Introduction à la sécurité des réseaux

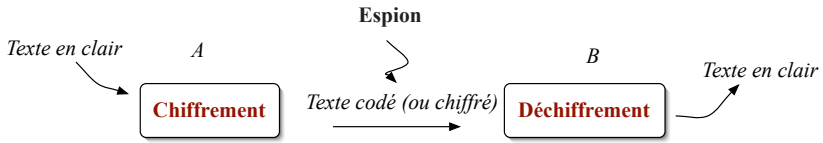
- **Deux grandes classes de problèmes de sécurité**
  - **Chiffrement permettant de garantir**
    - » La confidentialité des messages
    - » L'authentification des interlocuteurs
    - » L'intégrité et la non répudiation des messages
  - **Contrôle d'accès aux ressources (sécurité des communications)**
    - » Interdiction de l'accès à certaines ressources : applications, machines, réseaux
    - » Solution les pare-feux (firewall) et les proxys applicatifs

Introduction à la sécurité des réseaux

- **La confidentialité :**
  - Il faut garantir aux interlocuteurs qu'ils sont les seuls à pouvoir comprendre les messages qu'ils s'échangent
  - Technique de chiffrement (Cryptologie)
- **L'authentification :**
  - Il faut garantir aux interlocuteurs leur identité respective
  - Carte d'identité + signature
  - Dans le monde du numérique cela peut se faire aussi à l'aide du chiffrement: signature électronique
- **L'intégrité et la non répudiation des messages:**
  - Il faut garantir aux interlocuteurs
    - » que les messages ne peuvent pas subir de modification par un tiers et proviennent bien de la personne que l'on croit (authentifiée)
  - De plus il faut qu'un émetteur ne puisse pas nier avoir envoyé un message, ni le récepteur l'avoir reçu

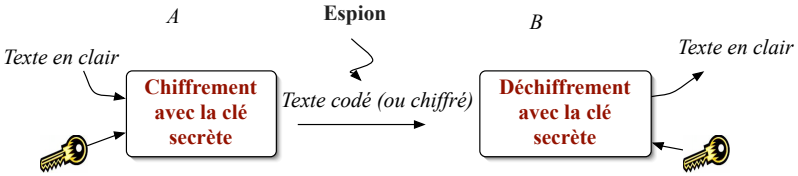
1- Principe du chiffrement: la cryptographie

- Comment s'échanger des données tel que seuls les interlocuteurs peuvent comprendre



- Remonte à des temps éloignés; Exemple du Chiffrement de J. César
  - Codage : décalage de 3 lettres dans l'alphabet En clair :"salut", chiffré :"vdoxw"
  - Besoin d'une clé connue par les interlocuteurs (ici nombre de décalage) et d'un algorithme de chiffrement (ici décalage dans l'alphabet)
  - On parle d'algorithme à **clé symétrique**

Principe du chiffrement



- Il est nécessaire que les algorithmes de chiffrement soient connus des interlocuteurs (qui ne se connaissent pas forcément)
- Normalisation des algorithmes de chiffrement
- Ils sont donc connus des espions

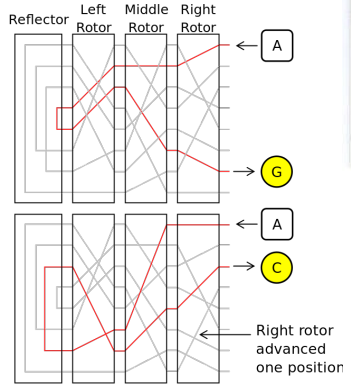
Robustesse d'un algorithme de chiffrement

- Exemple de César: seulement 25 clés différentes, facile à trouver
- On peut compliquer l'algorithme:
  - Chiffrement monoalphabétique : 1 lettre -> 1 lettre mais quelconque
    - » De l'ordre de 26 ! ± 4.10^{26} combinaisons possibles
    - » En fait pas très difficile à déchiffrer par une approche par statistique de l'apparition des lettres ou des combinaisons de lettres
  - Chiffrement polyalphabétique: On peut compliquer en chiffrant successivement les caractères avec différents chiffrements monoalphabétique.
    - » Exemple: C1 pour le 1er, 4ème, 7e..., C2 pour le 2ème, 5ème, 8ème..., C3 pour le 3ème, 6ème, 9ème

Un exemple: Enigma (1920)

- Trois rouleaux de 26 positions
- Un câblage manuel d'échange de lettres
- Changement de la position des rouleaux après chaque lettre
- Chiffrement et déchiffrement avec la même machine

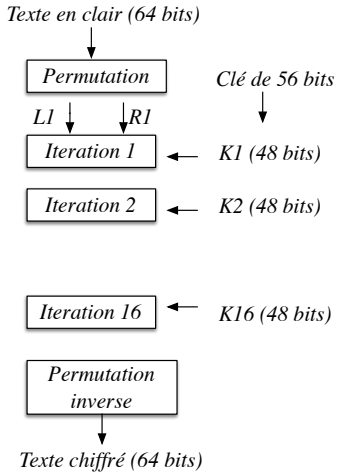
- Clé:
  - positions initiales des trois rouleaux
  - Câblage manuel de paire de lettre
- ± 10^{20} combinaisons



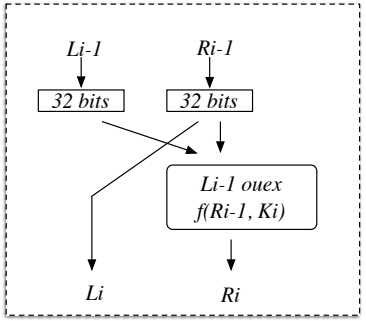
Méthodes modernes de chiffrement à clé symétrique

- DES (Data Encryption Standard) 1977-1993
- Basé sur le chiffrement de bloc de 64 bits à partir d'une clé de 64 bits (56 bits + 8 bits de parité)
- 16 clés sont calculées à partir de la clé de 56 bits (permutations successives)
- Calculs identiques et simples (permutations et ou exclusif) sur 16 étages
- Opérations inverses pour le déchiffrement

Principe du DES



Une itération



Méthodes modernes de chiffrement à clé symétrique

•DES reste un chiffrement monoalphabétique

•Concours en 1997 (10 000 dollars de récompense):

–cassage à partir d'une courte phrase

–En moins de 4 mois

–Essais de 10^{24} clés (soit un quart de l'ensemble des possibilités)

•Concours en 1999:

–à l'aide d'un super-ordinateur et 100 000 ordinateurs connectés

–En 22h et 15 minutes

Amélioration du DES

- Triple DES (1999) (3 clés et trois fonctions de calculs différentes)
- AES (Advanced Encryption Standard) 2001
  - Appelé aussi Rijndael du nom de ses auteurs (V. Rijmen et J. Daemen)
  - Bloc de 128 bits et clés de 128, 192 ou 256 bits
  - Un ordinateur pouvant casser une clé DES en 1 seconde mettrait 149 trillions (10^{18}) d'années pour casser une clé AES de 128 bits
- Problème: coûteux en calcul
- L'Iphone intègre un chiffrement AES 256 bits de ses données à l'aide d'une clé cachée dans le hardware (co processeur de chiffrement intégré)

Autres chiffrements connus

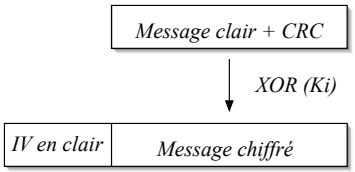
- RC4 (1987) :
  - Génération d'un flot de clés (Ki) à l'aide d'un vecteur d'initialisation (IV) et d'une clé secrète (K)
  - K1= chiffrement (IV, K) puis Ki+1= chiffrement (Ki, K)
  - Chiffrement simple à base de permutation, addition…
  - Chaque octet O de la trame est ensuite chiffré par O XOR Ki
  - Le vecteur d'initialisation change pour chaque paquet à chiffrer (tirage aléatoire) et est envoyé en clair pour permettre au récepteur de déchiffrer
  - Permet de limiter les calculs de chiffrement
  - RC4 réputé sûr si on ne réutilise pas le même vecteur d'initialisation lors d'un échange

Chiffrement pour réseaux sans fils

- WEP (Wired Equivalent privacy): Basé sur RC4, peut être facilement casser en quelques secondes aujourd'hui (voir plus loin)
- WPA (Wifi Protected Access) puis
  - TKIP (Temporel Key Integrity Protocol): nouvelle clé pour chaque paquet, même principe que le WEP mais avec un vecteur d'initialisation chiffré
  - WPA2 depuis 2004
    - » Chiffrement CCMP (Couter Mode CBC - Mac Protocol): basé sur AES réputé le plus sûr aujourd'hui

Exemple de chiffrement à clé symétrique : le WEP

- Principe basé sur RC4:
  - Une clé symétrique de 40 bits (K)
  - Pour chaque trame: l'émetteur fabrique une clé de 64 bits à partir de cette clé commune et de 24 bits tirés aléatoirement (appelés vecteurs d'initialisation IV).
  - Un flux de clés (Ki) sont générés à partir de K et IV à l'aide du chiffrement RC4



Exemple de chiffrement à clé symétrique : le WEP

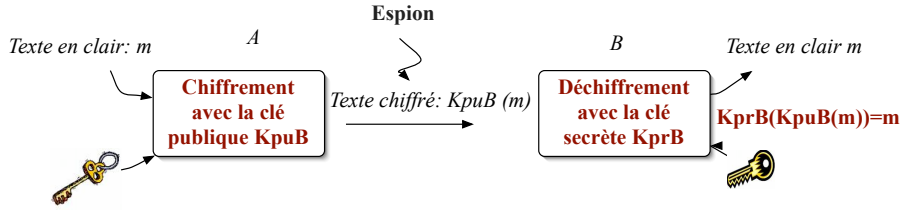
- Exemple d'Attaque:
  - 24 bits pour les IV est insuffisant, quelques milliers de trame envoyées successivement ont de fortes chances de réutiliser le même IV, donc le même flux de clé Ki
  - si on émet une requête permettant d'avoir une réponse connue Ri (par exemple ARP request), on peut récupérer Ci (la réponse chiffrée) et on peut calculer le flot de clé Ki:
    - »en effet : Ri xor Ki= Ci donc Ki= Ci xor Ri (xor est associatif et a xor a= 0 et a xor 0 = a)
  - A la prochaine utilisation de ce IV, on pourra déchiffrer le message
- Il existe des outils sur le WEB permettant de casser une clé WEP (retrouver la clé originale K) en quelques minutes voir quelques secondes (voir l'outil *aircrack-ng*)
- WPA plus sûr

Le chiffrement à l'aide de clé publique

- Le problème du chiffrement à clé symétrique est qu'il faut s'échanger la clé
- Si ce n'est pas possible via un autre moyen que le réseau. L'échange peut être espionné. Par exemple pour des échanges entre des entités qui ne se connaissent pas
- Invention des échanges basés sur des clés publiques par Diffie et Hellman en 1976

Principe du chiffrement à clé publique

- Les interlocuteurs possèdent un couple de clés, une privée (***Kpr***) et une publique (***Kpu***)
- La clé publique peut être connue de tous et donc envoyée en clair à tout interlocuteur potentiel
- Soit ***Kpu***(m) le message ***m*** chiffré à l'aide de la clé ***Kpu***
- Il existe des algorithmes de chiffrement tel que ***Kpr*** (***Kpu*** (***m***))=***m***



Principe du chiffrement à clé publique

- L'algorithme RSA (Ron Rivest, Adi Shamir, Leonard Adleman)
- Le calcul des clés secrètes et publiques:
  - Choix de 2 nombres premiers p et q
  - Calculs de **n=p*q** et **n1= (p-1)*(q-1)**
  - Choisir un nombre **e** tel que :
    - » p, q < **e** < n1 et ***pgcd*** **(n1, e)=1**
  - Trouvez un nombre **d** tel que
    - » p, q < **d** < n1 et **e*d modulo n1 = 1 (ou d= 1/ e modulo n1)**
  - La clé publique est égal à (**n, e**) et la clé privée est égal à (**n, d**)

Principe du chiffrement à clé publique

- **Chiffrement**:
  - soit **m** le texte en clair (un entier < n),
  - soit c le texte chiffré: **c= m^e modulo n**
  - Dans la pratique on découpe le message en bloc de taille inférieur à n
- **Déchiffrement**:
  - On peut montrer que **m^{e*d} modulo n** est forcement égal à **m**
  - Donc **m= c^d modulo n**
- Pour trouver la clé privée **(d, n)** en connaissant la clé publique **(e, n)** il faut trouver **p** et **q** tel **n= p*q**
- Si n est grand, c'est très long car le coût de la factorisation est exponentielle en fonction de n

Utilisation du chiffrement à clé publique

- Pour l'instant la factorisation n'est pas possible dans un temps raisonnable avec n ≥ 1024.
- Longueur des clés conseillée aujourd'hui > 2048 bits
- 2010, un clé RSA de 768 bits a été cassée : un entier n de 232 chiffres décimaux a été factorisé (travail de 500 PC pendant 1 an)
- Le RSA est beaucoup plus coûteux (rapport de 100 à 1000 pour une implémentation logicielle)
- Utilisé par exemple pour l'authentification des cartes bancaires
- On l'utilise aussi souvent pour s'échanger des clés symétriques (on parle alors de clé de **session**). Puis on continue ensuite avec un algorithme à clé symétrique comme le DES pour du chiffrement «volumineux»

2-Intégrité des messages et signature

- Signature numérique permettant de prouver qu'un message provient d'une personne
  - Vérifiable
  - Impossible à contrefaire
  - Non répudiable
- Exemples:
  - Déclaration des impôts
  - Mails
  - Contrats

Intégrité des messages à l'aide de clés privée-publique

- Bob envoie un message et une signature qui n'est rien d'autre que le message chiffré avec sa clé privée $\mathbf{K_{BPr}(m)}$
- Alice peut prouver que ce message provient de Bob en appliquant la clé publique de Bob à la signature: $\mathbf{K_{BPu}(K_{BPr}(m))= m}$
  - Le message provient forcément de Bob qui est le seul détenteur de sa clé privée
  - Le message n'a pu être modifié après réception
  - Personne ne peut contrefaire cette signature sans connaître la clé privée de Bob

Intégrité des messages par résumé de message

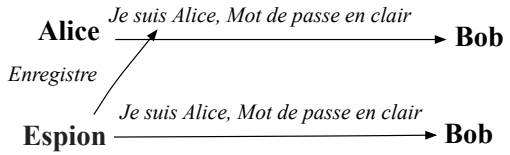
- Obtenir une signature moins coûteuse en calcul
- La signature (par chiffrement avec clé privée) est faite sur un **"*résumé*"** (ou ***empreinte***) du message (**H(m)**) beaucoup plus court que le message et de longueur fixe (fonction de hachage)
- Similaire à une somme de contrôle ou CRC
- Le message est envoyé avec la signature (m, Kpr(H(m)))
- Il faut que ce résumé H(m) soit tel que H(m) ≠ H(m') si m≠m' (sinon collision)
- Alice reçoit (m, Kpr(H(m))),
  - déchiffre Kpr(H(m)) obtient H(m) à l'aide de Kpu
  - Recalcule H(m-reçu) du message en clair
  - Vérifie que H(m-reçu)= H(m)

Intégrité des messages par résumé de message

- MD5 (Message Digest 5) :
  - Très répandu , basé sur des résumés de messages de 128 bits
  - Réputé non sûr (cassé par une équipe chinoise en 2004)
  - Aussi utilisé avec des clés symétriques
  - Par exemple dans RIPV2, stockage des mots de passe sous Linux...
- Algorithmes plus sûrs:
  - SHA-1 (Secure Hash Algorithm) basé sur des résumés de 160 bits
    - » première collision en 2017
  - Remplacé depuis 2010 par SHA-2 (SHA-256, SHA-512)

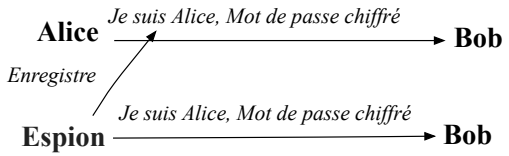
3- L'authentification

- Comment prouver à son interlocuteur que l'on est bien la personne que l'on prétend être
- Indispensable avant tout échange de donnée



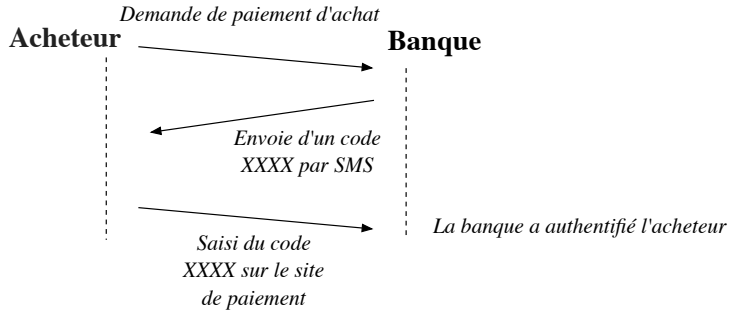
L'authentification

- Il faut chiffrer le mot de passe avec une clé symétrique
- Ne résout en rien l'authentification
- L'espion enregistre le mot de passe chiffré et peut le renvoyer sans connaître la clé symétrique



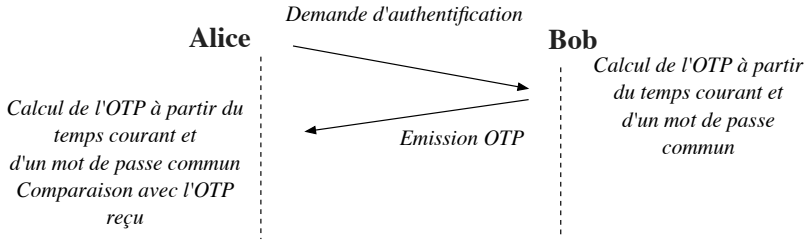
Authentification par OTP (One Time Password)

- La faille du protocole précédent vient du fait qu'il n'y a qu'un seul mot de passe
- Pour remédier à cela, il suffit changer de mot passe à chaque authentification
- Exemple authentification lors d'achat sur Internet par mot de passe via SMS:



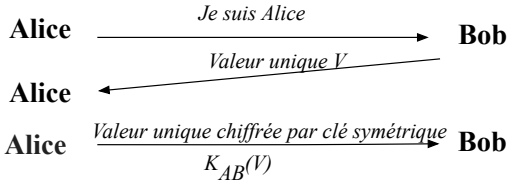
Authentification par OTP (One Time Password)

- Exemple d'OTP basé sur le temps:



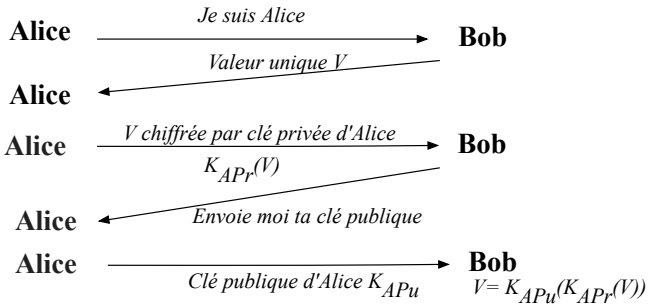
Authentification par clé symétrique

- Alice veut s'authentifier auprès de Bob
- Bob envoie une valeur unique à Alice à chaque authentification, Alice renvoie alors cette valeur chiffrée à Bob
- A priori Alice est la seule à connaitre la clé symétrique KAB



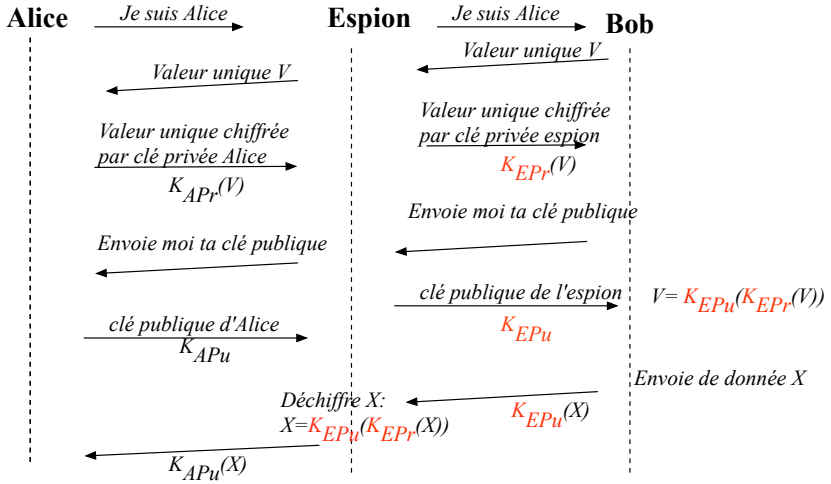
Authentification par clé publique/privée

- **La valeur unique est chiffrée grâce à la clé privée d'Alice (seule à la connaître)**



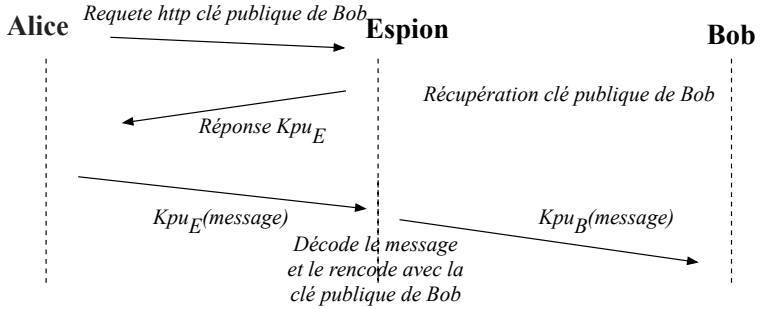
Authentification par clé publique

- **Mise en défaut du protocole: attaque par un intermédiaire ("homme au milieu", "man in the middle")**



Mise en défaut du chiffrement à clé publique

- L'attaque de "l'homme au milieu" est aussi possible pour la confidentialité des messages à l'aide de clé publique.
- Supposons que la clé publique de B soit accessible via son site WEB
- Supposons qu'un espion intercepte la requête *http* qui permet de récupérer la clé publique de Bob (K_B), il envoie en réponse sa propre clé publique K_E

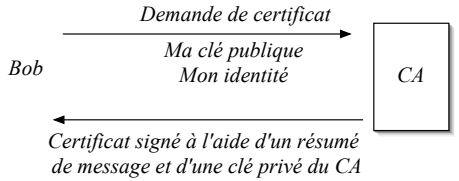


Solution pour se prémunir de l'attaque de l'intermédiaire

- Comment être sûr que la clé publique appartient à la personne à qui l'on s'adresse
- S'échanger les clés publiques par un canal de confiance (courrier ?, téléphone?, main propre...)
  - Du coup pas mieux que les clés symétriques
- S'adresser à un centre de certification qui peut nous garantir l'appartenance d'une clé publique à une personne

4- Certification

- Besoin d'assurer l'identité du possesseur d'une clé publique
- Centre de certification (CA)
- Délivre un certificat permettant de s'assurer de l'identité du possesseur d'une clé publique
- Un certificat mentionne la clé publique et l'identité
- Il ne peut être reproduit par un espion à cause de la signature du CA
- Bob peut demander son certificat signé par le CA (En général tierce personne pour vérifier la validité de la demande)

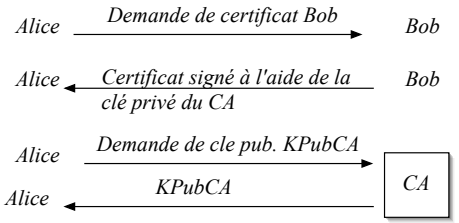


Exemple de certificat

- **Version de la norme X.509 utilisée**
- **Sicard Pascal**
- **Délivré par : CNRS standard expire le 14/9/8**
- **Algorithme de signature : MD5 avec chiffrement RSA**
- **Clé publique de l'utilisateur avec chiffrement RSA 256 octets:**
  - D0 A0 45 48 F8 B2 7C 05 AB D7 97 10 8B 28 0D 13 3E FD B4 79 2D E7 67 14 44 B2 18 04 E1 D3 7B C0 52 B1 74 D5 6A 6A 2B C2 8B 26 B3 A1 04 F0 E4 74 F7 F1 A1 C9 0A 0E 39 F7 D5 43 61 A9 92 C3 9F A4 53 AA 3E CC 8C A5 65 75 BF DD C8 A3 4C 09 99 4C 38 88 01 3A 80 75 DE C6 36 96 B6 82 50 B8 38 E1 CD 84 54 54 35 11 4D F8 22 97 B0 E4 66 40 7D BE 16 22 52 CB DB 59 CF 07 B0 94 E6 D2 3B 5A 17 71 40 B0 87 59 D4 50 54 D9 AB F1 54 D8 7E 71 4F 48 F0 96 4E 89 A6 20 71 3B E8 87 6B 0C 44 9C 57 B8 D8 10 4C 37 69 78 5A 16 D5 43 57 C0 3D EB B3 F8 E7 5C 41 1D B0 97 45 52 5C 2C C3 AE BB AC FC 16 91 04 58 47 D7 CF 21 DB AE AC 0C CF 08 76 BE 02 7D 45 2A D2 38 52 57 58 6A 51 41 4D A5 5D 93 55 5E 8D 39 0C 7D E3 07 4E 8E 91 29 B3 09 13 D6 A2 5B D2 AF 85 1D 90 3E E8 E0 84 F4 F2 76 2F 57 8D
- **....**
- **Signature:** F3 2A 1C 19 B1 FA A6 A7 4E AD 3C 69 A4 2A F4 9C

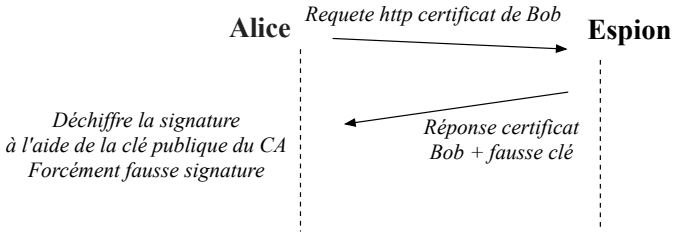
Sécurisation avec certificat

- Alice qui veut dialoguer avec Bob, lui demande son certificat (qui contient sa clé publique)
- Puis Alice demande la clé publique KPubCA du Centre de Certification
- Elle déchiffre la signature du certificat de Bob à l'aide de KPubCA
- Elle est sûre d'avoir la clé publique de Bob



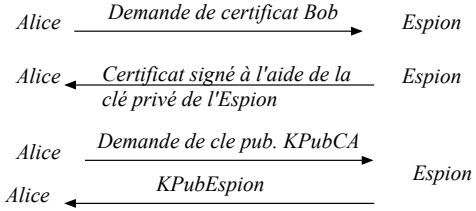
Attaque de l'intermédiaire en cas de certificat

- **Plus possible car l'espion ne peut fabriquer un certificat portant le nom de Bob et sa propre clé publique : impossible de fabriquée une fausse signature**



Attaque de l'intermédiaire en cas de certificat

- **Une faille persiste : la récupération de la clé publique du CA par Alice**



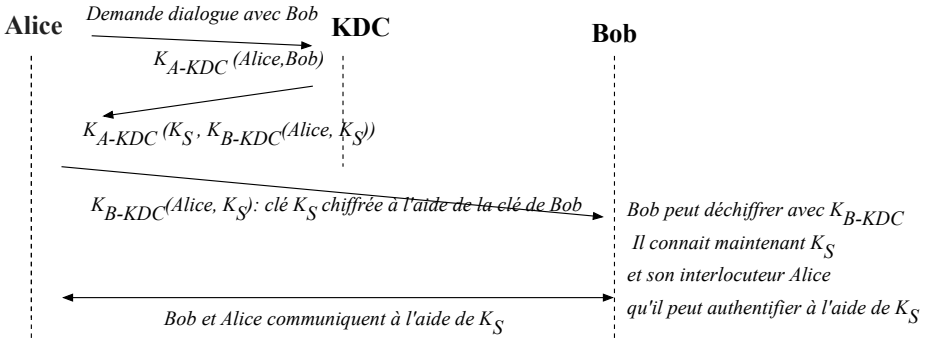
- **Problème récurent**
- **Solutions**
  - Les clés publiques des CA sont connues universellement et déjà dans les systèmes des machines (navigateurs...)
  - Soit chaîne de certificats (arrivant au CA voulu) et Clés des CA racine ("ancres de confiance") connues des machines
  - Annuaire normalisé : LDAP (Light Directory Access Protocol) contenant les certificats des personnes

Infrastructure de gestion de clé

- PKI (Public Key Infrastructure) ou IGC (Infrastructure de Gestion de Clé)
- Permet de délivrer des certificats signés de façon « à priori » sûre.
- Normes internationales définissent
  - Autorités d'enregistrement:
    - » Vérifie l'identité du demandeur de certificat (carte d'identité)
    - » Génère un couple clé privée/publique pour le demandeur
    - » Délivre au demandeur sa clé privée
    - » Envoie la demande à l'autorité de certification
  - Autorité de certification
    - » Crée les certificats à l'aide de sa clé privée (celle de l'autorité)
    - » Publie sa clé publique (par défaut par exemple dans les navigateurs)
    - » Envoie au service de publication les nouveaux certificats
  - Service de publication
    - » Publie la liste des certificats soit sur serveur WEB, soit par annuaire LDAP

5- Distribution de clés

- Permet de s'échanger de façon sûre une clé symétrique K_S
- Centre de distribution de clés (KDC)
- Bob et Alice s'inscrivent au KDC et possède donc une clé symétrique avec ce centre (notées $K_{A\text{-}KDC}$ et $K_{B\text{-}KDC}$)

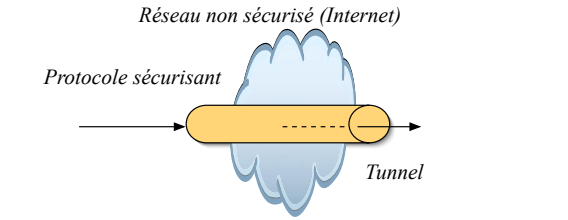


EAP : Extensible Authentification Protocol

- Mécanisme d'authentification universel
- Différentes versions « normalisées » avec des méthodes de chiffrements différentes et conçus pour différents niveaux de sécurisation
  - LEAP: version propriétaire de Cisco
  - EAP-TLS (Transport Layer Security): utilisé sur les réseaux sans fil (WPA) à base d'infrastructure à clés publiques
  - EAP-MD5
  - EAP-SIM utilisé par Free
  - EAP TTLS (Tunneled Transport Layer Security)
  - ….

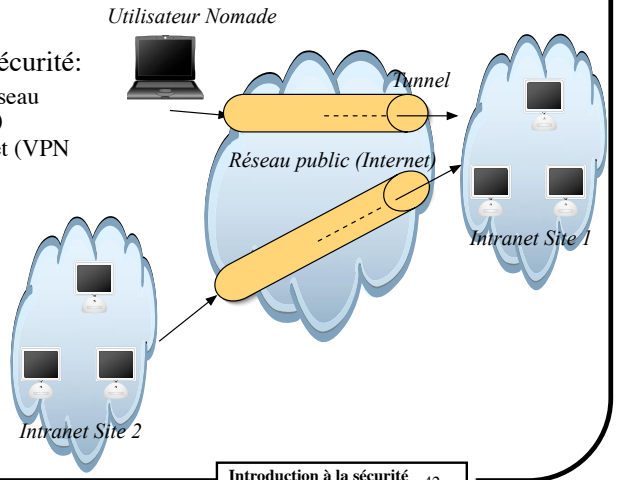
La mise en pratique des accès et échanges sécurisés

- VPN: Virtual Private Network
- La technique du tunneling
- Un protocole supplémentaire (appelé de Tunneling) implémente les fonctions de sécurité (authentification, chiffrage des données...)
- On crée à l'aide de ce protocole de sécurité un "Tunnel" à travers un réseau Public non sécurisé



VPN Définition

- Un VPN doit assurer:
  - » L'authentification
  - » L'intégrité
  - » La confidentialité
  - » La gestion des clés
- Il permet de relier en toute sécurité:
  - Deux sites éloignés à travers un réseau public non sécurisé (Intranet VPN)
  - Un utilisateur nomade à un Intranet (VPN d'accès)



Chiffrer à quel niveau de la couche OSI ?

- Différentes solutions à chaque niveau
  - Application :
    - » SSH (Secure Shell)
  - Transport:
    - » SSL/TLS(Secure Socket Layer/Transport Layer Security)
  - Réseau:
    - » IPSEC: IP sécurisé
  - Liaison de donnée:
    - » PPTP (Point-to-Point Tunneling Protocol)
    - » L2F (LayerTwo Forwarding) développé par CISCO et remplacé par L2TP
    - » L2TP (Layer Two Tunneling Protocol) : évolution de PPTP et L2F reprenant les avantages de chacun d'eux

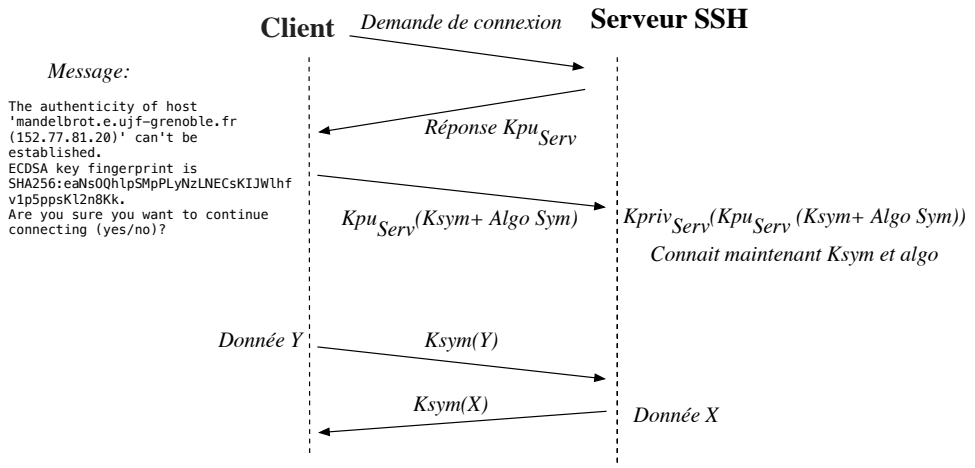
Tunnel SSH (niveau application)

- SSH: ensemble d'outils d'authentification et de chiffrements
- SSH permet de :
  - Accès sécurisé à un site distant : ssh remplace telnet, rlogin...
  - Copie de fichier sécurisé entre machine: scp au lieu de rcp
  - Transfert de fichier sftp au lieu de ftp
- Mais aussi de créer un Tunnel permettant le transfert de n'importe quelle application utilisant TCP
  - x11 (fenêtre X), SMTP, POP, IMAP...
  - Utile lorsque des règles de filtrage strictes sont installées dans un pare-feu

SSH (niveau application)

- SSH garantit:
  - L'authentification du serveur par le client par clés privée/publique (RSA)
    - » Sur le serveur génération d'un couple de clés privé/publique
    - » Soit le client connaît la clé publique du serveur (fichier .ssh/ Know_hosts), soit le serveur lui envoie (message particulier de mise en garde de ssh)
    - » Pour l'authentification le client envoie une clé symétrique (appelé clé de session) et l'algorithme de chiffrement utilisé chiffrés à l'aide de la clé publique du serveur
    - » Les données seront chiffrées à l'aide de cette clé symétrique
    - » Utilisation possible de certificats X509

SSH (niveau application)



- Clé publique du serveur sauvegardée dans .ssh/know_hosts

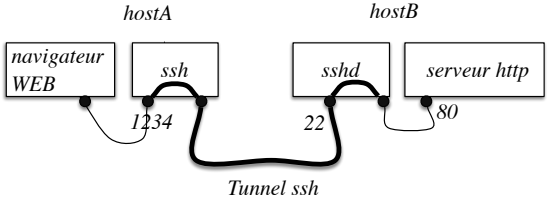
SSH (niveau application)

- **L'authentification du client par le serveur**
  - **Soit par login et mot de passe classique**
  - **Soit par couple de clé privé/publique appartenant cette fois au client**
    - » **Le client génère une paire de clé privée/publique (commande** *ssh-keygen***)**
    - » **La clé publique du client est rentrée sur le serveur (fichier** *.ssh/ know_hosts* contenant les noms DNS, les adresses IP , l'algo de chiffrement choisi et la clé publique*)*
    - » **Le serveur génère un message chiffré à l'aide de la clé publique du client, le client doit savoir déchiffrer ce message grâce à sa clé privée**
- **Compression éventuelle des données**

Exemple de tunnel SSH

- *Accès à un serveur WEB sur hostB*
- *Sur hostA:*

ssh -l Nomlogin -g -N -L 1234:hostB:80 hostB
sur navigateur http:// localhost[1234]



- *Pour le transfert de fichier il existe SFTP qui fait le tunnel automatiquement (le serveur SFTP écoute sur le port 115)*
- *On peut ainsi passer à travers un firewall qui filtre tout sauf ssh*

Commande tunnel SSH

Résumé syntaxe:

-L définition de tunnel

-N : pas de prompt du ssh

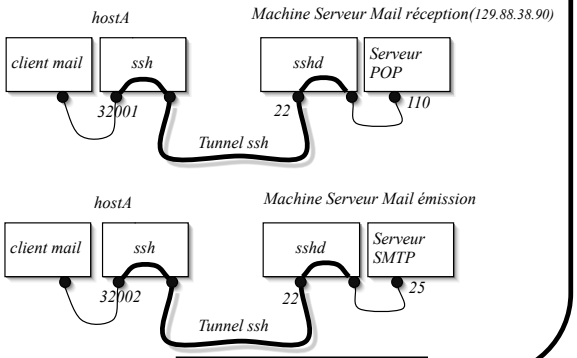
-l : Nom de compte sur la machine distante

ssh -l NomLogin -g -N -L
port_entree_tunnel:machine_destination_tunnel:port_destination_tunnel

On peut définir plusieurs tunnels dans la même commande

Accès aux serveurs de Mail sécurisé à l'aide de SSH

- Mise en place de deux tunnels : serveurs de d'émission (port 25) et de réception (port 110):
- *ssh -l sicard -N -L 32001:129.88.38.90:110 129.88.38.90 -L 32002:129.88.38.90:25 129.88.38.90*
- 129.88.38.90 est l'adresse de la machine sur laquelle tourne les serveurs de mail
- Il faut ensuite configurer le client mail :
  - Réception 32001 localhost
  - Emission 32002 localhost



Tunnel SSL/TLS (niveau transport)

- **SSL : Secure Socket Layer**
  - » Développé au départ par Netscape en 1994
- **TLS (Transport Layer Security)**
  - » concepts de SSL repris sous le nom TLS en 1999 par l'IETF (Internet Engineering Task Force) qui rédige les RFC d'Internet
- **Implémentés au dessus des sockets**
- Permet
  - » l'authentification du serveur
  - » le chiffrement des données échangées,
  - » l'intégrité des données
- Indépendant du protocole sous-jacent
- Applicables à toutes les applications développées sur TCP et UDP sans ré-écriture

Tunnel SSL/TLS (niveau transport)

- Utilisation généralisée aujourd'hui pour de nombreuses applications:
  - » https (port 443)
  - » pop3s, imaps
  - » ftps
  - » telnets
  - » ...
- Deux sous-protocoles
  - » Record Protocol : chiffrement et/ou signature des données
  - » Handshake Protocol: négociation de la session
- Il existe une version open source : **OpenSSL** (aujourd'hui: **LibreSSL**)
- Découverte d'une faille de sécurité importante dans OpenSSL en 2014

Principe Tunnel SSL/TLS

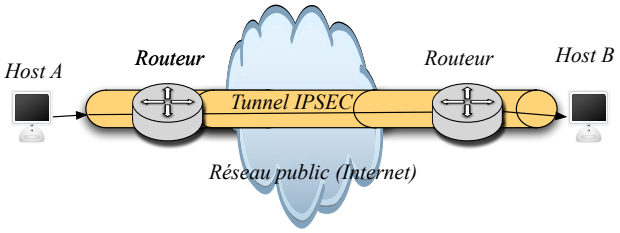
- Navigateur client envoie une demande de connexion sécurisé au serveur web
- Le serveur envoie au client son certificat contenant sa clé publique et une signature numérique
- Le Navigateur essaye de déchiffrer la signature grâce aux autorités de certification contenues dans le celui ci
- Si il n'y arrive pas il essaye avec la clé publique contenue dans la réponse mais avertit l'utilisateur du risque potentiel
- Le navigateur du client génère une clé de chiffrement symétrique qu'il chiffre à l'aide de la clé publique et l'envoie au serveur
- Les échanges se font ensuite à l'aide de cette clé symétrique

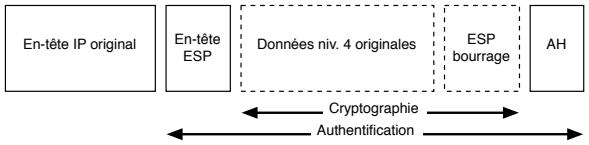
Sécurité et tunnel au niveau "réseau"

- **IP sécurisé (IPSEC)**
- **Plusieurs protocoles possibles**
  - Protocole d'authentification (entête AH) qui assure l'intégrité et l'authenticité des datagrammes IP (sans chiffrement)
  - Protocole ESP (Encapsulating Security Payload) peut aussi permettre l'authentification des données mais est principalement utilisé pour le chiffrement des données
  - IKE (Internet Key exchange) protocole qui permet d'assurer l'échange de clé de cryptage
- **Plusieurs modes possible:**
  - **Mode *transport* :** se fait au niveau des machines qui veulent échanger des données sécurisées (de bout en bout)
  - **Mode *Tunnel*:** IPSEC est implémenté au niveau d'un routeur intermédiaire (possible aussi de bout en bout)
    - » **Utilisation "classique" pour les VPN (Virtual Private Network)**
    - » **A noter: on emploie souvent le terme de VPN pour l'utilisation de tunnel de niveau "réseau"**

Modes d'IPSEC

- **Mode Transport (sécurité entre machines hôtes "de bout en bout")**



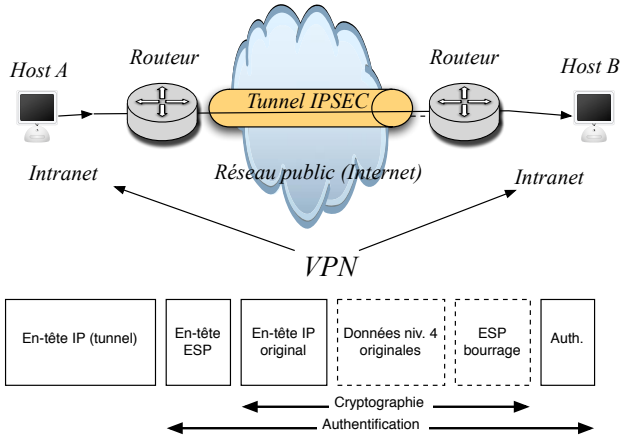


Possibilités supplémentaires de IPSEC

- On peut définir la sécurisation des paquets suivant des règles sur les paquets (~acces list) et des chiffrements divers
- Inconvénients dans le cas d'une sécurisation de bout en bout (machine hôte à machine hôte)
  - » L'utilisation de Pare-feux est impossible: Le routeur ne peut "voir" l'entête Transport
  - » Mise en place du NAT dynamique sur un routeur de sortie n'est pas possible car il ne peut pas changer les numéros de port de l'entête transport

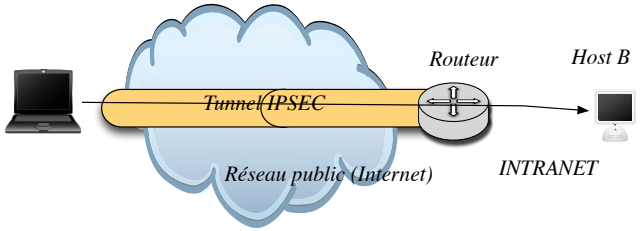
Modes d'IPSEC

- **Mode Tunnel**
  - » Définition d'un VPN composé de différents Intranets distants
  - » La communication est sécurisé à travers un Tunnel définit sur les routeurs d'entrée des Intranets
  - » L'entête IP originale est aussi chiffrée



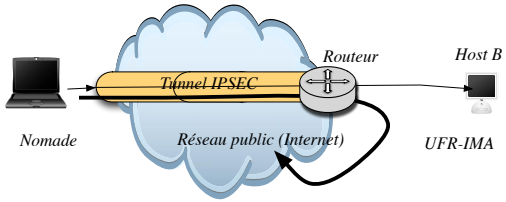
IPSEC pour utilisateur nomade

- **Accès à distance sécurisé pour utilisateur nomade**
- **Authentification par**
  - **Login/mot de passe**
  - **Certificat**



Dans la pratique
Exemple Accès UFR-IM2AG

- Accès à l'UFR IM2AG depuis l'extérieur
- Client CISCO VPN permet de créer un Tunnel IPSEC avec routeur d'entrée de l'UFR
- Configuration client CISCO sur machine nomade
  - » Adresse du routeur en bout de tunnel à l'UFR-IMA2G
  - » Certificat utilisateur (ou compte + mot de passe)
- Tout le trafic est sécurisé et passe par le routeur de l'UFR-IMA2G avant de ressortir éventuellement sur l'Internet d'une manière non sécurisée



Dans la pratique
VPN disponible sur le WEB

- Permet un accès anonyme au WEB; l'adresse IP des requêtes n'est pas la votre
- Confiance dans le fournisseur de VPN