STORMSHIELD

Network Endpoint Data

Transition vers la cryptographie post-quantique

Stormshield


WANTED
DEAD & ALIVE
SCHRÖDINGER'S CAT

Ordinateurs **Quantiques**

C’est quoi ces trucs ???

N’essayez pas de comprendre par l’intuition !

Qubit ?

Ou Qu-bit, ou Qbit (/'kju:bɪt/)

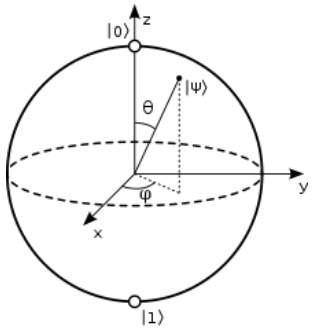
Différentes technologies: Superconducting, Trapped ion, Photonic, Silicon-based, Topological, Neutral atom, …

Propriétés amusantes

Superposition d’états quantiques,

Téléportation (Intrication quantique),

….



Etat quantique

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

α et β: probabilités d’amplitudes (nombres complexes)

L’informatique quantique n’est PAS

Le futur de l’informatique

« un système qui calcule toutes les valeurs possibles en parallèle »

A propos des **Qubits**

“Stable” vs Noisy

+ quantum annealing

Etat actuel

~1200 Qubits « stables »

Record de cohérence: 40 secondes ?

27 Qubits / 133 Qubits « vraiment exploitables »

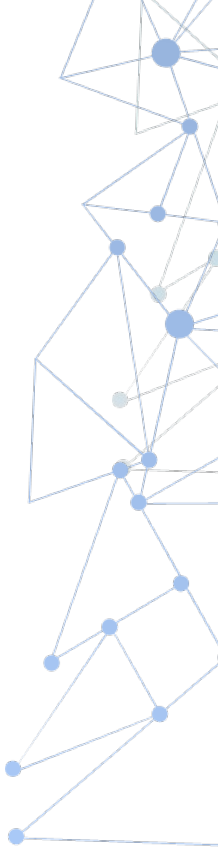
Cohérence: < 1 seconde ?

Conséquences de l'informatique quantique sur la **Cryptographie**

La très bonne nouvelle

Algorithmes
De hashes

Pas d'attaque quantique
efficace connue



La bonne nouvelle

Chiffrement **Symétrique**

L'algorithme de Grover réduit le temps d'attaque : nombre de bits de sécurité /2

AES 256 resterait **aussi solide qu'AES 128** l'est aujourd'hui
(ChaCha20 aussi)

Peu d'impact sur les utilisations de chiffrement symétrique

Implémentation de Grover pour AES ???

Cryptographie asymétrique

Les mauvaises nouvelles

Algorithme de **Shor**

Aide à calculer rapidement les clés privées RSA et ECC

(Non, il ne fait pas un “calcul simultané de toutes les valeurs possibles”)

Ordinateur quantique vs **RSA-2048**

~~4000~~ ! 2000 ?

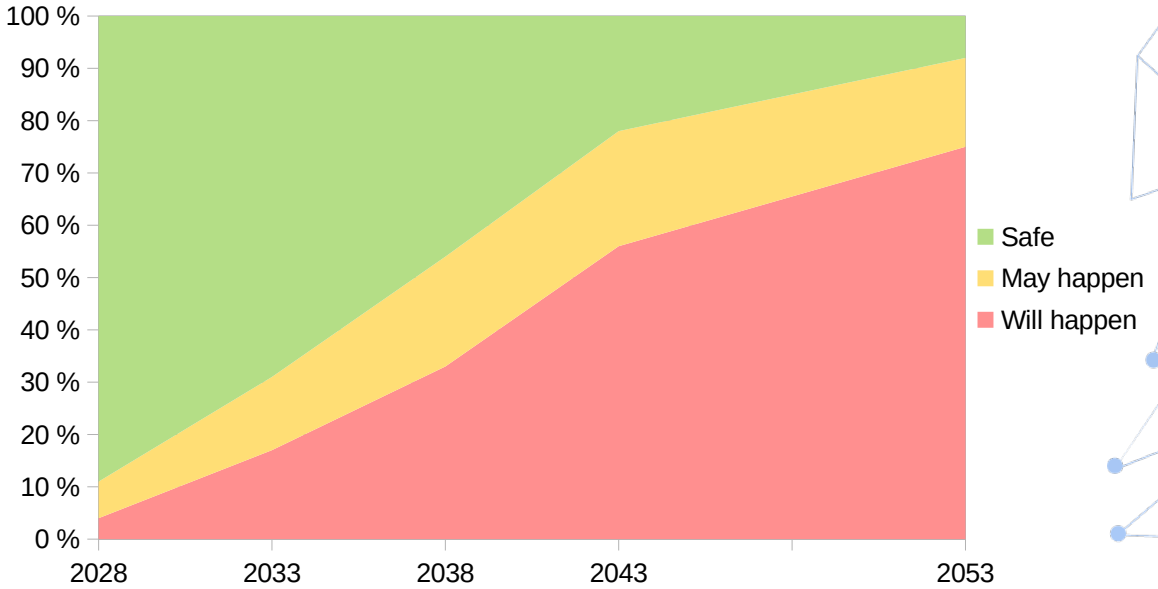
Nécessiterait ~ ~~6000+~~ Qubits (stables)

Nécessiterait au moins plusieurs minutes (heures ?) de fonctionnement

Quand RSA & ECC **seront cassés** ?

Un jour

Presque certain



Estimation d'opinions sur le risque qu'un ordinateur quantique soit capable de casser RSA-2048 en moins de 24h (étude de 2023)

Les conséquences

- Authentification → Usurpation d'une identité pour s'authentifier

- Certificats à durée de validité limitée (1 an?)
  - → Un certificat d'aujourd'hui ne sera plus valable dans 5 ans
- Renouvellement des PKIs à planifier
  - Doit être fini un peu avant le Qday (mois ? Semaines ?)

- Pire des cas
  - Suspendre les authentifications
  - Anéantissement du Web Of Trust ?

Les conséquences

- Signature → Signature frauduleuse

- Différents cas d'usage
  - Documents administratifs
  - Mises à jour (logiciel, firmware, …..)

- Inertie sur les autorités de confiance à anticiper
- Contre signature de masse quelques mois avant le Qday ?
- Pire des cas : perte de confiance en les données signées

Les conséquences

- Chiffrement → lecture de données confidentielles !

- Attaque « Stocker maintenant, déchiffrer plus tard »
  - La phase de stockage est déjà en cours
  - Problématique en fonction de la durée de confidentialité attendue des données

- Pire des cas : lecture de données confidentielles !

Théorème de **Mosca**

Y	**X**
Temps de standardisation et d'adoption	Durée de confidentialité des données

Z

Temps avant une attaque par ordinateur quantique

X + Y > Z = Problème !

Temps de standardisation et d'adoption:

- *Conception et implémentation des standards*
- *Déploiement des nouvelles versions de firmware / logiciel*
- *Arrêt complet d'utilisation des anciennes versions de firmware / logiciel*

Solutions

Pour un monde post quantique sécurisé

Algorithmes post quantiques

Fondations **Cryptographiques**

Réseaux euclidiens (« lattice »)

Polynômes multivariés

Utilisation de fonctions de hachage

Isogénies des courbes elliptiques supersingulières

Cryptographie basée sur les codes (y compris correction d'erreurs)

Utilisation de clés symétriques……

Fonctionnent sur des ordinateurs classiques

Standardisation des algorithmes

Chiffrement

ML-KEM /FIPS 203 (NIST, réseaux euclidiens)
Anciennement « CRISTALS-Kyber »

FrodoKEM (ISO, réseaux euclidiens)
Pas encore en version finale

Signature

ML-DSA / FIPS 204 (NIST, réseaux euclidiens)
Anciennement « CRISTALS-Dilithium »

SLH-DSA / FIPS 205 (NIST, hashes)
Anciennement « SPHINCS+ »

Falcon (NIST, réseaux euclidiens)
FIPS ???

Autres candidats en cours

NIST: versions finales publiées (aout 2024)

« Y’a plus qu’à ! »

Les problèmes

Des solutions.....

Question de confiance

Algorithmes	Résistance actuelle	Résistance post-quantique
RSA, ECC, DH	Oui	Non
Algorithmes post-quantiques	Oui ?	Oui (*)

(*): pas d'attaque quantique connue à ce jour

Questions d'implémentations

Algorithmes complexes

Risques d'erreurs d'implémentation

Fuite d'informations ?

Attaques par canaux auxiliaires (« side channel »)

Implémentations fiables ?

Aujourd'hui, surtout des implémentations en « démonstrateur »

Recommandations

Migrer rapidement aux algorithmes PQSafe
« To be completed by 2035 »

Les nouveaux algorithmes pourraient avoir des vulnérabilités classiques
(et c'est arrivé à certains !)

Pas de PQSafe seul pour l'instant

Mode hybride: (RSA|ECC) + PQSafe

PQSafe seulement « plus tard » (2030+ ?)

Questions de performances ?

Parfois plus rapides que RSA / ECC !

Mais peu de protections contre les attaques par canaux auxiliaires

Consommation mémoire ?

Surtout en environnements embarqués / contraints

Accélération matérielle ?

Nécessaire à terme dans certains cas d'usage

Tailles de données transmises

Algorithme de chiffrement	Post-quantique ?	Clé publique	Echange
X25519 (ECC)	Non	32 octets	32 octets
ML-KEM512 (Euclidien)	Oui	800 octets	768 octets
ClassicMcEliece384464 (Code) (candidat)	Oui	261120 octets	128 octets
BIKE (Code) (candidat)	Oui	1540 octets	1572 octets

Tailles de données transmises

Algorithme de signature	Post-quantique ?	Clé publique	Signature
Ed25519 (ECC)	Non	32 octets	64 octets
ML-DSA II (Euclidien)	Oui	1312 octets	2420 octets
Falcon-512 (Euclidien)	Oui	897 octets	666 octets
SPHINCS+-128s (hachages)	Oui	32 octets	7856 octets
SPHINCS+-128f (hachages)	Oui	32 octets	17088 octets

Conséquences

Importantes en utilisation **Réseau**

On dépasse (largement) un paquet réseau
(Au tout début de la négociation !)

- Impact sur les performances (latence de négociation)
- Problématique pour les protocoles UDP (IKEv2, QUIC, ….)

Encore (un peu) pire en mode hybride

Besoin de nouvelles normes

Protocoles **réseau**

Support de certificats hybrides

Solutions pour les grosses clés / données

Echange de clés hybride

Autres normes

PKIs / X.509 !

Mode hybride pour les formats de signature / chiffrement de données

Mode « déconnecté »: impossible de négocier avec le correspondant

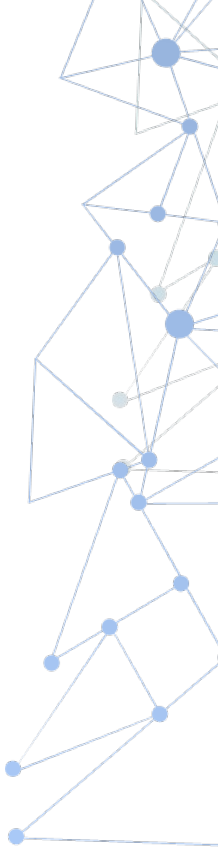
TLS / IPSec / réseau

RFCs + Drafts en cours

Echange de clés: Diffie-Hellman → Key Encapsulation Mechanism

Utilisation des nouveaux algorithmes
Mode hybride
Migration « facile »

Premiers échanges plus gros (pire pour IPSec) → latence supplémentaire
Pas d'impact sur le trafic suivant (chiffré en symétrique)



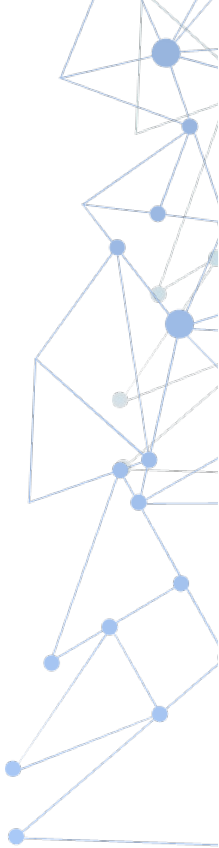
Et en pratique ?

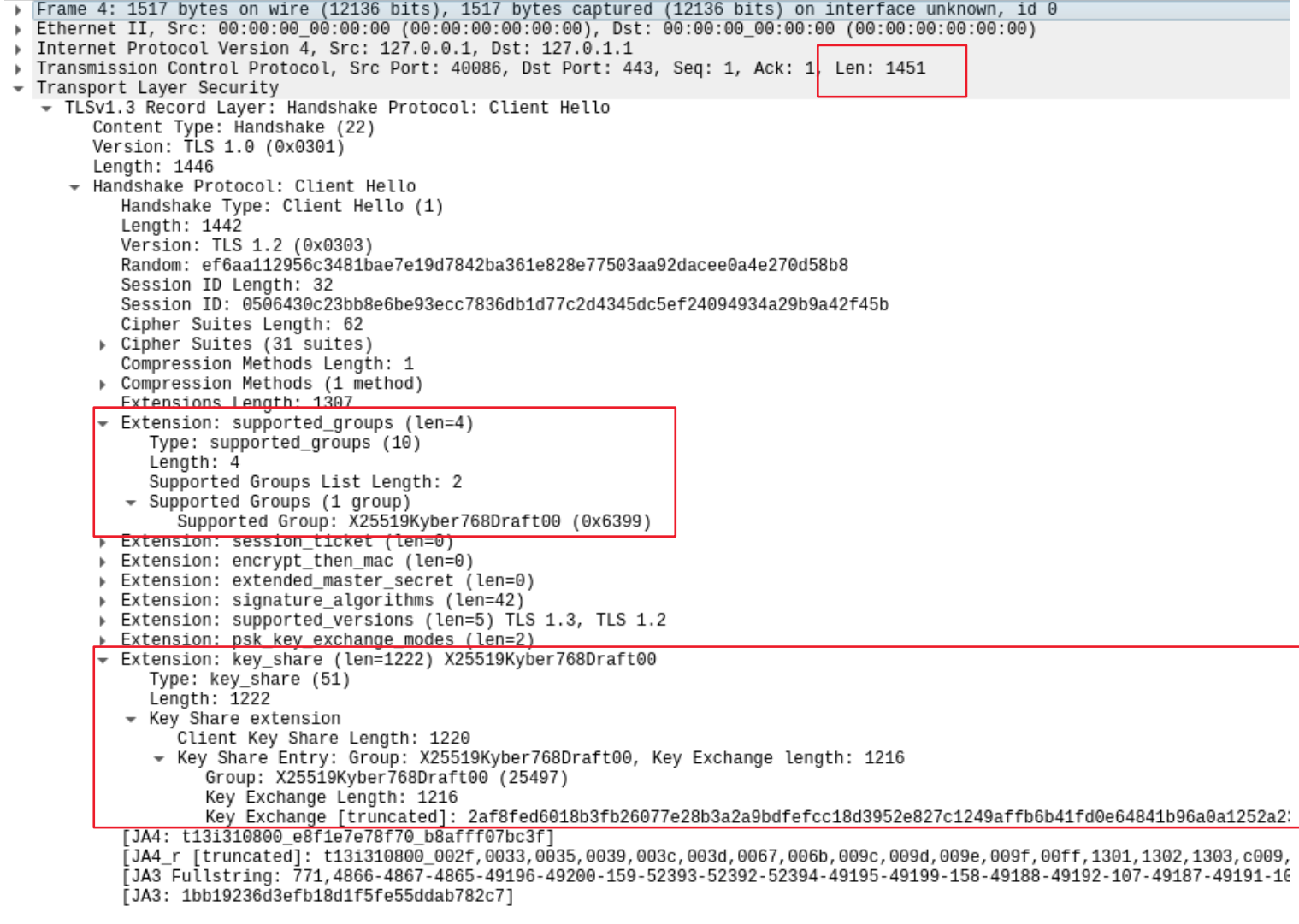
TLS:

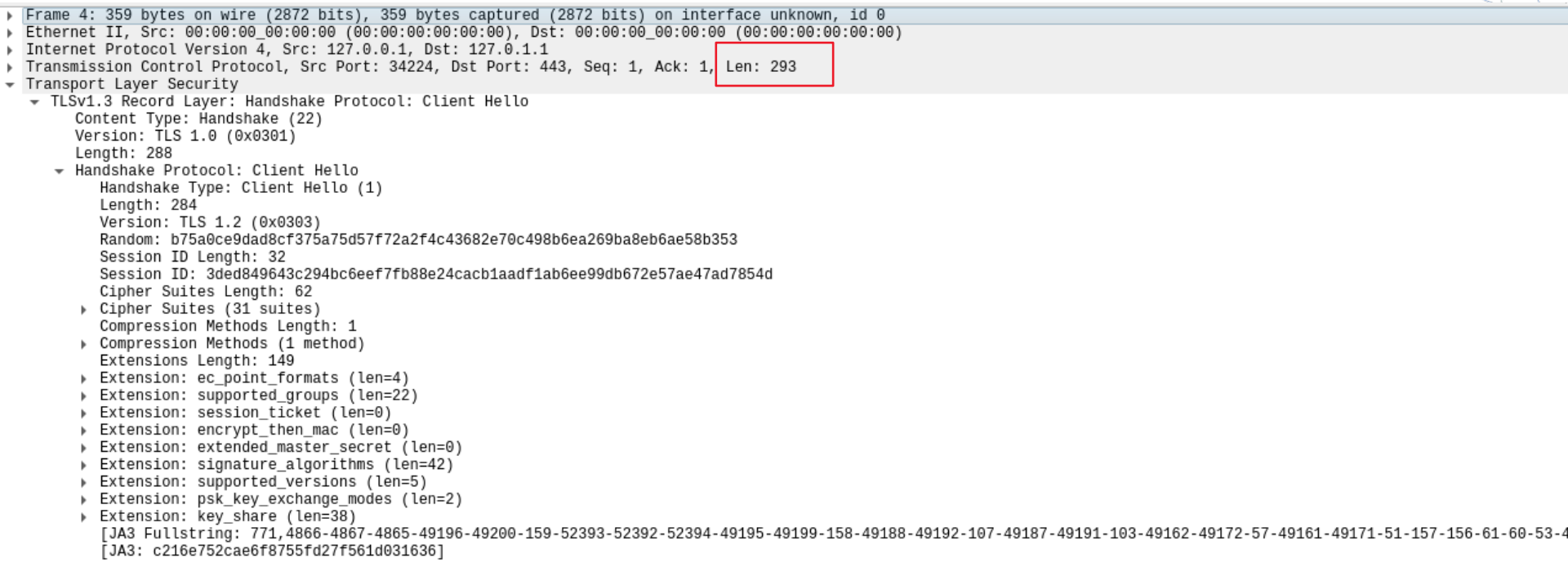
X25519Kyber768Draft00 → draft-kwiatkowski-tls-ecdhe-mlkem-01
Kyber+X25519 / ML-KEM+X25519|secp256r1 (hybride)

Déjà utilisé dans la nature:

- Chrome, Edge, Opera (activé par défaut)
- Firefox (désactivé par défaut)
- CloudFlare
- ……..




Frame 4: 1517 bytes on wire (12136 bits), 1517 bytes captured (12136 bits) on interface unknown, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.1.1
Transmission Control Protocol, Src Port: 40086, Dst Port: 443, Seq: 1, Ack: 1, Len: 1451
Transport Layer Security
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 1446
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 1442
Version: TLS 1.2 (0x0303)
Random: ef6aa112956c3481bae7e19d7842ba361e828e77503aa92dacee0a4e270d58b8
Session ID Length: 32
Session ID: 0506430c23bb8e6be93ecc7836db1d77c2d4345dc5ef24094934a29b9a42f45b
Cipher Suites Length: 62
Cipher Suites (31 suites)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 1307
Extension: supported_groups (len=4)
Type: supported_groups (10)
Length: 4
Supported Groups List Length: 2
Supported Groups (1 group)
Supported Group: X25519Kyber768Draft00 (0x6399)
Extension: session_ticket (len=0)
Extension: encrypt_then_mac (len=0)
Extension: extended_master_secret (len=0)
Extension: signature_algorithms (len=42)
Extension: supported_versions (len=5) TLS 1.3, TLS 1.2
Extension: psk_key_exchange_modes (len=2)
Extension: key_share (len=1222) X25519Kyber768Draft00
Type: key_share (51)
Length: 1222
Key Share extension
Client Key Share Length: 1220
Key Share Entry: Group: X25519Kyber768Draft00, Key Exchange length: 1216
Group: X25519Kyber768Draft00 (25497)
Key Exchange Length: 1216
Key Exchange [truncated]: 2af8fed6018b3fb26077e28b3a2a9bdfefcc18d3952e827c1249affb6b41fd0e64841b96a0a1252a2
[JA4: t13i310800_e8f1e7e78f70_b8afff07bc3f]
[JA4_r [truncated]: t13i310800_002f,0033,0035,0039,003c,003d,0067,006b,009c,009d,009e,009f,00ff,1301,1302,1303,c009,
[JA3 Fullstring: 771,4866-4867-4865-49196-49200-159-52393-52392-52394-49195-49199-158-49188-49192-107-49187-49191-16
[JA3: 1bb19236d3efb18d1f5fe55ddab782c7]


Frame 4: 359 bytes on wire (2872 bits), 359 bytes captured (2872 bits) on interface unknown, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.1.1
Transmission Control Protocol, Src Port: 34224, Dst Port: 443, Seq: 1, Ack: 1, Len: 293
Transport Layer Security
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 288
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 284
Version: TLS 1.2 (0x0303)
Random: b75a0ce9dad8cf375a75d57f72a2f4c43682e70c498b6ea269ba8eb6ae58b353
Session ID Length: 32
Session ID: 3ded849643c294bc6eef7fb88e24cacb1aadf1ab6ee99db672e57ae47ad7854d
Cipher Suites Length: 62
Cipher Suites (31 suites)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 149
Extension: ec_point_formats (len=4)
Extension: supported_groups (len=22)
Extension: session_ticket (len=0)
Extension: encrypt_then_mac (len=0)
Extension: extended_master_secret (len=0)
Extension: signature_algorithms (len=42)
Extension: supported_versions (len=5)
Extension: psk_key_exchange_modes (len=2)
Extension: key_share (len=38)
[JA3 Fullstring: 771,4866-4867-4865-49196-49200-159-52393-52392-52394-49195-49199-158-49188-49192-107-49187-49191-103-49162-49172-57-49161-49171-51-157-156-61-60-53-4
[JA3: c216e752cae6f8755fd27f561d031636]

Et en pratique ?

TLS:

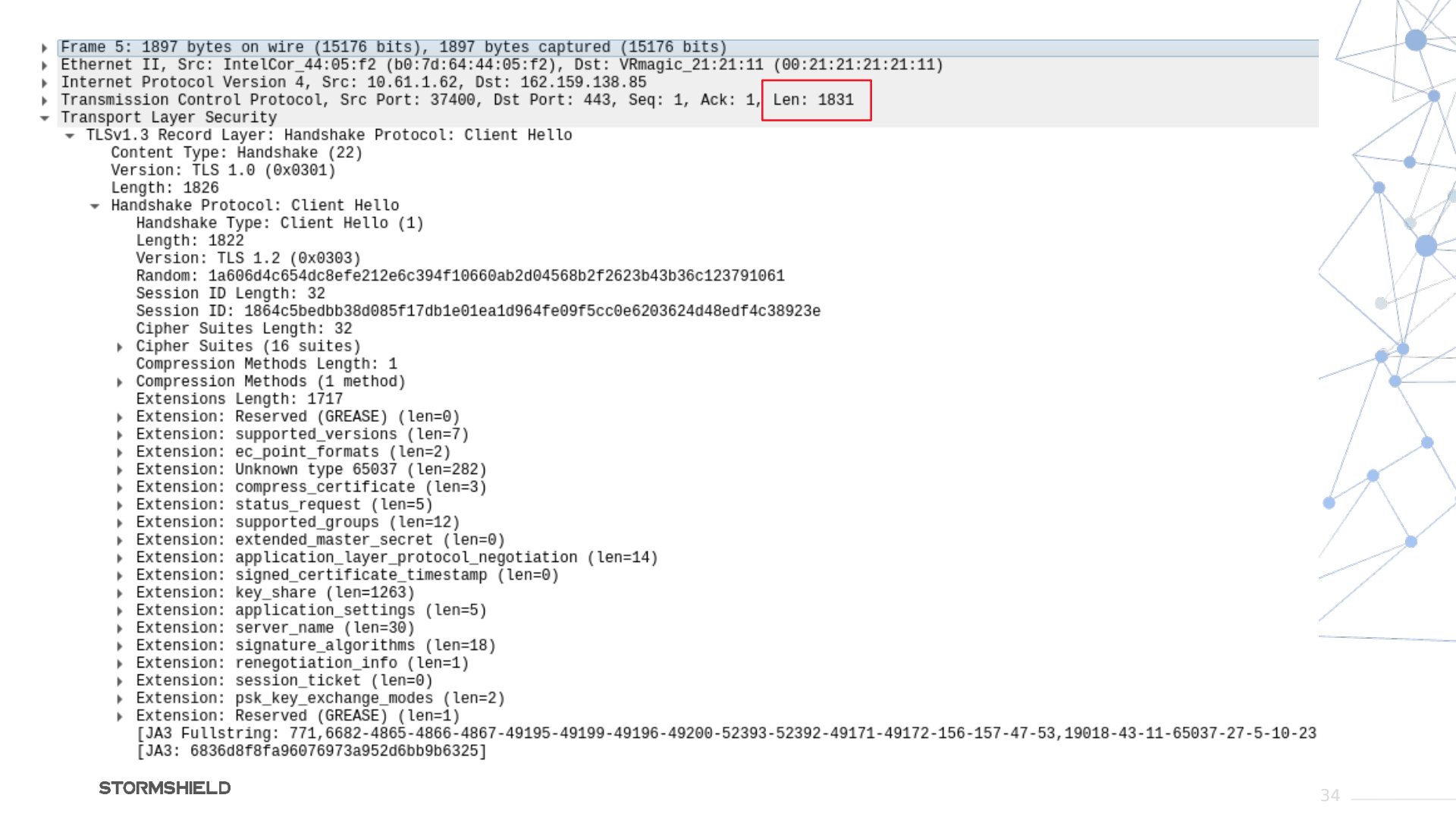
Négocié avec le correspondant

→ reste compatible avec les anciennes implémentations

Coté client, comment vérifier / maitriser ???
Impossible de forcer l'utilisation de ce KEM

Coté serveur, on peut forcer l'utilisation de ce KEM

→ Casse la compatibilité avec les « anciens » navigateurs !


Frame 5: 1897 bytes on wire (15176 bits), 1897 bytes captured (15176 bits)
Ethernet II, Src: IntelCor_44:05:f2 (b0:7d:64:44:05:f2), Dst: VRmagic_21:21:11 (00:21:21:21:21:11)
Internet Protocol Version 4, Src: 10.61.1.62, Dst: 162.159.138.85
Transmission Control Protocol, Src Port: 37400, Dst Port: 443, Seq: 1, Ack: 1, Len: 1831
Transport Layer Security
TLSv1.3 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 1826
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 1822
Version: TLS 1.2 (0x0303)
Random: 1a606d4c654dc8efe212e6c394f10660ab2d04568b2f2623b43b36c123791061
Session ID Length: 32
Session ID: 1864c5bedbb38d085f17db1e01ea1d964fe09f5cc0e6203624d48edf4c38923e
Cipher Suites Length: 32
Cipher Suites (16 suites)
Compression Methods Length: 1
Compression Methods (1 method)
Extensions Length: 1717
Extension: Reserved (GREASE) (len=0)
Extension: supported_versions (len=7)
Extension: ec_point_formats (len=2)
Extension: Unknown type 65037 (len=282)
Extension: compress_certificate (len=3)
Extension: status_request (len=5)
Extension: supported_groups (len=12)
Extension: extended_master_secret (len=0)
Extension: application_layer_protocol_negotiation (len=14)
Extension: signed_certificate_timestamp (len=0)
Extension: key_share (len=1263)
Extension: application_settings (len=5)
Extension: server_name (len=30)
Extension: signature_algorithms (len=18)
Extension: renegotiation_info (len=1)
Extension: session_ticket (len=0)
Extension: psk_key_exchange_modes (len=2)
Extension: Reserved (GREASE) (len=1)
[JA3 Fullstring: 771,6682-4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53,19018-43-11-65037-27-5-10-23
[JA3: 6836d8f8fa96076973a952d6bb9b6325]

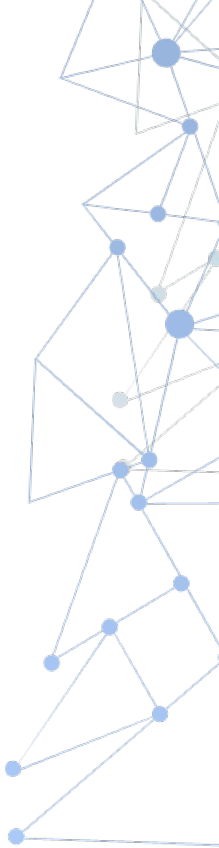
Et en pratique ?

TLS:

En pratique, certaines implémentations ont des problèmes

- ClientHello sur plusieurs paquets
- Extensions du ClientHello inconnues des serveurs

→ https://tldr.fail/



Et en pratique ?

IPSec (IKEv2):

RFC 7383: IKE Frag (pour les grosses clés)
RFC 9242: Intermediate Exchange (pour utiliser IKE Frag au tout début de la négociation)

RFC 9370: Multiple Key Exchanges (pour le mode hybride) → Jusqu'à 8 échanges

draft-kampanakis-ml-kem-ikev2-06: pour ML-KEM
Numéros IANA en attente d'attribution

Mode transition (négociation IKEv2)
Mode hybride imposé

IPSecDR ?
Sujet en cours (ANSSI)

Négociation classique:

```
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: parent_sa ikev2_init[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: parent_sa ikev2_init[R]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: parent_sa ikev2_init[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: parent_sa ikev2_init[R]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  ikev2_auth[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  ikev2_auth[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  ikev2_auth[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  ikev2_auth[R]
IP 192.168.156.31 > 192.168.156.21: ESP(spi=0xc2225292,seq=0x1), length 136
IP 192.168.156.21 > 192.168.156.31: ESP(spi=0xcc9666ce,seq=0x1), length 136
```

ECDSA+ML-KEM1024:

```
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: parent_sa ikev2_init[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: parent_sa ikev2_init[R]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: parent_sa ikev2_init[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: parent_sa ikev2_init[R]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  ikev2_auth[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  ikev2_auth[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  ikev2_auth[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  ikev2_auth[R]
IP 192.168.156.31 > 192.168.156.21: ESP(spi=0xcc659a2f,seq=0x1), length 136
IP 192.168.156.21 > 192.168.156.31: ESP(spi=0xc18939c3,seq=0x1), length 136
```

ECDSA+Frodo640:

```
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: parent_sa ikev2_init[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: parent_sa ikev2_init[R]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: parent_sa ikev2_init[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: parent_sa ikev2_init[R]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  #43[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  #43[R]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  ikev2_auth[I]
IP 192.168.156.31.500 > 192.168.156.21.500: isakmp: child_sa  ikev2_auth[I]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  ikev2_auth[R]
IP 192.168.156.21.500 > 192.168.156.31.500: isakmp: child_sa  ikev2_auth[R]
```

Les PKIs

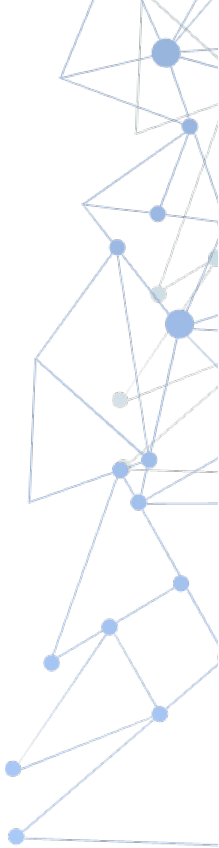
RFC 5280:

X.509v3

Une seule clé privée
Une seule clé publique
Une seule signature

PKI Post-quantique pure: Ok

PKI Hybride ?????
x.509 est extensible !



Les PKIs

Approche
Multi certificats

Gros changements dans les protocoles

Compatibilité avec les anciens logiciels ?

Variante: double encapsulation (mais gros impact performances)

Les PKIs

Approche **Hybride**

« Catalyst » / « Alt »

Draft-truskovsky-lamps-pq-hybrid-x509-02 (expiré en février 2024)
Variante ITU normalisée (en mode « transition »)

AltSignatureAlgorithm, SubjectAltPublicKey, AltSignatureValue

Certificats gérés par les anciens logiciels (si les extensions ne sont pas « critical »)

Aucune garantie de vérification PQSafe par le correspondant

Les PKIs

Approche **Composite**

Draft-ounsworth-pq-composite-[sigs|keys]

Nouveaux « algorithmes »: Dilithium+ECC, Dilithium+RSA, Falcon+ECC

Incompatibles avec les anciens logiciels

Garantie de vérification PQSafe par le correspondant

Pas de gros impact sur les protocoles

Les PKIs

Approche **Caméléon**

Draft-bonnell-lamps-chameleon-certs-03
Idée principale: un certificat « classique » + un certificat « delta »
Certificat « classique » compatible avec les anciens logiciels
Pas de garantie de vérification PQSafe par le correspondant
Impact sur les protocoles ?
Gros impact sur les implémentations

Les PKIs

Quelle norme choisir ?

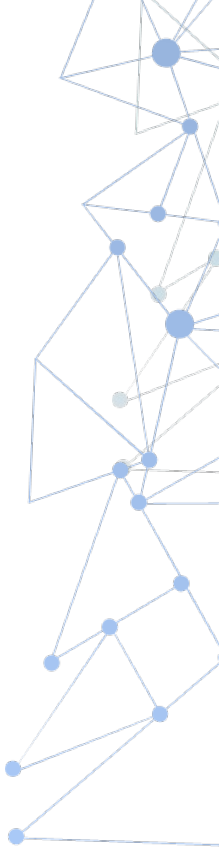
Hybride / Composite / Caméléon:

Chaque approche a ses avantages et inconvénients !

Choix en fonction:

- Des cas d'utilisation (chiffrement ? Authentification/Signature ?)
- Des contraintes de déploiement (Composite est plus exigeant)
- Des exigences de sécurité (Composite apporte des garanties)
- Des normes supportées par vos outils ?

Pertinence de mixer des normes sur une seule PKI ?

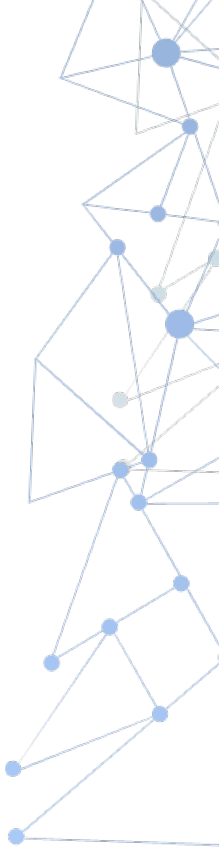


Les PKIs

Impact sur la taille !

```
2,0K root-rsa.crt
3,2K root-rsa.pkey
1,5K Bill-rsa.crt
1,7K Bill-rsa.pkey
```

```
48K root-sphincs.crt
282 root-sphincs.pkey
9,6K exploitation-ecc-dil.crt
8,2K exploitation-ecc-dil.pkey
8,0K Bill-ecc-dil.crt
8,2K Bill-ecc-dil.pkey
```



Le hardware ?

TPMs ?

HSMs ?

Accélération matérielle ?

Secureboot ?

Cartes à puces ?

……

Théorème de **Mosca**

Vous êtes ici

Y

Temps de standardisation et d'adoption

X

Durée de confidentialité des données

Z

Temps avant une attaque par ordinateur quantique

X + Y > Z = Problème !

Temps de standardisation et d'adoption:

- *Conception et implémentation des standards*
- *Déploiement des nouvelles versions de firmware / logiciel*
- *Arrêt complet d'utilisation des anciennes versions de firmware / logiciel*

Les prochaines étapes ?

Se préparer !

Inventaires des usages cryptographiques

- Au moins au niveau « macro »
- Identifier les cas critiques
- Crypto-agilité
- Roadmaps de vos outils

Premières migrations

2025 ?

Chiffrement (KEM)

Données sensibles / critiques

En mode hybride

Phase de migration / transition !

La transition post-quantique est un marathon !

La roadmap

Stormshield

Priorité au chiffrement !

SNS

V5.0: KEM hybride pour IPSec

- PoC interne + PoC TheGreenBow

V5.x: autres usages KEM (TLS)

Point d'attention: composants matériels !

- TPM (signature de mises à jour)

Signature et authentification: V5.x, V6.x

SDS

Dépendant de la PKI pour chiffrer

Synchronisation les fournisseurs PKI

Cible 2025 (hybride)

SES

Moins vulnérable à l'attaque « store now, decrypt later »
(flux à durée de vie relativement courte)

Migration TLS en KEM hybride prévue

Thank you

STORMSHIELD