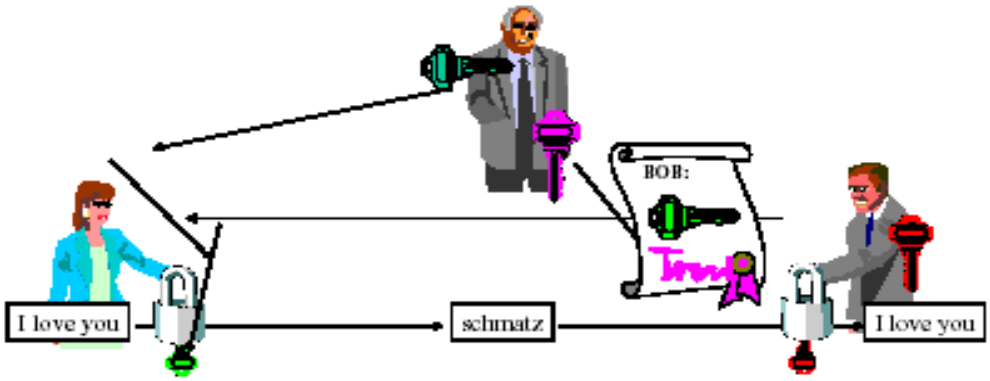
La signature du certificat

- S'effectue en deux étapes :
  - Un "message digest" est générée par hachage sur les données du certificat (Tbs Certificate) en utilisant l'algorithme SHA-1 ou MD5, le plus utilisé étant SHA-1.
  - Le haché (ou l'empreinte) est ensuite cryptée avec la clé privée de la CA qui génère le certificat.
- Vérification de la signature :
  - le destinataire, qui connaît la clé publique de signature de CA, l'utilise pour retrouver la valeur du haché du certificat. Cette opération permet de s'assurer de l'identité de l'expéditeur.
  - ensuite, le destinataire calcule le haché du certificat à partir de message reçu. Si les deux empreintes sont identiques, cela signifie que les données signées sont intègres et authentifiées.

Vérification des certificats

• Un tiers à signé un certificat liant la clé de Bob à son nom.

• Si Alice est en possession de la clé publique du tiers elle peut vérifier le certificat.



Les PKIs

• Une PKI est un Infrastructure à Clé Publique (Public Key Infrastructure)

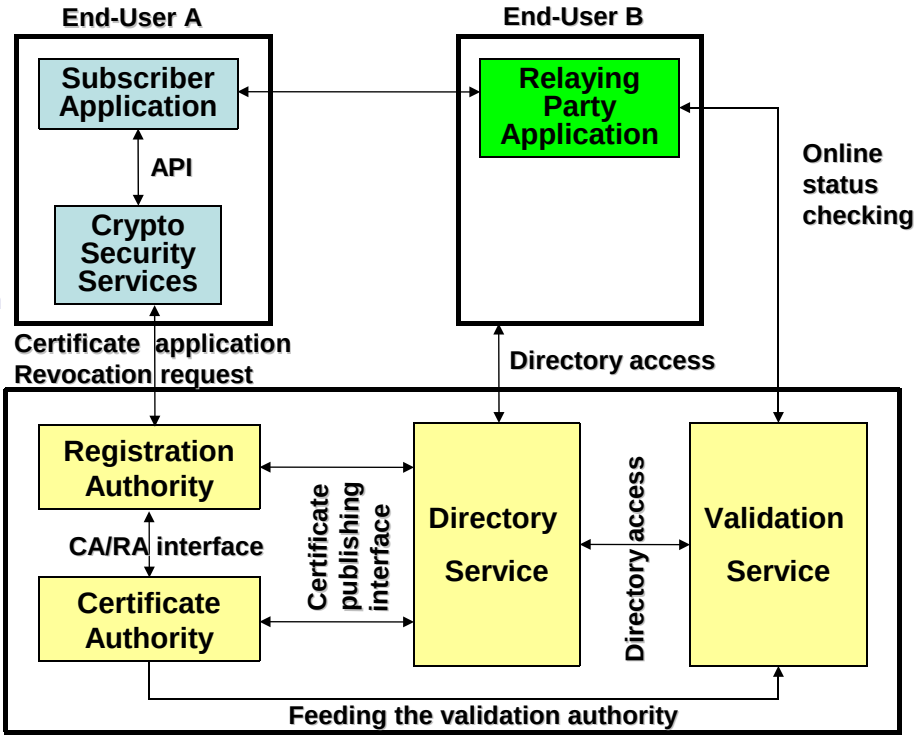
• C'est le dispositif nécessaire pour gérer la génération et la distribution contrôlée de certificats.

• Une PKI est faite des éléments suivants:
  - **Des autorités de certification (CA)**
  - **Des autorités d'enregistrement (RA)**
  - **Des annuaires de certificats**

• Une PKI fournit quatre services principaux:
  - fabrication de bi-clés.
  - certification de clé publique et publication de certificats.
  - Révocation de certificats.
  - Gestion la fonction de certification.

Composants d'une PKI

- **Utilisateurs finaux** (possesseurs des clés)
- **Autorité d'enregistrement** (RA, Local RA)
- **Autorité de certification** (Certificate Authority- CA)
- **Annuaire**
- **Service de validation**
- **Hiérarchie : plusieurs niveaux de certification**



L'autorité de certification (CA)

- La CA crée et signe les certificats :
  - authentifie physiquement le participant,
  - fait générer une paire de clés publique/privée par le participant,
  - crée un certificat avec l'identité du participant, sa clé publique, une date d'expiration et la signature de la CA,
  - fournit une copie de sa propre clé publique au participant.

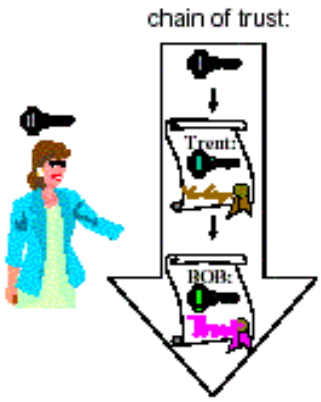
- Muni de son certificat et de la clé publique de la CA, le nouveau participant peut communiquer avec tous les autres participants certifiés par la même CA.

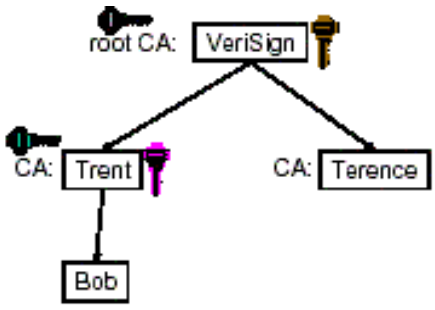
L'autorité de certification (CA)

- Une CA peut faire certifier sa clé publique par une autre CA :
  - **Trent peut faire certifier sa clé publique par VeriSign**
  - **Si Alice fait confiance à VeriSign, elle peut :**
    - ✓ vérifier le certificat de Trent avec la clé publique de VeriSign,
    - ✓ vérifier la certificat de Bob avec la clé publique de Trent.

Hiérarchie de CA

- CA Hiérarchiques : chaîne de confiance





L’autorité d’enregistrement

• La CA peut déléguer l'enregistrement de nouveaux participants à des Autorités d'enregistrement (Registration Authorithy).

• La RA ne possède pas la clé privée de la CA.

• La CA lui fait confiance pour l'authentification physique des participants.

• Après avoir authentifié le nouveau participant, la RA lui fait générer une paire de clé et envoie la clé publique à la CA pour création du certificat.

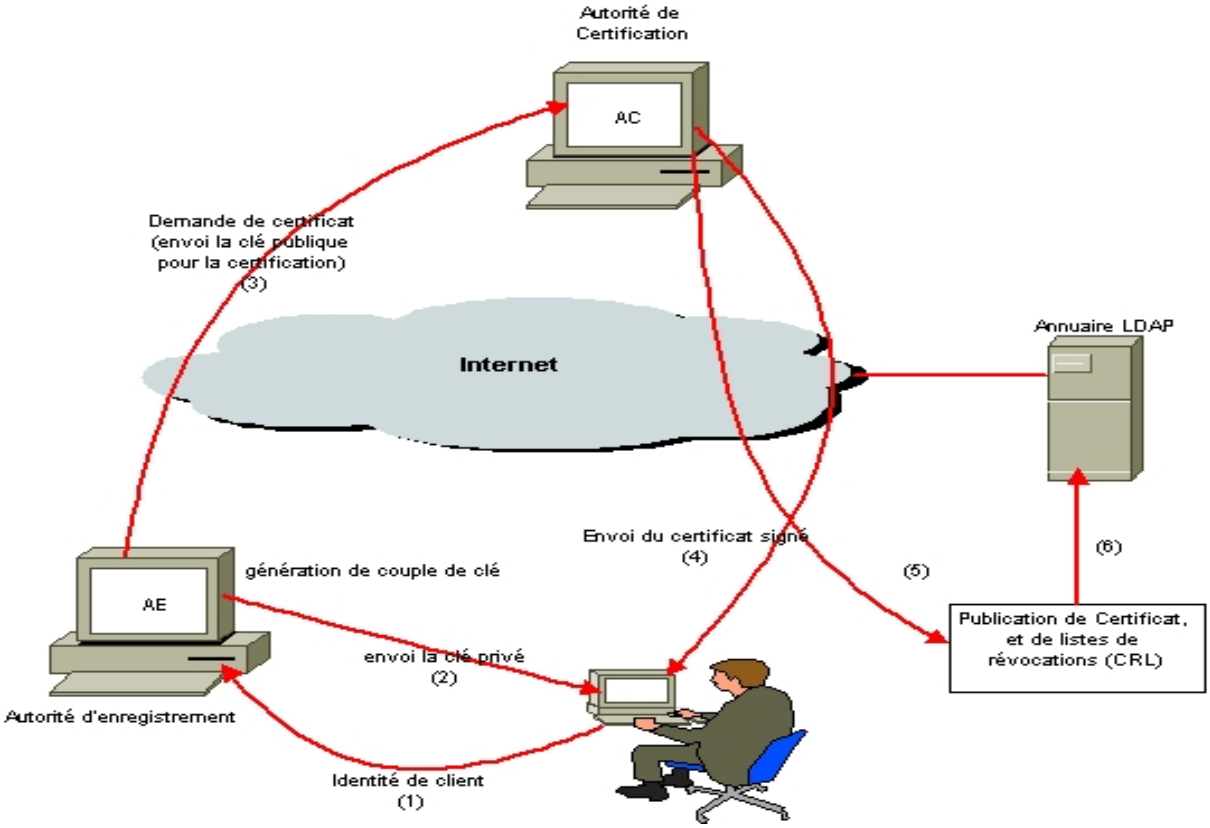
L’annuaire de certificats

• Pour faciliter l'accès aux certificats, la CA met à disposition un annuaire (LDAP)

• L'annuaire fournit aussi la liste des certificat révoqués (Certificate Revokation List, CRL)

• Un certificat peut être révoqué avant son échéance pour cause de vol, de perte ou de changement de statut (Alice quitte son employeur)

Cycle de vie d'un certificat



7

Exemple de crypto-systèmes : SSL/TLS

- **SSL** (Secure Socket Layer) est un protocole de sécurisation des échanges au niveau de la couche application. Il est implémenté au dessus de la couche TCP.
- **SSL permet d'assurer les services de sécurité suivants:**
  - confidentialité : assurée par **les algorithmes à chiffrement symétrique de blocs** comme DES, ou 3DES
  - intégrité : assurée par l'utilisation de MAC (Message Authentication Code) basés sur **les fonctions de hachage** MD5 (16 octets) ou SHA-1 (20 octets).
  - authentification : permet l'authentification des 2 entités (authentification client facultative) **basé sur des certificats**, et l'authentification des données grâce aux MAC.

Quelques Liens

- Network Security with OpenSSL (O'REILLY Éditions)
- Sécurité et Internet (transparents Didier Donsez)
- La cryptographie moderne, transparents de Michel Van Caneghem
- Une introduction à la cryptographie (Network Associates Inc.)
- Introduction à la cryptographie (www.hsc.fr)
- http://www.apprendre-en-ligne.net/crypto/activites/index.html
- Chapitre de Livre : la sécurité des systèmes d'information
- Cryptographie – Généralités (Jean Berstel, Univ. Marne-la-Vallée)
- Site: http://www.apprendre-en-ligne.net/crypto/vigenere/index.html