Applications réseau

Cours 5 : Quelques notions de sécurité

Florian Sikora
florian.sikora@dauphine.fr

LAMSADE

M1 apprentissage

Adapté des slides de Kurose & Ross

Cours 5 : Sécurité

- Qu'est-ce que la sécurité des réseaux ?
- Principes de la cryptographie
- Authentification
- Intégrité des messages
- Certificats
- Sécuriser l'email
- Sécuriser TCP avec SSL
- Sécurisation de réseaux sans-fil
- Firewalls

Rappels

- Domaine de la sécurité :
  - Comment des gens mal intentionnés peuvent attaquer un réseau.
  - Comment se défendre contre ces attaques.
  - Comment construire des architecture “immunes” à ces attaques.
- Internet conçu sans beaucoup de sécurité...

Rappels

- Domaine de la sécurité :
  - Comment des gens mal intentionnés peuvent attaquer un réseau.
  - Comment se défendre contre ces attaques.
  - Comment construire des architecture “immunes” à ces attaques.
- Internet conçu sans beaucoup de sécurité...
  - Vision originelle : un groupe de personnes se faisant mutuellement confiance, attachés à un réseau transparent (1970 : no-password `http://youtu.be/wr-9g0T4jpk`).

Rappels

- Domaine de la sécurité :
  - Comment des gens mal intentionnés peuvent attaquer un réseau.
  - Comment se défendre contre ces attaques.
  - Comment construire des architecture “immunes” à ces attaques.
- Internet conçu sans beaucoup de sécurité...
  - Vision originelle : un groupe de personnes se faisant mutuellement confiance, attachés à un réseau transparent (1970 : no-password `http://youtu.be/wr-9g0T4jpk`).
  - Des problèmes de sécurité à toutes les couches.

Objectifs

- Comprendre les principes généraux de la sécurité des réseaux.
  - Notions rapides de cryptographie et son utilisation.
  - Authentification.
  - Intégrité de messages.
- En pratique.
  - Firewalls, détection d'intrusion.
  - Sécurité au niveau des couches.

Cours 5 : Sécurité

Chiffres

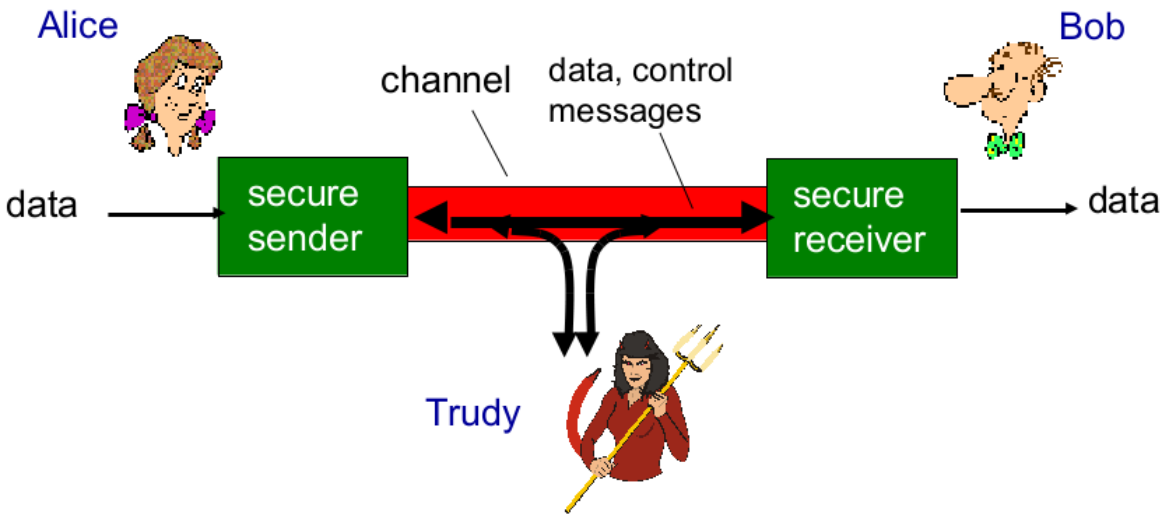
- 80% du trafic e-mail est du spam (parfois vérolé).(APWG)
  - 20 milliards de messages / jour en 2005.
- Cybercrime coûte des milliards de dollars aux entreprises.
  - Aussi un marché de la protection...

Qu'est-ce que la sécurité des réseaux ?

- **Confidentialité**.
  - Seuls l'émetteur et le destinataire devraient "comprendre" le message, voir même savoir qu'il y a message !
  - Émetteur chiffre (ou crypte, ou code...), destinataire déchiffre le message.
- **Authentification**.
  - Émetteur et destinataire veulent être sûrs de l'identité de l'autre.
- **Intégrité des messages**.
  - Émetteur et destinataire veulent être sûrs que leur message n'a pas été modifié sans détection.
- **Disponibilité et contrôle d'accès**.
  - Cas de DoS, personnes illégitimes pour empêcher une communication légitime...

Ami / ennemi

- Le monde de la sécurité utilise souvent Alice (A), Bob (B) et Trudy (T) (ou Ennessa ?).
- Alice et Bob sont amoureux (illégitimes) et veulent s'envoyer des messages (sans que la femme de Bob le sache).
- Mais Trudy (intrus, "intruder")...



Alice et Bob

- Q : Que peuvent représenter Alice et Bob ?

Alice et Bob

- Q : Que peuvent représenter Alice et Bob ?
- Des Alice et Bob de la vraie vie.
- Navigateur/serveur pour lors d'une transaction sécurisée.
- Banque en ligne.
- Serveur DNS.
- Routeurs qui s'échangent des tables de routage.
- ...

Trudy

- Q : Que peut être un méchant ?
- Interception de message.
- Insertion de messages.
- Changer l'adresse du paquet.
- DoS...

Cours 5 : Sécurité

Historique

- Vieux comme ~~le monde~~ Jules César (au moins).
  - Q : Comment ?

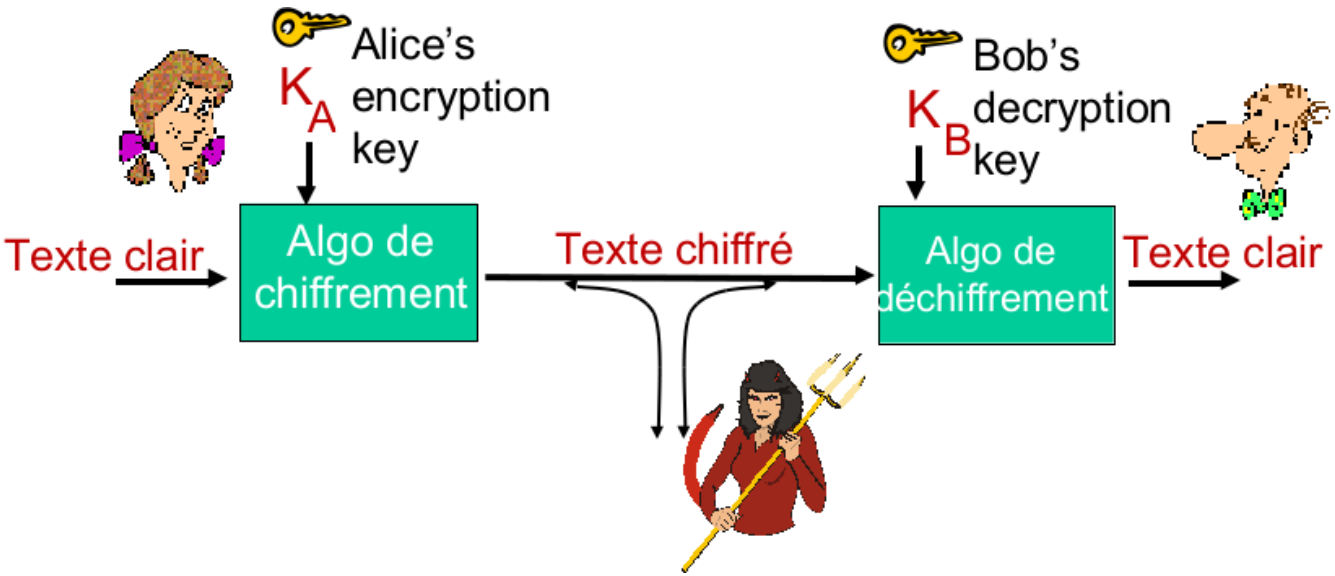
Historique

- Vieux comme ~~le monde~~ Jules César (au moins).
  - Q : Comment ?
  - Décalage fixe dans l'alphabet du même côté.
  - Q : Problèmes ?

Historique

- Vieux comme ~~le monde~~ Jules César (au moins).
  - Q : Comment ?
  - Décalage fixe dans l'alphabet du même côté.
  - Q : Problèmes ?
  - Petit nombre de clés (taille du décalage).
  - Langage naturel n'a pas une fréquence uniforme : détection aisée.
- Bien amélioré depuis !

Vocabulaire



- m : texte en clair.
- $K_A(m)$: texte chiffré par la clé K_A.
- $m = K_B(K_A(m))$.

Codage avec clés symétriques

- $K_A = K_B = K_S$.
- Alice et Bob ont la même clé qu'ils partagent, K_S.
  - Par exemple, codage César.
  - $K_S = 3$.
  - "Bob, je t'aime. Alice" $\rightarrow$ "ere, mh w'dlph. dolfh".

Codage avec clés symétriques

- $K_A = K_B = K_S$.
- Alice et Bob ont la même clé qu'ils partagent, K_S.
  - Par exemple, codage César.
  - $K_S = 3$.
  - "Bob, je t'aime. Alice" $\rightarrow$ "ere, mh w'dlph. dolfh".
  - Problème : 25 tests maximum et on retrouve le message.

Codage avec clés symétriques

- $K_A = K_B = K_S$.
- Alice et Bob ont la même clé qu'ils partagent, K_S.
  - Par exemple, codage César.
  - $K_S = 3$.
  - "Bob, je t'aime. Alice" $\rightarrow$ "ere, mh w'dlph. dolfh".
  - Problème : 25 tests maximum et on retrouve le message.
- Problème, comment Alice et Bob se mettent d'accord pour la clé ?

Clés symétriques : chiffrement par substitution

- Déterminer pour chaque lettre de l'alphabet la manière dont on va le coder.
- Nombre de clefs possible ?

Clés symétriques : chiffrement par substitution

- Déterminer pour chaque lettre de l'alphabet la manière dont on va le coder.
- Nombre de clefs possible ?
- $26! = 2^{88}$ = beaucoup beaucoup.

Clés symétriques : chiffrement par substitution

- Déterminer pour chaque lettre de l'alphabet la manière dont on va le coder.
- Nombre de clefs possible ?
- $26! = 2^{88}$ = beaucoup beaucoup.
- Exemple :
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - AZERTYUIOPQSDFGHJKLMWXCVBN
  - SUBSTITUTION devient LWZLMOMWMOGF

Clés symétriques : chiffrement par substitution

- Déterminer pour chaque lettre de l'alphabet la manière dont on va le coder.
- Nombre de clefs possible ?
- $26! = 2^{88}$ = beaucoup beaucoup.
- Exemple :
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - AZERTYUIOPQSDFGHJKLMWXCVBN
  - SUBSTITUTION devient LWZLMOMWMOGF
- Problème ?

Clés symétriques : chiffrement par substitution

- Déterminer pour chaque lettre de l'alphabet la manière dont on va le coder.
- Nombre de clefs possible ?
- $26! = 2^{88}$ = beaucoup beaucoup.
- Exemple :
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - AZERTYUIOPQSDFGHJKLMWXCVBN
  - SUBSTITUTION devient LWZLMOMWMOGF
- Problème ?
- Dans un texte en langage naturel, certaines lettres sont plus répétées que d'autres : donc aussi dans le mot chiffré ! (analyse de fréquence)

Clés symétriques : Plus sophistiqué (années 1500)

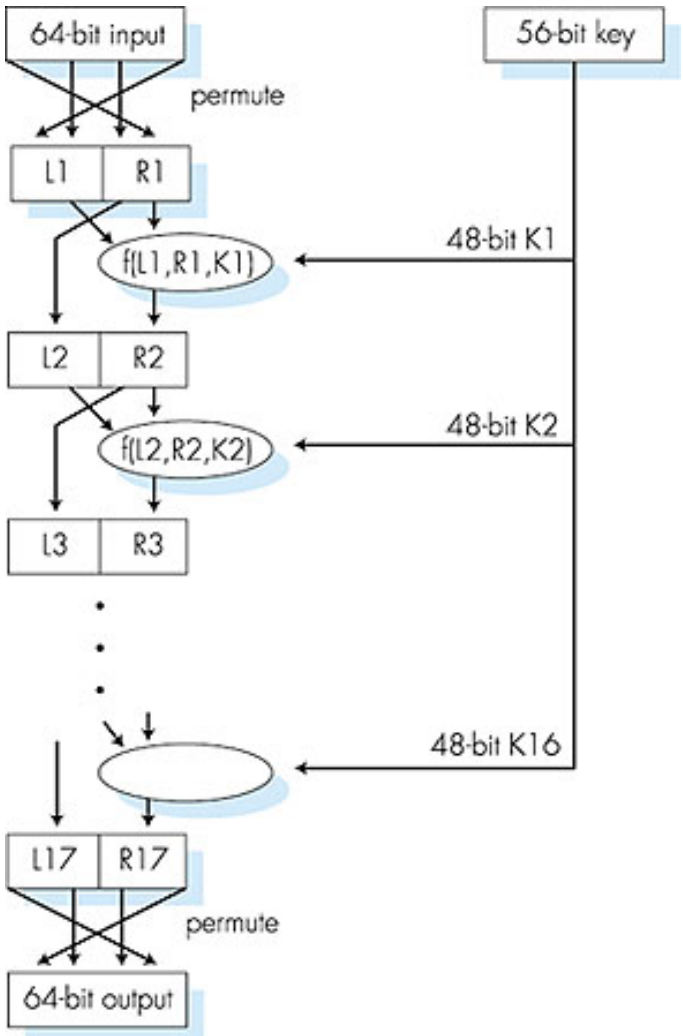
- n codages lettre à lettre différents : $M_1, M_2, ..., M_n$.
- Un **pattern cyclique** pour le codage.
  - Par exemple, pour $n = 4$,
    $M_1, M_3, M_4, M_2, M_3; M_1, M_3, M_4, M_2, M_3$
- On code chaque caractère du texte en utilisant le codage lettre à lettre **dans l'ordre du pattern**.
  - Bob : b dans M_1, o dans M_3, b dans M_4.
- Clé : les n codages lettre à lettre et le pattern cyclique.
- Évite le problème du codage par substitution. Voir aussi chiffre de Vigenère.

Clés symétriques : DES

- DES : Data Encryption Standard.
- Très largement utilisé dans les années 70 jusqu'aux années 90. **Non recommandé depuis**.
- En 1997-1999, la société RSA veut montrer qu'il n'est plus bon, en plus d'être lent. Lance un concours de crackage.
  - 1997 : 4 mois.
  - 1999 : Moins de 24 heures (100 000 machines distribuées, brute-force à raison de 245 milliards de clés à la seconde).

Clés symétriques : DES

- Chiffre le texte clair en morceaux de 64 bits, à l'aide d'une clé de 56 bits (input de 64 bits, output de 64 bits). Chaque morceau est encodé indépendamment.
- Trois étapes :
  1. Permutation du bloc.
  2. 16 étapes de transformations avec 48 bits de la clé (tables de mapping, XOR, etc)
  3. Repermutation du bloc selon le schéma initial.



Clés symétriques : DES - AES

- DES remplacé par AES en 2001.
- Blocs de 128 bits.
- Clés de 128, 192 ou 256 bits.
- Résolution par brute-force qui prenait 1 seconde pour DES (2^{55} essais) prend 149 mille milliards d'années.

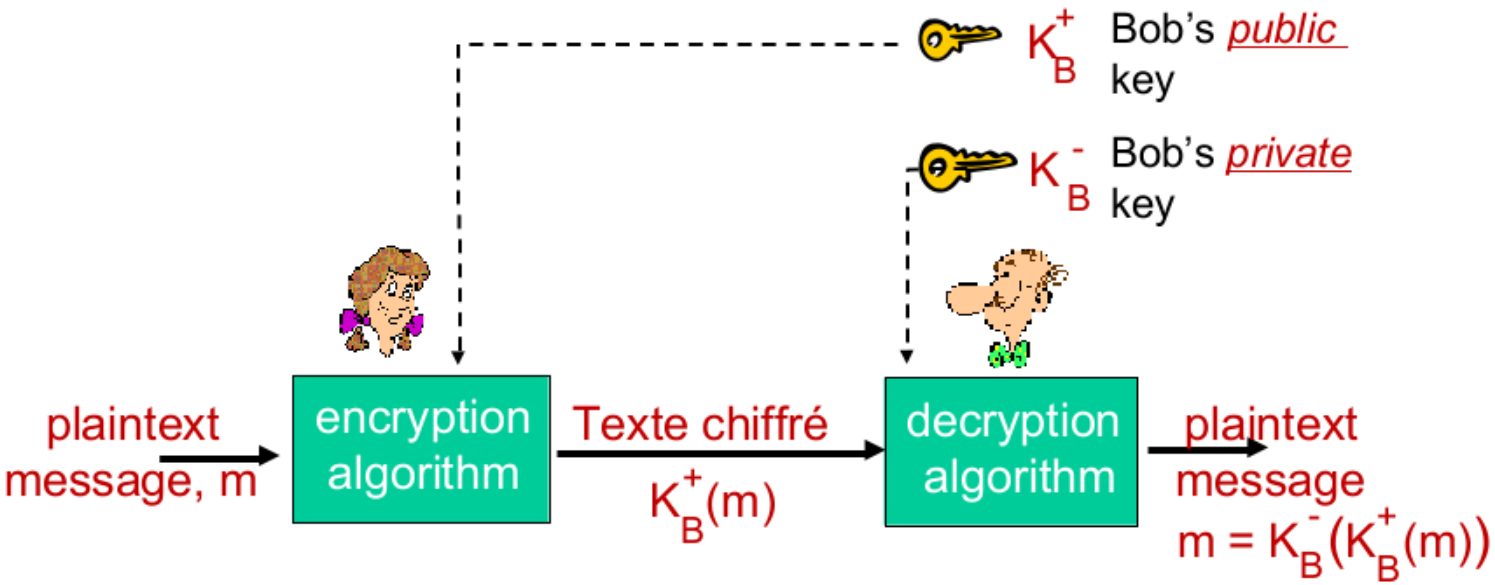
Codage avec clés publique/privée

- Clé symétrique (pendant 2000 ans)
  - L'émetteur et le destinataire doivent connaitre une **clé secrète partagée**.
  - Problème : **comment décider de la clé en premier lieu**, surtout s'ils ne se "rencontrent" jamais ?

Codage avec clés publique/privée

- Clé symétrique (pendant 2000 ans)
  - L'émetteur et le destinataire doivent connaitre une **clé secrète partagée**.
  - Problème : **comment décider de la clé en premier lieu**, surtout s'ils ne se "rencontrent" jamais ?
- Clé publique
  - Approche totalement différente ! (Rivest, Shamir, Adleman, 1978).
  - Émetteur et destinataire **ne partagent rien de secret**.
  - Clé pour le chiffrement connue de tous.
  - Clé pour le déchiffrement connue seulement par le receveur.

Codage avec clés publique/privée



- La clé privée est secrète.
- Clé privée et publique **sont liées** par une fonction mathématique.
- Il doit être (quasi) impossible de retrouvé la privée depuis la publique.

Codage RSA

- Indispensable :
  1. Avoir des clés K_B^+ et K_B^- telles que $K_B^+(K_B^-)) = m$.
  2. En ayant la clé publique K_B^+, il doit être impossible de calculer la clé privée K_B^-.

Codage RSA

- Idée :
  - Choix de deux grands nombres premiers et un exposant.
  - Clés privés et publique issues d'un calcul à sens unique.
  - Utilise le fait que **factoriser le produit de grands nombres premiers est très très long**.
- Longueurs de clés :
  - 512 bits : cassable avec une 100aine de machines.
  - 1024 bits : actuellement conseillé mais cassable à l'avenir.
  - 2048 bits : conseillés par les experts.

RSA : Pré-requis, un peu d'arithmétique

- (Rappel de primaire) $x \bmod n$ = reste de x quand il est divisé par n.
- Faits :
  - $(a \bmod n + b \bmod n) \bmod n = (a + b) \bmod n$.
  - $(a \bmod n - b \bmod n) \bmod n = (a - b) \bmod n$.
  - $(a \bmod n \cdot b \bmod n) \bmod n = (a \cdot b) \bmod n$.
- Donc,
  - $(a \bmod n)^d \bmod n = a^d \bmod n$.

RSA : Pré-requis, un peu d'arithmétique

- (Rappel de primaire) x mod n = reste de x quand il est divisé par n.
- Faits :
  - $(a \bmod n + b \bmod n) \bmod n = (a + b) \bmod n$.
  - $(a \bmod n - b \bmod n) \bmod n = (a - b) \bmod n$.
  - $(a \bmod n \cdot b \bmod n) \bmod n = (a \cdot b) \bmod n$.
- Donc,
  - $(a \bmod n)^d \bmod n = a^d \bmod n$.
- Par exemple,
  - $a = 14, n = 10, d = 2$:
    - $(a \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$.
    - $a^d \bmod n = 14^2 \bmod 10 = 196 \bmod 10 = 6$.

RSA

- Message : un pattern de bits.
- Un pattern de bits peut être représenté par un entier unique.
- Donc, chiffrer un message = chiffrer un entier.

RSA

- Message : un pattern de bits.
- Un pattern de bits peut être représenté par un entier unique.
- Donc, chiffrer un message = chiffrer un entier.

- Par exemple :
  - $m = 10010001$, représenté par 45.
  - Pour chiffrer m, on chiffre le nombre correspondant, ce qui nous donne un nouveau nombre, le texte chiffré.

RSA : créer les clés publiques et privées

1. Choisir deux grands entiers premiers, p, q (par exemple, sur 1024 bits).
2. Calculer $n = p \cdot q, z = (p-1) \cdot (q-1)$.
3. Choisir e (avec $e < n$) t.q. e n'a pas de facteur commun avec z (e et z sont premiers par rapport à l'autre).
4. Choisir d tel que $e \cdot d - 1$ soit exactement divisible par z, i.e. $ed \bmod z = 1$.
5. Clé publique : (n, e). Clé privée : (n, d).

RSA : chiffrement et déchiffrement

- $K_B^+ = (n, e)$, $K_B^- = (n, d)$
- Alice veut envoyer m ($m < n$). Elle calcule $c = m^e \bmod n$.
- Pour déchiffrer, Bob calcule $m = c^d \bmod n$.
- Si Trudy écoute, elle ne peut pas retrouver m sans avoir d...

RSA : chiffrement et déchiffrement

- $K_B^+ = (n, e)$, $K_B^- = (n, d)$
- Alice veut envoyer m ($m < n$). Elle calcule $c = m^e \bmod n$.
- Pour déchiffrer, Bob calcule $m = c^d \bmod n$.
- Si Trudy écoute, elle ne peut pas retrouver m sans avoir d...
- Magie !
  - $c^d \bmod n = (m^e \bmod n)^d \bmod n = m$.
  - Explication dans 2 slides.

RSA : exemple

- Pour créer K_B^+ et K_B^-, Bob choisit $p = 5, q = 7$ (entiers premiers, beaucoup trop petit pour la pratique).
- Donc, $n = p \cdot q = 35$, et $z = (p-1) \cdot (q-1) = 24$.
- $e = 5$ (e et z premiers entre eux).
- $d = 29$ ($e \cdot d - 1 = 144$ divisible par z) ($144/24 = 6$).

RSA : exemple

- Pour créer K_B^+ et K_B^-, Bob choisit $p = 5, q = 7$ (entiers premiers, beaucoup trop petit pour la pratique).
- Donc, $n = p \cdot q = 35$, et $z = (p-1) \cdot (q-1) = 24$.
- $e = 5$ (e et z premiers entre eux).
- $d = 29$ ($e \cdot d - 1 = 144$ divisible par z) ($144/24 = 6$).
- Rend publique ($n = 35, e = 5$), garde privé ($n = 35, d = 29$).

RSA : exemple

- Bob rend public $(n = 35, e = 5)$, garde privé $(n = 35, d = 29)$.
- Alice veut envoyer le message 8-bits 00001100.
  1. $m = 12$
  2. $m^e = 24832$
  3. $c = m^e \bmod n = 17$.

RSA : exemple

- Bob rend public $(n = 35, e = 5)$, garde privé $(n = 35, d = 29)$.
- Alice veut envoyer le message 8-bits 00001100.
  1. $m = 12$
  2. $m^e = 24832$
  3. $c = m^e \bmod n = 17$.
- Bob reçoit c, il utilise sa clé privée.
  1. $c = 17$
  2. $c^d = 48196857210675091509141182522307169 7$.
  3. $m = c^d \bmod n = 12$.
- Message retrouvé !

RSA : pourquoi ça marche ?

- On doit montrer que $c^d \bmod n = m$, avec $c = m^e \bmod n$.
- On sait que pour tout x, y, $x^y \bmod n = x^{y \bmod z} \bmod n$.
  - avec $n = p \cdot q$, p et q premiers et $z = (p-1) \cdot (q-1)$.
  - Non prouvé ici, vous devez y croire !

RSA : pourquoi ça marche ?

- On doit montrer que $c^d \bmod n = m$, avec $c = m^e \bmod n$.
- On sait que pour tout x, y, $x^y \bmod n = x^{y \bmod z} \bmod n$.
  - avec $n = p \cdot q$, p et q premiers et $z = (p-1) \cdot (q-1)$.
  - Non prouvé ici, vous devez y croire !
- Donc :
  - $c^d \bmod n = (m^e \bmod n)^d \bmod n$ (par def).
  - $= m^{ed} \bmod n$ (par les rappels).
  - $= m^{ed \bmod z} \bmod n$ (par le fait).
  - $= m^1 \bmod n$ (car d choisi tel que $ed \bmod z = 1$).
  - $= m$.

RSA : pourquoi c'est sécurisé ?

- En tant que Trudy, vous connaissez la clé publique de Bob, (n, e).
- Comment trouver d ?

RSA : pourquoi c'est sécurisé ?

- En tant que Trudy, vous connaissez la clé publique de Bob, (n, e).
- Comment trouver d ?
- Trouver les facteurs de n, sans connaitre les deux facteurs p et q.
- Mais, si développer est facile (faire $p \cdot q$)...
- Factoriser est dur ! (trouver p et q tels que $p \cdot q = n$).

RSA : pourquoi c'est sécurisé ?

- En tant que Trudy, vous connaissez la clé publique de Bob, (n, e).
- Comment trouver d ?
- Trouver les facteurs de n, sans connaitre les deux facteurs p et q.
- Mais, si développer est facile (faire $p \cdot q$)...
- Factoriser est dur ! (trouver p et q tels que $p \cdot q = n$).
- Mais on ne sait pas s'il est possible de le faire rapidement un jour. Si oui, RSA non sécurisé !
  - Ordinateur quantique peut le faire en temps polynomial (algorithme de Shor) (15 factorisé en 2001, 21 en 2008, 143 en 2012...).

RSA : pourquoi c'est sécurisé ?

- En tant que Trudy, vous connaissez la clé publique de Bob, (n, e).
- Comment trouver d ?
- Trouver les facteurs de n, sans connaitre les deux facteurs p et q.
- Mais, si développer est facile (faire $p \cdot q$)...
- Factoriser est dur ! (trouver p et q tels que $p \cdot q = n$).
- Mais on ne sait pas s'il est possible de le faire rapidement un jour. Si oui, RSA non sécurisé !
  - Ordinateur quantique peut le faire en temps polynomial (algorithme de Shor) (15 factorisé en 2001, 21 en 2008, 143 en 2012...).
  - Mais la cryptographique quantique existe (BB84) par ex. `http://www.idquantique.com/`

RSA - Performances...

- Très gourmand en ressources.
- DES est au moins 100 fois plus rapide.

RSA - Performances...

- Très gourmand en ressources.
- DES est au moins 100 fois plus rapide.
- Utiliser RSA pour établir une connexion sécurisée afin de s'échanger une seconde clé, symétrique cette fois.

RSA - Questions...

- Dans l'exemple, avec p et q petit, c^d déjà très grand !
- En pratique, p et q de l'ordre de plusieurs centaines de bits !
- Comment trouver des nombres premiers élevés ?
- Comment choisir e et d ?
- Comment mettre à la puissance de grands nombres ?
- Hors sujet...

Cours 5 : Sécurité

Authentification

- But : prouver à quelqu'un que l'on est bien la personne que l'on prétend être.
- Vie réelle ? :

Authentification

- But : prouver à quelqu'un que l'on est bien la personne que l'on prétend être.
- Vie réelle ? :
  - Par l'apparence (rencontre).
  - Voix (téléphone).
  - Photo (passeport).
  - ...

Authentification

- But : prouver à quelqu'un que l'on est bien la personne que l'on prétend être.
- Vie réelle ? :
  - Par l'apparence (rencontre).
  - Voix (téléphone).
  - Photo (passeport).
  - ...
- Ici, ce type de données non disponible.
- Exemple où Alice doit s'authentifier auprès de Bob.

Authentification - 1er essai

- Protocole : Alice dit à Bob : “Je suis Alice”.

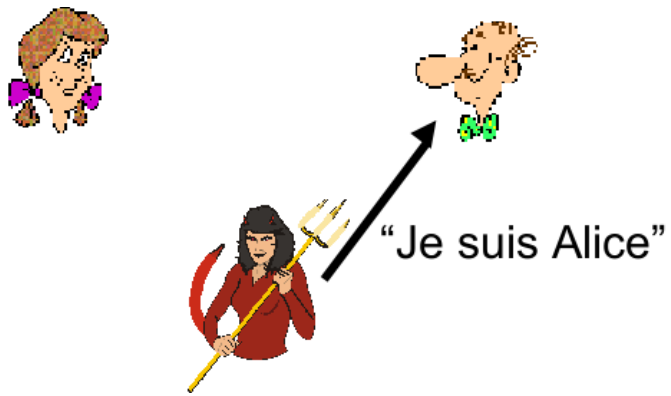
Authentification - 1er essai

- Protocole : Alice dit à Bob : “Je suis Alice”.

- Problème ?

Authentification - 1er essai

- Dans un réseau, Bob ne peut pas voir Alice.
- Trudy peut simplement dire qu'elle est Alice.



Authentification - 2ème essai

- Protocole : Alice dit à Bob : “Je suis Alice” dans un paquet IP contenant en source son adresse IP.

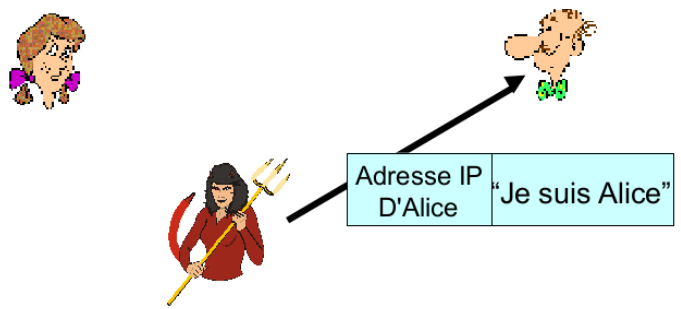
Authentification - 2ème essai

- Protocole : Alice dit à Bob : “Je suis Alice” dans un paquet IP contenant en source son adresse IP.

- Problème ?

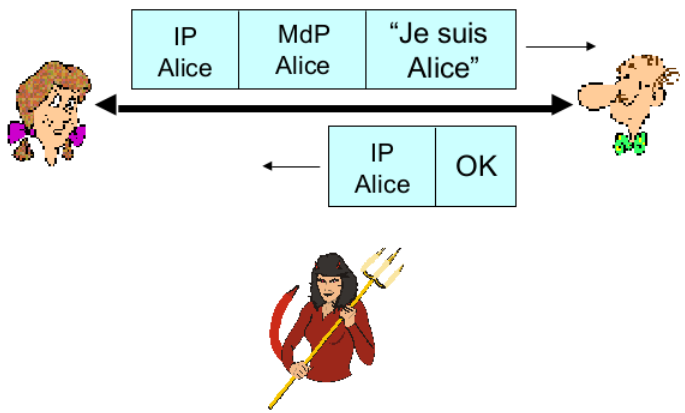
Authentification - 2ème essai

- Trudy peut imiter un paquet avec l'adresse d'Alice.



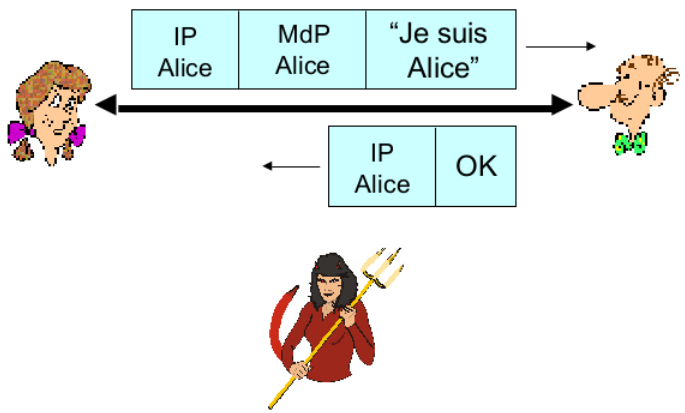
Authentification - 3ème essai

- Protocole : Alice dit à Bob : “Je suis Alice” avec un mot de passe (“clé symétrique”) pour le “prouver”.



Authentification - 3ème essai

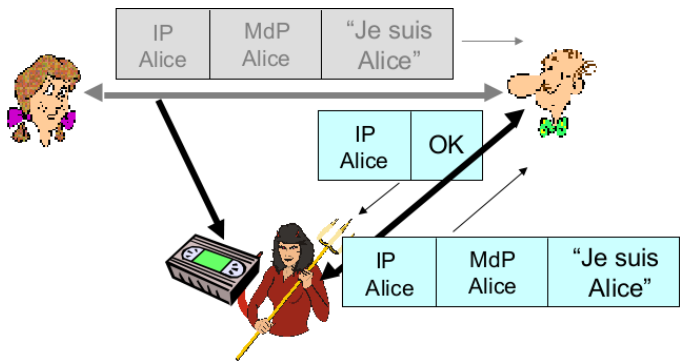
- Protocole : Alice dit à Bob : “Je suis Alice” avec un mot de passe (“clé symétrique”) pour le “prouver”.



- Problème ?

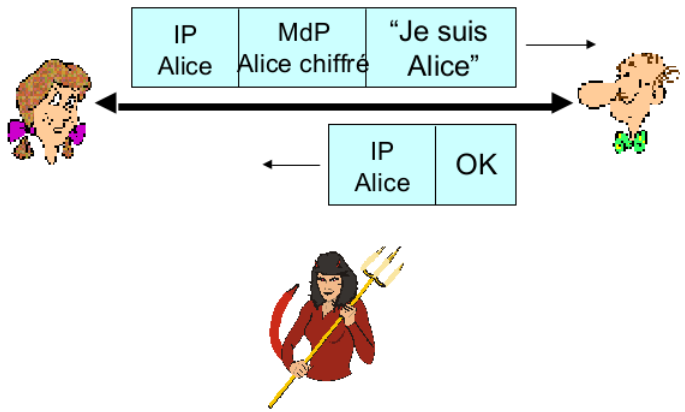
Authentification - 3ème essai

- Trudy peut “sniffer”, enregistrer ce qu’il se passe.
- Connaît alors le mot de passe, peut recréer un paquet.



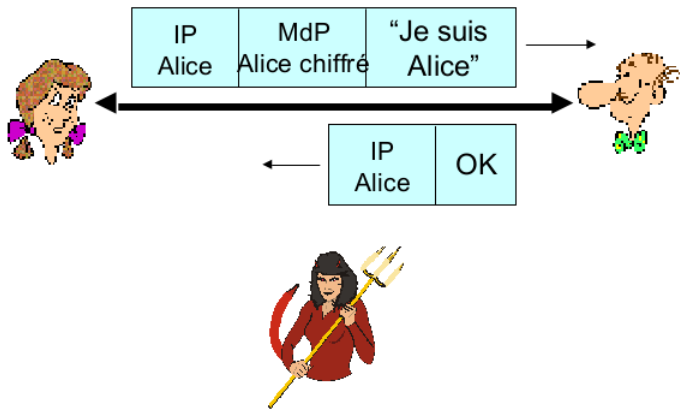
Authentification - 4ème essai

- Protocole : Alice dit à Bob : “Je suis Alice” avec un mot de passe *chiffré* pour le “prouver”.



Authentification - 4ème essai

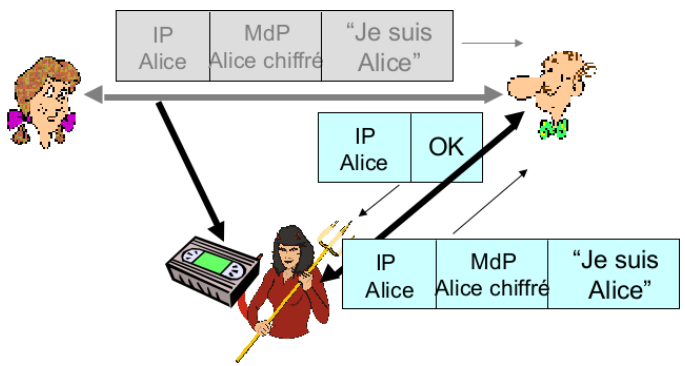
- Protocole : Alice dit à Bob : “Je suis Alice” avec un mot de passe *chiffré* pour le “prouver”.



- Problème ?

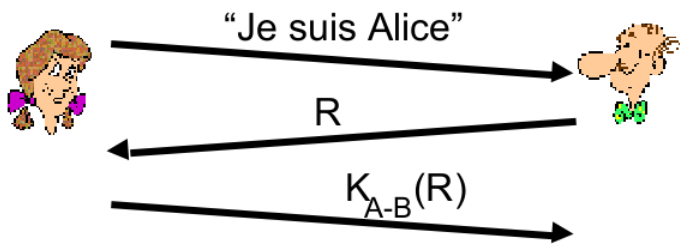
Authentification - 4ème essai

- Trudy peut toujours “sniffer”, enregistrer ce qu’il se passe.
- Ne connaît pas le mot de passe, mais peut renvoyer le même paquet tel quel et se faire passer pour Alice !



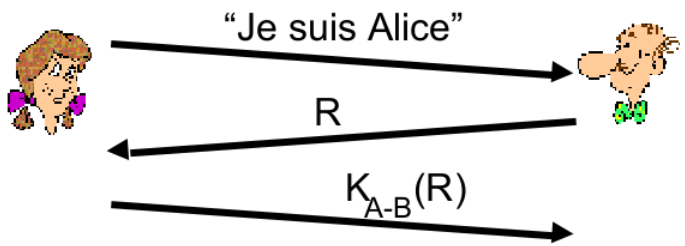
Authentification - 5ème essai

- But : éviter l'attaque par reproduction.
- Avoir un nouveau mot de passe à chaque message (un ordre dans les MdP, un algo générateur de MdP...).
- Utiliser un "nonce" (valeur jetable) : nombre utilisé qu'une seule fois.
- Bob répond à Alice avec un nombre R qu'il choisit.
- Alice répond avec R chiffré selon une clé secrète partagée : elle est donc "en vie".
- Bob reçoit et déchiffre : c'est bien Alice qui est la seule à savoir encoder R.



Authentification - 5ème essai

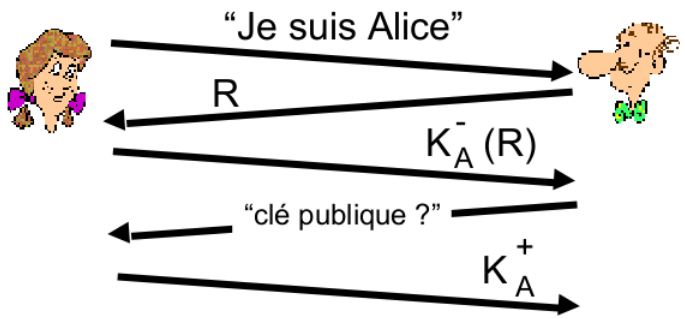
- But : éviter l'attaque par reproduction.
- Avoir un nouveau mot de passe à chaque message (un ordre dans les MdP, un algo générateur de MdP...).
- Utiliser un "nonce" (valeur jetable) : nombre utilisé qu'une seule fois.
- Bob répond à Alice avec un nombre R qu'il choisit.
- Alice répond avec R chiffré selon une clé secrète partagée : elle est donc "en vie".
- Bob reçoit et déchiffre : c'est bien Alice qui est la seule à savoir encoder R.



- Problème : demande le partage d'une clé secrète.

Authentification - 6ème essai

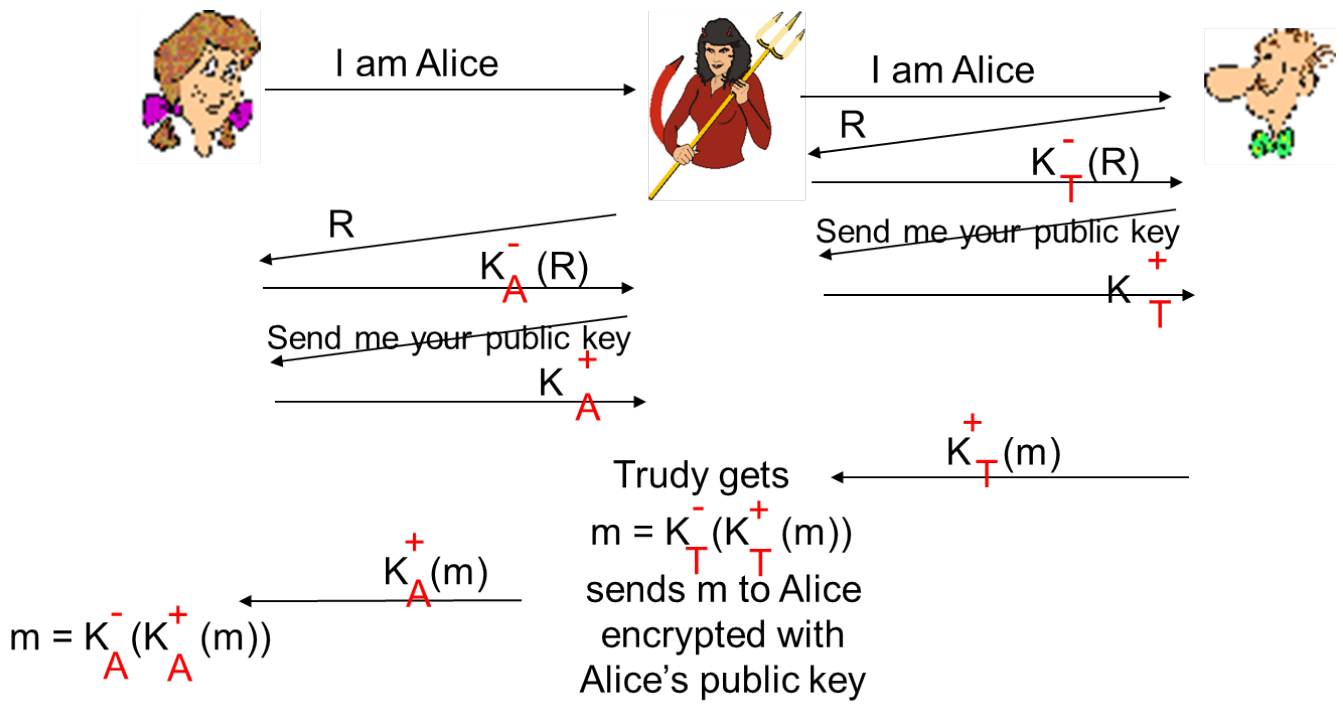
- But : utiliser un chiffrement à clé publique.



- Bob calcule $K_A^+(K_A^-(R)) = R$, authentifiant Alice.

Authentification - 6ème essai - Man in the middle

- Trudy au milieu d'Alice et Bob.
- Se fait passer pour Alice auprès de Bob, et pour Bob auprès d'Alice.



Authentification - 6ème essai - Man in the middle

- Difficile à détecter !
- Bob reçoit tout ce que lui envoie Alice et inversement !
- ... sauf que Trudy lit tout !

Cours 5 : Sécurité

Signature électronique

- Besoin d'attester l'auteur d'un message, le propriétaire. Analogue à la signature physique.
- Signature numérique avec chiffrement.
- Doit être vérifiable, impossible à contrefaire, non répudiable.

Signature électronique

- Signature électronique simple pour un message m :
  - Bob signe m en le chiffrant avec sa clé privée K_B^-, créant un message chiffré $K_B^-(m)$.

Signature électronique

- Alice reçoit le message signé.
- Alice vérifie que m est bien signé par Bob en appliquant la clé publique de Bob K_B^+ à $K_B^-(m)$ et retrouve bien m.
- Si m est retrouvé, c'est que m a été signé avec la clé privée de Bob.

- Alice vérifie donc :
  - Bob a signé m.
  - Personne d'autre n'a signé m.
  - Bob a signé m et non m'.

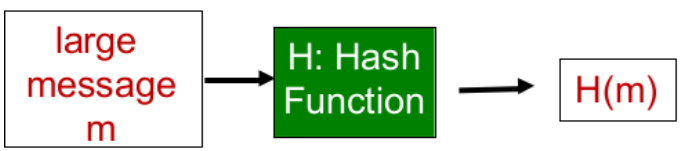
Signature électronique

- Problème, générer cette signature est très couteux en temps et ne se justifie pas pour tous les documents.
- Routeurs, e-mail... n'ont pas besoin de données chiffrées, veulent juste être sûr de l'auteur et de l'intégrité du message.

Signature électronique

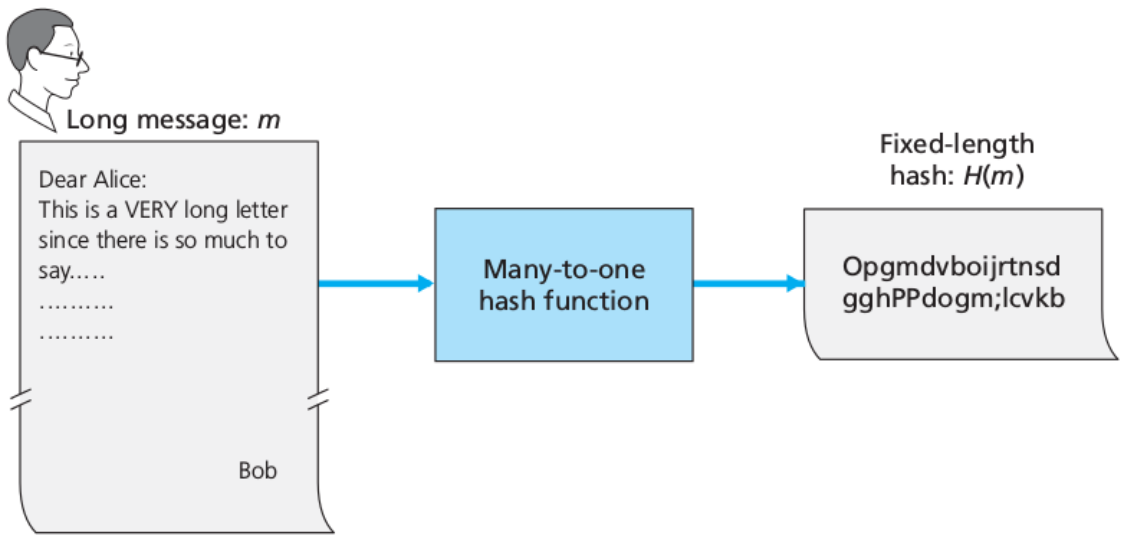
- Problème, générer cette signature est très couteux en temps et ne se justifie pas pour tous les documents.
- Routeurs, e-mail... n'ont pas besoin de données chiffrées, veulent juste être sûr de l'auteur et de l'intégrité du message.

- Approche : résumé de message.
- Une fonction de hachage "résume" tout le message m sur une longueur fixe $H(m)$. (fingerprint)
- Seul $H(m)$ est chiffré.



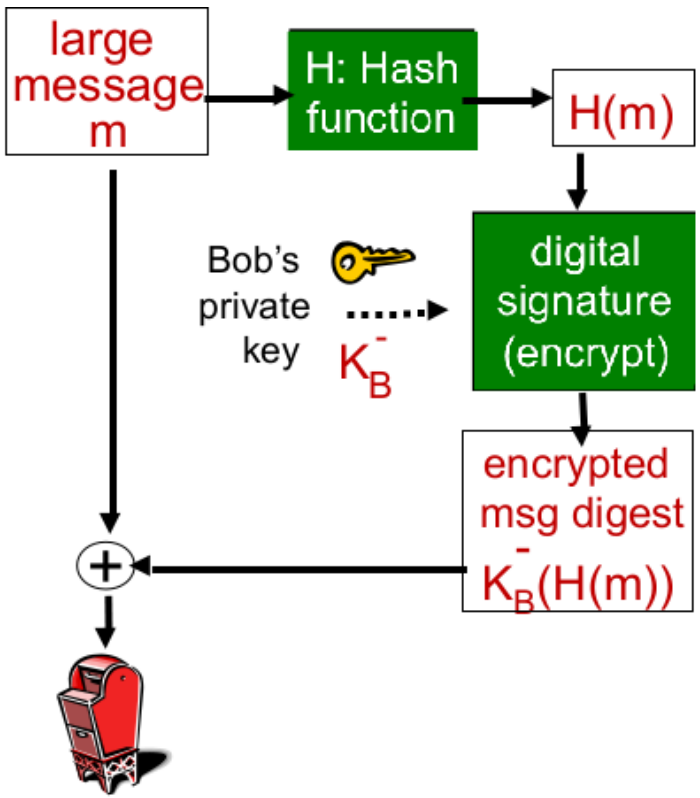
Signature électronique - Hachage

- Le hachage doit :
  - Si $m \neq m'$, $H(m) \neq H(m')$.
  - Étant donné $H(m)$, très difficile d'obtenir m.

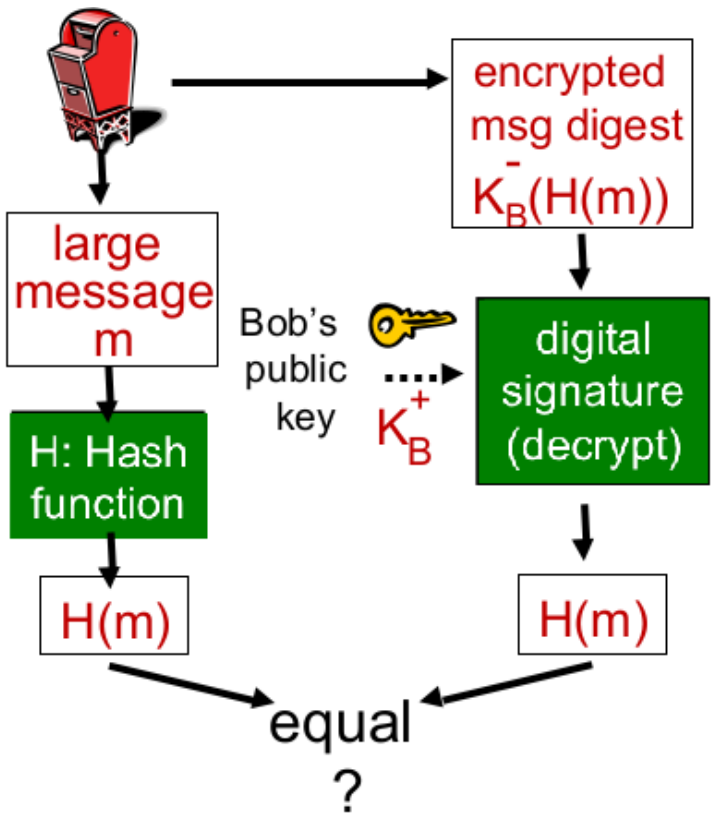


Signature électronique - Hachage

- Bob signe son message.



- Alice vérifie la signature et l'intégrité du message reçu.



Signature électronique - Hachage

- Algorithmes de hachage utilisés.
- MD5 (vérification d'ISO linux...)
  - Calcule des résumés de 128 bits.
  - Facile à calculer : on peut vérifier s'il y a égalité entre le hachage recalculé et le hachage supposé.
- SHA-1

Cours 5 : Sécurité

Blague de la pizza

- Trudy fait une “blague” à Bob, Alice travaille dans une pizzeria.
- Trudy envoie à Alice : “Chère Alice, merci de m’envoyer 13 pizzas à l’adresse de Bob”.
- Trudy signe avec sa clé privée.
- Trudy envoie à Alice sa clé publique, en disant que c’est celle de Bob.
- Alice vérifie la signature : OK.
- Envoie 13 pizzas à Bob...
- ... qui déteste les pizzas.

Problèmes

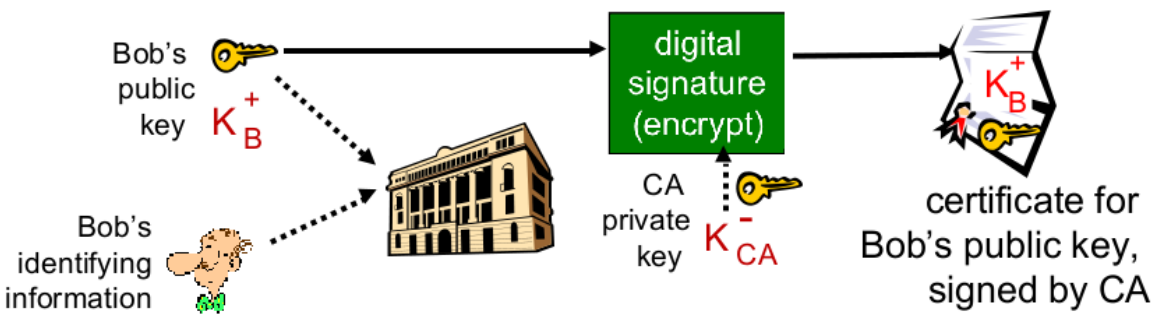
- Chiffrement à clé publique, rien ne prouve qu'on l'on a la "bonne" clé publique.
- Attaque man-in-the-middle précédente : Trudy envoie sa clé publique, mais Bob croit que c'est celle d'Alice.
- Blague de la pizza...

Problèmes

- Chiffrement à clé publique, rien ne prouve qu'on l'on a la "bonne" clé publique.
- Attaque man-in-the-middle précédente : Trudy envoie sa clé publique, mais Bob croit que c'est celle d'Alice.
- Blague de la pizza...
- Besoin d'un *tiers de confiance*, **autorité de certification**.

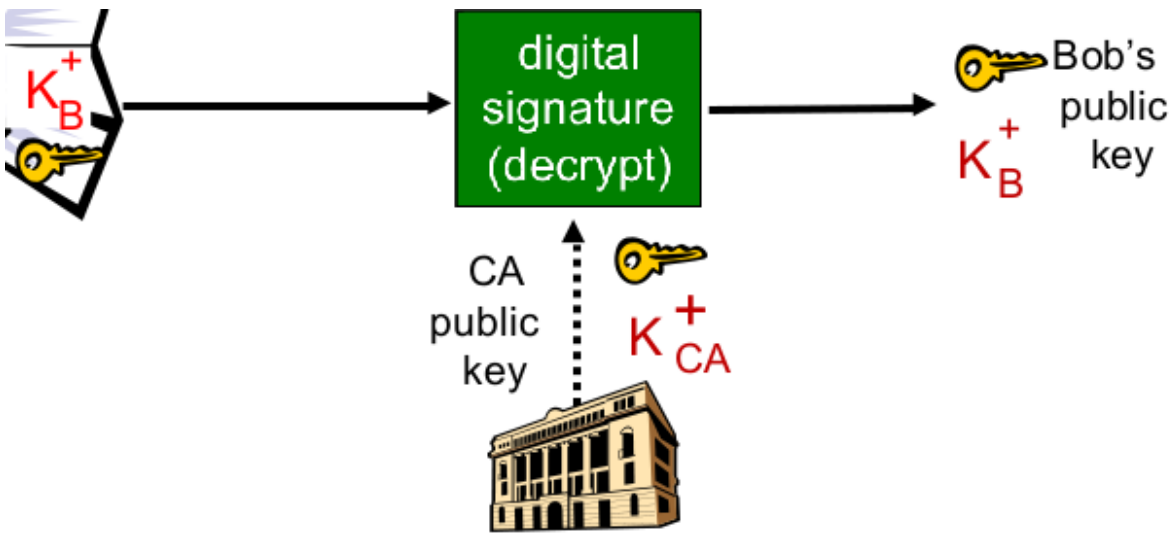
Certificats

- Autorité de certification.
  - Valide des identités et émet des certificats.
- Bob (une personne, un routeur...) enregistre sa clé publique au CA.
  - Bob envoie une preuve de son identité au CA (pas de démarche prédéfinie, il faut faire confiance au CA...).
  - CA crée un certificat liant Bob à sa clé publique.
  - Le certificat contenant la clé publique de Bob est signé par le CA.
- Évite d'associer Bob à la clé publique de Trudy.



Certificats

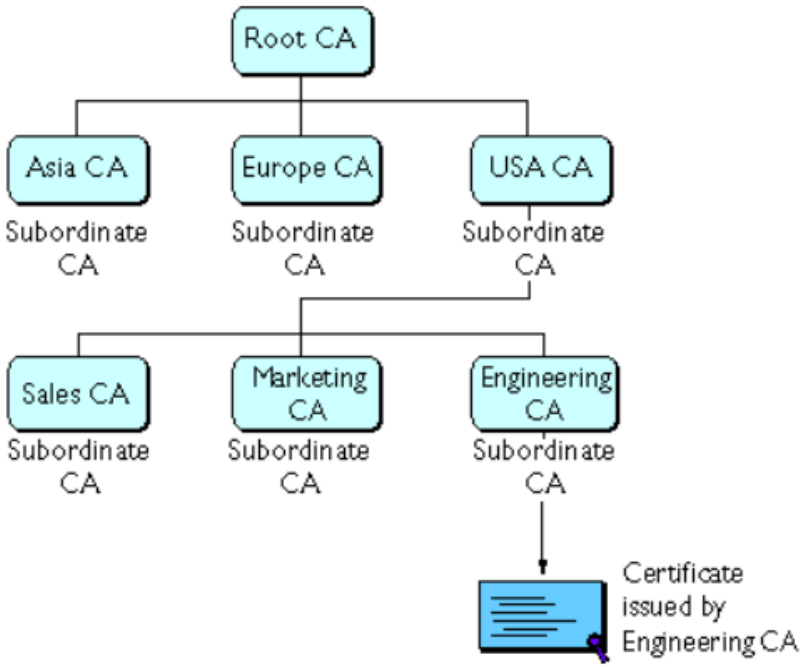
- Quand Alice veut la clé publique de Bob.
  - Récupère le certificat de Bob.
  - Applique la clé publique du CA au certificat de Bob pour obtenir la clé publique de Bob.
- Si Alice ne connait pas Bob mais connait le CA associant la clé publique de Bob à Bob et lui fait confiance, Alice accepte le fait que la clé fournie est celle de Bob.
- Reporte le problème sur le CA : notion de chaîne de certificats.



Certificats

- Clients ont des listes de CA de confiance.
- Organisation hierarchique des CA.
  - Racine auto-certifiée, supposée de confiance.
  - Chaque niveau inferieur est certifié par le CA du niveau supérieur.

Certificats



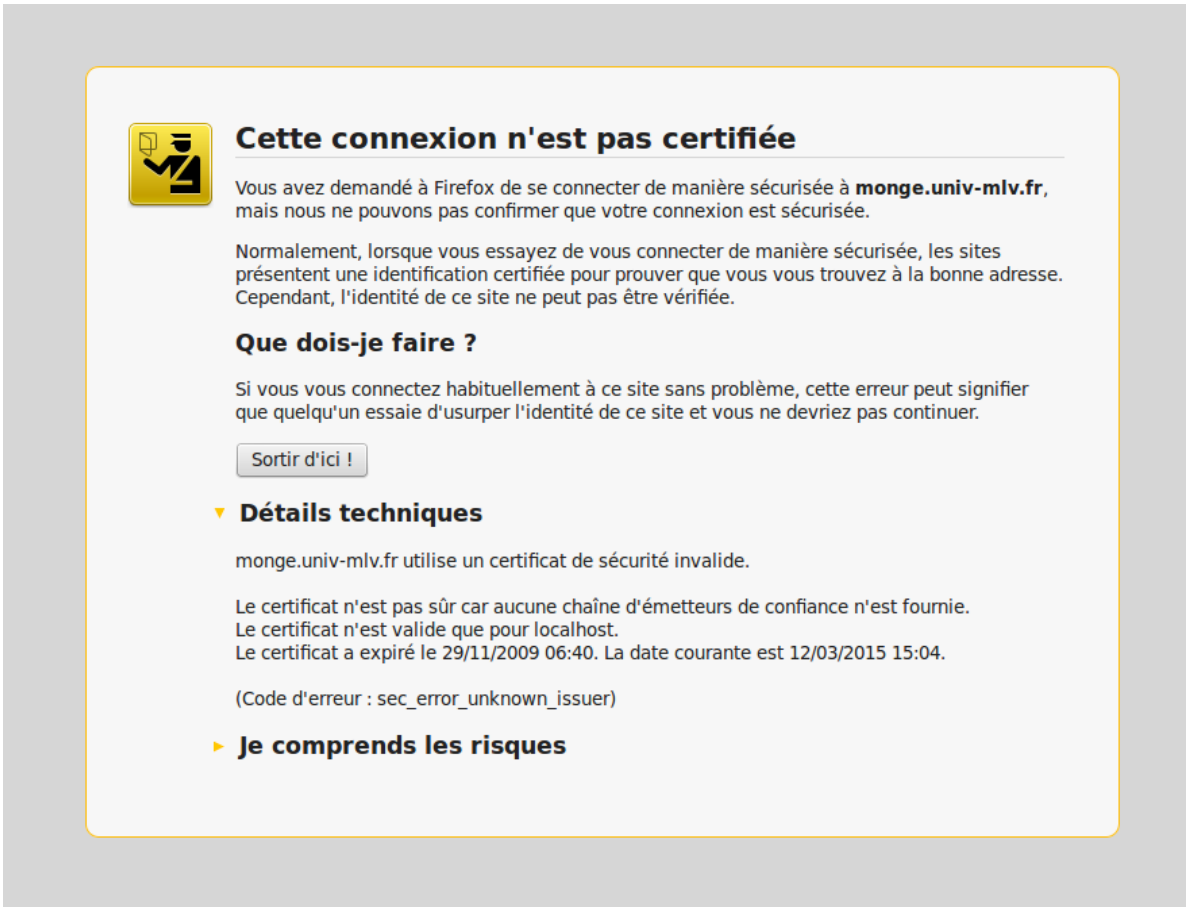
- Root CA certifié par Root CA.
- Certificat de USA CA certifié par Root CA.
- Chaines de certificats.
  - Quand on reçoit un certificat, on vérifie si l'émetteur est de confiance.
  - Si on ne sait pas, on remonte au parent.

Certificats

- Vous pouvez lire les certificats de sites webs stockés dans votre navigateur (préférences > avancée dans firefox par ex.).
- Date d'expiration...

Certificats

- Si on ne connait personne ou qu'il n'y a personne (parfois inutile de payer un certificat pour un votre propre serveur) :



Cours 5 : Sécurité

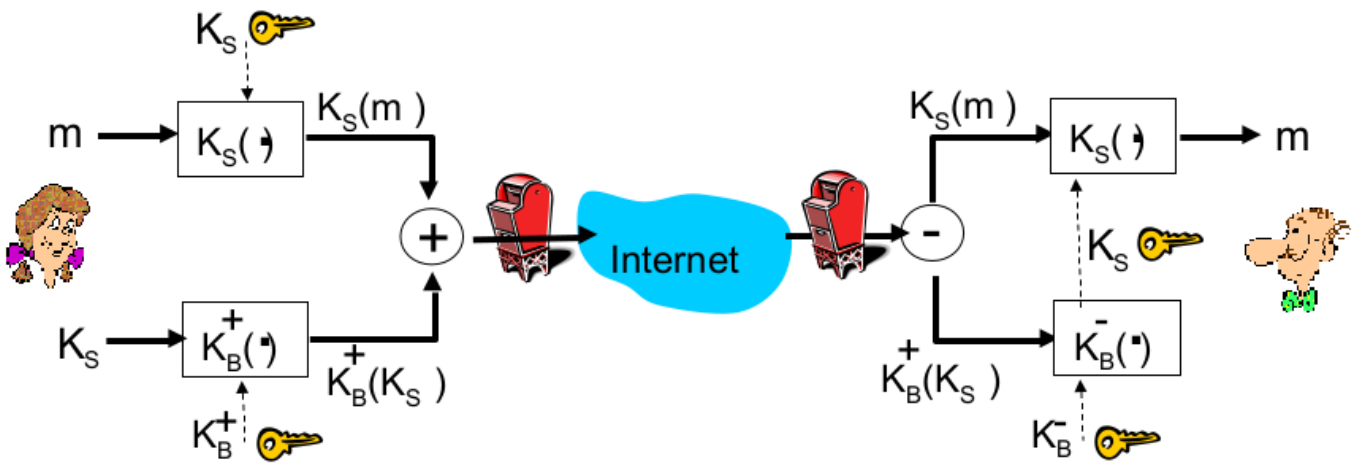
Email sécurisé

- Alice veut envoyer un email confidentiel m à Bob.
- Veut la confidentialité.
  - La femme de Bob ne doit pas avoir accès à leurs échanges.
- Authentification de l'expéditeur.
  - Si reçoit "je ne t'aime plus", veut être sûr que ça vient d'Alice.
- Intégrité.
  - Message arrive dans l'état d'envoi.

Email condifentiel

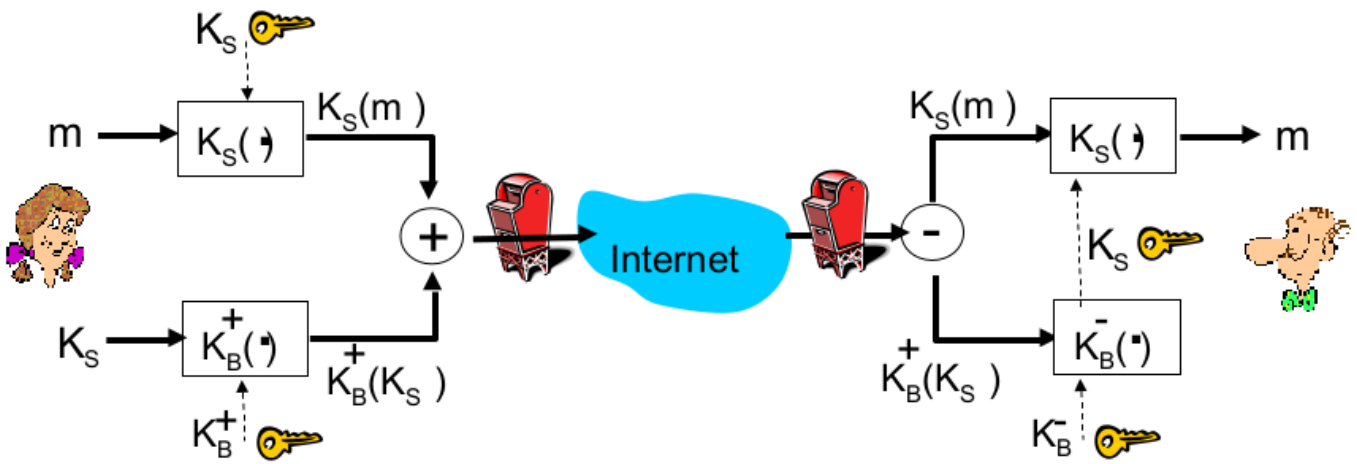
- Peut chiffrer par clé symétrique comme vu avant.
  - Toujours le problème de partager un secret...
- Peut utiliser clé privée / publique.
  - Peu efficace car message gros (pièces jointes...)

Email condifentiel



1. Alice génère au hasard une clé de session symétrique K_S.
2. Alice utilise K_S pour chiffrer son message m.
3. Alice chiffre K_S avec la clé publique de Bob K_B^+ (stockée sur sa page web, donné en personne, un serveur de clés (MIT ?)...).
4. Alice attache le message chiffré et la clé chiffrée ensemble et envoie à Bob.

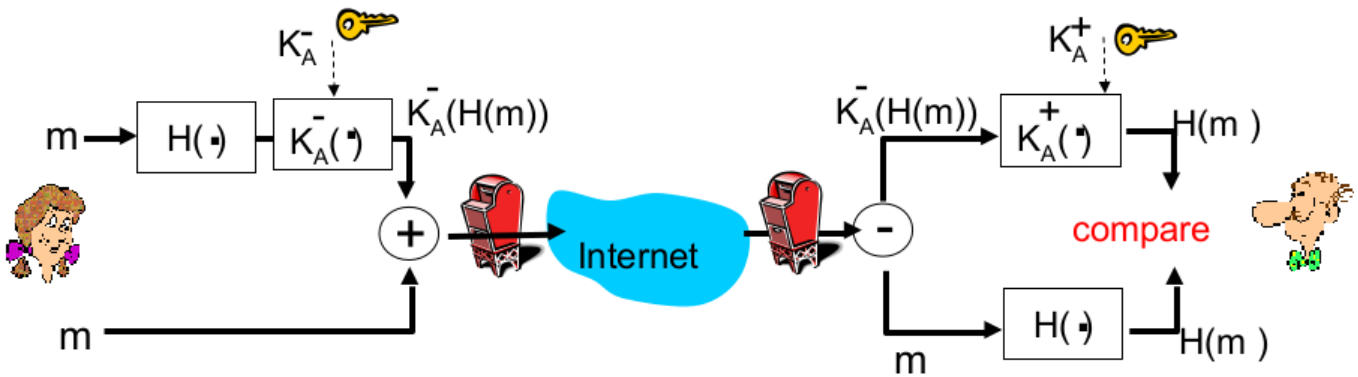
Email condifentiel



1. Bob reçoit l'ensemble.
2. Utilise sa clé privée pour obtenir K_S.
3. Déchiffre m avec K_S

Email sécurisé

- Avoir l'authentification et l'intégrité (on oublie la confidentialité pour le moment).

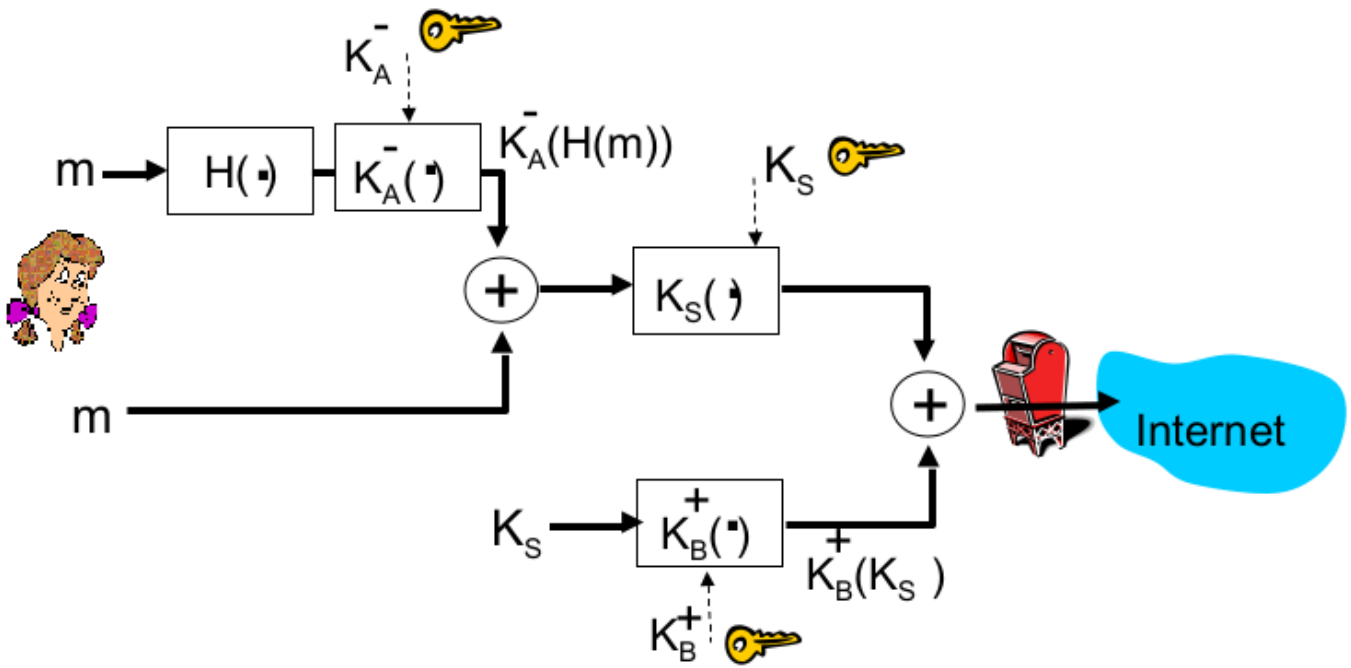


- Alice "signe" le message résumé (hachage + clé privée).
- Alice envoie le message en clair et le message signé.

- Bob applique la clé publique d'Alice au message résumé.
- Compare à son propre hachage du message.

Email sécurisé++

- Avoir l'authentification et l'intégrité et la confidentialité.



- Alice utilise 3 clés !
  - Sa clé privée, la clé publique de Bob et la clé de session symétrique crée.

PGP

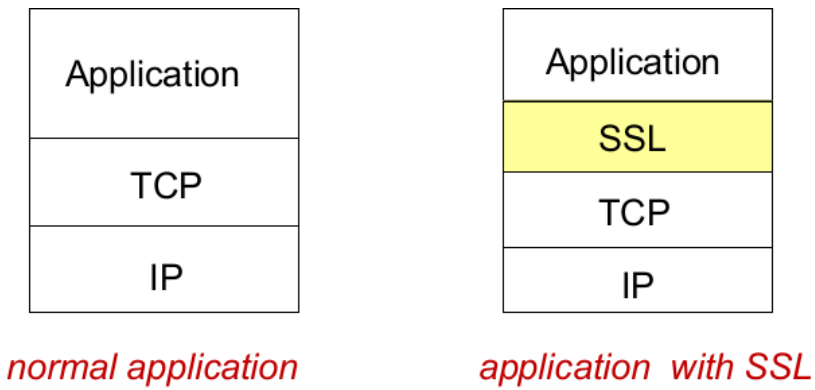
- Doit résoudre le problème de l'échange de clé publique (certificat).
  - L'idéal est de recevoir la clé publique en main propre / page personnelle.
  - "Réseau de confiance" (Alice certifie Bob/CléBob si elle est sûre).
- PGP (Pretty Good Pricacy, 1991), standard pour l'email.
  - Enquête judiciaire pendant 3 ans des USA pour avoir distribué gratuitement un logiciel de chiffrement (diffusion interdite aux USA).
- Utilise le principe précédent :
  - Avec MD5 ou SHA pour le hachage.
  - Avec CAST, triple DES ou IDEA pour le chiffrement clé symétrique.
  - Avec RSA pour chiffrement clé publique.

Cours 5 : Sécurité

SSL (Secure Sockets Layer)

- Largement utilisé.
  - Supporté par la plupart des navigateurs (https).
  - Des milliards de dollars par an passent par SSL.
- Initialement développé par Netscape.
- Buts initiaux :
  - E-Commerce sur Internet.
  - Chiffrement (cartes de crédit).
  - Authentification serveur Web (et en option auth. client).

SSL et TCP/IP



- SSL donne une API aux applications.
- Librairies C, classes Java disponibles.

SSL - Vue générale

- Bob envoyé sur une page sécurisée du serveur web Alice.
- Navigateur et serveur exécutent le protocole d'échange SSL grâce à RSA :
  - Authentification du serveur.
  - Génération d'une clé symétrique partagée.

SSL - Un peu + détaillé (simplifié quand même)

1. Navigateur envoie au serveur sa version de SSL et ses préférences de chiffrement pour la suite.
2. Le serveur fait de même et y ajoute son certificat (contient la clé publique RSA du serveur, signé avec la clé privé du CA).
3. Navigateur reçoit et si CA est de confiance, OK, récupère la clé publique. Sinon, pas de connexion.
4. Navigateur crée une clé de session symétrique (comme pour PGP), la chiffre avec la clé publique du serveur, et l'envoie.
5. Navigateur indique au serveur que ses prochains messages seront chiffrés avec la clé de session. Envoie un message chiffré indiquant que les présentations sont terminées.
6. Serveur indique que ses prochains messages seront chiffrés avec la clé de session et envoie un message indiquant que ses présentations sont terminées.
7. Échanges de messages chiffrés, utilisation de la clé de session symétrique.

SSL et e-commerce : Limites

- SSL pas spécialement crée pour l'e-commerce.
  - Mais pour sécuriser tout type de communication entre clients.
- Protocole général !
- Certificat reçut par Bob ne dit pas si la société Alice est fiable, réputée, à le droit d'accepter des paiements...
- Certificat de Bob ne dit pas si la carte de Bob est OK (cartes volées...)

Cours 5 : Sécurité

Sans-fil

- Ondes vulnérables :
  - Normes de sécurité supplémentaires nécessaires pour les réseaux sans fil.

WEP (IEEE 802.11)

- Plateforme mobile veut se connecter à un point d'accès.
- On suppose qu'ils connaissent tous les 2 la clé symétrique (mot de passe).
  - 40 bits dans WEP 64 bits, 104 dans le WEP 128 bits.
  - Soit 5 ou 13 caractères ASCII.
- Un vecteur d'initialisation (IV) de 24 bits est ajouté par l'émetteur pour chiffrer une seule trame et ajouté en clair dans la trame.
- Changement d'IV à chaque trame.

WEP

- Utilisation de cette clé 64 ou 128 bits (mot de passe + IV de 24 bits) pour initialiser un **générateur pseudo-aléatoire** (aléatoire impossible informatiquement), chiffrant le message (RC4).
- Idée : XOR entre les octets du message et les octets du générateur.

- A réception, on extrait le IV que l'on ajoute au mot de passe partagé.
- Initialise le générateur pseudo-aléatoire.
- XOR à l'envers.

WEP - Problèmes

- IV de 24 bits : 2^{24} IV différents.
- IV change à chaque trame.
- 99% de chances d'avoir le même IV après 12 000 trames.
  - Quelques secondes pour 1 Ko à un débit de 11 Mbit/s.
- IV codé en clair dans la trame : espion sait quand il y a répétition.

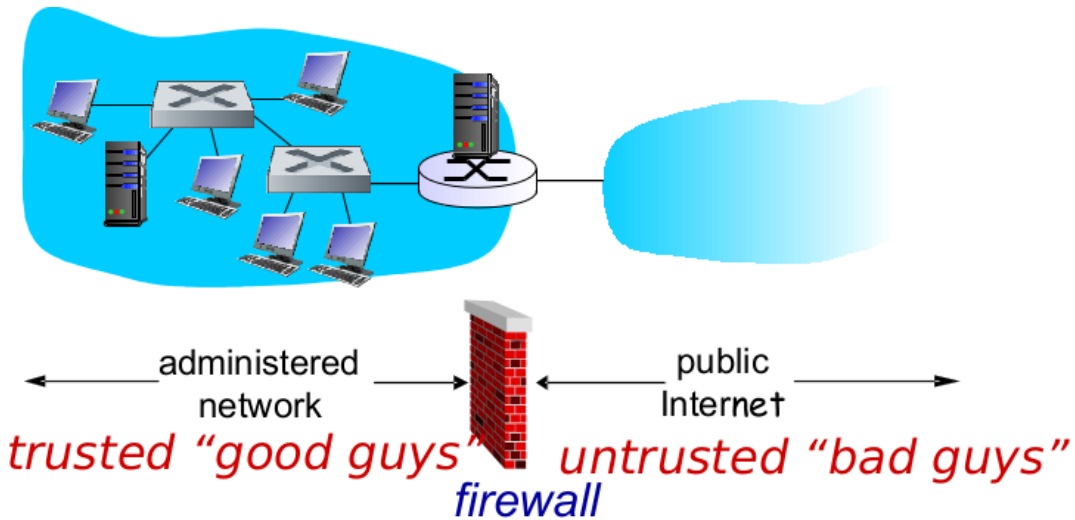
WEP - Problèmes

- IV de 24 bits : 2^{24} IV différents.
- IV change à chaque trame.
- 99% de chances d'avoir le même IV après 12 000 trames.
  - Quelques secondes pour 1 Ko à un débit de 11 Mbit/s.
- IV codé en clair dans la trame : espion sait quand il y a répétition.

- Préférer WPA, ou WPA2 (802.11i).

Cours 5 : Sécurité

Firewalls

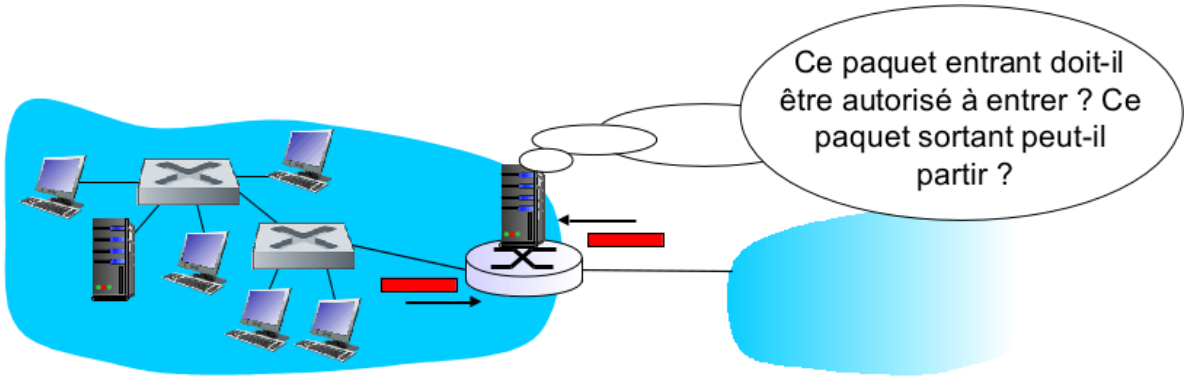


- Isole le réseau d'une organisation (les gentils) d'Internet (les méchants, le reste), en autorisant certains paquets à passer et en en bloquant d'autres.
- Comme dans les château forts, souvent qu'une seule entrée entre l'intérieur et l'extérieur (contrôle au pont-levis !).

Firewalls : pourquoi

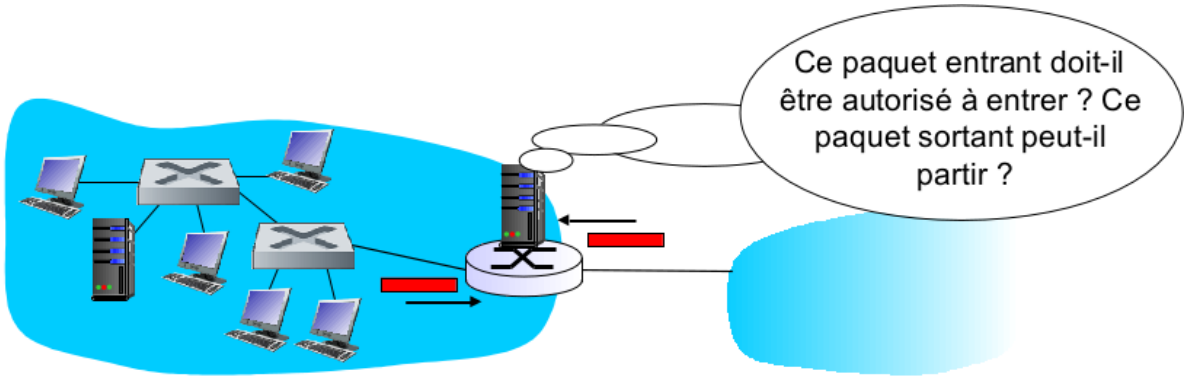
- Protection contre les dénis de service.
- Autorise l'accès au réseau interne à seulement certaines personnes.
- Trois types de firewalls :
  - Filtre de paquets sans état (couche réseau).
  - Filtre de paquets avec état (idem).
  - Passerelle d'applications (couche application).

Filtrage de paquets sans état



- Paquet par paquet, le routeur décide.
- Selon :
  - L'IP source, l'IP destination.
  - Le port TCP/UDP source, le port destination.
  - Le type de message.
  - Les bits ACK/SYN de TCP.
  - ...
- Liste de conditions...

Filtrage de paquets sans état



- Paquet par paquet, le routeur décide.
- Selon :
  - L'IP source, l'IP destination.
  - Le port TCP/UDP source, le port destination.
  - Le type de message.
  - Les bits ACK/SYN de TCP.
  - ...
- Liste de conditions...
- Par exemple, on peut décider de bloquer tous les segments TCP sauf ceux portant sur le port 80.
  - Évite toute intrusion étrangère et fuite vers l'extérieur.

Filtrage de paquets sans état - autres exemples

1. Bloquer les datagrammes entrant et sortant avec pour port source ou destination 23.
   - Résultat : bloque toutes les connections Telnet.
2. Bloquer les connexions TCP avec bit ACK à 0.
   - Résultat : empêche les clients extérieur de faire des connexions TCP avec les clients interne, mais autorise les clients interne à se connecter à l'extérieur.

Filtrage de paquets avec état

- Sans état : autorise des paquets sans “sens”.
  - Ex : Port destination : 80, bit ACK=1 mais pas de connexion TCP établie...
- Avec état : conserve le statut de chaque connexion TCP.
- Permet de déterminer si un paquet “a du sens”.
- Peut éteindre des connexions inactives.

Passerelles d'application

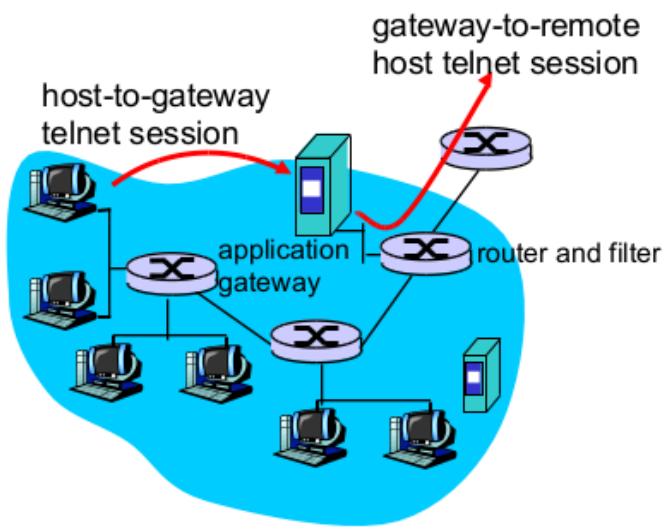
- Filtrage précédent par adresse IP.
- Peut vouloir filtrer par "employés" (procédure d'identification).
  - Données contenues dans la couche application.

Passerelles d'application

- Filtrage précédent par adresse IP.
- Peut vouloir filtrer par "employés" (procédure d'identification).
  - Données contenues dans la couche application.
- Passerelles d'application : un serveur spécifique que toutes les données d'applications doivent traverser avant de quitter ou d'entrer dans le réseau.
- Décision de la passerelle d'application selon les données.

Passerelles d'application - Exemple

- Veut permettre à certains utilisateurs d'établir des connexion Telnet avec l'exterieur, et bloquer tous les clients externes.
- On utilise une passerelle + un filtre de paquet sur routeur.
  - Le filtre bloque toutes les tentatives de Telnet sauf celles venant de la passerelle :
  - Un utilisateur doit d'abord passer par la passerelle (qui lui demande id+pwd et l'autorise uniquement si...).
  - La passerelle ouvre ensuite une connexion telnet avec l'extérieur et relai les données avec le client (client/serveur).
    - Nécessaire pour passer le filtre...



Passerelles d'application

- Souvent utilisé en entreprise pour HTTP, pour l'e-mail...
- Proxy-cache est une passerelle...

- Perte de performances.
- Effort de configuration (par ex. IP du proxy dans les navigateurs...).

Limites

- Compromis entre niveau de communication autorisé et sécurité.
- Filtre par IP/port : possible de manipuler son IP ou les ports.
  - Doit faire des règles sans exception (par ex. bloquer tout UDP) pour une sécurité plus grande.
- Bug logiciel sensible aux attaques sur les passerelles possible...

Conclusion

- Techniques de base.
  - Cryptographie (symétrique, clé publique).
  - Intégrité de messages.
  - Authentification.
- Plusieurs scénarios.
  - E-mail sécurisé.
  - SSL.
  - WEP.

Pseudos

http://www.quizzoodle.com/session/
be555ddd15894ee7a522f6db07c91677