INF111 - Initiation aux Réseaux

Licence 3 Informatique de Bordeaux - 2005-2006

A. Pêcher
pecher@labri.fr
http://www.labri.fr/perso/pecher

Bureau 360
Laboratoire Bordelais de Recherche en Informatique
Université de Bordeaux 1

Licence 3 Info - Bordeaux

Plan du cours (12/12) - vendredi 16 décembre 2005

Sécurité des réseaux

1. Cours 12 : sécurité des réseaux
   - Préambule
   - Algorithmes à clef symétrique
   - Algorithmes à clef publique
   - Signatures numériques
   - Par chiffrage du message
   - Avec des condensats de message
   - Certificats
   - Quelques applications

Plan

- confidentialité : protéger les informations de ceux qui ne sont pas autorisés à les connaitre (ex. cryptage) ;
- authentification : s'assurer de l'identité de l'interlocuteur, avant de révéler des information confidentielles ;
- non-répudiation : s'assurer qu'un message validé par les deux parties ne peut être remis en cause ultérieurement (ex. par apposition de signatures) ;
- contrôle d'intégrité : s'assurer qu'un message n'a pas été modifié (ou créé) lors de l'acheminement.

Introduction (2)
Au niveau de chaque couche

La sécurité doit être prise en charge au niveau de chaque couche :

- couche physique : ex. mise en place d'une surveillance des lignes de transmissions ;
- couche liaison : ex. chiffrage sur un segment (entre 2 routeurs) - inconvénient : déchiffrage au niveau de chaque routeur, mais néanmoins utile (ex. WIFI) ;
- couche réseau : ex. filtrage des paquets valides (murs pare-feu = firewalls) ;
- couche transport : connexions cryptées en totalité, de bout-en-bout ;
- couche applications : gestion de l'authentification et de la non-répudiation.

Introduction (3)
Deux principes fondamentaux de la cryptographie

- redondance : les messages doivent contenir un certain degré de redondance. Un minimum de redondance est nécessaire pour que le destinataire puisse distinguer un vrai message, d'un message créé par un intrus, par une simple inspection de son contenu (ex. ajout d'un code CRC). Cependant trop de redondance facilite le cassage d'un code ;
- fraîcheur : mettre en oeuvre une méthode permettant de déceler le rejeu d'anciens messages (par ex. en incluant une date de validité (courte)).

Plan

Algorithme à clef symétrique

Utilise la même clef pour le chiffrement et le déchiffrement.

- développé par IBM, adopté en 1977 par le gouvernement américain (documents non-classifiés) ;
- utilise une clef de 56 bits ;
- chiffrage par blocs de 64 bits ;
- Algorithme
  - transposition initiale ;
  - 16 itérations réversibles paramétrées par la clef ;
  - permutation des 32 bits de gauche avec ceux de droite ;
  - transposition inverse.
- Commentaires : clef initiale de 128 bits, réduits à 56 à la demande de la NSA. Repose sur un nombre "privé" Lucifer. "Cassé" en 1977. EN 1979, évolution "Triple DES" (3xDES avec 2 clés).

AES
Advanced Encryption Standard - Rijnadael

- issus d'un concours en 1998 : vainqueurs Rijndael, Serpent, Twofish, RC6, MARS ;
- blocs de taille 128 bits ;
- clefs de 128, 192 ou 256 bits ;
- fondé sur la théorie des corps de Galois ;
- Algorihtme :
  - un bloc de 128 bits = 4 lignes de 4 octets ;
  - 10 rondes :
    - Substitution octet par octet ;
    - Rotation de chacune des 4 lignes vers la gauche ;
    - Mélange de chacune des colonnrs, indépendamment des autres, via un calcul matriciel sur $GF(2^8)$;
    - XOR avec la clef de la ronde courante.
- rapide : chiffrement/déchiffrement de 700 Mo/s (processeur 2GHz).

Caractéristiques des algorithmes les plus courants

Algorithme	Auteurs	Clef	Commentaires
Blowfish	Bruce Schneier	$\leq$448 bits	Vieux et lent
DES	IBM	56 bits	Faible
IDEA	Massey et Xuejia	128 bits	Efficace, breveté
RC4	Ronald Rivest	$\leq$2048 bits	certaines clefs faibles
RC5	Ronald Rivest	128 à 256 bits	Efficace, breveté
Rijndael	Daemen et Rijmen	128 à 256 bits	Le meilleur
Serpent	Anderson, Biham et Knudsen	128 à 256 bits	Très fort
Triple DES	IBM	168 bits	Second
Twofish	Bruce Schneier	128 à 256 bits	Très fort très utilisé

Plan

Dans un algorithme à clef symétrique, la distribution des clefs est le point délicat : la personne qui chiffre le message et celle qui le déchiffre utilise la même clef.

Algorithme à clef publique

chaque utilisateur X possède une clef publique E_X, librement accessible et une clef privée D_X.

1. A chiffre son message avec E_B et l'envoi à B ;
2. B déchiffre le message avec D_B ;
3. et inversement...

Requis : il est excessivement difficile de déduire la clef privée D_B de la clef publique E_B.

RSA
Rivest, Shamir et Aldeman

Principe :

1. p et q deux grands nombre premiers ;
2. $n = p * q$ et $z = (p-1) * (q-1)$;
3. d choisi tels d et z soient premiers entre eux ;
4. e solution de l'équation $ed = 1 \pmod{z}$;
5. cryptage de $P < n$: $C = P^e \pmod{n}$;
6. décryptage : $D = C^d \pmod{n}$;

Commentaires :

- clef publique : (n, e) ;
- clef privée : (n, d) ;
- clefs d'au moins 1024 bits, assez lent, largement utilisé pour distribuer des clefs ;
- connaissant (n, e), pour trouver d, il faut factoriser n en $p * q$, pour déterminer z...

Plan

Signatures numériques
Principes

Une signature numérique doit permettre :

- au destinataire de vérifier l'identité de l'expéditeur ;
- d'empêcher l'expéditeur de nier ultérieurement le contenu du message ;
- d'empêcher le destinataire de fabriquer de toute pièce un tel message.

Deux familles :

- Signatures à clef symétrique ;
- Signatures à cléf publiques.

Plan

Signatures à clef symétrique

Utilise une autorité centrale *BB* (tiers de confiance) auquelle sont confiées l'ensemble des clefs privés (symétriques). *A* envoie le message crypté à *BB* qui le décode et le recode avec la clef de *B* et le relaie à *B*, avec une copie chiffrée avec la clef de *BB* :

- *BB* est garant de l'identité de *A* par apposition de la clef de *A* ;
- la copie créée avec la clef de *BB* est garante du contenu ;

Inconvénient : tout le monde doit accorder sa confiance à l'autorité centrale et tous les messages transitent par celui-ci.

Signatures à clef publiques

Si l'algorithme choisi est commutatif : $D(E(P)) = E(D(P)) = P$ (ex. RSA)

1. A encode le message P avec sa clef privée D_A, puis encode le résultat avec la clef publique E_B de B ;
2. B reçoit $E_B(D_A(P))$, le décode avec sa clef privée D_B et obtient donc $D_A(P)$.
3. B stocke $D_A(P)$ et le déchiffre avec la clef publique de A.

Commentaires :

- $D_A(P)$ est la preuve pour B de la validité du contenu du message, car il est construit avec la clef secrète de A ;
- si le secret autour de la clef de A est rompu ou si A change de clef, B n'a plus de preuve valide.

Plan

Condensats de message
Message Digest

Les deux méthodes de signature précédentes chiffrent le message tout entier. Pour garantir le contenu d'un message, il est plus adapté d'utiliser une fonction de hachage MD telle que :

1. pour tout P, MD(P) est facile calculer ;
2. connaissant MD(P), on ne peut pas retrouver P ;
3. à partir de P, on ne peut pas trouver P' tel que $P \neq P'$ et MD(P)=MD(P') ;
4. si P subit une modication (même infime) alors MD varie fortement.

Exemple avec des clefs publiques :

- A envoie P et $D_A(MD(P))$;
- B récupère $MD(P)$ en appliquent E_A la clef publique de A ;
- $D_A(MD(P))$ est la preuve de B, de l'envoi de A ;
- B vérifie que P n'a pas été modifié en calculant lui même MD(P).

MD5
par Ronald Rivest

Il est très utilisé pour s'assurer qu'un fichier n'a pas été corrompu lors d'un transfert (ex. images ISO de CDs). Ceci permet entre autres de garantir qu'un paquet Mandrake, Debian etc est valide avant de l'incorporer dans le système.

```
$ echo "greg is great" | md5sum
4344b3ba300c03cd4b3e771a1c9576a2  -
$ echo "Greg is great" | md5sum
82f42a50b06f6e4d62c6dcda5b46289a  -
```

SHA1
par la NSA

```
$ echo "greg is great" | sha1sum
ea4e0543e1093b0f7c7d5aa94cd02466dcca7f1b  -
$ echo "Greg is great" | sha1sum
98d1d74cc65c6a8f0576083e91cbf1e1e049afd9  -
```

Plan

Certificats
Transfert des clefs publiques

Dans la cryptographie à clef publique, *A* a besoin de récupérer la clef publique de *B*. Le risque est que *C* intercepte la communication et substitue sa clef publique en place de celle de *B*. Puis continue en tant qu'intermédiaire transparent (man in the middle) dans l'échange chiffré entre *A* et *B*.
Un certificat sert à transmettre une clef publique : il est délivé par un organisme certificateur CA. Il contient les informations suivantes :

- la clef publique de *B* ;
- des informations spécifiques sur *B* : par exemple, nom,prénom, adresse postale, adresse email etc...
- un condensat SHA-1 des 2 précédents points, signé avec la clef privée du CA.

Pour récupérer la clef publique de *B*, *A* déchiffre le certificat avec la clef publique du CA, et vérifie l'intégrité en calculant lui-même le SHA-1. *C* est incapable de créer un tel certificat car il ne connait pas la clef privée du CA, et les informations spécifiques de son propre certificat ne peuvent passer pour celles de *B*.

Le format d'un certificat a été standardisé sous le nom de X.509.
Principaux champs :

- Version : version de X.509 utilisée ;
- Numéro de série : identifiant unique du certificat ;
- Algorithme de signature ;
- Délivré par : nom X.500 de l'organisme certificateur ;
- Période de validité ;
- Clef publique ;
- Idéntité de l'émetteur ;
- Signature : signature du certificat authentifiée par la clef publique du CA.

Plan

IPsec
IP security

services à la carte : secret, intégrité des données, protection contre le rejeu. Ces services sont fondés sur la cryptographie à clef symétrique.

- orienté connexion : une connexion est appelée SA (Security Association). Les identificateurs de sécurité sont transportés dans les paquets.
- 2 modes :
  - transport : dans chaque paquet IP, l'entête IP-Sec est inséré juste après l'entête IP. Le champ protocole IP est modifié pour indiquer la présence de l'entête IP-Sec ;
  - tunnel : on ajoute un nouvel entête IP suivi de l'entête IP-Sec suivi de l'entête IP original. Utile si l'extrémité du tunnel n'est pas le destinataire. Inconvénient : surcoût de l'entête IP supplémentaire.

IPsec (2)
IP security

Entête IP-Sec : version ESP (Encapsulating Security Payload), la plus récente :

- deux mots de 32 bits :
  - index des paramètres de sécurité : identificateur de connexion (permet au destinataire de déterminer quelle clef utiliser comme clef partagée) ;
  - numéro de séquence : chaque paquet possède un numéro unique (la retransmission porte un numéro différent) ;

L'entête est suivi habituelllement d'un troisième mot le vecteur d'initialisation VI, utilisé lorsque les données sont chiffrées (sinon il est omis).
A la fin du paquet est ajouté une section HMAC (Hashed Message Authentification Code qui sert au contrôle d'intégrité : calcul du hachage sur le paquet + la clef partagée (non transmise).

Murs pare-feu
Firewalls

Un Mur pare-feu sert à filtrer :

- des paquets (le plus courant) : placé généralement au niveau d'un routeur. Inspection des paquets pour ne laisser passer que ceux qui sont valides selon des règles définies par l'administrateur réseau ; ex. filtrage en fonction du port de destination - pour des paquets à destination d'un serveur web, n'autoriser que les paquets vers les ports 80 (HTTP) et 443 (HTTPS) etc ...
- des applications : ex. filtre antispam au niveau d'un client de courrier électronique ...

Remarque : un mur pare-feu est mal adapté aux services qui utilisent des ports variables (ex. FTP, voix sur IP ...)

- chaque station partage une clef avec la station de base ;
- une fois enregistrée, les clefs demeurent stables plusieurs mois ;
- chiffrement basé sur RC4 ;
- principes :
  - on ajoute aux données un polynôme de contrôle CRC-32 ;
  - à partir de la clef partagée et d'un vecteur d'initialisation VI sur 24 bits, on génère un keystream (flot) ;
  - on effecture un XOR ;
  - on ajoute en entête le VI.
- décodage : récupération du VI, génération du keystream, XOR, contrôle du CRC.

Sécurité en WIFI (2)
WEP - Wired Equivalent Privacy

Commentaires :

- seulement 2^{24} possibilités pour le VI : une écoute sur plusieurs minutes permet de récupérer plusieurs paquets avec le même VI et la même clef ;
- permet de déterminer le keystream associé au VI ;
- ... et on recommence ...
- faux paquets valides faciles à constituer : contrôle CRC inefficace ;
- faille dans RC4.

Sécurité du courrier électronique
PGP - Pretty Good Privacy

- créé en 1991, par Phil Zimmermann ;
- prend en charge la confidentialité, l'authentification, les signatures numériques et la compression ;
- sources libres ;
- chiffrage avec IDEA (International Data Encryption Algorithm), utilise des clefs de 128 bits ;
- gestion des clefs avec RSA ;
- contrôle de l'identité avec MD5 ;

Sécurité du courrier électronique (2)
PGP - algorithme

1. calcul du hachage du message avec MD5 ;
2. chiffrage du hachage avec RSA (clef privée de A) ;
3. concaténation du message et du hachage chiffré ;
4. compression ZIP ;
5. A donne une clef de message aléatoire K_M ;
6. chiffrage du flux compressé avec IDEA en fonction de K_M et de la clef publique de B ;
7. conversion en base 64 et envoi.

SSL & HTTPS
Secure Sockets Layer

- se place entre la couche Transport et la couche Application ;
- 2 phases : Etablissement de la connexion & Utilisation de la connexion ;
- plusieurs algorithmes de cryptographie :
  - chiffrage avec Triple DES et contrôle de l'intégrité avec SHA-1 ;
  - chiffrage avec RC4 et contrôle avec MD5 (moins sûr, plus rapide).

HTTPS = HTTP sur SSL : le port utilisé est en général 43 (au lieu de 80).

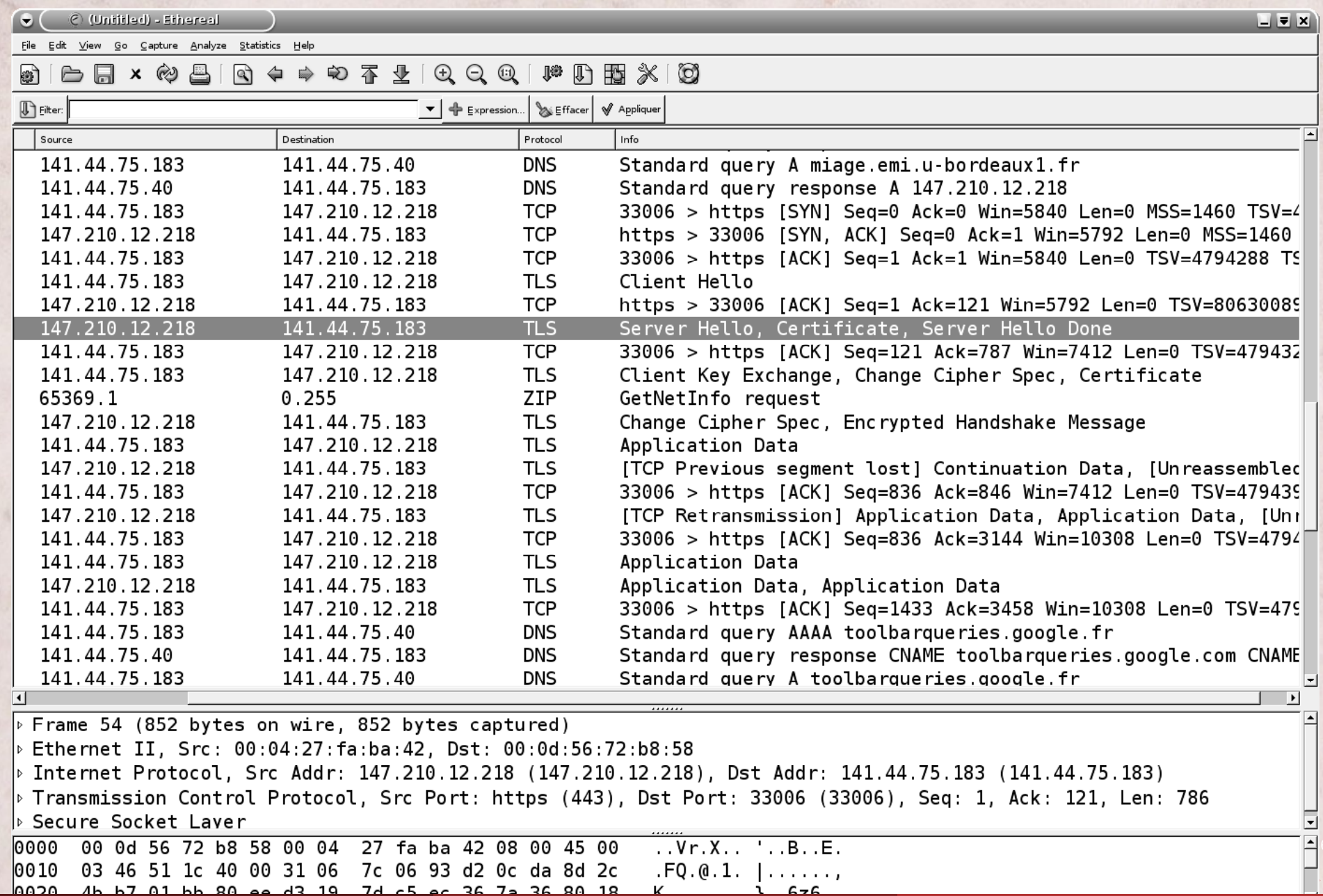
la page
https ://miage.emi.u-bordeaux1.fr/miageBx/stages/

```
<HTML>
<HEAD>
<TITLE>Les Stages a la MIAGe</TITLE>
<LINK Rel="stylesheet" type="text/css" href="./Styles/style2.css" ></HEAD>
<BODY>
<HR>
<p Class=titrePrincipal>Les Stages a la MIAGe</p>
<HR>

<TABLE ALIGN="center" BORDER="2" WIDTH="95%" bordercolor="white" cellpaddi
<TR Class=TexteNormal HEIGHT="35%">
         <TD WIDTH="45%" >

    <B> <A HREF="InfosGenerales/InfoGenerales.php">Informations G^^e9n^^e9
    <P>
     Les stages a la MiAge, Plan et contenu de ce site,
     Calendrier des stages, Calendrier des soutenances,
     News, Permanence, Exemple de convention de stage.<BR>
...
```

Ethereal en action



Ethereal en action (2)