Architecture des réseaux - R3.06
HTTP & HTTPS & SSH

IUT-2
Département Informatique

24 septembre 2024

Sommaire

IUT2 Université Grenoble Alpes
Département INFO

Hypertext Transfer Protocol et le web

Hypertext Transfer Protocol (HTTP) a été inventé en 1989 par Tim Berners-Lee (alors au CERN) avec les URLs et le langage HTML pour créer le World Wide Web (WWW) .

> *« Je n'ai fait que prendre le principe d'hypertexte et le relier au principe du TCP et du DNS et alors, boum ! ce fut le World Wide Web ! »*
>
> *"**I just had to take the hypertext idea and connect it to the TCP and DNS ideas and – ta-da !***
>
> ***– the World Wide Web**." https://www.w3.org/People/Berners-Lee/Kids.html*

Il fonde en 1990 le World Wide Web Consortium (W3C) pour promouvoir et assurer l'interopérabilité des technologies du WWW.

Trois inventions ont permis de former le web :

1. URL : référence mondiale unique
2. HTML : inspiré de Standard Generalized Markup Language (SGML), Hypertextes
3. HTTP : protocole textuel très simple

IUT2 Université Grenoble Alpes
Département INFO

Modèle Client/Serveur

Client
un « navigateur »
(p.ex : Firefox, Google Chrome...)

Serveur
un « serveur web »
(p.ex : Apache, Nginx, ...)

Echange typique :

1. requête : GET www...
2. réponse : un fichier HTML, XML, JPEG...

IUT2 Université Grenoble Alpes
Département INFO

le protocole HTTP

HTTP

Le protocole HTTP défini par la RFC2068 [Fielding et coll., 1997] a été conçu pour :

- transférer les documents hypertextes d'un serveur vers un client
- transférer n'importe quel autre type fichier.

Ressources

Ces documents sont appelés des « **ressources** ».
Chaque **ressource** est « localisée » sur le serveur par une Uniform Resource Locator (URL).

IUT2 Université Grenoble Alpes
Département INFO

Uniform Resource Locator

URL : adresse identifiant toute ressource (document) sur le web.

Syntaxe d'une URL :

```
<protocole><nomserveur ou adresse IP>:<port>/<chemin ressource>
```

Exemples :

```
http://www.ietf.org
http://www.iut2.univ-grenoble-alpes.fr/index.html
http://localhost:8080/ap5/page.html#chap1
http://216.58.206.67:80/search?source=hp&q=iut&oq=iut
```

Bien d'autres protocoles possibles : https, ftp, file...

IUT2 Université Grenoble Alpes
Département INFO

Uniform Resource Locator

URL : adresse identifiant toute ressource (document) sur le web.

Syntaxe d'une URL :

```
<protocole><nomserveur ou adresse IP>:<port>/<chemin ressource>
```

Exemples :

```
http://www.ietf.org
http://www.iut2.univ-grenoble-alpes.fr/index.html
http://localhost:8080/ap5/page.html#chap1
http://216.58.206.67:80/search?source=hp&q=iut&oq=iut
```

Bien d'autres protocoles possibles : https, ftp, file...

IUT2 Université Grenoble Alpes
Département INFO

Format d'une URL

Identité du gestionnaire de ressource

`http://www.monsite.fr/un_chemin/un_fichier?argument`

Nom DNS ou adresse IP

`http://193.55.51.32/un_chemin/un_fichier?argument`

Numéro de port : 80 par défaut

`http://www.monsite.fr:8080/un_chemin/un_fichier?argument`

Autres ports : 8000, 631 (CUPS), etc. (services avec interface web)

IUT2 Université Grenoble Alpes
Département INFO

Format d'une URL

Localisation de la ressource

`http://www.monsite.fr/un_chemin/un_fichier?argument`

Chemin → Organisation du site
Fichier → Document, script (Site statique vs. site dynamique)

Paramètres

`http://www.monsite.fr/un_chemin/un_fichier?argument`

Spécifique à l'application web → Dans le code du script

IUT2 Université Grenoble Alpes
Département INFO

HTTP – requête

Format texte, très simple, codage ASCII

GET : demande une ressource par son URL (pas de modification)

PUT : inverse de get, place une ressource sur un serveur

POST : envoie de données binaires (encodées en texte) sur un serveur pour traitement

DELETE : supprime une ressource

PATCH : modification partielle d'une ressource

et d'autres : HEAD, OPTION, CONNECT, TRACE ...

PUT, DELETE, PATCH : nécessitent un accès avec privilège (mot de passe)

IUT2 A
Université Grenoble Alpes
Département INFO

HTTP 1.1 : requête

Format requête :

```
Ligne de commande (Commande, URI, Version de protocole)
En-tête de requête
[Ligne vide]
Corps de requête
```

Exemple :

```
GET /index.html HTTP/1.1
Host: exemple.com
User-Agent: Mozilla/5.0 Firefox/100.0
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3

[vide -- pas de corps de requête]
```

HTTP 1.1 : réponse

Format Réponse :

```
Ligne de statut (Version, Code-réponse, Texte-réponse)
En-tête de réponse
[Ligne vide]
Corps de réponse
```

Exemple :

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2021 23:59:59 GMT
Server: Apache/0.8.4
Content-Type: text/html
Content-Length: 59
Expires: Sat, 01 Jan 2022 00:59:59 GMT
Last-modified: Fri, 09 Aug 2020 14:21:40 GMT

<TITLE>Exemple</TITLE>
<P>Ceci est une page d'exemple.</P>
```

HTTP code de réponse

Les codes d'état :

- 1xx - Information
- 2xx - Succès
- 3xx - Redirection
- 4xx - Erreur du client HTTP
- 5xx - Erreur du serveur

Exemples :

- 200 : succès
- 301 et 302 : redirection (permanente ou temporaire)
- 403 : accès refusé
- 404 : ressource non trouvée
- 504 : pas de réponse

IUT2 A
Université Grenoble Alpes
Département INFO

HTTP Type de ressource

Document

- Page HTML statique, fichier image, audio, etc.

Script ou programme CGI

- Common Gateway Interface
- Génère dynamiquement des ressources (HTML, image, archive, etc.)
- Programmé en C, Java, Perl, Python, etc.

Hybride : document + script

- Côté serveur : PHP, JSP, ASP, etc. (Interprété/Exécuté avant réponse)
- Côté client : Javascript, Java, Flash, ActiveX, etc.

IUT2 A
Université Grenoble Alpes
Département INFO

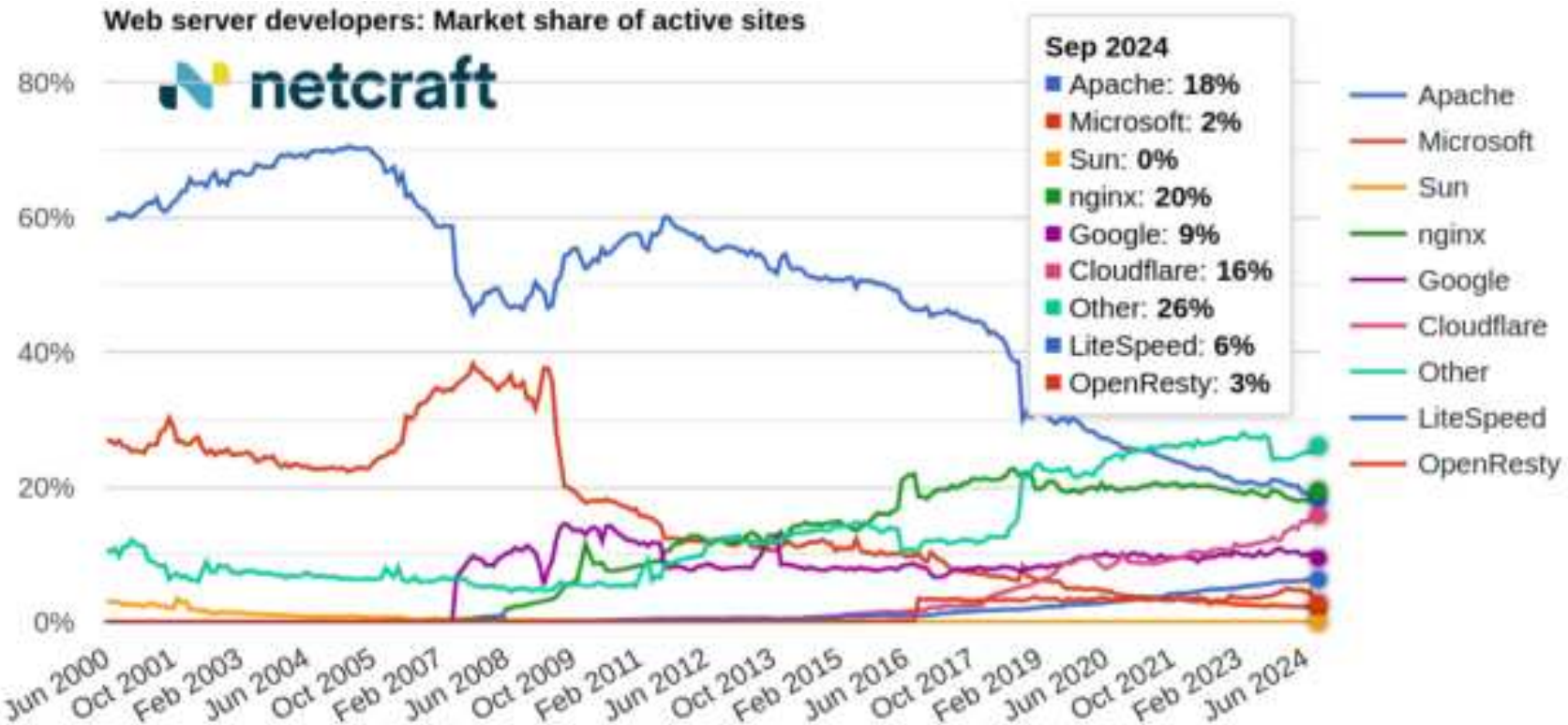
HTTP Caractéristiques clefs

- Protocole "textuel"
- TCP/80
- Simple : requête / réponse
- Sans état
- Connexion et échanges à l'initiative du client

IUT2 A
Université Grenoble Alpes
Département INFO

Apache

Un des serveurs les plus utilisés



source : `netcraft.com`

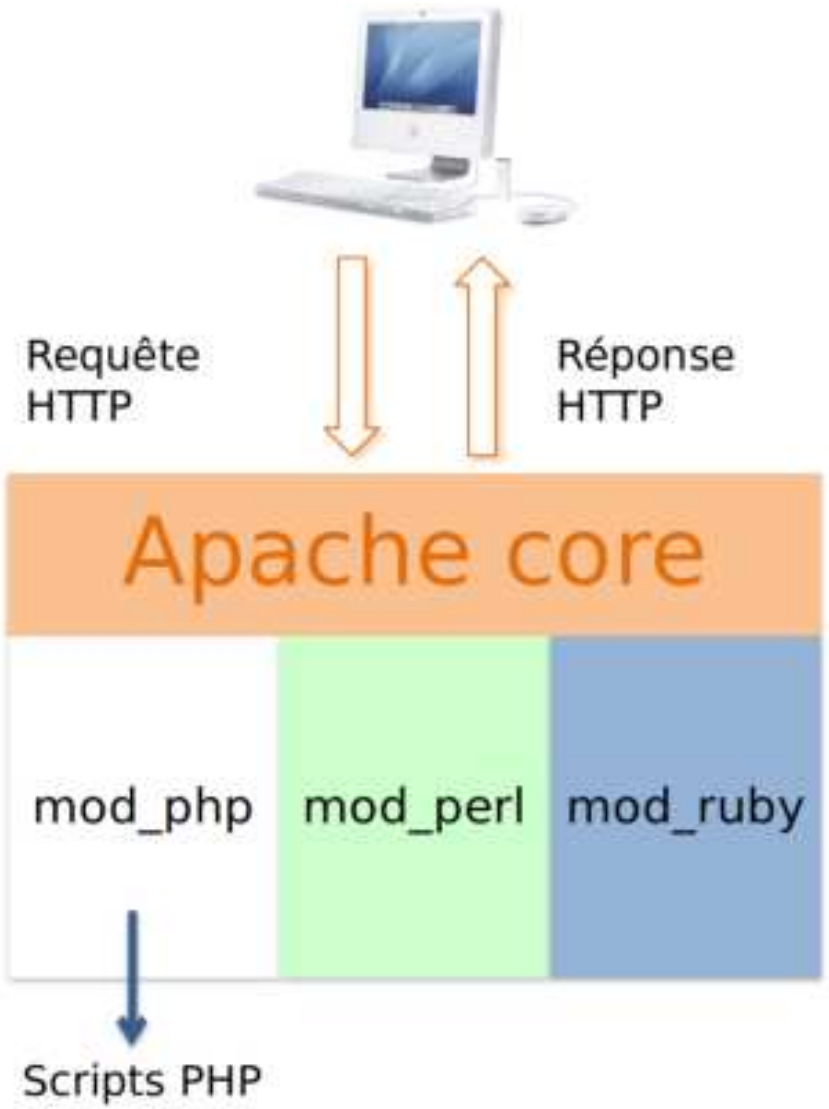
IUT2 Université Grenoble Alpes
Département INFO

Apache – Caractéristiques

- Créé en 1995
- Libre
- Multiplateforme
  - Unix, Linux, *BSD, Windows, OS X, etc.
- Supporte IPv4 et IPv6
- Hébergements de multiples sites
  - notion de Virtual Hosts
- Architecture modulaire

IUT2 Université Grenoble Alpes
Département INFO

Apache – Architecture



IUT2 Université Grenoble Alpes
Département INFO

Apache – Configuration (Debian GNU/Linux)

/etc/apache2/

```
apache2.conf   => Fichier de configuration principal
ports.conf     => Numéros de ports
envvars        => Variables d'environnement
conf.d/        => Fragments de la configuration

mods-available/ => Fichiers pour l'activation de modules
mods-enabled/   => Modules activés (liens symboliques)

sites-available/ => Configuration spécifique / site web
sites-enabled/   => Sites actifs (liens symboliques)
```

IUT2 Université Grenoble Alpes
Département INFO

Configuration des modules : Ex. PHP5

```
/etc/apache2/mods-available/
    php5.conf => configuration du module
    php5.load => chemin vers le binaire du module
/etc/apache2/mods-enabled/
    php5.conf -> ../mods-available/php5.conf
    php5.load -> ../mods-available/php5.load
```

IUT2 A
Université Grenoble Alpes
Département INFO

Configuration des sites

```
/etc/apache2/sites-available/
    default => site HTTP par défaut
    default-ssl => site HTTPS par défaut
/etc/apache2/sites-enabled/
    000-default -> ../sites-available/default
    default-ssl -> ../sites-available/default-ssl
```

IUT2 A
Université Grenoble Alpes
Département INFO

Activater des modules et des sites : commandes

Modules :

- `a2enmod` : active un module (qui doit être installé)
- `a2dismod` : désactive un module

Sites

- `a2ensite` : active un site (qui doit avoir un fichier de config)
- `a2dissite` : désactive un site

Redémarrage du service

- `service apache2 reload`
- à faire après tout changement de configuration

IUT2 Université Grenoble Alpes
Département INFO

Faiblesses du protocole HTTP

HTTP n'est pas sécurisé.

$\rightarrow$ vulnérable à :

- Surveillance
- Usurpation d'identité
- *Man in the middle*

IUT2 Université Grenoble Alpes
Département INFO

Écoute/Surveillance

IUT2 Université Grenoble Alpes
Département INFO

Usurpation d'identité

IUT2 Université Grenoble Alpes
Département INFO

Man in the middle

Hypertext Transfer Protocol Secure (HTTPS)

Hypertext Transfer Protocol Secure (HTTPS) permet :

1. de vérifier l'**identité** des acteurs :
   - Le serveur : est-ce bien le site www.mabanque.fr ?
   - (Le client)

2. d'assurer la **confidentialité** et l'**intégrité** des échanges des échanges :
   - Échange d'une clé secrète de codage entre le navigateur et le client

La sécurité est assurée par un protocole de sécurisation des échanges : Transport Layer Security (TLS).

IUT2 A
Université Grenoble Alpes
Département INFO

Transport Layer Security (TLS)

Fonctionne suivant un mode client-serveur. Il permet :

- l'authentification du serveur (parfois celle du client)
- la confidentialité des données échangées (ou session chiffrée)
- l'intégrité des données échangées ;

Il existe plusieurs version de TLS (de 1.0 à 1.3 aujourd'hui).
le client et le serveur négocient la « meilleure » version du protocole à utiliser dans une phase de négociation

IUT2 A
Université Grenoble Alpes
Département INFO

HTTPS

HTTPS = *HTTP Over TLS* ; RFC 2818 [Rescorla, 2000]

→ Implanté comme HTTP au dessus d'une couche TLS (couche session)

- **Port 443**

Besoin :

- Initialement conçu pour les transactions financières en ligne
- Progressivement étendu à presque tous le web
  - Depuis 2017, Chrome et Firefox signalent les sites n'utilisent pas HTTPS pour recueillir des informations
  - Les site HTTPS sont généralement mieux indexés par les moteurs de recherche

sources : google.com

IUT2 Université Grenoble Alpes
Département INFO

Sommaire

IUT2 Université Grenoble Alpes
Département INFO

Principe du chiffrement

Dans les communications, le chiffrement peut être utilisé pour rendre un message illisible pour toute personne étrangère à l'échange.

Les étapes sont généralement :

(**C**) Chiffrement : Un message clairement lisible (« bonjour Maman ») est chiffré en un message chiffré illisible (« erqmrxu#Pdpdq »)

(**T**) Transmission : le message est envoyé sur le réseau (p.ex. via HTTP, SMTP)

(**D**) Déchiffrement : Le message (« erqmrxu#Pdpdq ») est reçu par le destinataire qui possède la **clé** permettant de le déchiffrer sous sa forme originale (« bonjour Maman »).

IUT2 Université Grenoble Alpes
Département INFO

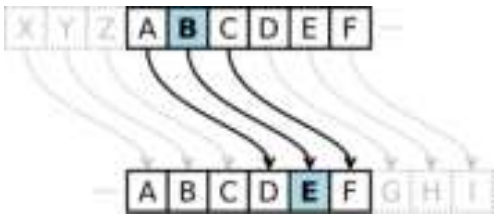
Exemple : chiffrement de César

Avec une **clé** = 3 on décale les lettres de 3 rangs dans une table ASCII.

« bonjour Maman »

devient

« erqmrxu#Pdpdq »



sources : wikipedia

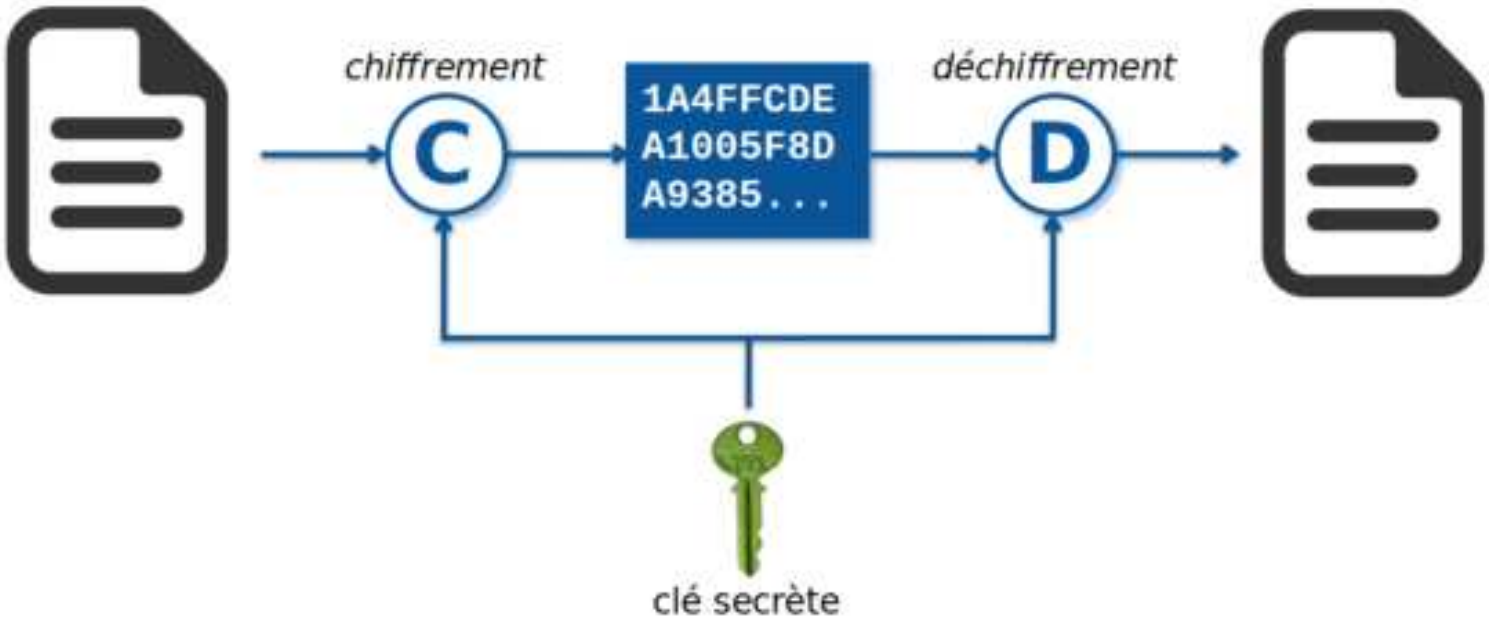
La même clé permet de *déchiffrer* le message (clé symétrique).

Exemples :
Jules César avec le chiffre de César
Turing avec Enigma.

IUT2 Université Grenoble Alpes
Département INFO

Chiffrement symétrique

La même clé permet de *chiffrer* et *déchiffrer* le message.



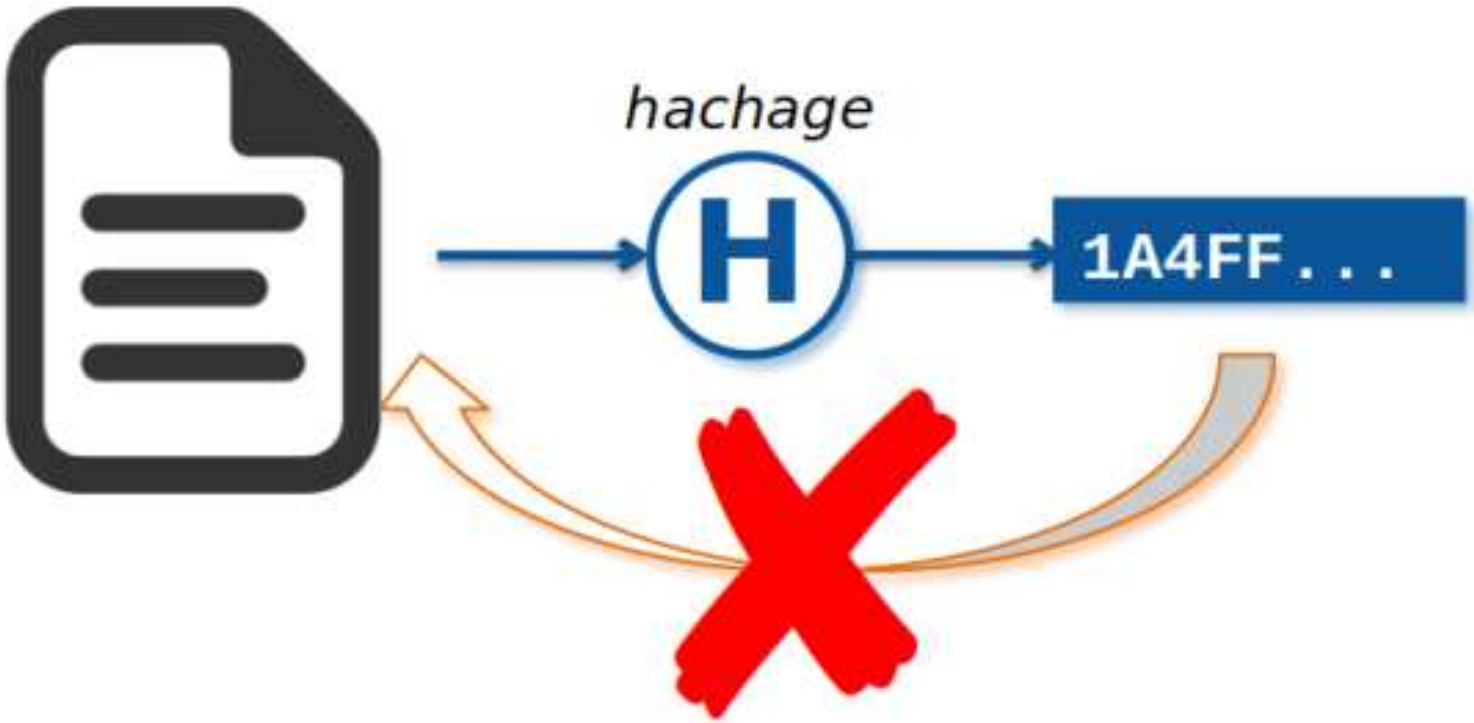
Algorithmes : ~~DES~~, ~~3DES~~, AES, Blowfish, ~~RC4 (WEP)~~

IUT2 Université Grenoble Alpes
Département INFO

Hachage : intégrité

Condensat, empreinte, hachage

- Vérifier la non-modification d'un document
- Résultat d'un calcul irréversible



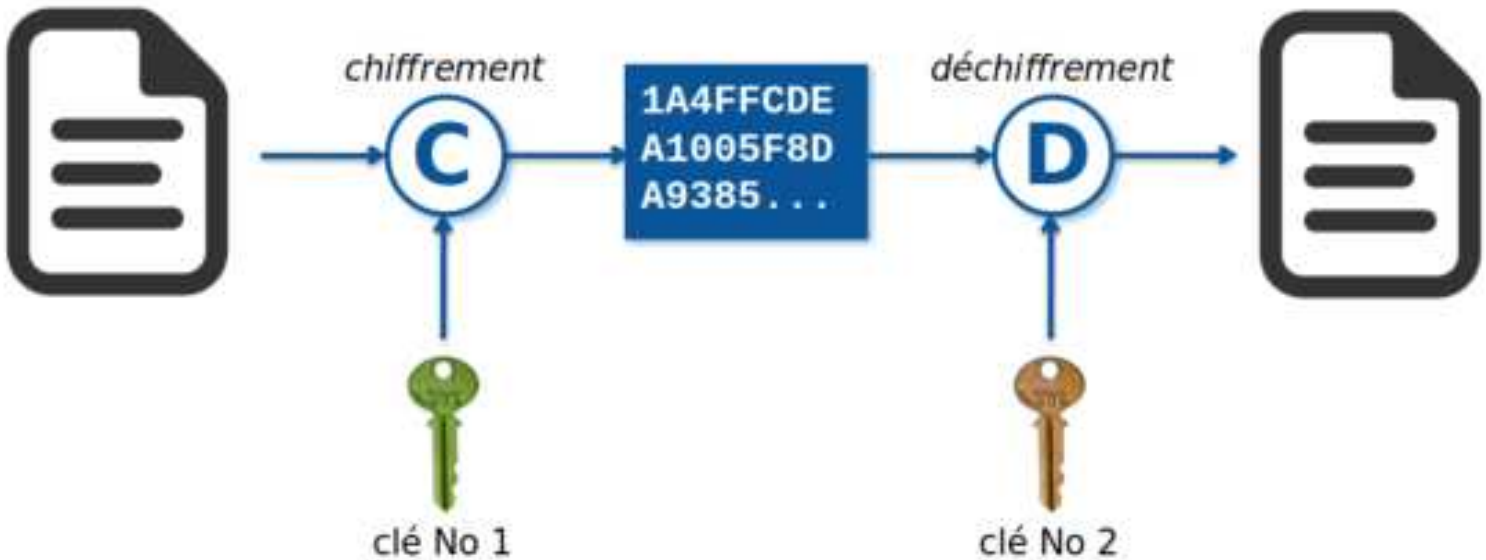
Algorithmes : ~~MD5~~, ~~SHA-1~~, SHA-256, SHA-384, chacha20-poly1305

IUT2 Université Grenoble Alpes Département INFO

Chiffrement asymétrique

Deux clés distinctes :

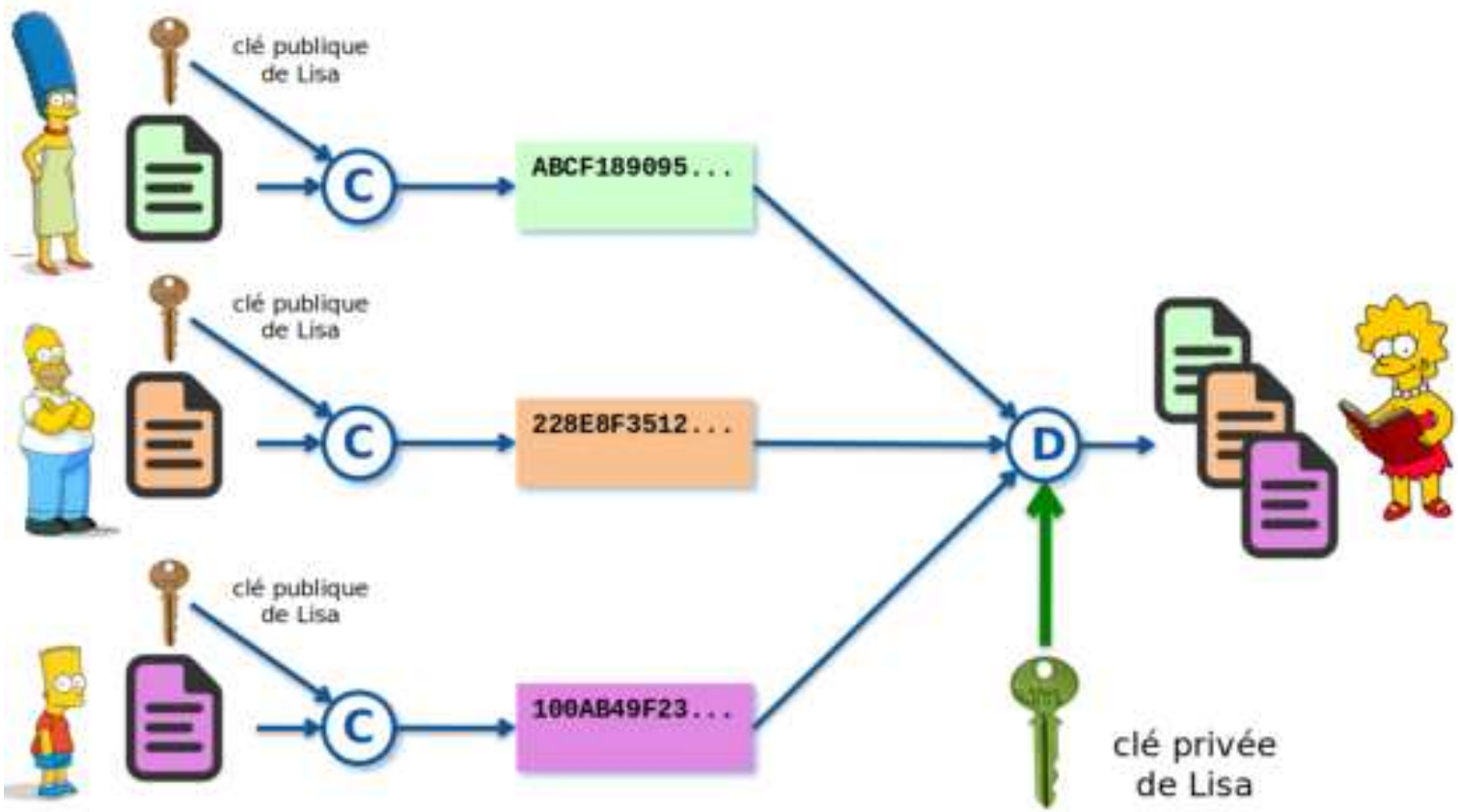
1. clé publique (distribuée)
2. clé privée (personnelle)



Algorithmes : RSA, El Gamal (PGP, DSA), ECDSA, ED25519

IUT2 A
Université Grenoble Alpes
Département INFO

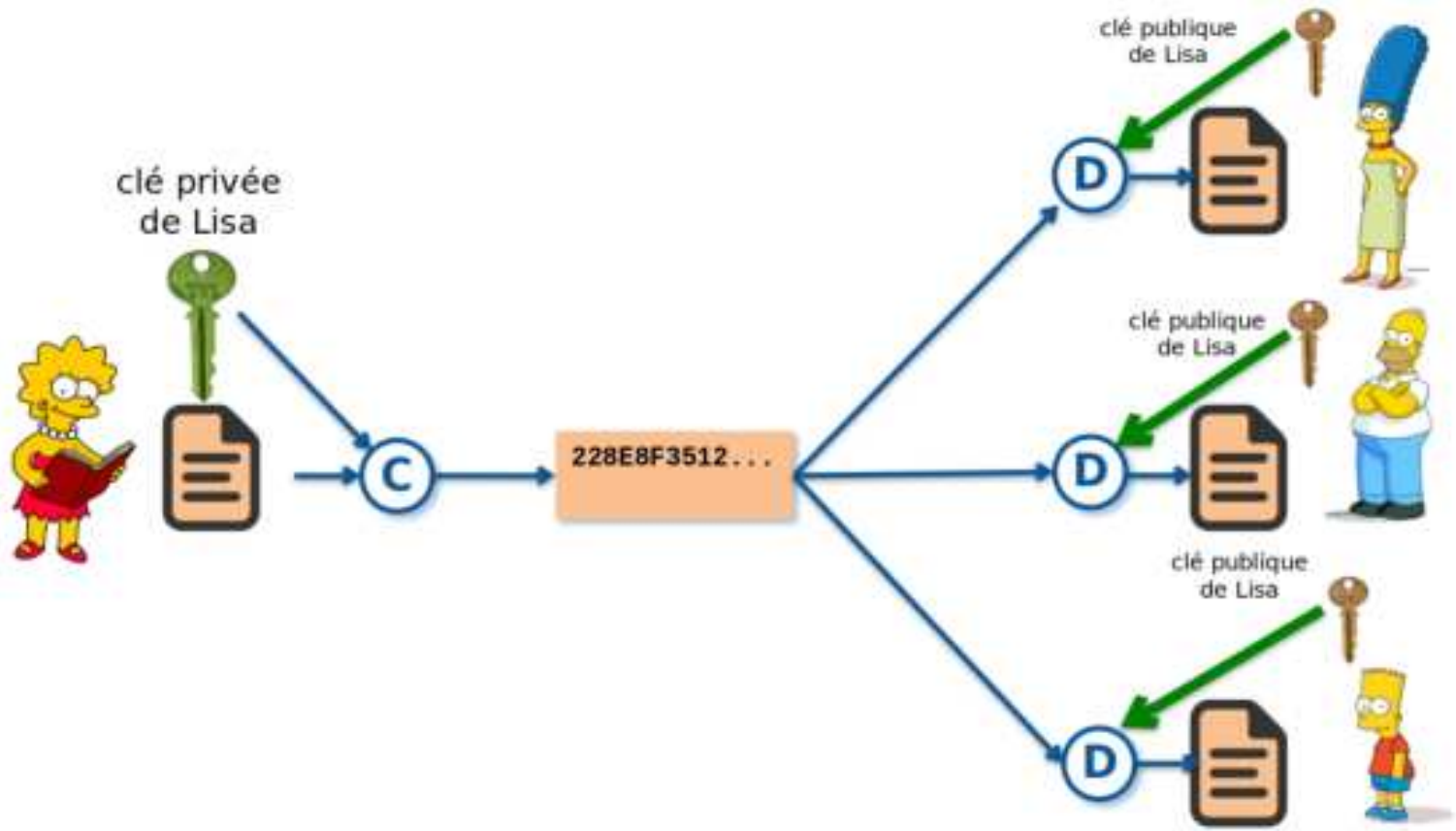
Chiffrement asymétrique : Confidentialité



seul le destinataire peut déchiffrer
Clé publique => chiffrement
Clé privée => déchiffrement
Ex. certificats

IUT2 Université Grenoble Alpes
Département INFO

Chiffrement asymétrique : authentification

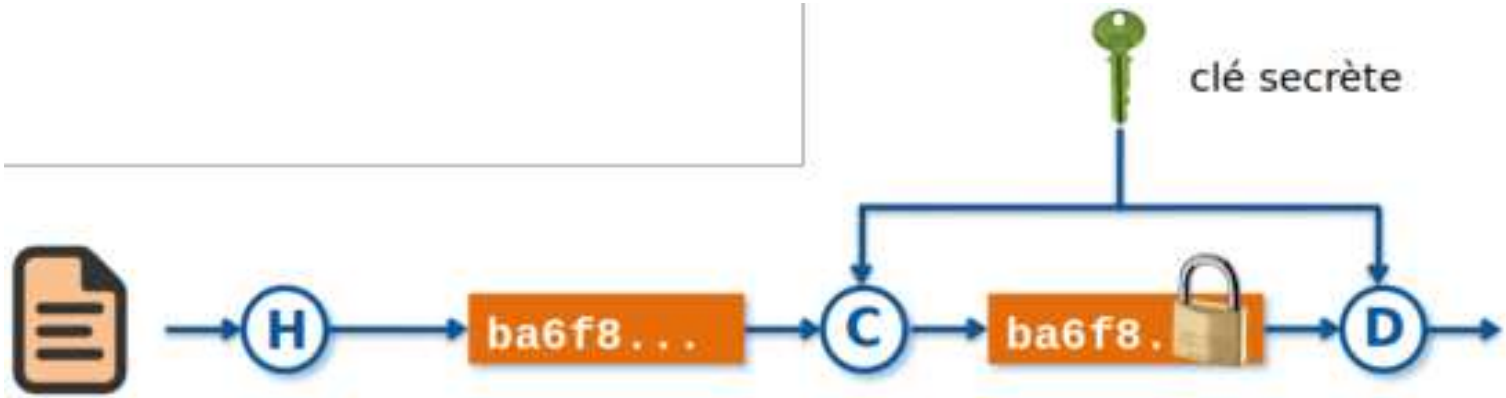


seul la clé publique permet de déchiffrer
Ex. Signature électronique

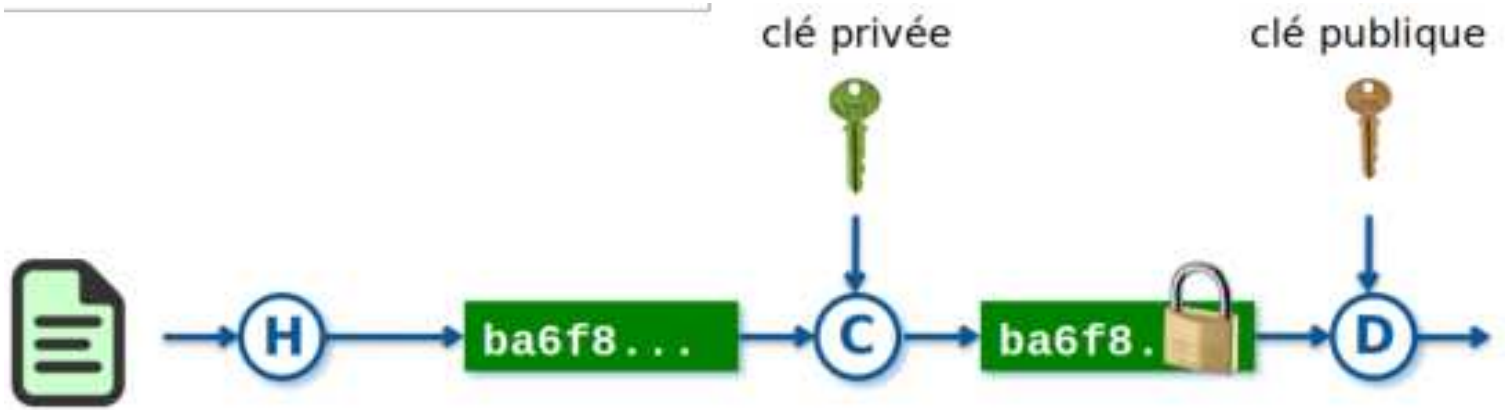
IUT2
Université Grenoble Alpes
Département
INFO

Condensats : usages

Code d'authentification (HMAC)
Chiffrement symétrique



Signature électronique
Chiffrement asymétrique



UT2 Université Grenoble Alpes
Département INFO

Force du chiffrement

Coûts : mathématique

- Factorisation des nombres premiers (RSA)
- Courbes elliptiques (ECDH)
- Logarithme discret (DH)

Cryptanalyse

- Déchiffrer sans connaître la clé
- Temps de calcul "infini", lié à
  - La qualité de l'algorithme de chiffrement (ex. MD5, DES)
  - La taille de clé

  → RSA : 4 ans pour casser une clé de 768 bits (2009)
  - 2048 bits recommandé, 112 bits pour du chiffrement symétrique
  - ne résistera pas à une attaque quantique [Chen et coll., 2016]

IUT2 Université Grenoble Alpes
Département INFO

Sommaire

IUT2 Université Grenoble Alpes
Département INFO

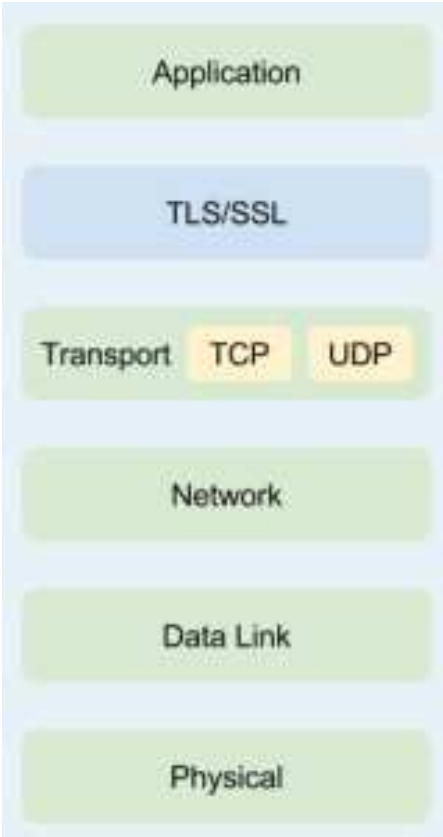
TLS (ex Secure Sockets Layer (SSL))

TLS procure un canal de communication sécurisé entre deux terminaux avec les propriétés suivantes :

Authentification : Le serveur est toujours authentifié (client optionnel). Par cryptographie asymétrique (p. ex. RSA,ECDSA), par signature numérique (EdDSA) ou par clé prépartagée symétrique (PSK).

Confidentialité : Les données envoyées sur le canal après son établissement ne sont visibles que par les terminaux.

Intégrité : Les données envoyées sur le canal après l'établissement ne peuvent pas être modifiées par des attaquants sans être détectées.



IUT2 Université Grenoble Alpes
Département INFO

Identité et Certificats

Carte d'identité

- CN : Common Name (nom du site, domaine, organisme, individu)
- O : Organisation
- Date de validité
- Autorité de certification (Certificate Authority : CA)

Clé publique

- Chiffrement asymétrique

Signature électronique

- Condensat chiffré : Carte d'identité + clé publique

IUT2 A
Université Grenoble Alpes
Département INFO

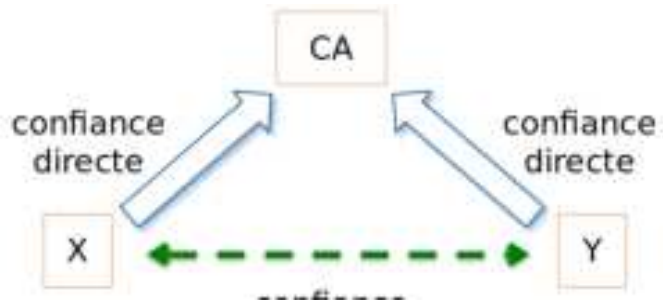
Certificats

Formats

- X.509 (RFC 1422)
- OpenPGP (RFC 4880)
- PKCS#1 (Public Key Cryptographic Standard)

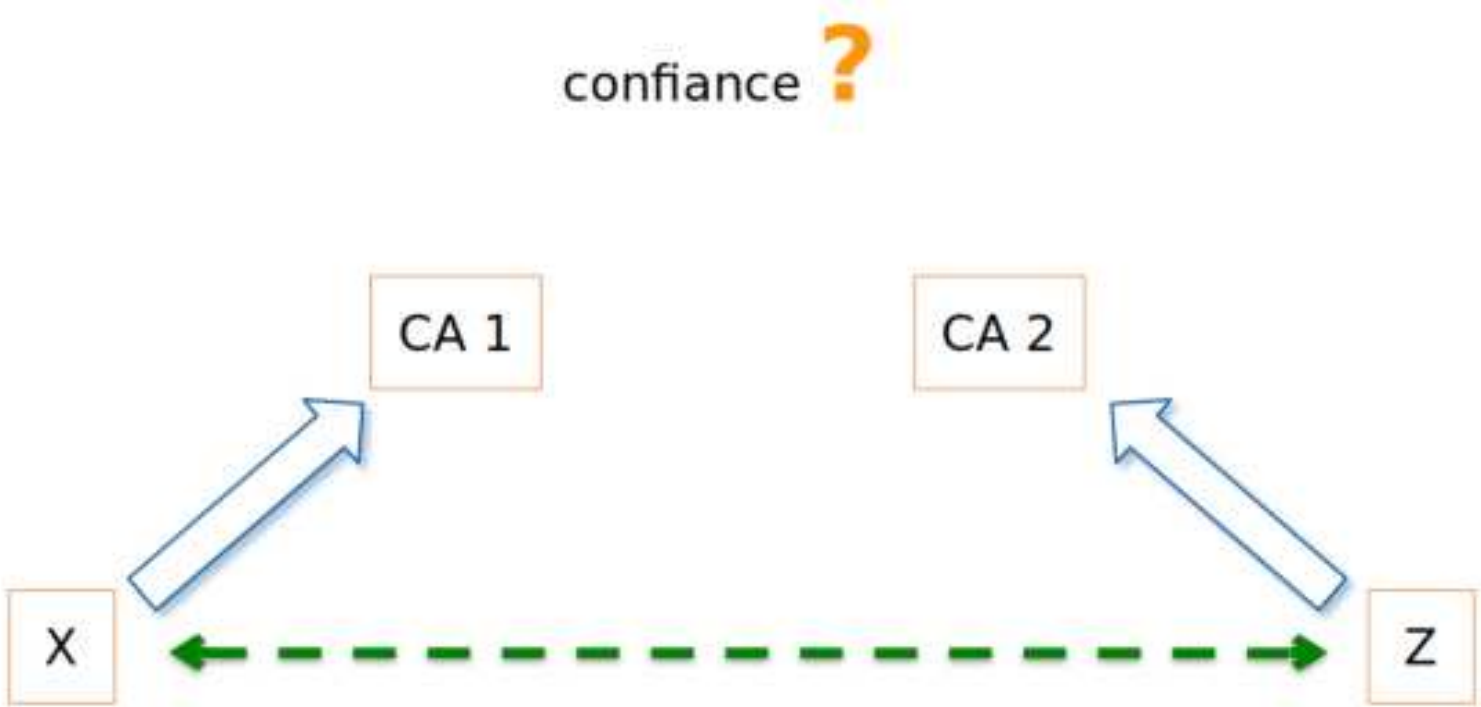
Autorité de certification (AC)

- Signature de certificat
- Principe de confiance réciproque

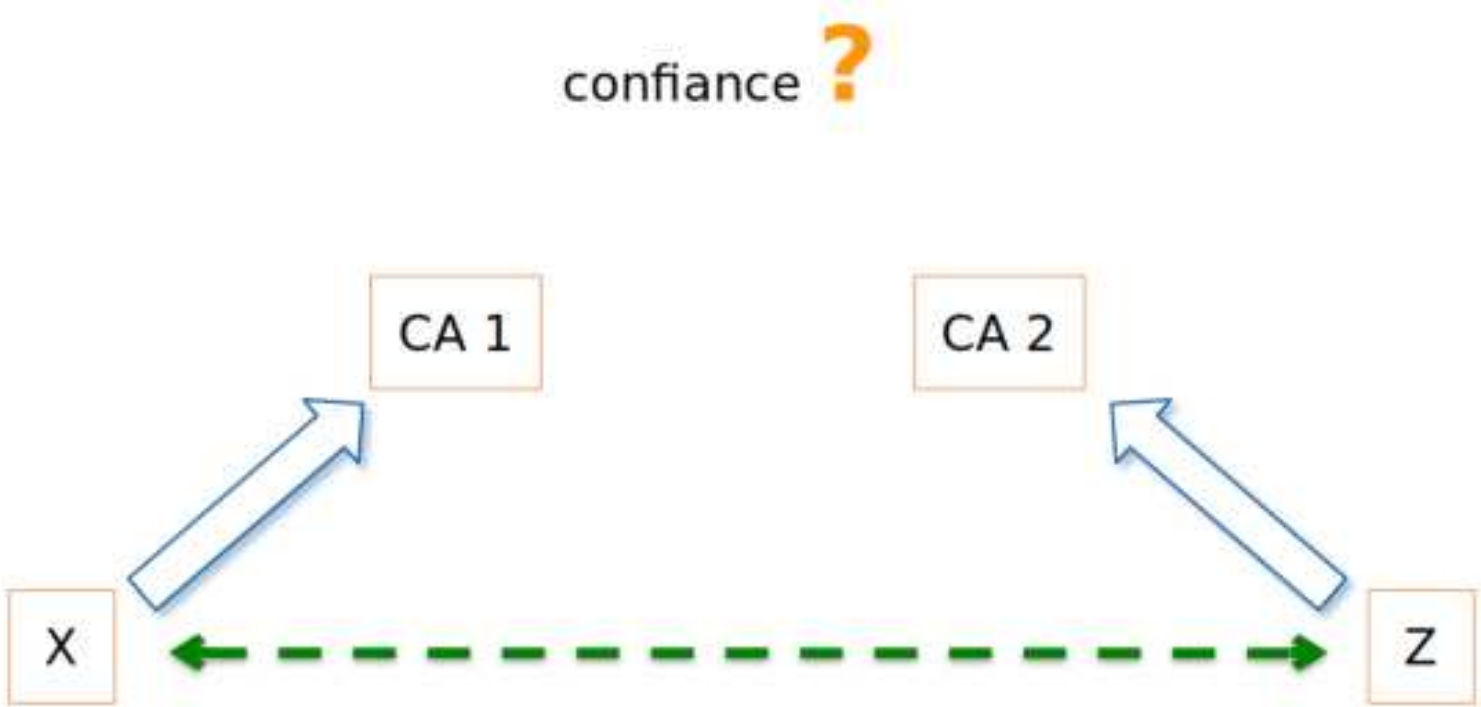


IUT2 Université Grenoble Alpes
Département INFO

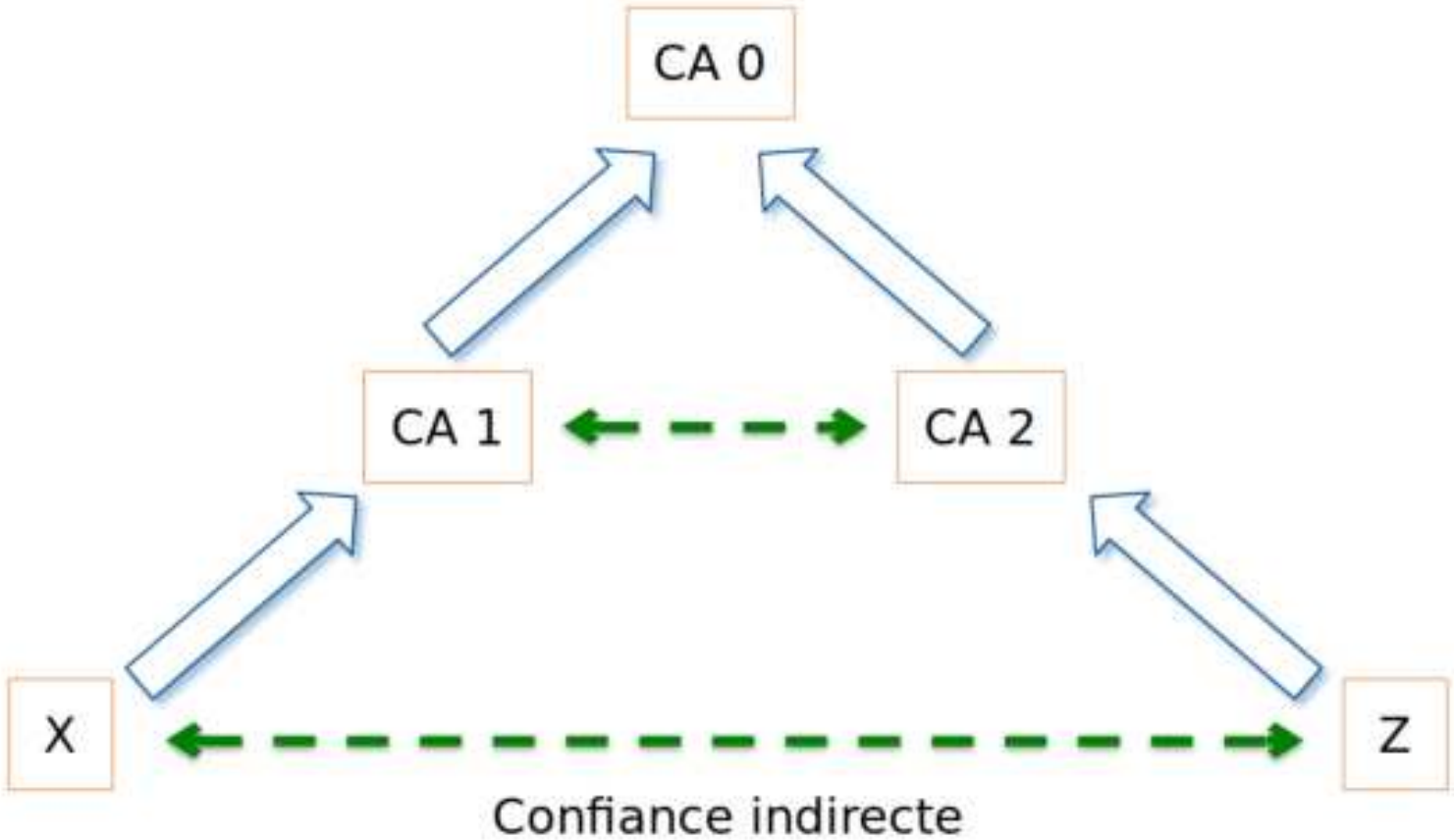
Certificats – chaîne de certification



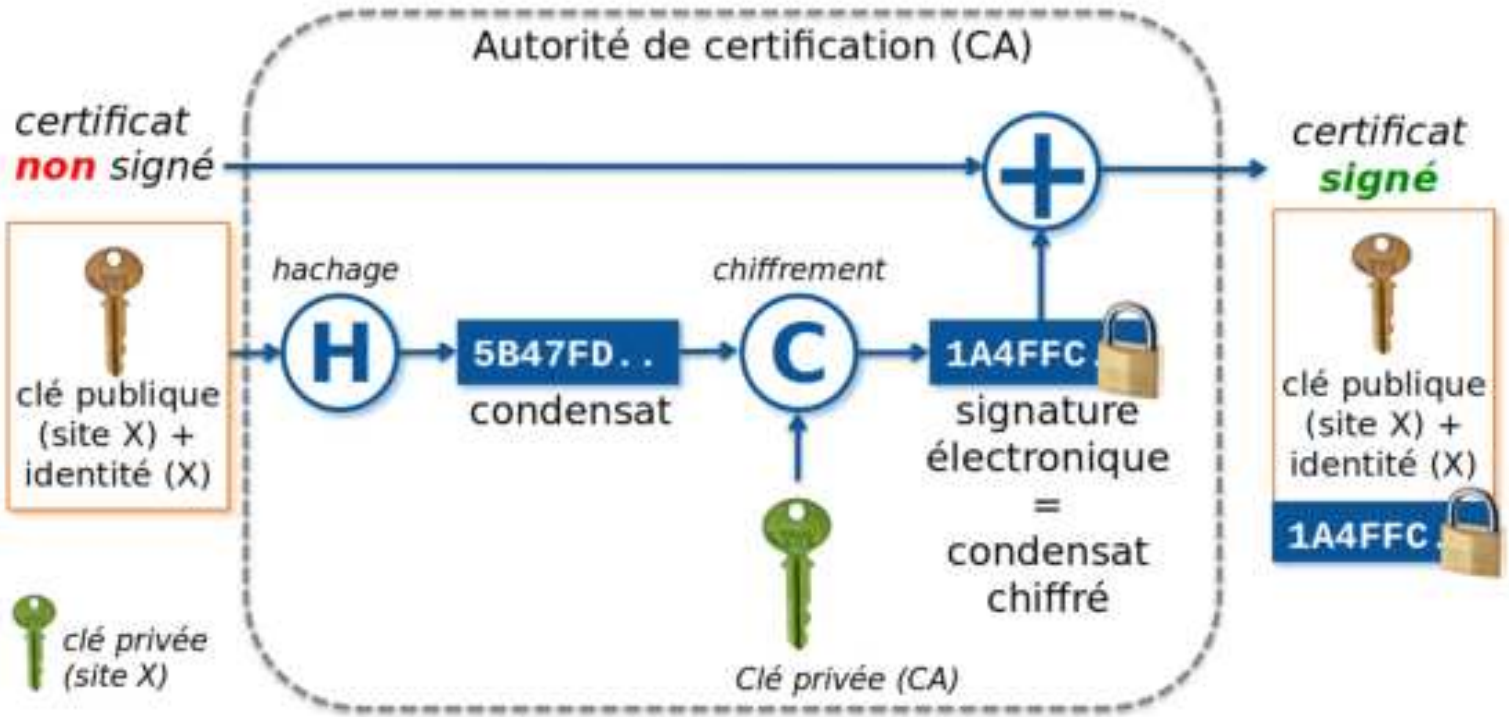
Certificats – chaîne de certification



Certificats – chaîne de certification



Certificats – signature par un CA



Sommaire

IUT2 Université Grenoble Alpes
Département INFO

Phases TLS

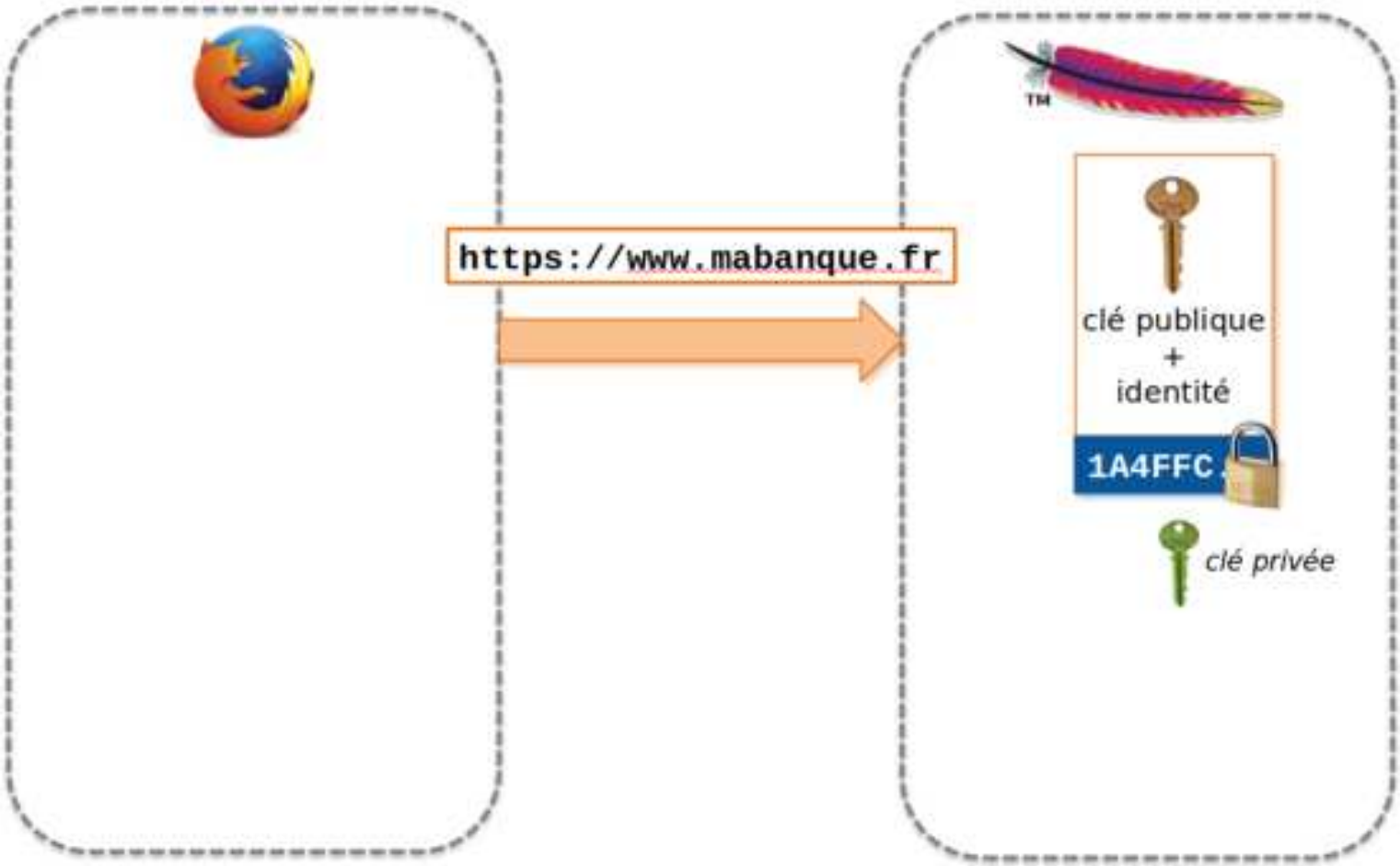
TLS se compose de deux éléments principaux :

Handshake Un protocole d'établissement (poignée de main) qui authentifie les parties communicantes, négocie les modes et les paramètres cryptographiques et établit des clés partagées. Le protocole est conçu pour résister à la falsification.

Échange utilise les paramètres établis par le *handshake* pour protéger le trafic entre les parties communicantes à l'aide des clés de trafic.

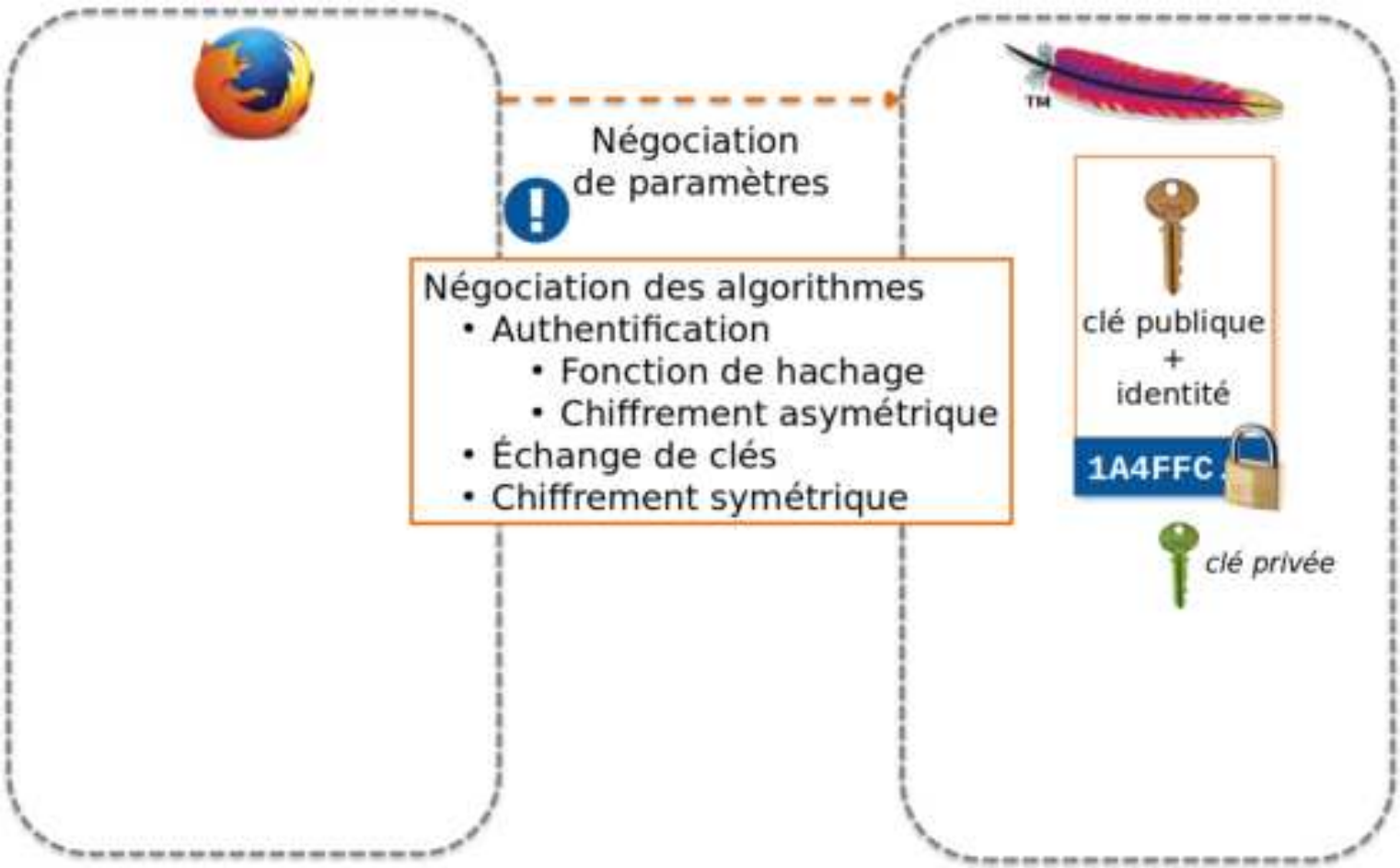
IUT2 A
Université Grenoble Alpes
Département INFO

Échanges HTTPS (TLS 1.2) – *Handshake*



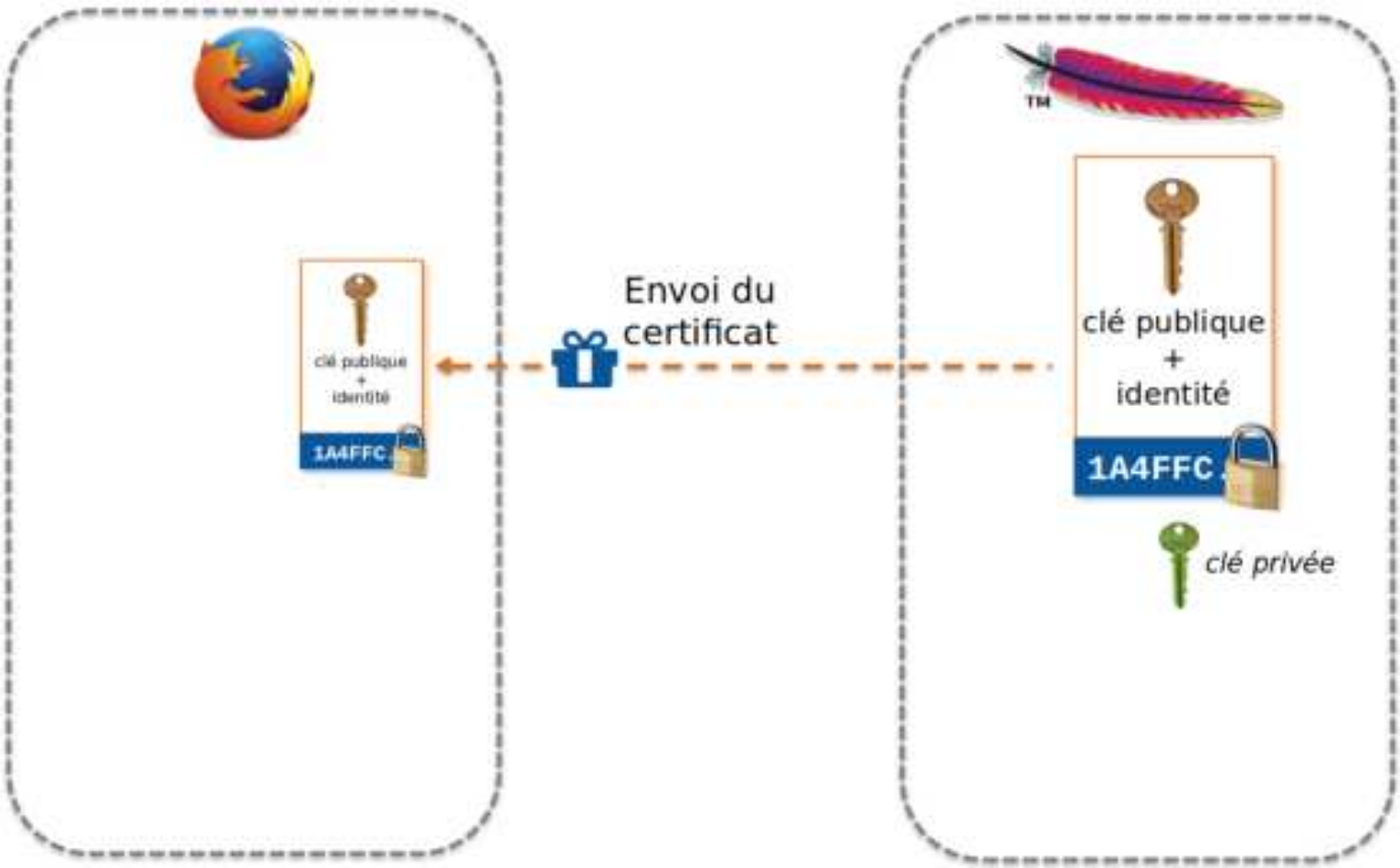
IUT2 Université Grenoble Alpes
Département INFO

Échanges HTTPS (TLS 1.2) – *Handshake*

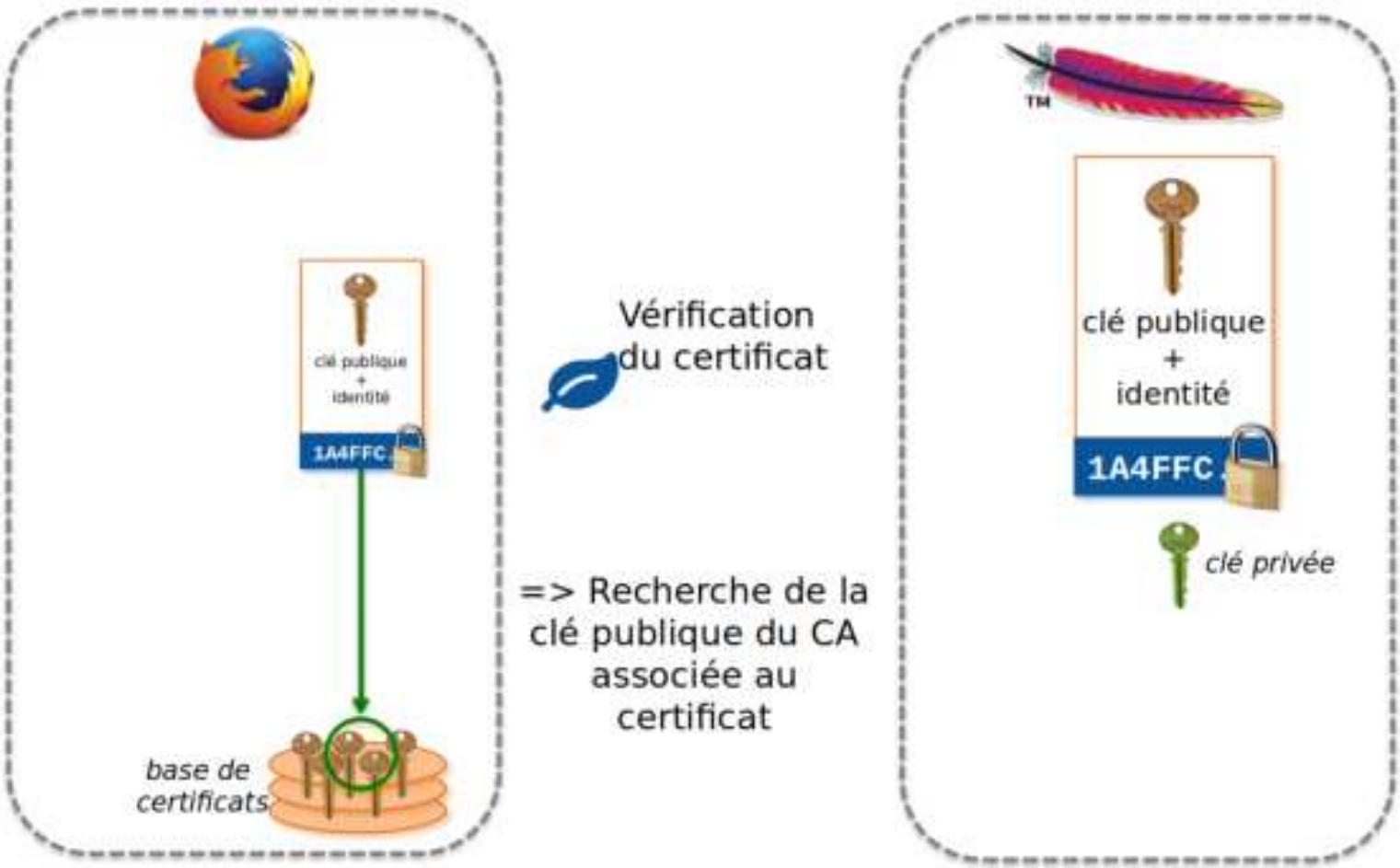


IUT2 Université Grenoble Alpes
Département INFO

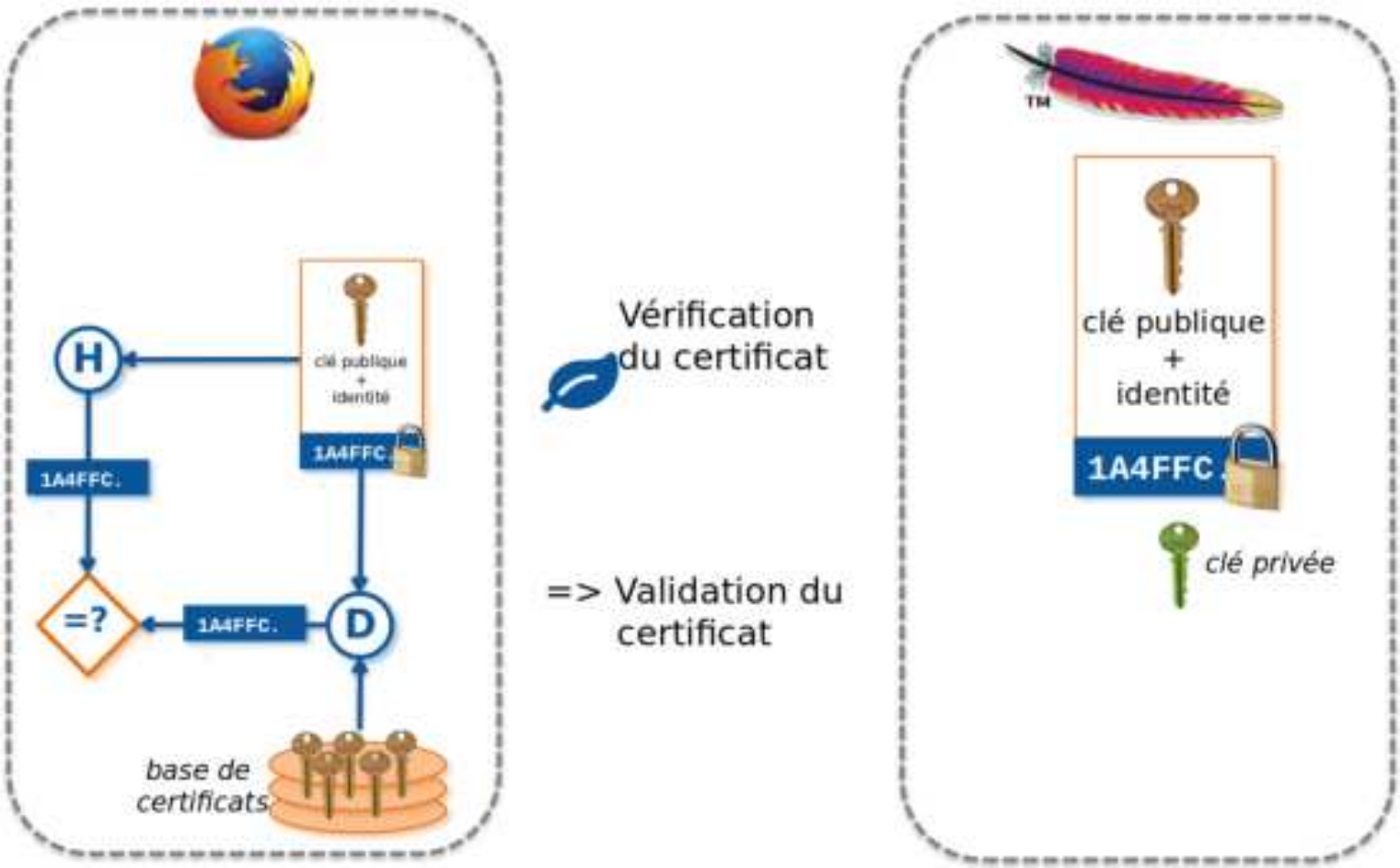
Échanges HTTPS (TLS 1.2) – *Handshake*



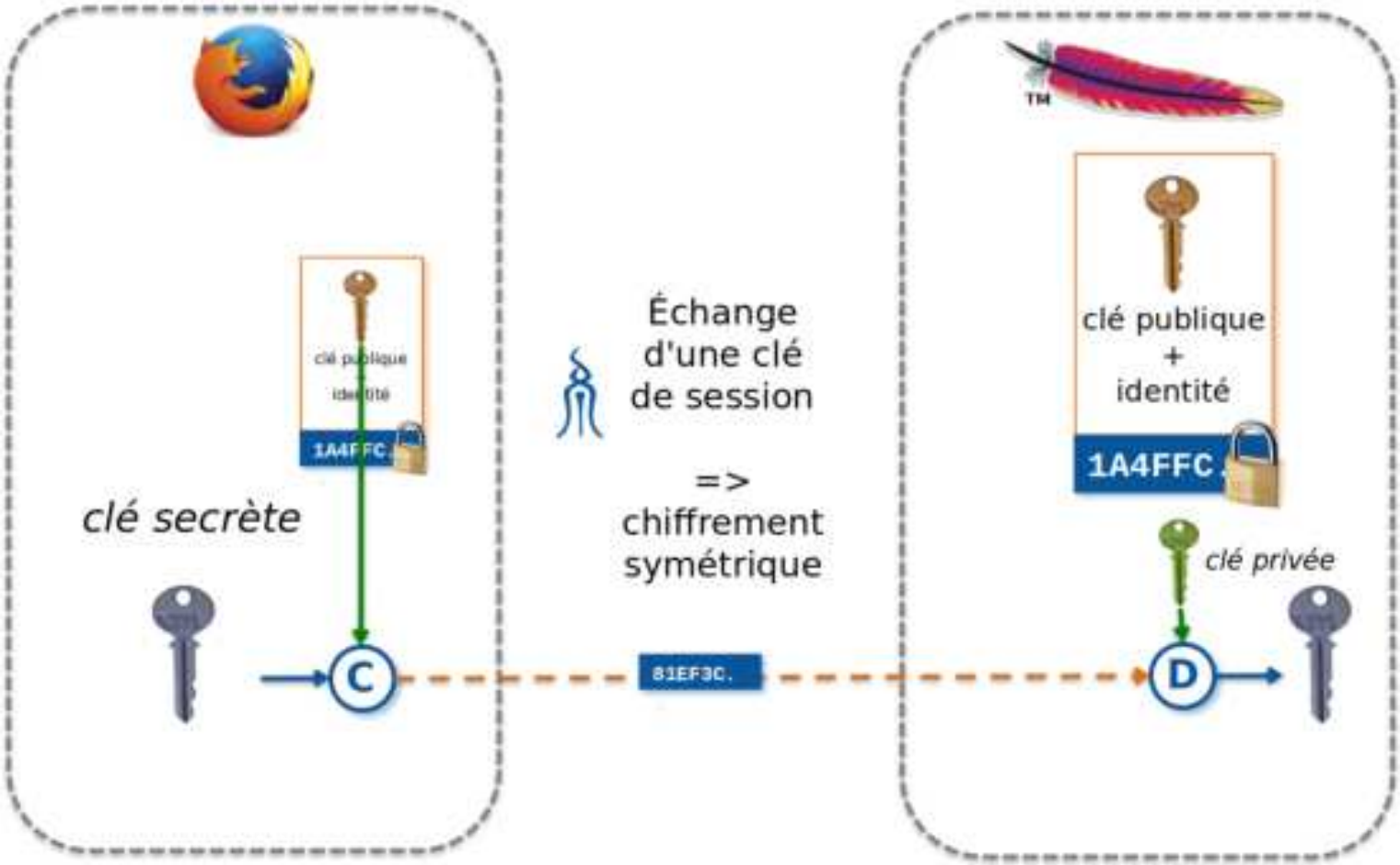
Échanges HTTPS (TLS 1.2) – *Handshake*



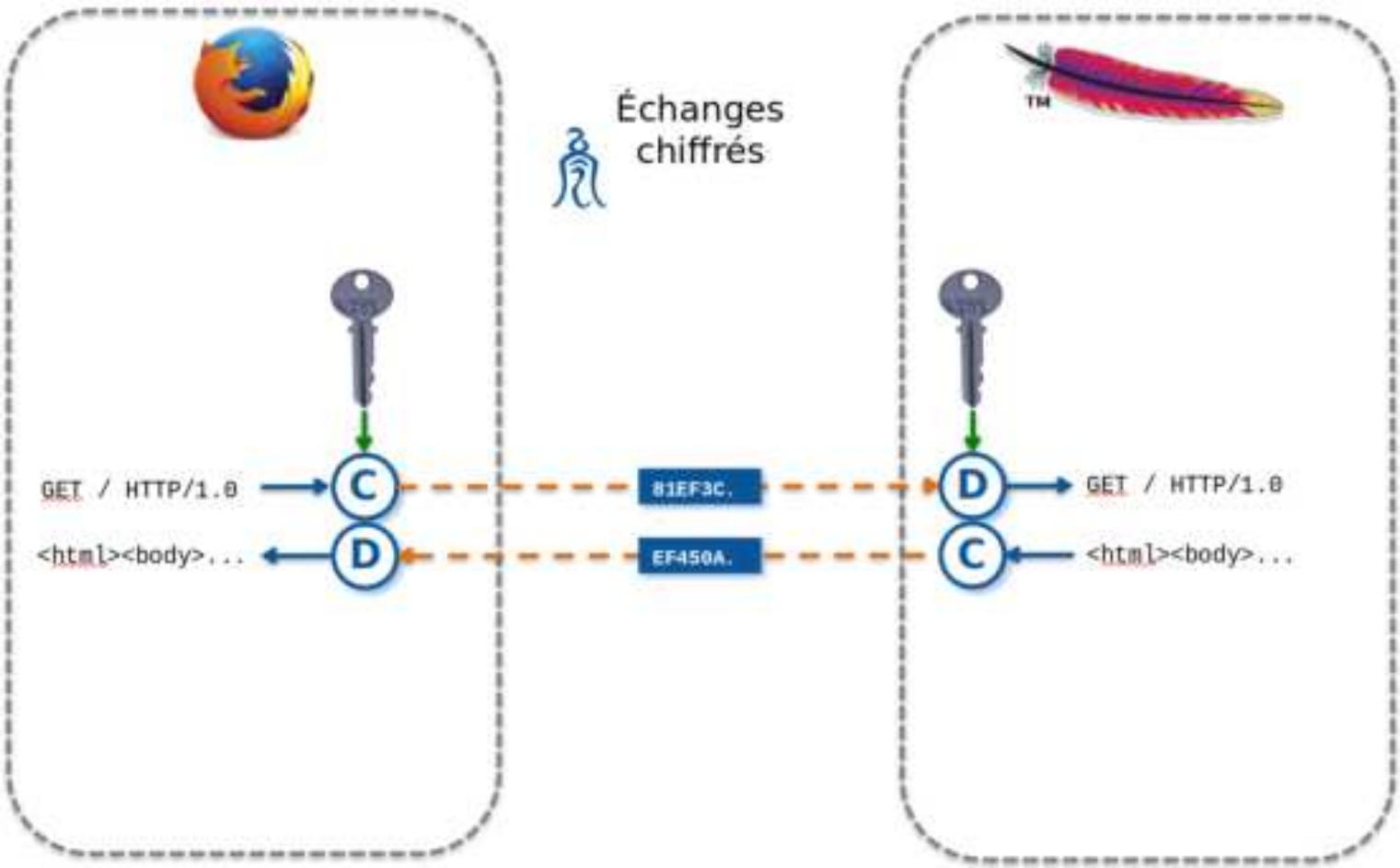
Échanges HTTPS (TLS 1.2) – *Handshake*



Échanges HTTPS (TLS 1.2) – *Handshake*



Échanges HTTPS (TLS 1.2)



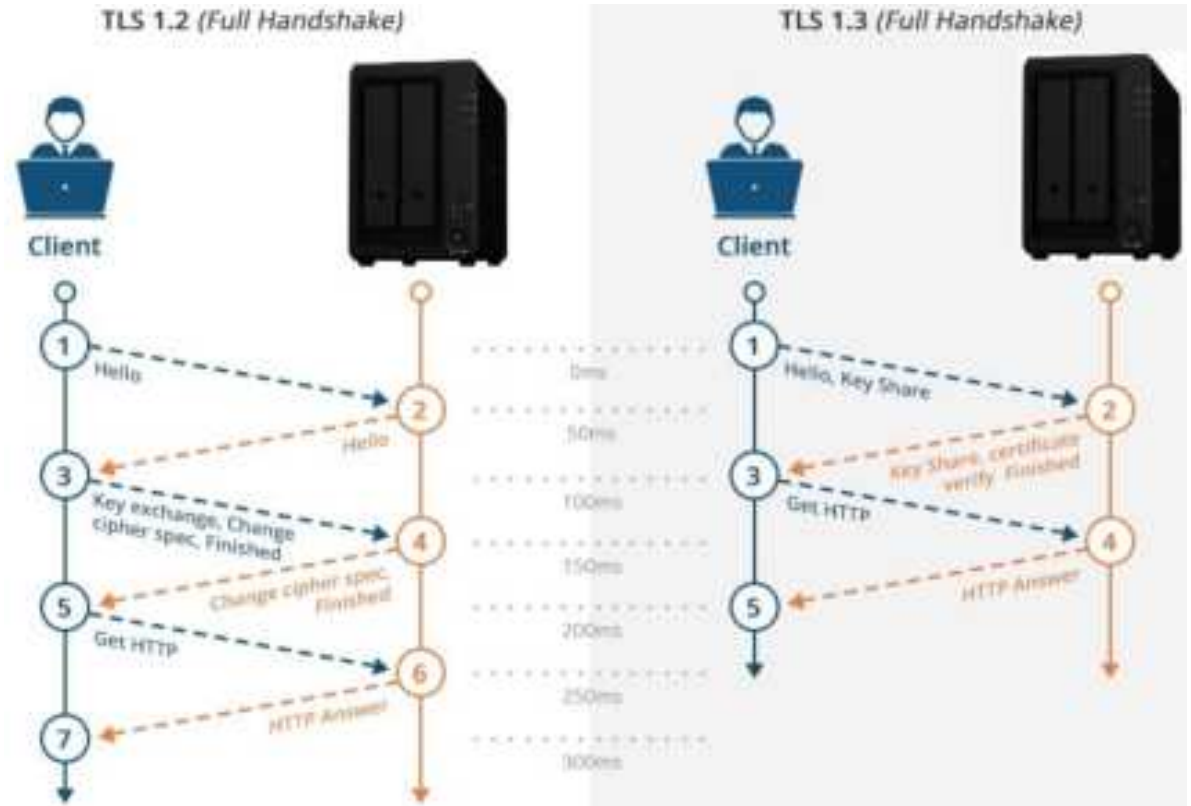
IUT2
Université Grenoble Alpes
Département
INFO

Sommaire

IUT2 Université Grenoble Alpes
Département INFO

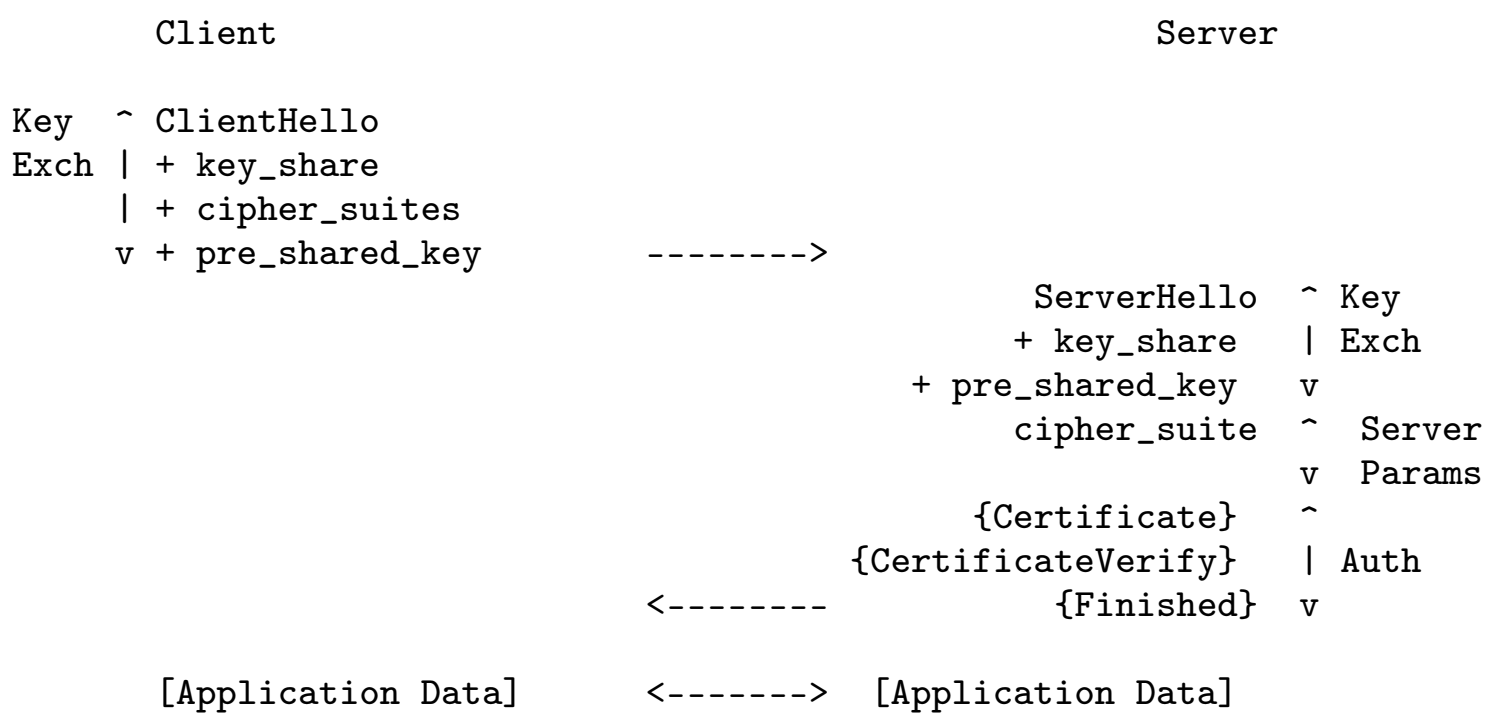
Échanges HTTPS Comparaison TLS 1.2 vs TLS 1.3

TLS 1.3 [8446, 2018] simplifie le handshake initial. Le chiffrement commence dès après la reception du message **ServeurHello**.



source : mariushosting.com

TLS 1.3 handcheck (HTTPS)



ClientHello : version du protocole, la suite aléatoire du client (`key_share` ou `pre_shared_key`) et une liste de suites cryptographiques (`cipher_suites`).

ServeurHello : les paramètres cryptographiques négociés (`cipher_suite`), la suite aléatoire du serveur (`key_share` ou `pre_shared_key`),le certificat (`Certificate`), une signature de l'ensemble de l'échange + le certificat version du protocole (`CertificateVerify`), Le hash (MAC) de tout le handshake (`Finished`)

Chiffrement symétrique à l'aide des clés échangées (`key_share` ou `pre_shared_key`)

IUT2 Université Grenoble Alpes
Département INFO

TLS 1.3 : *Cipher suites* (suites cryptographiques)

```
$ /usr/bin/openssl ciphers -v
TLS_AES_256_GCM_SHA384          TLSv1.3 Enc=AESGCM(256)             Mac=AEAD
TLS_CHACHA20_POLY1305_SHA256    TLSv1.3 Enc=CHACHA20/POLY1305(256) Mac=AEAD
TLS_AES_128_GCM_SHA256          TLSv1.3 Enc=AESGCM(128)             Mac=AEAD
```

Enc = Encrypt
Mac = Message Authentication Code

IUT2 Université Grenoble Alpes
Département INFO

HTTPS (TLS 1.3) resumé

HTTPS = HTTP transporté par TLS.

Authentification par certificat

TLS assure :

- Authentification (Certificat + MAC) : s'assure que les parties qui échangent des informations sont bien celles qu'elles prétendent être.
- Chiffrement (Enc) : chiffre les données échangées pour les cacher aux tiers.
- Intégrité (MAC) : vérifie que les données n'ont pas été altérées durant l'échange.

TLS est également utilisé avec SMTP (Simple Mail Transfer Protocol Secure (SMTPS)) ou avec FTP (FTP Secrure (FTPD))

IUT2 A
Université Grenoble Alpes
Département INFO

Sommaire

IUT2 Université Grenoble Alpes
Département INFO

Secure Shell (SSH)

Secure SHell (SSH) est un protocole pour obtenir des communications sécurisées (RFC4251 [Ylonen et Lonvick, 2006]).

Il chiffre et compresse un tunnel de session qui sécurise les données transmises

- le mot de passe est chiffré
- les informations échangées sont chiffrées

Fonctionnement en mode client serveur. Les clients ssh demandent une ouverture de connexion au serveur sshd

- Serveur : sshd
- Clients : ssh, scp, sftp (ssh = slogin)
- Outils de gestion : ssh-add, ssh-agent, ssh-keygen
- Fichiers de configuration : /etc/ssh et $HOME/.ssh

IUT2 Université Grenoble Alpes
Département INFO

Shell interactif

- Ouvrir un shell :
  `ssh [user@]adresse`
- Pour quitter :
  `exit`
- Par défaut, utilise le **port 22** (= port d'écoute de sshd).
- Changer le port :
  `ssh -p port [user@]adresse`
- Exécuter une commande à distance :
  `ssh [-p port] [user@]adresse commande [arguments]`

IUT2 A
Université Grenoble Alpes
Département INFO

Étapes du protocole

- Connexion TCP (3-way handshake) au **port 22** du serveur.

KEYINIT Key Exchange Initialization.

⇒ Échange des différents algorithmes de cryptographie supportés par le client et le serveur.

- Négociation des algorithmes de chiffrement : *ciphers* du client vers le serveur (ctos) et du serveur vers le client (stoc).
- Négociation des algorithmes d'intégrité **Message Authentication Code (MAC)** *ctos* et *stoc*.

KEX Key Exchange.

⇒ Création d'un secret partagé symétrique qui servira a crypter le trafic et échanger les clefs.

- Les échanges suivants sont cryptés par cette nouvelle clef de session.
- Le client envoie une clef publique en utilisant la clef publique du serveur (s'il ne l'a pas, il doit d'abord l'accepter, ou la récupérer sur un serveur tiers.)
- La clef de session est changée régulièrement.

IUT2 A
Université Grenoble Alpes
Département INFO

Sommaire

IUT2 Université Grenoble Alpes
Département INFO

Organisation des clés – Client

- Une paire de clés publique et privée créées avec `ssh-keygen`
  - `~/.ssh/id_ed25519` (privée) Ne JAMAIS partager
  - `~/.ssh/id_ed25519.pub` (public)
- Les clés publiques de "confiance", acceptées au premier log in, sont stockées dans
  - `~/.ssh/known_hosts`

IUT2 Université Grenoble Alpes
Département INFO

Organisation des clés – Serveur

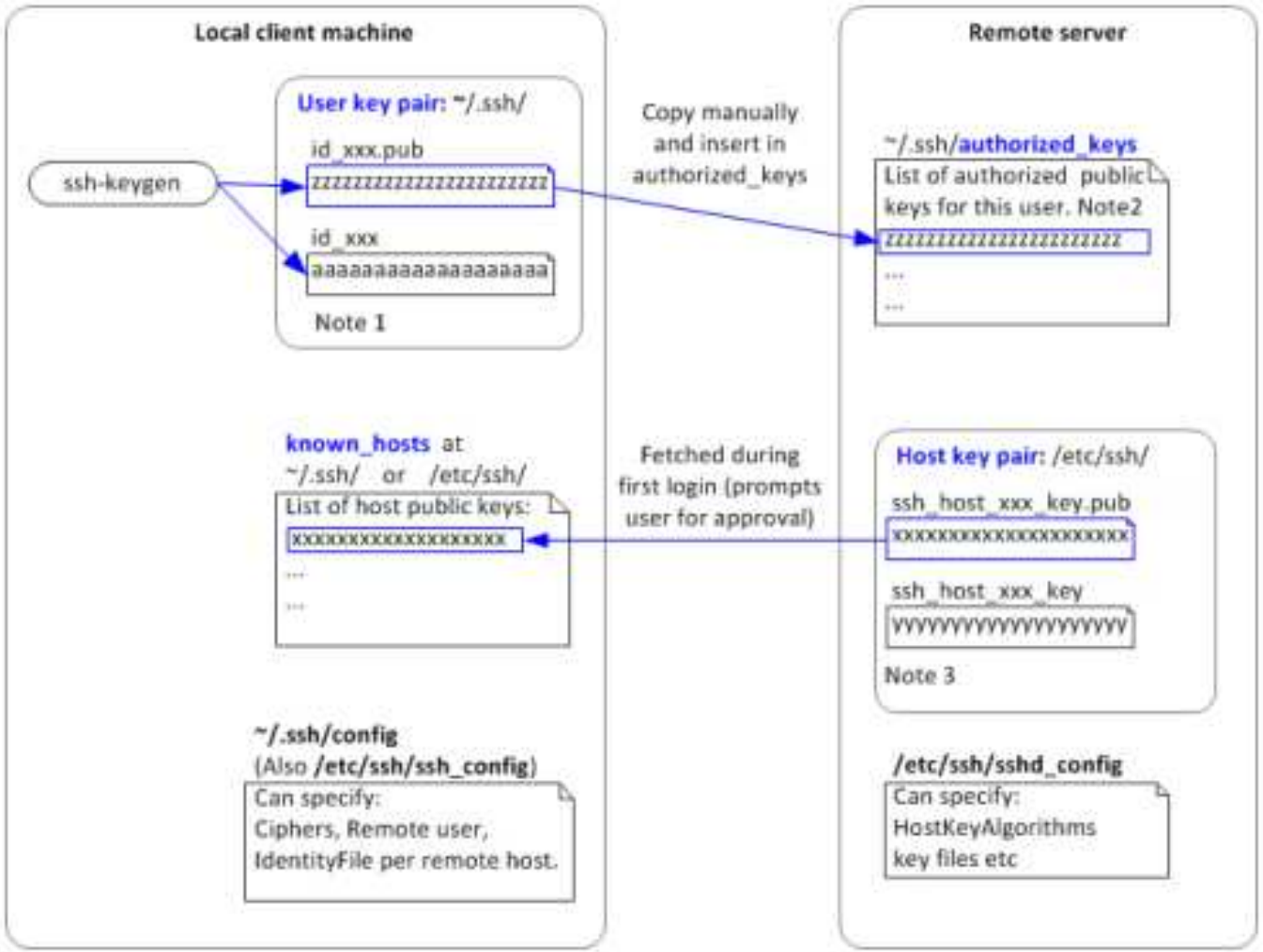
- Une paire de clés publique et privée créées à l'installation
  - `/etc/ssh/ssh_host_ed25519_key` (private) n'est JAMAIS partagé
  - `/etc/ssh/ssh_host_ed25519_key.pub` (public) → envoyée et stockée dans le `known_hosts` du client
- Les clés publiques des clients de "confiance", sont préalablement écrites dans
  - `~/.ssh/authorized_keys`

IUT2 Université Grenoble Alpes
Département INFO

Sommaire

IUT2 Université Grenoble Alpes
Département INFO

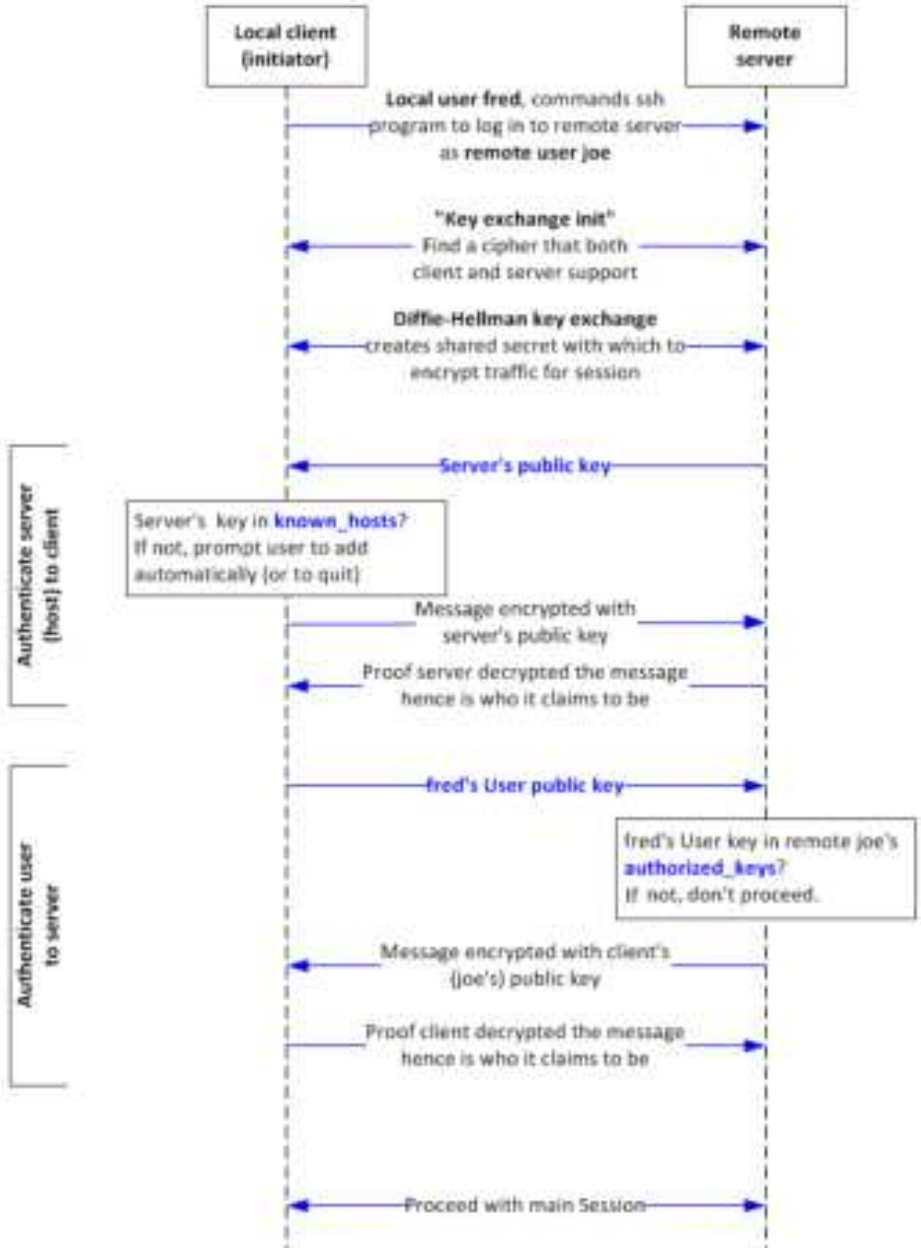
Échange des clefs



crédit : gwideman 2018-10-14

IUT2 Université Grenoble Alpes
Département INFO

SSH – connexion



IUT2 Université Grenoble Alpes
Département INFO

crédit : gwideman 2018-10-14

SSH resumé

SSH = Accès distant sécurisé : Fournit une méthode sécurisée d'accès à un ordinateur sur un réseau non sécurisé.

SSH assure :

- Authentification : que l'ordinateur distant et l'utilisateur sont bien celui qu'ils prétendent être.
- Chiffrement : Utilise le chiffrement pour sécuriser les commandes envoyées à un ordinateur et les réponses reçues.
- Intégrité : vérifie que les données n'ont pas été altérées durant l'échange.

SSH est également utilisé pour Secure Copy Protocol (SCP), SSH File Transfer Protocol (SFTP) et SSH Filesystem (SSHFS)

IUT2 A
Université Grenoble Alpes
Département INFO

Glossaire I

FTPD FTP Secrure. 61

HTTP Hypertext Transfer Protocol. 3, 5, 23
HTTPS Hypertext Transfer Protocol Secure. 27, 29

MAC Message Authentication Code. 65

SCP Secure Copy Protocol. 72
SFTP SSH File Transfer Protocol. 72
SGML Standard Generalized Markup Language. 3

SMTPS Simple Mail Transfer Protocol Secure. 61
SSH Secure SHell. 63
SSHFS SSH Filesystem. 72
SSL Secure Sockets Layer. 41

TLS Transport Layer Security. 27, 29, 30, 40, 41, 48, 49, 57

URL Uniform Resource Locator. 5–7

W3C World Wide Web Consortium. 3
WWW World Wide Web. 3

IUT2 A
Université Grenoble Alpes
Département INFO

Références I


8446, R. (2018).
The Transport Layer Security (TLS) Protocol Version 1.3.
RFC 8446.
Eric Rescorla.


Chen, L., Jordan, S., Liu, Y., Moody, D., Peralta, R., Perlner, R. et Smith-Tone, D. (2016).
Report on Post-Quantum Cryptography, NIST Interagency/Internal Report (NISTIR).
National Institute of Standards and Technology, Gaithersburg, MD, USA).


Fielding, R. T., Nielsen, H., Mogul, J., Gettys, J. et Berners-Lee, T. (1997).
Hypertext Transfer Protocol – HTTP/1.1.
RFC 2068.


Rescorla, E. (2000).
HTTP Over TLS.
RFC 2818.


Ylonen, T. et Lonvick, C. (2006).
The Secure Shell (SSH) Protocol Architecture.
RFC 4251.

IUT2 A
Université Grenoble Alpes
Département INFO