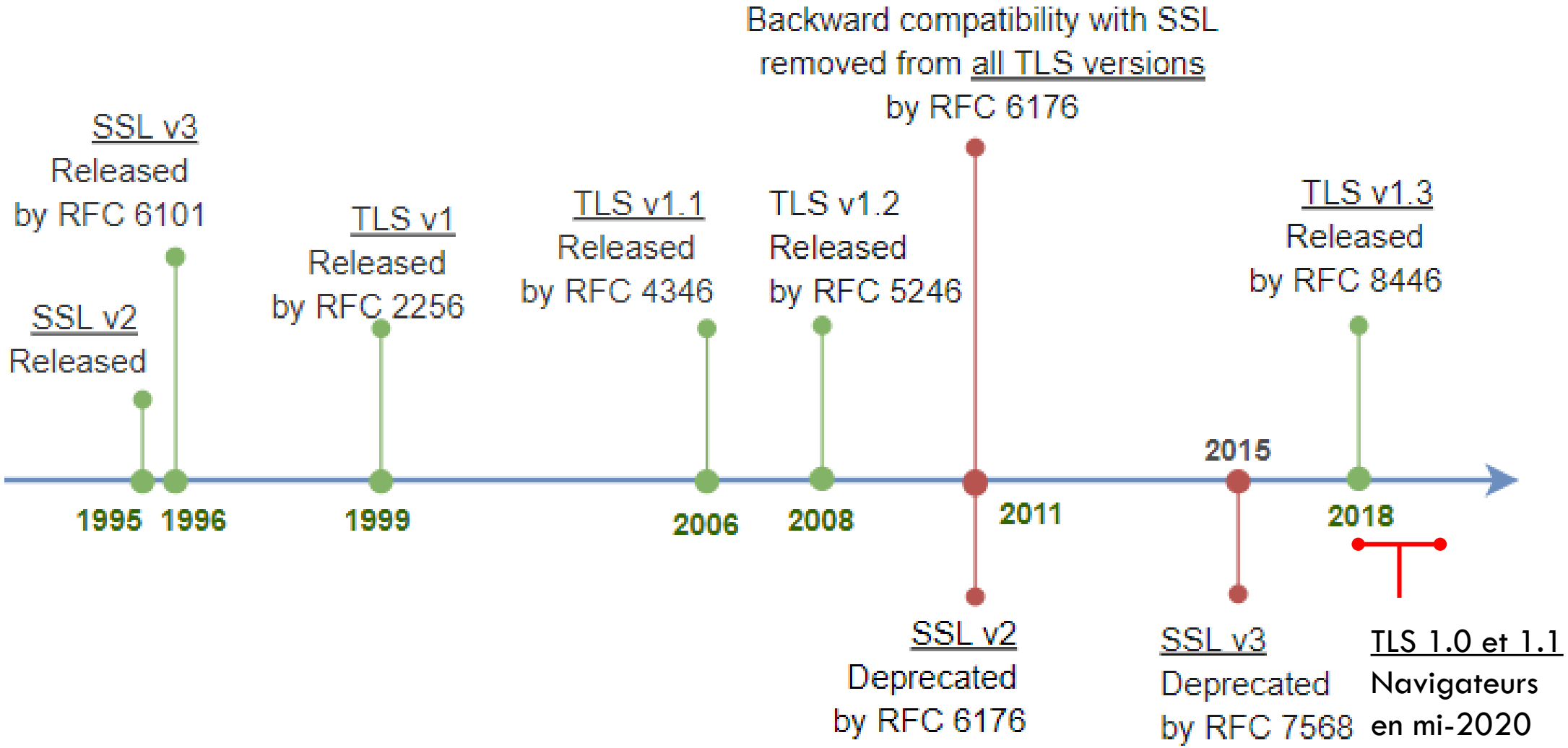

Bib'INSA
Restaurant
Cafétéria
Service Scolarité
Service Relations
Internationales
Salle des Thèses

TLS 1.3

UN PEU D'HISTOIRE



POURQUOI TLS 1.3 ?

Combler définitivement les failles historiques

- Ce sont aujourd'hui des failles utilisateur (version de la librairie SSL, ciphers faibles..)
- Nettoyage et départ sur une base propre

Améliorations de sécurité, vie privée et performance (on en parle après)

Mais: compatibilité à assurer (migration de la pile tls est longue)

LE DÉVELOPPEMENT DE TLS 1.3 (1)

La version finale de TLS est été livrée en aout 2018

- Et pourtant…

LE DÉVELOPPEMENT DE TLS 1.3 (2)

Comment on développe un protocole aussi important ?

- Nécessité de test à grande échelle
- Système de « drafts », version incrémentales du protocole
  - Chaque draft a un numéro de version protocolaire (cohabitation)
  - Implémenté dans une grand partie des libraires en version RC
  - Implémenté dans Chrome / Firefox (pas toujours le même draft)

- Fin 2017, la plupart des services de l'Université Fédérale (SCOUT / CAS ..) répondaient en TLS 1.3 (compilation de openssl pre-1.1.1)

LE DÉVELOPPEMENT DE TLS 1.3 (3)

Beaucoup de correctifs concernent des problèmes d'implémentation

- Firewall / analyseurs de trafic qui attendent du TLS 1.2
- Bugs de certaines libraires

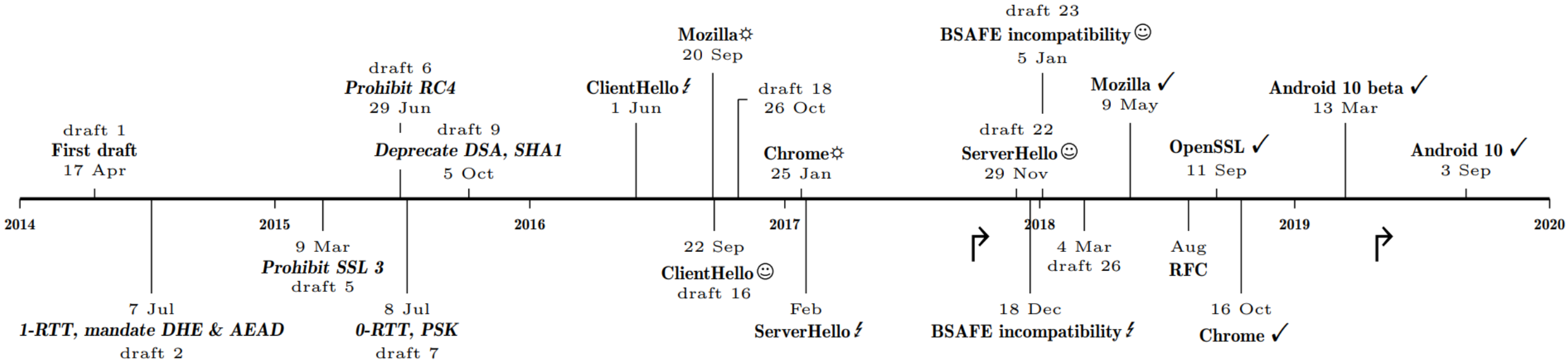


Figure 1: Timeline for TLS 1.3; important changes in italic (see [50]). We highlight some critical issues (↯) found in trials and fixed (☺). Support was sometimes first optional (☼), later a default (✓). The arrows (↱) show the start times of draft/RFC scans.

RÉCAP:

TLS 1.3 ressemble protocolairement à TLS1.2

- KeyShare, supported_versions, PSK are extensions
- HRR looks like ServerHello, distinguished by Random field
  - Random set to SHA-256 of "HelloRetryRequest"
- Real protocol version in supported_version extension
- session_id and compression restored back in ServerHello
- Dummy Change Cipher Spec (CCS)
- GREASE mechanism

TLS 1.3 mitige toutes les failles TLS1.2 apparues avant sa version définitive

Lien utile détaillant les trames: https://tls13.ulfheim.net/

MODIFICATIONS SÉCURITÉ (1)

Algorithmes d'échange:

- plus que 5 groupes DHE + ECDHE

Ciphers:

- Elimination de l'échange de clé et de l'authentification dans la spécification
- On passe de :
  - 37 cipher suite en TLS 1.2
  - 5 cipher suite en TLS 1.3 (pour le moment)
    - TLS_AES_128_GCM_SHA256
    - TLS_AES_256_GCM_SHA384
    - TLS_CHACHA20_POLY1305_SHA256
    - TLS_AES_128_CCM_SHA256
    - TLS_AES_128_CCM_8_SHA256

Perfect Forward Secrecy systématique (exit RSA key exchange)

MODIFICATIONS SÉCURITÉ (2)

- Key Exchange
  - RSA
- Encryption algorithms:
  - RC4, 3DES, Camellia.
- Cryptographic Hash algorithms:
  - MD5, SHA-1.
- Cipher Modes:
  - AES-CBC.
- Other features:
  - TLS Compression & Session Renegotiation.
  - DSA Signatures (ECDSA ≥ 224 bit).
  - ChangeCipherSpec message type & “Export” strength ciphers.
  - Arbitrary/Custom (EC)DHE groups and curves.

RC4
- Roos’s Bias 1995
- Fluhrer, Martin & Shamir 2001
- Klein 2005
- Combinatorial Problem 2001
- Royal Holloway 2013
- Bar-mitzvah 2015
- NOMORE 2015

RSA-PKCS#1 v1.5 Encryption
- Bleichenbacher 1998
- Jager 2015
- DROWN 2016

Renegotiation
- Marsh Ray Attack 2009
- Renegotiation DoS 2011
- Triple Handshake 2014

3DES
- Sweet32

AES-CBC
- Vaudenay 2002
- Boneh/Brumley 2003
- BEAST 2011
- Lucky13 2013
- POODLE 2014
- Lucky Microseconds 2015

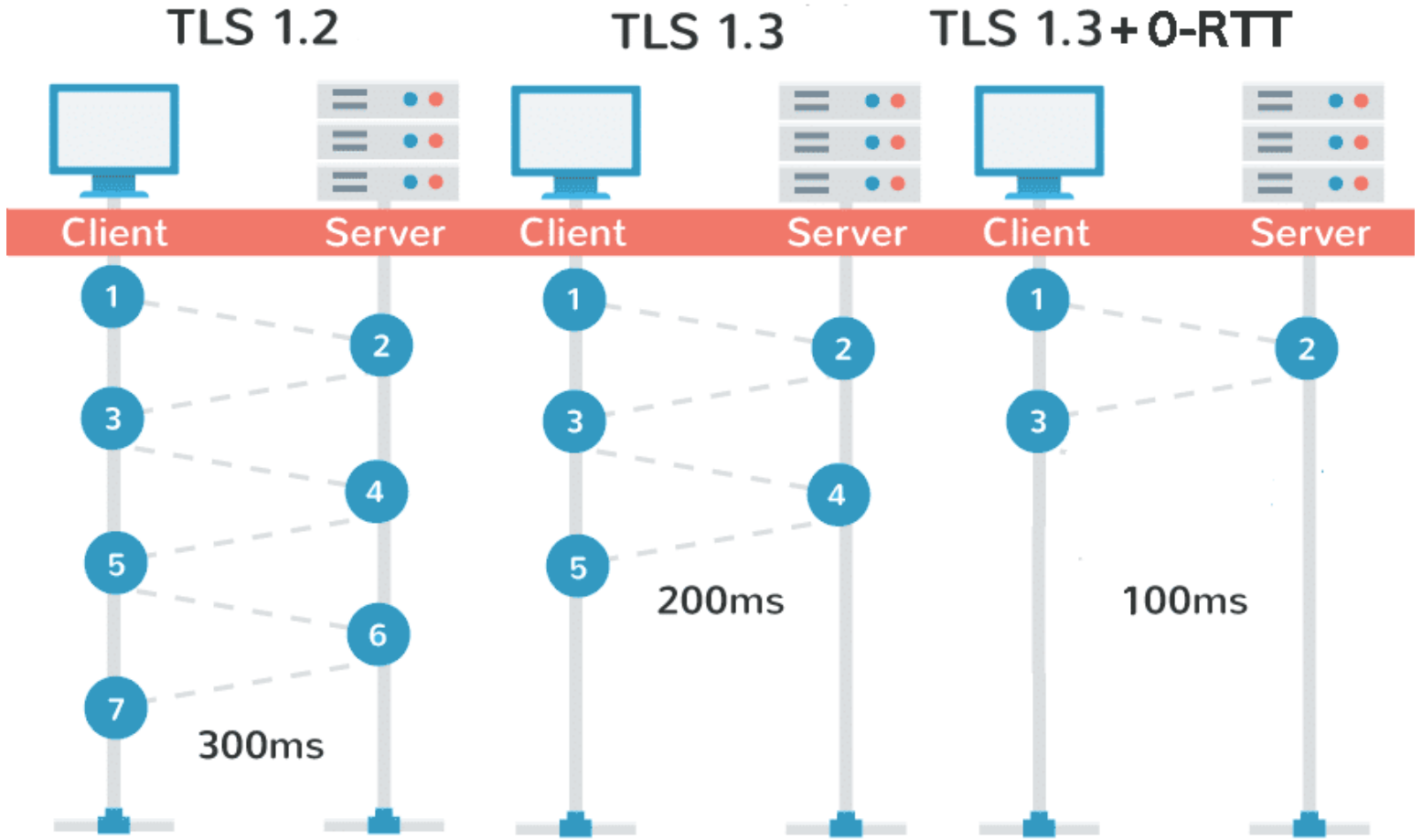
Compression
- CRIME 2012

MD5 & SHA1
- SLOTH 2016
- SHAttered 2017

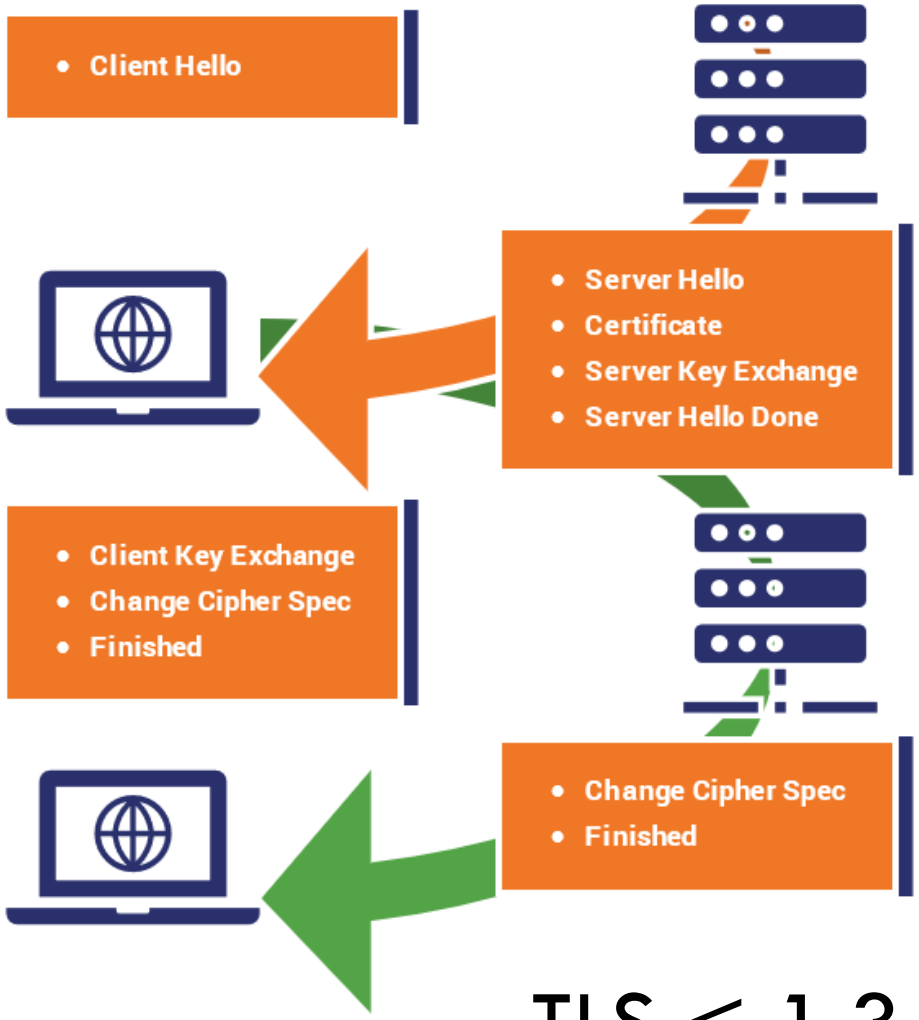
CHANGEMENTS SUR LE PROTOCOLE (1)

- TLS1.2 = 2 RTT
  TLS1.3 = 1 RTT
  ou 0 RTT

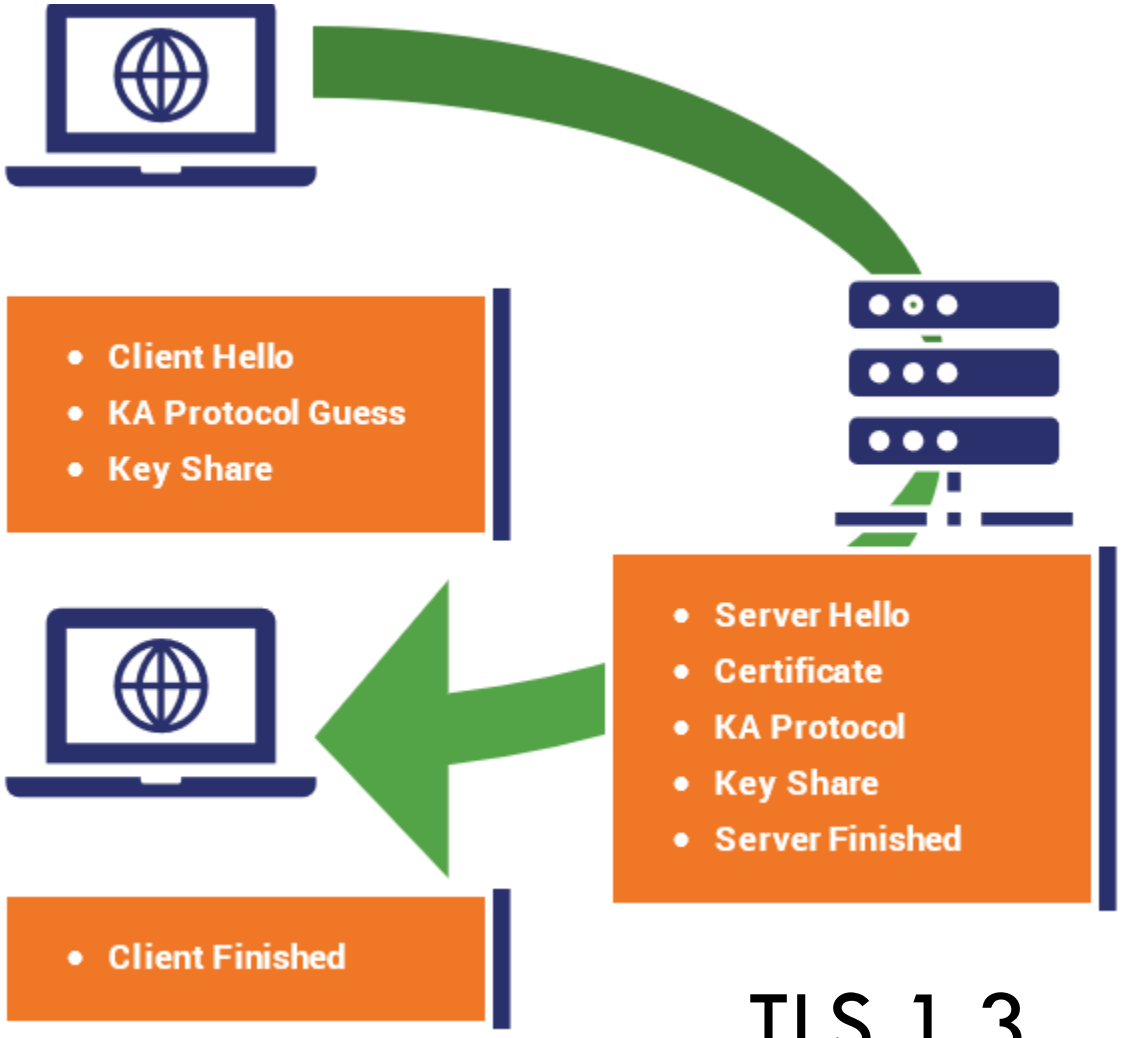
- Intérêt sur les réseau à latence



Key exchanges, and by extension the digital signature scheme no longer require negotiation.
=> Un échange de moins



TLS < 1.3



TLS 1.3

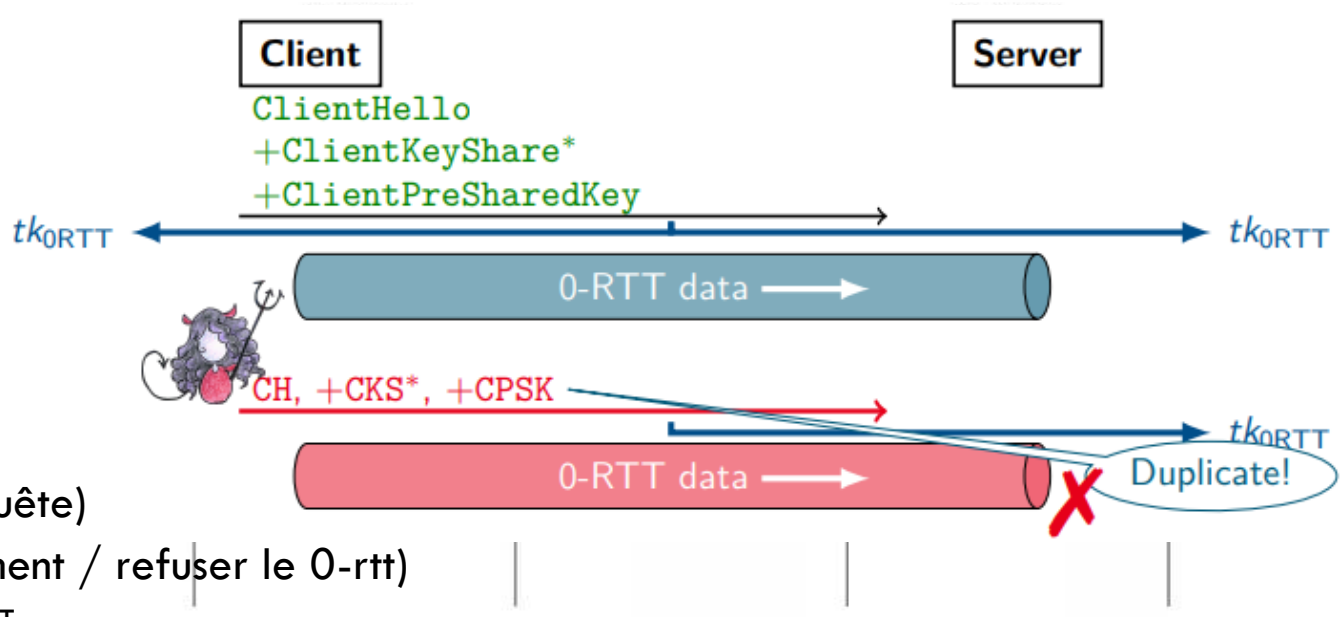
CHANGEMENTS SUR LE PROTOCOLE (3)

De 1 RTT à 0 RTT (Resumption)

- Envoyer des données AVANT l'authentification
- Cloudflare estime que 40% des connexions sont des « reprises »

Mais

- Pas de PFS (utilisation de la précédente clé)
- PSK: stockage de la clé
- Replay attack



Solution

- Protection applicative (identifiant unique de requête)
- Protection serveur web (attendre conditionnellement / refuser le 0-rtt)
  - Ex HAProxy: http-request wait-for-handshake if METH_POST
  - Cloudflare has chosen to disallow 0-RTT for all but GET requests with no query string

CHANGEMENTS SUR LE PROTOCOLE (4)

Poignée de main complètement chiffrée

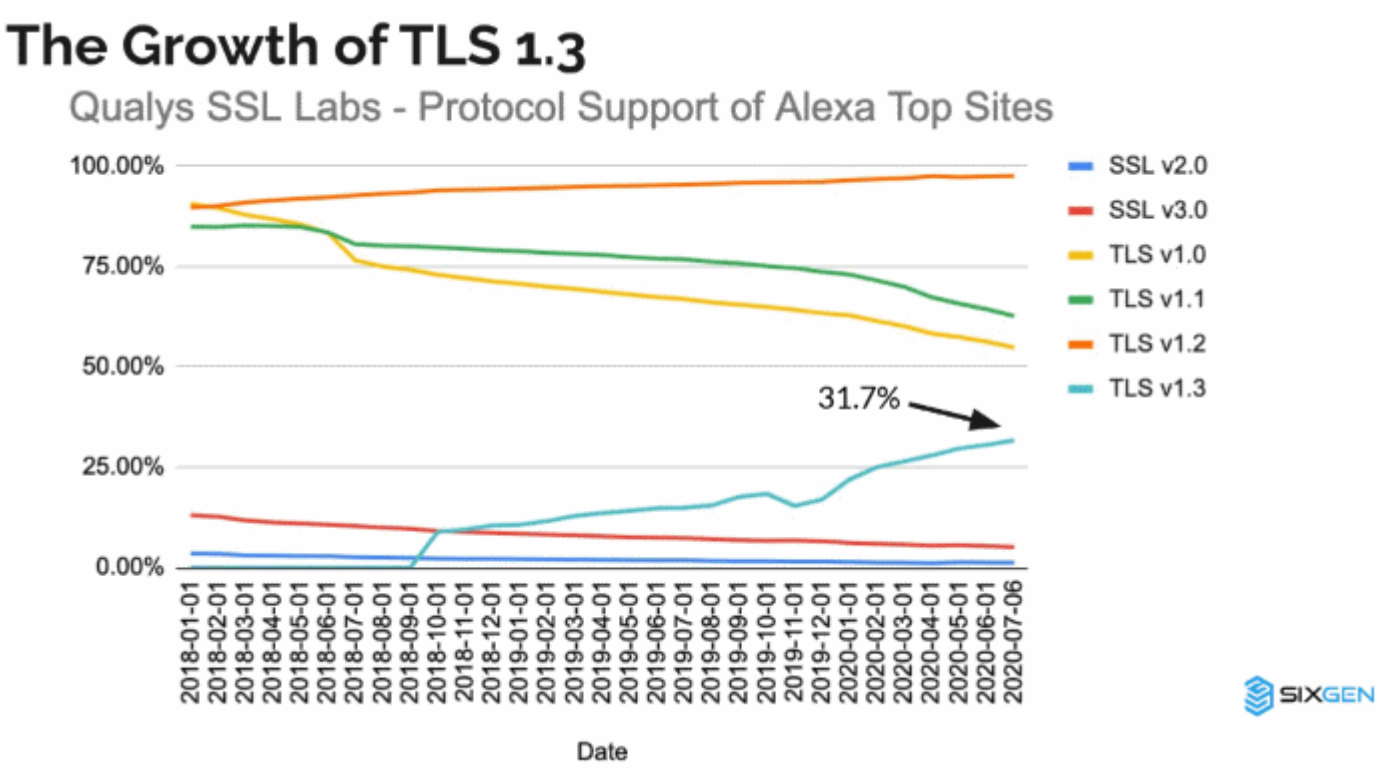
- Le certificat y compris
- Complexifie le travail de certains firewall

Downgrade detection

- Si pas TLS1.3, ServerHello.random termine par 44 4F 57 4E 47 52 44 01 (⇔ DOWNGRD)
  - Le client est censé reconnaître la supercherie

TLS 1.3 AUJOURD'HUI (1)

Quelques stats



TLS 1.3 AUJOURD'HUI (2)

Librairies

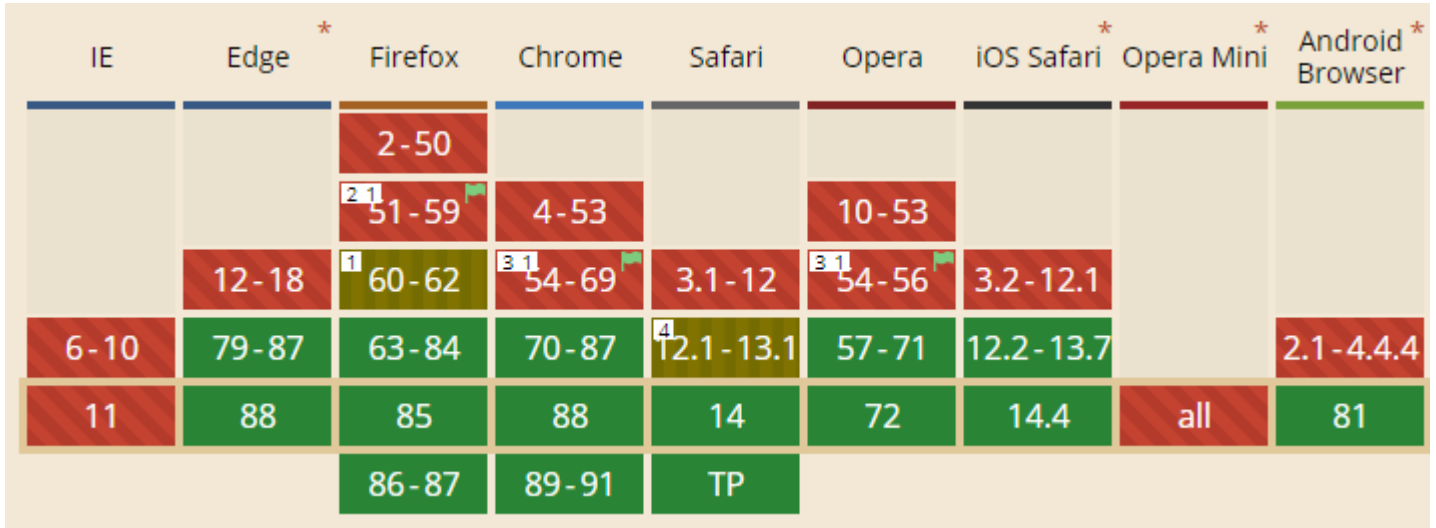
- Openssl 1.1.1 (Ubuntu 18.04, R)
- GnuTLS: 3.5.x
- NSS: oui
- Schannel (Microsoft): Windows 10 21H1 ? Pas serveur
- Secure transport (Apple): yes

Navigateurs

OS

- Ubuntu 18.04+
- Debian buster
- RedHat: ??
- CentOS (RIP): ??

Windows OS	TLS 1.0 Client	TLS 1.0 Server	TLS 1.1 Client	TLS 1.1 Server	TLS 1.2 Client	TLS 1.2 Server	TLS 1.3 Client	TLS 1.3 Server
Windows 10, version 1903	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Not supported	Not supported
Windows 10, version 1909	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Not supported	Not supported
Windows 10, version 2004	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Not supported	Not supported
Windows 10, version 20H2	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Not supported	Not supported
Windows 10, version 21H1	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled



TLS 1.3 AUJOURD'HUI: EXEMPLE DE CONFIG.

Apache

SSLCipherSuite ECDH+AESGCM:ECDH+CHACHA20:ECDH+AES256:ECDH+AES128:!aNULL:!SHA1:!AESCCM
SSLCipherSuite TLSv1.3 TLS_AES_256_GCM_SHA384:TLS_AES_128_GCM_SHA256:TLS_CHACHA20_POLY1305_SHA256

HAProxy

ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
ssl-default-bind-ciphers ECDH+AESGCM:ECDH+CHACHA20:ECDH+AES256:ECDH+AES128:!aNULL:!SHA1:!AESCCM
bind fqdn:443 tfo ssl crt fqdn.pem allow-0rtt ssl-min-ver TLSv1.2
http-request wait-for-handshake if METH_POST

BILAN

Doit-on migrer urgemment sur TLS 1.3 ?

- Non, une bonne configuration de TLS 1.2 est sécurisée aujourd'hui !

Peut-on migrer sur TLS 1.3 ?

- No problem' (si on est sage sur l'inspection protocolaire…)
- Vous ne ferez pas pire que les SMTP d'Orange qui sont toujours en TLS v1.0 (1999)

Facteurs de décision:

- Pour clients avec latence
- Simplifier la configuration (mais TLS 1.2 doit encore être supporté)

VRAC

DTLS 1.3: pas encore là

- Draft 41
- Proche d'une publication ?
- https://datatracker.ietf.org/doc/draft-ietf-tls-dtls13/

Encrypted SNI (Server Name Indication)

- Draft IETF: extension https://tools.ietf.org/html/draft-ietf-tls-esni-09
- Chiffrement via une clé dans DNS(SEC)
- CloudFlare le supporte
- Le Great Firewall bloque TLS 1.3 + ESNI

FIN

systemctl stop louis.worker

systemctl start philippe.worker

systemctl start capitoul-plaintext.service

SOURCES

https://ralphholz.science/publications/TrackingTheDeploymentOfTls13OnTheWebAStoryOfExperimentationAndCentralization.pdf

https://blog.cloudflare.com/why-tls-1-3-isnt-in-browsers-yet/

https://connect.ed-diamond.com/MISC/MISC-105/RFC-8446-TLS-1.3-que-faut-il-attendre-de-cette-nouvelle-version

https://ccronline.sigcomm.org/wp-content/uploads/2020/08/sigcomm-ccr-paper430-with-open-review.pdf

https://www.slideshare.net/Vtzslavek/tls-13final13-99191807

https://owasp.org/www-chapter-london/assets/slides/OWASPLondon20180125_TLSv1.3_Andy_Brodie.pdf

https://dev.to/techschoolguru/a-complete-overview-of-ssl-tls-and-its-cryptographic-system-36pd

https://www.cryptologie.net/article/390/keeping-up-with-tls-13/

https://www.fasterize.com/fr/blog/tls-1-3-la-nouvelle-version-du-web-securise/

https://www.haproxy.com/fr/blog/tls-1-3-0-rtt-haproxy/