Cybersécurité, une initiation: Mécanismes de sécurité

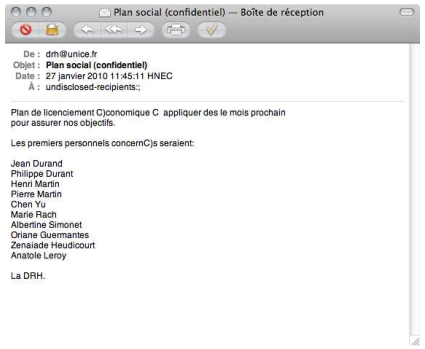
Bruno Martin

Université Côte d'Azur

M2 MIAGE SIRIS

Contenu

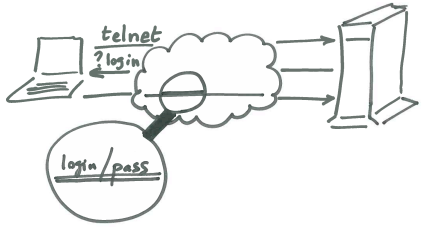
Illustration



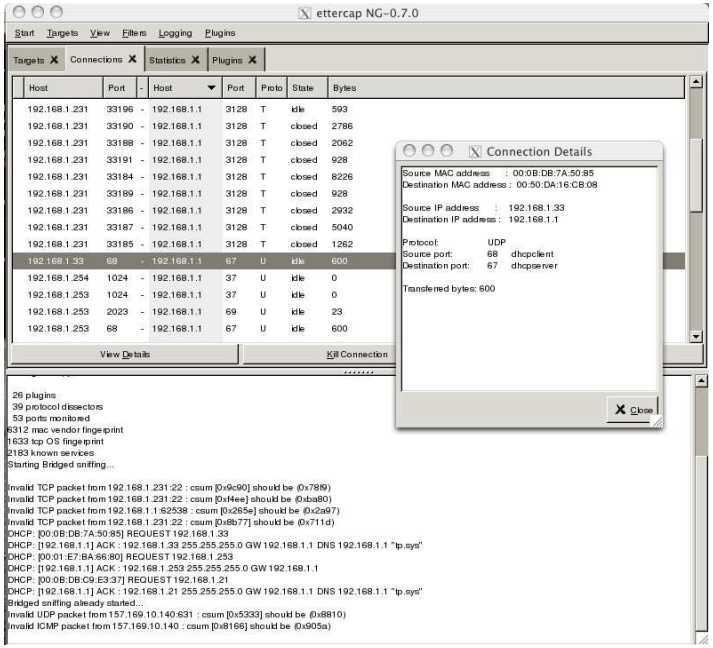
- L'information est publique ?
- Provient-elle de la DRH ?
- A-t-elle été modifiée ?

Explication

- Attaque passive
- Vol d'information
- Malveillance
- Possible avec :
  - telnet
  - pop
  - imap
  - http



Ettercap



Ettercap- description

- suite for MIM attacks on LAN
- features sniffing of live connections, content filtering on the fly and many other interesting tricks
- supports active and passive dissection of many protocols (even ciphered ones) and includes many features for network and host analysis

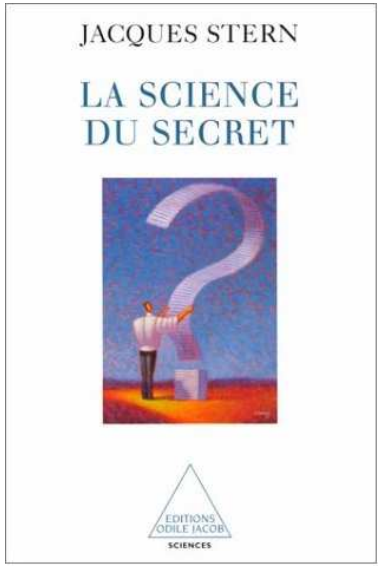
Plan

Cryptographie
- Age artisanal
- Age technique
- Distribution de clés
- Chiffres à clé publique
- Signatures numériques
- Hachage cryptographique
- Certification
- Identification & authentification

Protocoles sécurisés
- SSL
- IPSEC

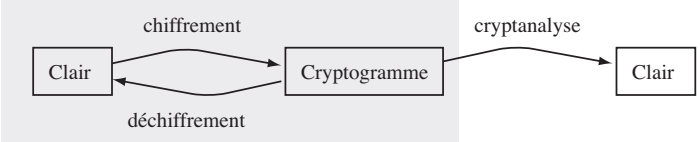
Historique

3 âges de la crypto. pour J. Stern :
- artisanal : modifier l'écriture
- technique : machines à chiffrer
- paradoxal : cryptographie à clé publique



Cryptologie = cryptographie + cryptanalyse

science de la communication en présence d'adversaires.



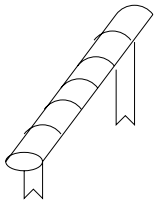
- **chiffrer** un **clair**→ **cryptogramme** (**confidentialité**).
- Destinataire légitime **déchiffre** le cryptogramme → clair.
- **cryptanalyste** ne peut **décrypter** le cryptogramme.

Âge artisanal – César

le clair `toute la gaule` devient `WRXWH OD JDXOH`.

(`A` devient `d`, `B` devient `e`...) **Substitution**

Transposition : scytale



Transposition simple à tableau

A partir d'une phrase clé, on définit une clé numérique :

T	R	A	N	S	P	O	S	I	T	I	O	N	S	I	M	P	L	E
18	14	1	8	15	12	10	16	3	19	4	11	9	17	5	7	13	6	2

On chiffre, «le chiffrement est l'opération qui consiste à transformer un texte clair, ou libellé, en un autre texte inintelligible appelé texte chiffré ou cryptogramme» [2].

18	14	1	8	15	12	10	16	3	19	4	11	9	17	5	7	13	6	2
l	e	c	h	i	f	f	r	e	m	e	n	t	e	s	t	l	o	p
é	r	a	t	i	o	n	q	u	i	c	o	n	s	i	s	t	e	à
t	r	a	n	s	f	o	r	m	e	r	u	n	t	e	x	t	e	c
l	a	i	r	o	u	l	i	b	e	l	l	é	e	n	u	n	a	u
t	r	e	t	e	x	t	e	i	n	i	n	t	e	l	l	i	g	i
b	l	e	a	p	p	e	l	é	t	e	x	t	e	c	h	i	f	f
r	é	o	u	c	r	y	p	t	o	g	r	a	m	m	e			

On prend ensuite par blocs de 5 lettres les colonnes prises dans l'ordre défini par la clé.

Âge technique – Enigma

le clair `alles in ordnung` devient `EDCGZVRRIOVRAY`

Sécurité parfaite – Vernam 1917

Ce one-time pad est-il un chiffre «parfait» ?

A et *B* partagent une suite aléatoire de *n* bits : la clé secrète *K*.
A chiffre *M* de *n* bits en $C = M \oplus K$.
B déchiffre *C* en $M = K \oplus C$.

Exemple

$M = 0011$, $K = 0101$
$C = 0011 \oplus 0101 = 0110$
$M = K \oplus C$.

Non-réutilisation : à chaque nouveau message, engendrer une nouvelle clé.

Chiffres produits et itérés [3]

Amélioration : combiner substitutions et transpositions.

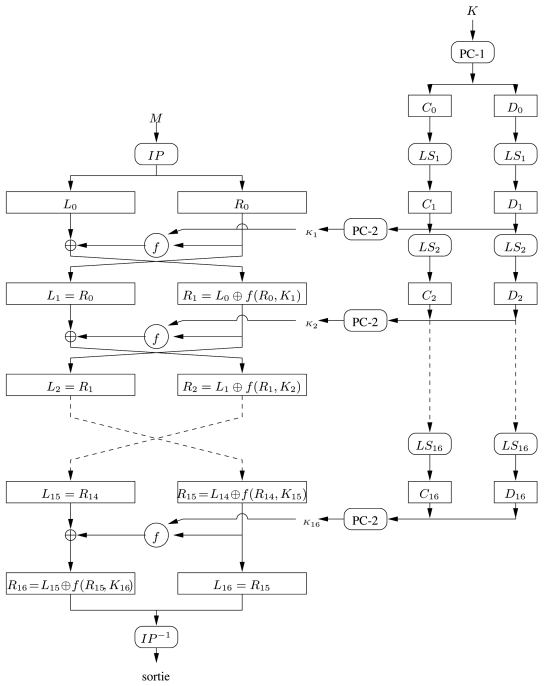
Un chiffre est **itéré** si le chiffré est obtenu par applications itérées d'une fonction de tour. A chaque tour, on combine le texte d'entrée avec une clé de tour.

Définition

Dans un chiffre itéré à r tours, le chiffré est calculé par application itérée au clair d'une ***fonction de tour*** *g t.q.*

$$C_i = g(C_{i-1}, K_i) \quad i = 1, \ldots, r$$

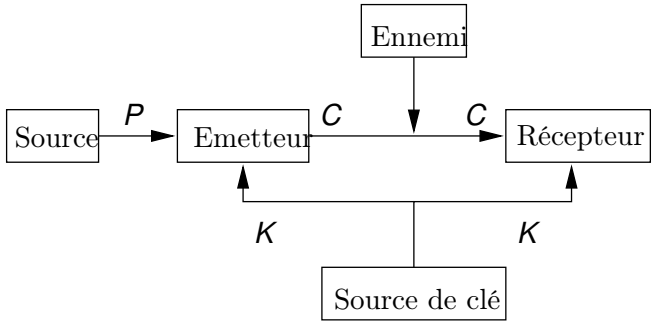
où C_0 est le clair, K_i une clé de tour et C_r le chiffré.
Déchiffrement en inversant l'équation précédente : pour une clé fixée K_i, g doit être inversible.



Le résultat

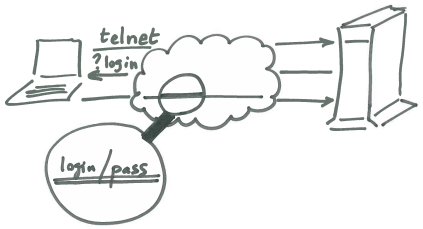
Chiffre à clé secrète

Modèle de Shannon pour le secret [3] :



Retour à l'exemple

- Attaque passive
- Vol d'information
- Malveillance
- **Plus possible**
- Distribution des clés ?



Tentative

Pré-distribuer une clé à chaque couple d'utilisateurs dans OS...

Petit calcul :

- 2 utilisateurs : 2 clés
- 4 utilisateurs : 6 clés
- n utilisateurs : $\frac{1}{2}\binom{n}{2}$

Environ 4.10^9 machines connectées...
Nombre de clés....
Mémoire pour les stocker : ...
Evolution ?

Les engendrer et les transmettre
Comment ?

Protocole de Diffie-Hellman

Procédé qui permet d'établir une clé partagée entre plusieurs entités de telle sorte qu'aucune d'entre elle ne puisse établir sa valeur par avance.
On cherche une solution qui permet à deux entités :

- qui ne se sont jamais rencontrés
- qui ne possèdent pas d'information partagée

de construire une clé secrète commune

- connue d'eux seuls
- inconnue de quiconque, même d'un indiscret qui écouterait leurs communications.

L'idée
Imaginer une solution facile à calculer pour les utilisateurs légaux et difficile pour un indiscret : fonction **à sens unique**.

Mise en accord par Diffie Hellman [4]

- **Etape préliminaire**
  - On choisit q un grand premier
  - On choisit a, $1 < a < q$
- **Les clés :** Chaque utilisateur U :
  - choisit aléatoirement X_U, $1 < X_U < q$ conservée secrète
  - publie $Y_U = a^{X_U} \mod q$

A et B construisent une clé commune avec : Y_A et Y_B.

- A calcule $K = Y_B^{X_A} \mod q$
- B calcule $K = Y_A^{X_B} \mod q$

A et B ont alors une clé (secrète) commune K :

$$Y_B^{X_A} \equiv (a^{X_B})^{X_A} \equiv a^{X_B X_A} \equiv Y_A^{X_B} \mod q$$

Chiffres à clé publique

Invention de Diffie et Hellman [1] ; phrase prophétique :

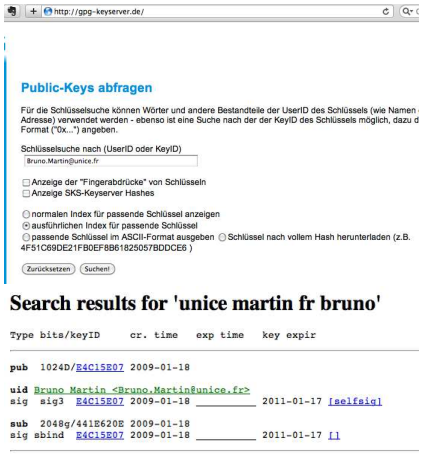
> *Nous sommes aujourd'hui à l'aube d'une révolution en cryptographie.*

Idée géniale : asymétrique ; chiffrement $\neq$ du déchiffrement.

- chiffrement par clé **publique**.
- déchiffrement avec clé **privée**.

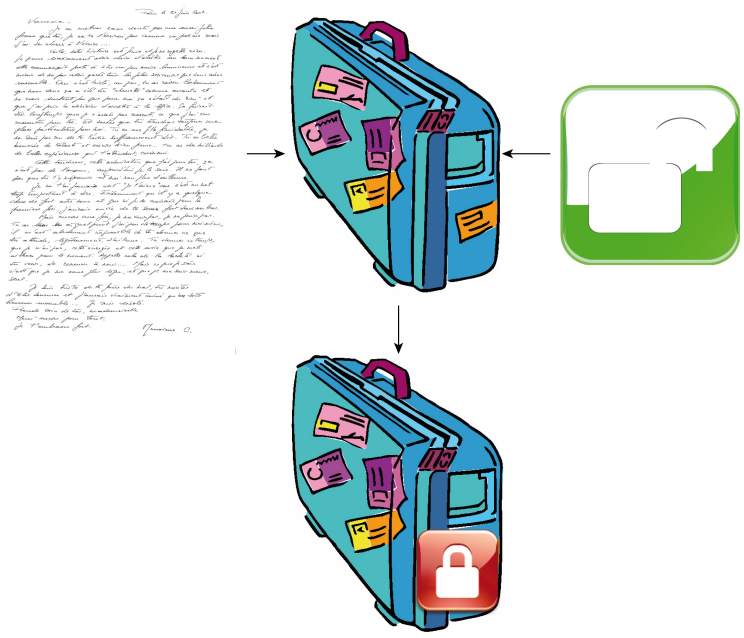
Les clés sont... publiques

Enfin, celles qui servent à chiffrer ; elles sont dans un annuaire :

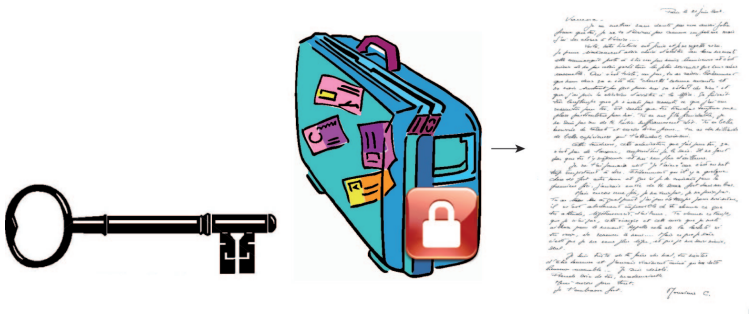


Bruno.Martin@unice.fr

Envoi d'un message



Réception d'un message



Paire de clés

Clé privée

$\{.\}_{SK}$

Clé publique

$\{.\}_{PK}$

Chiffre à clé publique

m → $\{m\}_{PK_B}$ → c → c → $\{c\}_{DK_B}$ → m

PK_B

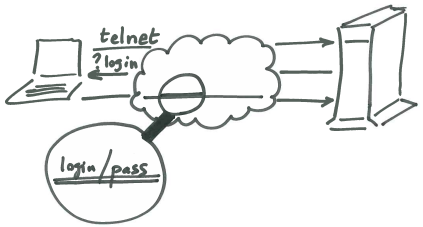
PK_B

DK_B

Annuaire de clés publiques

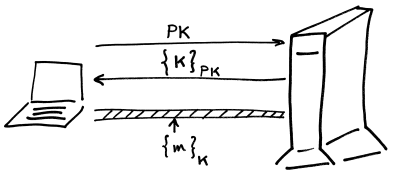
Retour à l'exemple

- Attaque passive
- Vol d'information
- Malveillance
- **Plus possible**
- Distribution clés par PKC

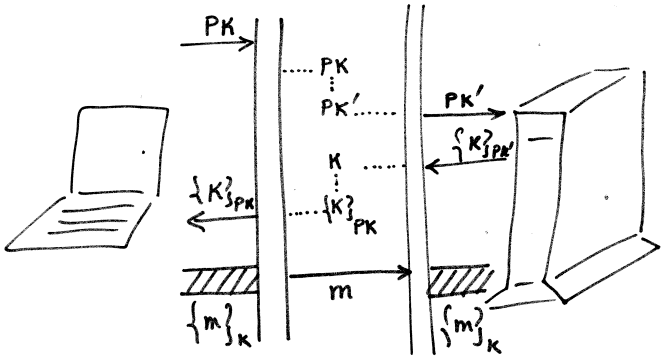


La solution ?

- $\{.\}_K$ pour chiffrer
- K transmis avec $\{K\}_{PK}$
- On a fini ?



Nouvelle attaque !



Man In the Middle

Pourquoi le chiffrement hybride ?

A disadvantage of asymmetric ciphers over symmetric ciphers is that they tend to be about "1000 times slower." By that, I mean that it can take about 1000 times more CPU time to process an asymmetric encryption or decryption than a symmetric encryption or decryption.

Synthèse

Le chiffrement

- garantit la confidentialité
- pas l'authentification
- d'autres attaques possibles

Empêcher l'attaque MIM...

Assurer l'authentification...

Avec les signatures

Objectif des signatures

- Seul l'expéditeur doit pouvoir signer
- N'importe qui peut vérifier la signature
- La signature dépend uniquement :
  - de l'identité de l'expéditeur
  - du message
- Garantit :
  - authentification de l'expéditeur
  - integrité du message

Signature

Principe : échanger les rôles de *pk* et de *sk*

- signature (privée) notée **sig** qui, pour la clé *sk*, retourne une signature *s* d'un message *m* ;

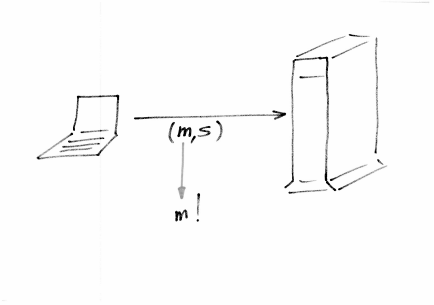
$$\text{sig}_{sk}(m) = \{m\}_{sk} = s$$

- vérification notée **ver** qui, à une clé publique *pk* et pour tout couple message/signature (m, s) vérifie que la signature correspond bien au message.

$$\text{ver}_{pk}(m, s) = \begin{cases} \text{vrai si } s = \text{sig}_{sk}(m) \Leftrightarrow \{s\}_{pk} = m \\ \text{faux si } s \neq \text{sig}_{sk}(m) \end{cases}$$

Inconvénients

- On « voit » *m*
- Perte de confidentialité
- Taille *s* $\propto$ taille *m*



Diminuer la taille de la signature...

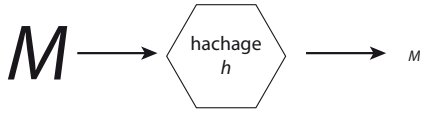
Utiliser le hachage

Hachage

h calcule

$$z = h(M)$$

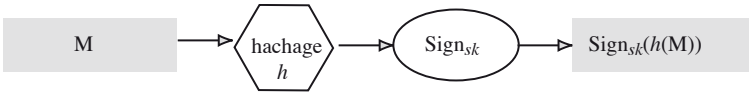
M de taille arbitraire,
z empreinte de taille fixe.



h est **à sens unique**, i.e.

- $h(M)$ rapide à calculer
- z difficile à inverser.

Signature avec hachage



Bob envoie M signé.
Clés : $(pk, \underline{sk})$
Il calcule

1. $h(M)$
2. $s = \{h(M)\}_{sk}$

Il envoie (M, s).

N'importe qui :

- reçoit (M, s)
- récupère pk de Bob
- vérifie :
  1. $z = h(M)$
  2. s signe $M \Leftrightarrow z = \{s\}_{pk}$

h connue de tout le monde (~~`md5`~~, `SHA-1`, `SHA-3` aka Keccak).

Rappel

On voulait contrer l'attaque MIM et assurer l'authentification.

But :

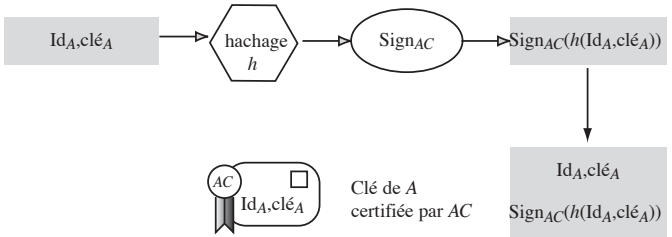
- transmettre une clé publique pk
- garantir la relation entre Id et pk

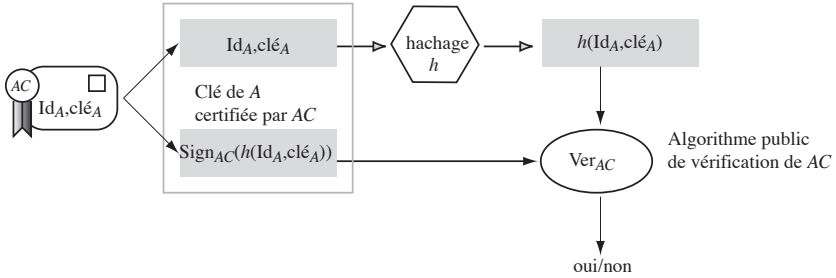
Comment faire ?

Principe

Une autorité va **garantir** la relation (Id, pk)

Certification & Vérification



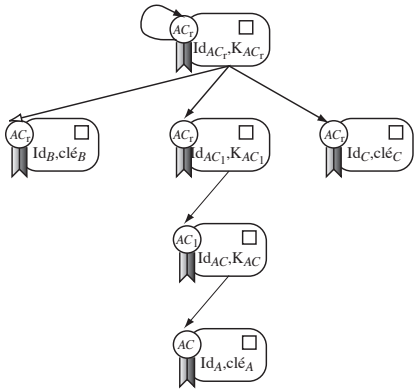


Paradoxe

Comment connaît-on l'algorithme de vérification de l'autorité de certification ?

Chaîne de certification

Une AC certifie une autre AC. Bob remonte une chaîne de certification jusqu'à une AC en qui il a confiance.



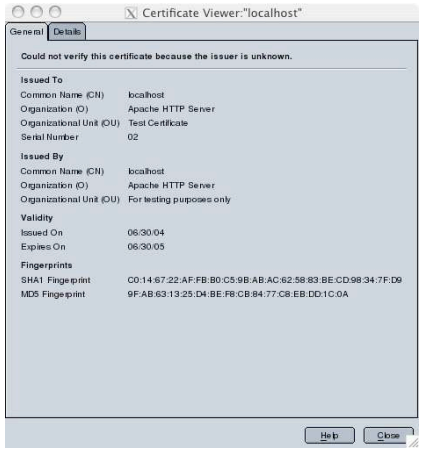
Mauvaise AC : attaques possibles (2011 : DigiNotar)

Création d'une AC « racine »

Problème : il faut une AC « racine ».
Certificat auto-délivré. L'entité qui délivre le certificat est identique au sujet certifié.

- confiance : large distribution de la clé publique de l'AC.
- possible de se déclarer comme AC « racine »

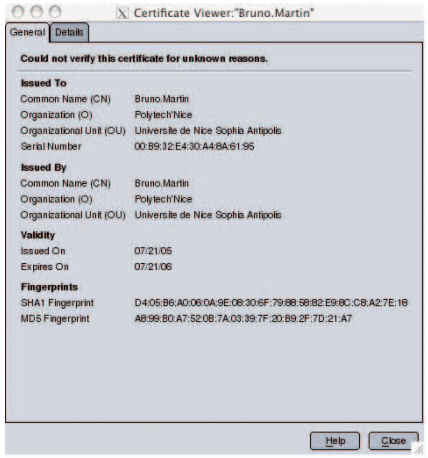
Rupture dans la chaîne



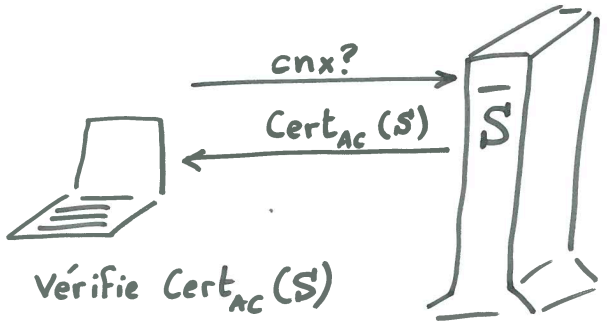
Une attaque MIM de plus haut niveau

Porte sur la transmission d'un certificat dont l'AC n'est pas connue. C'est le cas, par exemple, d'un certificat "auto-délivré".

Certificat original et celui falsifié



Transmettre une clé publique



Rapide synthèse

Ce qu'on sait faire pour le moment :

- Transmettre une clé publique
- Transmettre une clé secrète
- Sécuriser un canal

Encore une attaque

- DOS
- empêcher la communication
- transmettre des $\text{Cert}_{AC}(S)$ erronés
- client passe son temps à faire des vérifications inutiles

Comment s'assurer de l'identité de S ?

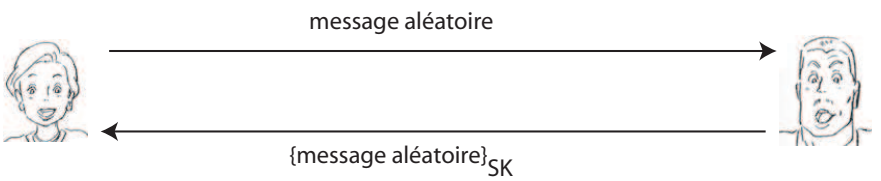
Identification et authentification

- **identification** affirmation d'une identité : "je suis Bruno"
- **authentification** : vérifie l'identité "je peux prouver que je suis Bruno"

Service d'authentification vérifie l'identité à différents niveaux :

- applicatif : `http, ftp`
- transport : `ssl, ssh`
- réseau : `ipsec`
- transmission : `pap, chap` (qui utilisent `md5`)

Exemple d'authentification « asymétrique »



- Sans calcul h(msg-aléa), KPA possible : (m/$\{\text{m}\}_{SK}$) connus, ...
- **Hyp :** Alice connaît la clé publique de Bob au préalable.

Authentification

C'est le moyen pour Alice de vérifier l'identité de Bob.
On appelle *sk* la clé privée de Bob et *pk* sa clé publique.

$$\begin{array}{l|l} A \to B & r = \text{un message aléatoire} \\ B \to A & c = \{r\}_{sk} \end{array}$$

Signer un message aléatoire r fourni par un tiers et le réexpédier peut s'avérer dangereux.
Une idée serait d'utiliser une fonction de hachage h afin que Bob signe en chiffrant $h(r)$. Mais le danger persiste.

Authentification

Mieux vaut que Bob signe un message de son cru

$$\begin{array}{l|l} A \to B & \textit{"Bonjour, est-ce Bob ?"} \\ B \to A & m = \textit{"Alice, je suis bien Bob"} \\ & c = \{h(m)\}_{sk} \end{array}$$

On rappelle que $\{.\}_{sk}$ est la signature de Bob.

Identification

Alice ne connait pas forcément PK_B. Comment informer sûrement quelqu'un de sa clé publique ?

$$\begin{array}{l|l} A \to B & \textit{"Bonjour"} \\ B \to A & \textit{"Bonjour, je suis Bob. Voici ma clé publique"}\ pk \\ A \to B & \textit{"Prouve-le."} \\ B \to A & m = \textit{"Alice, c'est bien Bob"} \\ & c = \{h(m)\}_{sk} \end{array}$$

N'importe qui peut se faire passer pour Bob aux yeux d'Alice, en fournissant sa propre clé publique.

Transmettre un certificat

Certificat garantit la relation entre une identité et clé publique.

$$\begin{array}{l|l} A \to B & \textit{"Bonjour"} \\ B \to A & \textit{"Bonjour, je suis Bob. Voici mon certificat"}\ cert_B \\ A \to B & \textit{"Prouve-le."} \\ B \to A & m = \textit{"Alice, c'est bien Bob"} \\ & c = \{h(m)\}_{sk} \end{array}$$

Melchior pourrait se substituer à Bob dans les trois premiers échanges, mais échouera au delà.

Echanger un secret

La communication par clés publiques est coûteuse, une fois l'authentification finie on échange une clé pour utiliser un chiffre symétrique.

$A \to B$	*"Bonjour"*
$B \to A$	*"Bonjour, je suis Bob. Voici mon certificat"* $cert_B$
$A \to B$	*"Prouve-le"*.
$B \to A$	$m =$ *"Alice, c'est bien Bob"*
	$c = \{h(m)\}_{sk}$
$A \to B$	*"Ok Bob, voici notre secret :"*
	$s = \{secret\}_{pk}$
$B \to A$	$m' = \{\text{message de Bob}\}_{secret}$

Attaque

L'homme du milieu Melchior peut s'interposer dans les 5 premiers échanges. Arrivé au sixième, il peut brouiller le message de Bob, quitte à ne pas envoyer un message très intelligible à Alice :

$B \to M$	$m' = \{\text{message de Bob}\}_{secret}$
$M \to A$	altération de m'

Alice n'a aucune certitude quant à l'existence de Melchior, même si elle trouve suspect le dernier message de Bob.

SSL

Pour éviter cette incertitude, mieux vaut utiliser un MAC :

$M = h($ un message de Bob, *secret*$)$

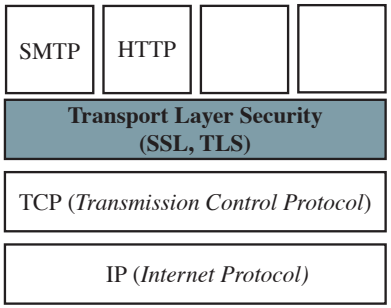
$A \to B$	*"Bonjour"*
$B \to A$	*"Bonjour, je suis Bob. Voici mon certificat"* $cert_B$
$A \to B$	*"Prouve-le"*.
$B \to A$	$m =$ *"Alice, c'est bien Bob"*
	$c = \{h(m)\}_{sk}$
$A \to B$	*"Ok Bob, voici notre secret :"*
	$s = \{secret\}_{pk}$
$B \to A$	$m' = \{\text{message de Bob}\}_{secret}$
	$M = h($message de Bob, *secret*$)$

Melchior peut pertuber ce qu'il veut, M aura au moins l'avantage d'en avertir le destinataire.

Sécurité de TCP

Protocoles pour sécuriser TCP :

- **Secure Socket Layer**
- **Transport Layer Security** standardisé par l'IETF



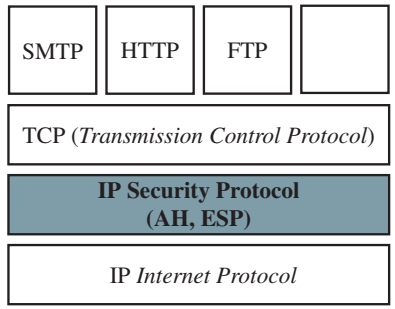
Synthèse

- SSL est un exemple de **protocole** : ensemble de règles permettant d'établir une communication entre deux entités
- fournit certains **services** de sécurité
  - identification, authentification
  - confidentialité
  - intégrité
- mis en place par les **mécanismes** de sécurité
  - chiffrement, signature
  - contrôle d'accès
  - hachage
  - certification

Sécurité d'IP

Protocoles pour sécuriser IP :

- **IPSEC**
- **SSH**



Bibliographie

W. Diffie and M.E. Hellman.
New directions in cryptography.
IEEE Trans. on Inform. Theory, 22(6) :644–654, 1976.

D. Kahn.
La guerre des codes secrets.
InterEditions, 1980.

L.R. Knudsen.
Block ciphers – a survey.
In Springer Verlag, editor, State of the art in applied cryptography, number 1528 in LNCS, pages 18–48, 1998.

D. Stinson.
Cryptographie, théorie et pratique.
International Thomson Publishing, 1995.