

Plan du cours

- 1 Introduction SSL/TLS
- 2 Certification
 - Généralités
 - X.509
- 3 Protocole TLS
 - TLS Handshake
 - TLS Record
- 4 Attaques
 - BEAST

Plan du cours

- 1 Introduction SSL/TLS
- 2 Certification
 - Généralités
 - X.509
- 3 Protocole TLS
 - TLS Handshake
 - TLS Record
- 4 Attaques
 - BEAST

SSL/TLS

Secure Socket Layer

- Protocole (au niveau applicatif) de sécurisation des échanges sur Internet
- Développé par Netscape (SSL version 1, 2 et 3)
- Rebaptisé **Transport Layer Security** (TLS) par l'IETF suite au rachat du brevet à Netscape
- Actuellement, TLSv1.2 (RFC 5246 2008, RFC 6176 2011).

SSL/TLS

Secure Socket Layer

- Protocole (au niveau applicatif) de sécurisation des échanges sur Internet
- Développé par Netscape (SSL version 1, 2 et 3)
- Rebaptisé **Transport Layer Security** (TLS) par l'IETF suite au rachat du brevet à Netscape
- Actuellement, TLSv1.2 (RFC 5246 2008, RFC 6176 2011).
 - **SSL v2.0** (1995)
 - **SSL v3.0 & TLS v1.0**
 - TLS v1.1 & TLSv1.2

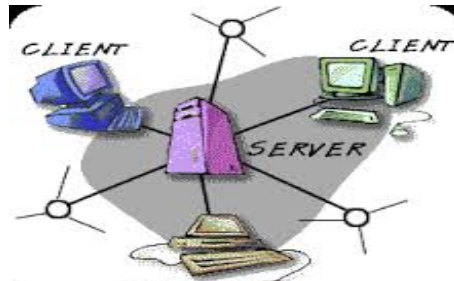
TLS 1.3

Objectifs

⇒ Établir un canal de communication sécurisé (chiffré) entre deux machines (type client-serveur) après une étape d'authentification.

Services

- **Authentification du serveur** (utilisation d'un **certificat numérique**)
- **Confidentialité**
- **Intégrité**
- optionnelle, **authentification** du client



Points forts

- sur-couche sécurisée des protocoles courants :
 - HTTP (HTTPS)
 - POP3/IMAP (POP3S/IMAPS)
 - ...
- SSL/TLS est transparent pour l'utilisateur.



Plan du cours

- 1 Introduction SSL/TLS
- 2 Certification
 - Généralités
 - X.509
- 3 Protocole TLS
 - TLS Handshake
 - TLS Record
- 4 Attaques
 - BEAST

Plan du cours

- 1 Introduction SSL/TLS
- 2 Certification
 - Généralités
 - X.509
- 3 Protocole TLS
 - TLS Handshake
 - TLS Record
- 4 Attaques
 - BEAST

Motivation

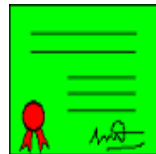
- ⇒ Garantir la provenance de la clef publique.
- ⇒ Annuaire de clef publiques **certifiées** par une **autorité** de "confiance".
 - L'**autorité** de "confiance" **signe** l'identité d'Alice ainsi que sa clef publique.



Certificat

Définition

- Un **certificat** permet d'associer une clef publique à une entité (une personne, une machine, . . .) pour en assurer la **validité**.
- Le certificat est un **lien** entre l'entité physique et l'entité numérique.
- Il est délivré par une **autorité de certification**
 - **Certification Authority (CA)**.

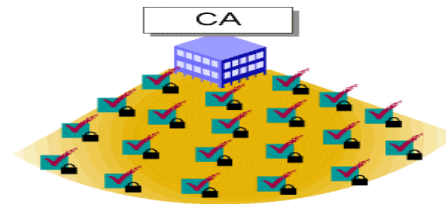


Autorité de certification – (I)

- C'est un tiers de confiance

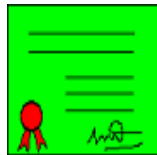
Mission

- Génère les certificats
- Stocke et distribue les certificats
- Emet des listes de révocation
 - CRL : Certificate Revocation List

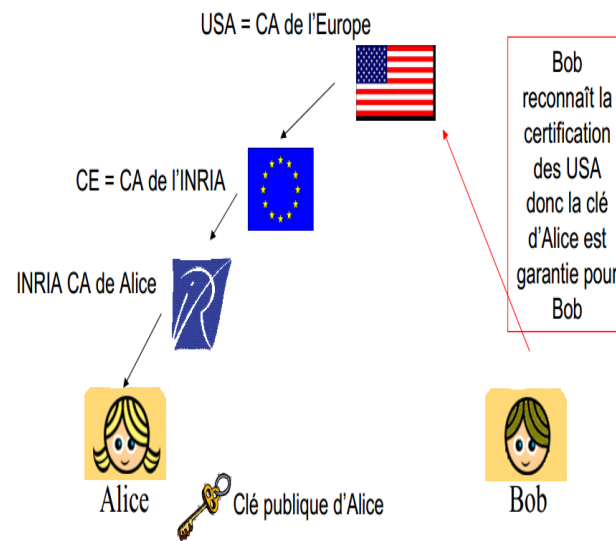


Autorité de certification – (II)

- Une CA possède elle-même un certificat.
 - Le certificat d'une CA peut-être auto-signé : **autorité racine**.
 - Le certificat peut avoir été émis par une autre CA (**relation hiérarchique**).
- Il existe des CA :
 - privées (intranet d'une entreprise, ...)
 - organisationnelles (CNRS, ...),
 - commerciales (Thawte, Verisign, ...),



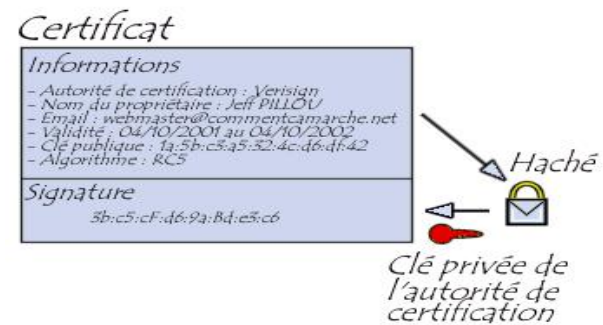
Chemin de certification



Structure d'un certificat

Ce sont des petits fichiers divisés en deux parties :

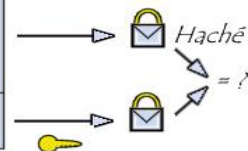
- La partie contenant les **informations**
- La partie contenant la **signature de l'autorité de certification**



Vérification

Certificat

Informations
- Autorité de certification : Verisign
- Nom du propriétaire : Jeff PILLOU
- Email : webmaster@commentcamarche.net
- Validité : 04/10/2001 au 04/10/2002
- Clé publique : 1a:5b:c5:a5:52:4c:d6:d1:42
- Algorithme : RSA
Signature
5b:c5:cF:d6:9a:8d:e3:c6



Déchiffrement à l'aide
de la clé publique de
l'autorité de certification

Plan du cours

- 1 Introduction SSL/TLS
- 2 Certification
 - Généralités
 - X.509
- 3 Protocole TLS
 - TLS Handshake
 - TLS Record
- 4 Attaques
 - BEAST

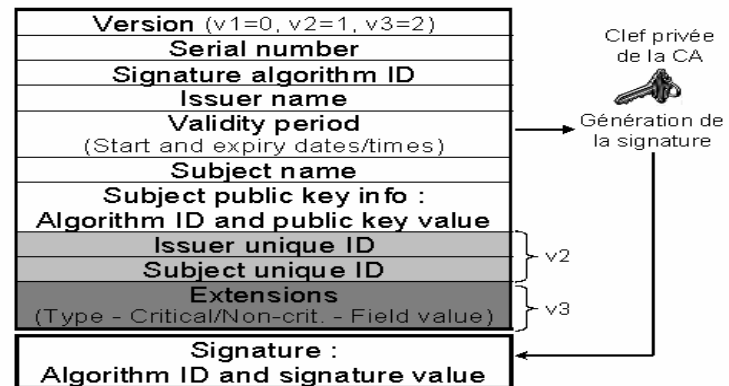
Certificats X.509

- X.509 est la **norme** (1998) de certificat la plus utilisée.
 - Forme des certificats & algorithme pour la validation du **chemin de certification**.
- Il repose sur un système hiérarchique de CA.

Format d'un Certificat X.509 – (I)

- Version (1, 2, ou 3).
- Numéro de série (*unique, permet d'identifier de certificat de manière unique*)
- Algorithme de signature du certificat
- Nom du signataire du certificat (*qui a généré le certificat*)
- Validité (dates limite : pas avant, pas après)
- Détenteur du certificat
- Informations sur la clef publique :
 - Algorithme à clef publique, Clef publique proprement dite
- Identifiant unique du signataire (optionnel, X.509 version 2)
- Identifiant unique du détenteur du certificat (optionnel, X.509 version 2)
- Extensions (optionnel, X.509 version 3)
 - Liste des extensions
- **Certificat**

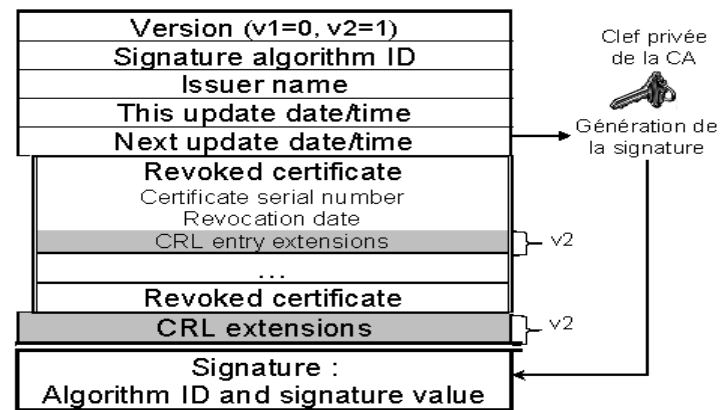
Format d'un Certificat X.509 – (II)



Exemple

```
Certificate:
Data:
  Version: 1 (0x0)
  Serial Number: 7829 (0x1e95)
  Signature Algorithm: md5WithRSAEncryption
  Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
         OU=Certification Services Division,
         CN=Thawte Server CA/emailAddress=server-certs@thawte.com
  Validity
    Not Before: Jul  9 16:04:02 1998 GMT
    Not After : Jul  9 16:04:02 1999 GMT
  Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
         OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
      Modulus (1024 bit):
        00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
        33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
        66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
        70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
        16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
        c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
        8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
        d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
        e8:35:1c:9e:27:52:7e:41:8f
      Exponent: 65537 (0x10001)
  Signature Algorithm: md5WithRSAEncryption
    93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
    92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
    ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
    d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
    0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
    5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
    8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
    68:9f
```

X509 – Format des Listes de Révocation



Exemple de CRL

Certificate Revocation List (CRL):

Version 1 (0x0)

Signature Algorithm: md5WithRSAEncryption

Issuer: /C=FR/L=Paris/O=Hervé E9 Schauer Consultants
/OU=Certificate Authority/CN=HSC CA/Email=ca@hsc.fr

Last Update: Aug 26 12:13:35 1999 GMT

Next Update: Sep 25 12:13:35 1999 GMT

Revoked Certificates:

Serial Number: 07

Revocation Date: Aug 26 12:12:31 1999 GMT

Signature Algorithm: md5WithRSAEncryption

c4:92:09:bd:ca:9f:cd:56:bd:ef:05:85:f7:b8:01:a6:f5:69:

Collision sur MD5



Xiaoyun Wang, Hongbo Yu.

How to Break MD5 and Other Hash Functions.


EUROCRYPT 2005.

```
d131dd02c5e6eec4693d9a0698aff95c 2fcab58712467eab4004583eb8fb7f89
55ad340609f4b30283e488832571415a 085125e8f7cdc99fd91dbdf280373c5b
d8823e3156348f5bae6dacd436c919c6 dd53e2b487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080a80d1e c69821bcb6a8839396f9652b6ff72a70
```

and

```
d131dd02c5e6eec4693d9a0698aff95c 2fcab50712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325f1415a 085125e8f7cdc99fd91dbdf7280373c5b
d8823e3156348f5bae6dacd436c919c6 dd53e23487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080280d1e c69821bcb6a8839396f965ab6ff72a70
```

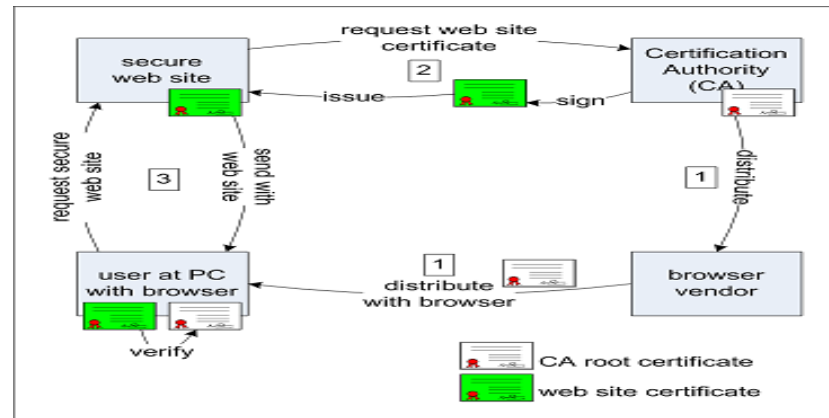
Collision sur les certificats X.509

 Arjen Lenstra and Xiaoyun Wang and Benne de Weger.
Colliding X.509 Certificates based on MD5-collisions.
EUROCRYPT 2005.

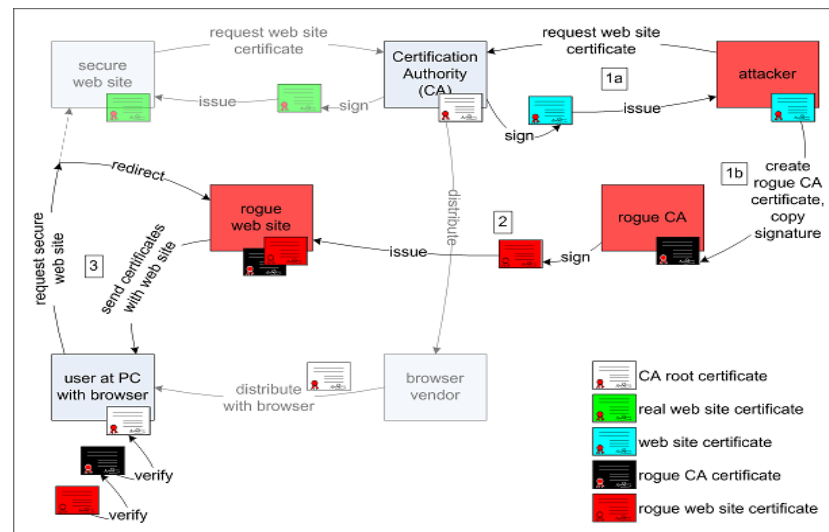
⇒ Il est possible de trouver deux certificats avec la même signature.



CA dévoyé



CA dévoyé



Plan du cours

- 1 Introduction SSL/TLS
- 2 Certification
 - Généralités
 - X.509
- 3 Protocole TLS
 - TLS Handshake
 - TLS Record
- 4 Attaques
 - BEAST

Déroulement du protocole

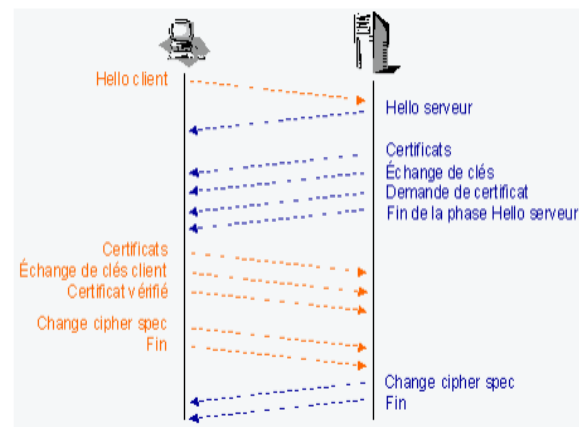
- **TLS Record**
 - Compression des données
 - Chiffrer les connexions avec un algorithme symétrique
 - Vérifier l'intégrité à l'aide avec un MAC
- **TLS Handshake**
 - Authentifier une partie (voir les deux)
 - Négocier les algorithmes et les clefs de session utilisées par TLS
 - Remonter des alertes



Plan du cours

- 1 Introduction SSL/TLS
- 2 Certification
 - Généralités
 - X.509
- 3 Protocole TLS
 - TLS Handshake
 - TLS Record
- 4 Attaques
 - BEAST

TLS Handshake



Authentification du Serveur

- Après la demande du client, le serveur envoie :
 - son certificat,
 - liste les algorithmes cryptographiques.
- Le client vérifie la validité du certificat.
- Si le certificat est valide, le client génère :
 - un **pre-master secret (PMS)** de 48 octets
 - dérive ensuite un **master secret (MS)** de même taille
- Le PMS est chiffré avec la clef publique du serveur puis transmis à ce dernier (typiquement, $\text{PMS}^{e_s} \bmod N_s$).

Les données échangées par la suite entre le client et le serveur sont chiffrées et authentifiées à l'aide des clefs dérivées de MS.

Authentification (optionnelle) du Client

- Le serveur – et seulement lui – peut demander au client de s'authentifier en lui demandant son certificat.
- Le client réplique en envoyant ce certificat puis en signant un message avec sa clef privée.
 - Ce message contient des informations sur la session et le contenu de tous les échanges précédents.

Génération des Clefs (< Août 2008)

- Création du **MS**
 - Le PMS a été échangé pendant le Handshake
 $MS = f(r_s, r_c, PMS)$, avec r_s, r_c des aléas échangés pendant le **ClientHello** et **ServerHello**.
 - f utilise plusieurs appels à MD5 et SHA1 (ou SHA256).

Soit :

$$f_{Char'}(r_s, r_c, PMS) = MD5(PMS \| SHA1('Char' \| PMS \| r_c \| r_s)),$$

alors :

$$f(r_s, r_c, PMS) = f_{A'}(r_s, r_c, PMS) \| f_{BB'}(r_s, r_c, PMS) \| f_{CCC'}(r_s, r_c, PMS).$$

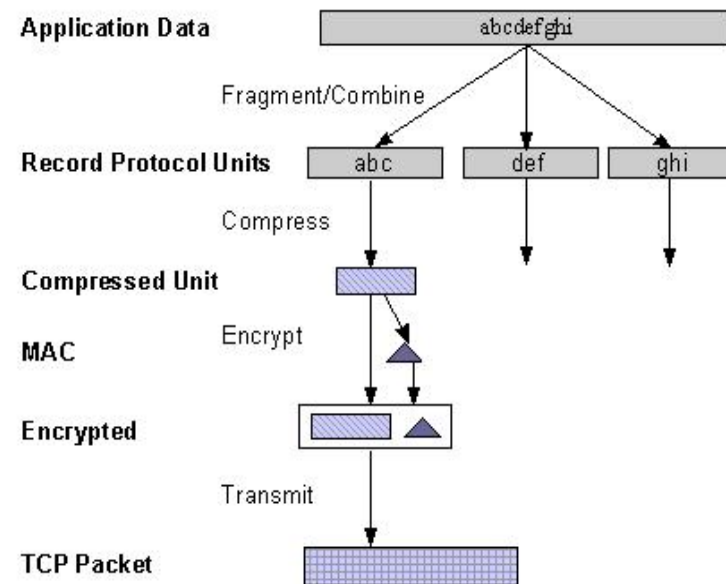
Plan du cours

- 1 Introduction SSL/TLS
- 2 Certification
 - Généralités
 - X.509
- 3 Protocole TLS
 - TLS Handshake
 - TLS Record
- 4 Attaques
 - BEAST

TLS Record – (I)

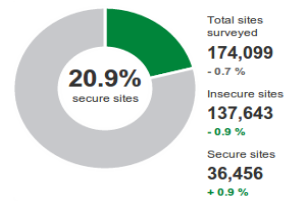
- Services
 - Fragmentation
 - Transfère
 - Compression
 - Confidentialité
 - Chiffrement par blocs (DES, 3DES, AES128, AES192, AES256, ...) en mode CBC **dégradé**
 - ...
 - Intégrité (utilisation d'un HMAC)
 - MD5, SHA1, et SHA256.

TLS Record – (II)



SSL Pulse (www.trustworthyinternet.org/ssl-pulse/)

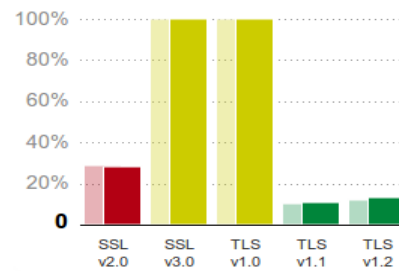
Bilan



SSL Pulse (www.trustworthyinternet.org/ssl-pulse/)

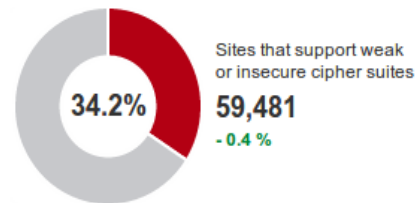
Protocoles

- SSL v2.0
- SSL v3.0 & TLS v1.0
- TLS v1.1 & TLSv1.2.



SSL Pulse (www.trustworthyinternet.org/ssl-pulse/)

Faiblesse du Chiffrement – Phase Record (limite 128 bits)



SSL Pulse (www.trustworthyinternet.org/ssl-pulse/)

Resistance aux Attaques Connues

