RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

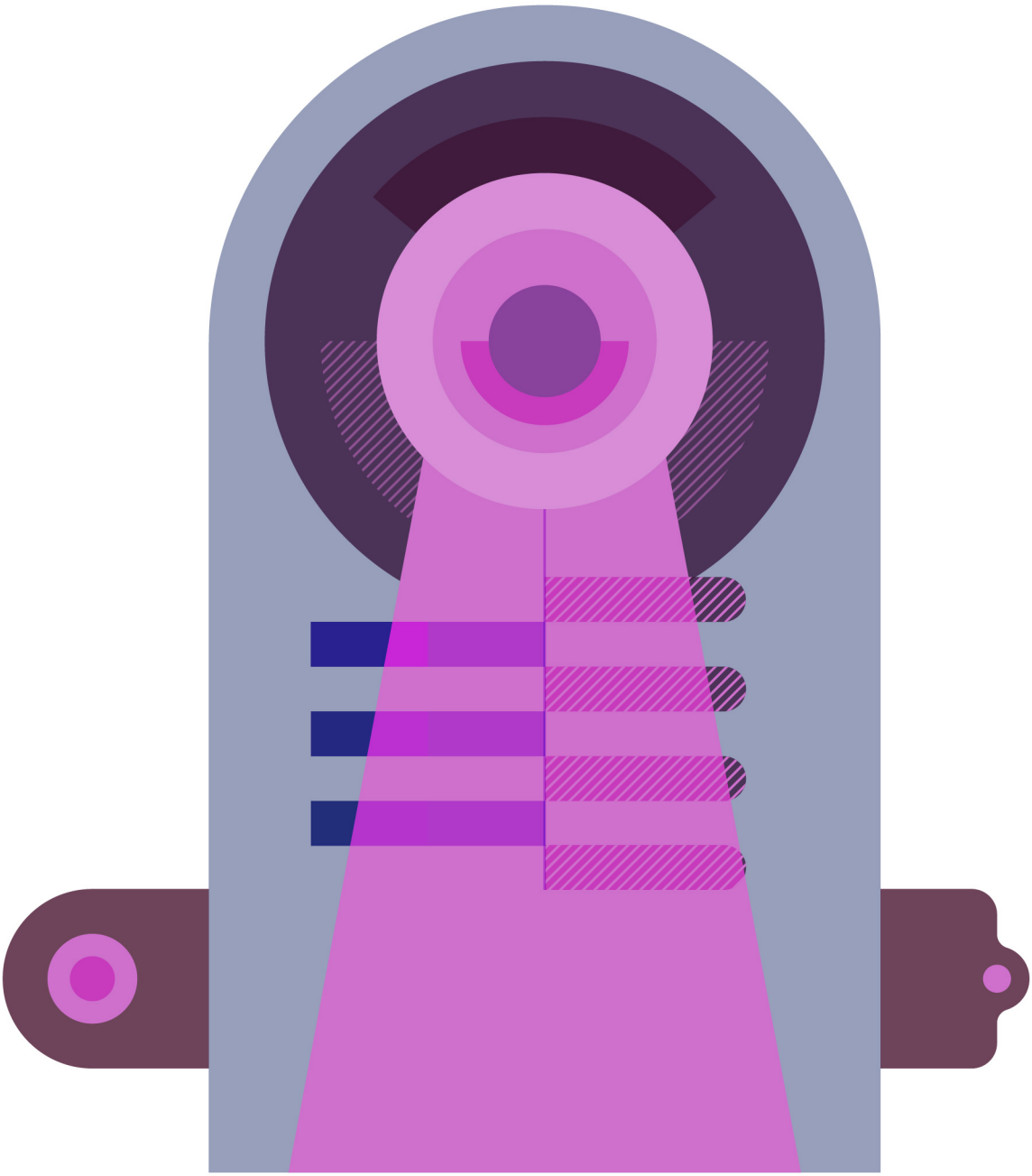
TRANSITION POST-QUANTIQUE DE TLS 1.3

FICHE TECHNIQUE ANSSI

ANSSI-FT-115
02/02/2026

PUBLIC VISÉ :

Développeur | Administrateur | RSSI | DSI | Utilisateur



Informations

Attention

Ce document rédigé par l'ANSSI s'intitule **« Transition Post-Quantique de TLS 1.3 »**. Il est téléchargeable sur le site `cyber.gouv.fr`.

Évolutions du document :

VERSION	DATE	NATURE DES MODIFICATIONS
1.0	02/02/2026	Version initiale

Table des matières

1
Introduction

Ce document vise à présenter un état des lieux de la transition post-quantique de la version 1.3 du protocole TLS (Transport Layer Security), définie dans la RFC 8446 [14]. Le document présente d'abord le fonctionnement général du protocole. Ensuite, il identifie les parties du protocole concernées par la menace quantique et des solutions de transition associées. Il présente également les travaux existants sur le sujet. Ce document vient en complément de l'avis de l'ANSSI sur la transition post-quantique [5, 6].

Les mécanismes de cryptographie asymétrique utilisés aujourd'hui sont vulnérables à la menace quantique. Ainsi, le but de la transition post-quantique est de les remplacer par des mécanismes de cryptographie asymétrique dite post-quantique, supposée résistante à des attaques qui seraient réalisées sur un ordinateur classique et sur un ordinateur quantique. En particulier, les signatures de cryptographie classique sont remplacées par des signatures de cryptographie post-quantique, et les échanges de clés, souvent basés sur le schéma Diffie-Hellman (DH), sont remplacés par des échanges de clés basés sur des mécanismes d'encapsulation de clés (ou *Key Encapsulation Mechanism* (KEM)) post-quantiques. Pendant la transition post-quantique, l'hybridation est recommandée par l'ANSSI [5, 6]. Cela signifie l'utilisation d'un mécanisme de cryptographie asymétrique classique éprouvé, mais vulnérable à des attaques quantiques, combiné avec un mécanisme de cryptographie asymétrique supposé résistant aux attaques quantiques, mais pour lequel l'assurance de robustesse vis-à-vis des attaques réalisées à l'aide d'ordinateurs classique et quantique est moindre. Une conséquence de l'utilisation des mécanismes de cryptographie post-quantique dans les protocoles est une augmentation importante de la taille des messages échangés pendant les phases d'échange de clés et d'authentification. En effet, les mécanismes de cryptographie post-quantique existants possèdent des tailles de clés, de chiffrés et de signatures très grandes par rapport aux mécanismes de cryptographie classique. Quant aux mécanismes de cryptographie symétrique, la robustessse de ces derniers n'est pas menacée par l'ordinateur quantique. Toutefois, l'ANSSI recommande [5, 6] d'augmenter les tailles des clés du chiffrement symétrique et des sorties des fonctions de hachage.

2
Présentation de TLS 1.3

Le protocole TLS 1.3 (Transport Layer Security) permet d'établir un canal de communication sécurisé entre un client et un serveur. Il assure principalement la confidentialité et l'intégrité des données applicatives, ainsi que l'authentification du serveur, et optionnellement celle du client. Il est composé de trois protocoles principaux, comme présenté dans la Figure 1 : Handshake, Record et Alert. La partie impactée par la transition post-quantique est indiquée en rouge, il s'agit du protocole Handshake. Le protocole TLS 1.3 est décrit plus en détails dans le guide de l'ANSSI [7].

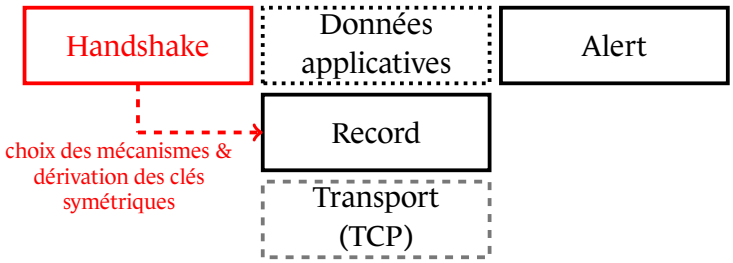


FIGURE 1 – Protocole TLS 1.3

- Le protocole Handshake permet de négocier les suites cryptographiques devant être utilisées entre le client et le serveur. Il est exécuté à l'initiation des échanges. Il permet également d'établir un secret partagé, tout en authentifiant le serveur, et optionnellement le client. Les clés symétriques de chiffrement et d'intégrité sont dérivées à partir de ce secret partagé. Dans la suite du document, seule l'authentification par signature, recommandée par l'ANSSI [7], sera considérée.

Attention

Le protocole *handshake* de TLS 1.3 est impacté par la menace quantique. En effet, des mécanismes asymétriques sont utilisés pour l'échange de clés (DH) et l'authentification par signature du serveur et du client.

- Le protocole Record permet de transmettre les messages TLS 1.3 et les données applicatives. Il se charge de fragmenter les données, les encapsuler dans des trames structurées et les envoyer. Il protège les données applicatives et les messages TLS 1.3 en confidentialité et intégrité en utilisant des mécanismes symétriques, négociés dans le protocole *handshake*. Le fonctionnement du protocole Record n'est pas impacté par la menace quantique, puisqu'il n'utilise pas de mécanismes asymétriques. Ainsi, il ne sera pas traité dans la suite de ce document. Toutefois, la sécurité des clés utilisées dans ce protocole repose sur le protocole Handshake, pendant lequel la dérivation de clés est calculée.

- Le protocole Alert permet d'échanger des messages d'alerte et d'erreur. Les messages de ce protocole sont envoyés au protocole Record qui les protège avant transmission. Le protocole Alert n'est pas impacté par la menace quantique. Il n'utilise pas de mécanismes de cryptographie et ne sera donc pas traité dans la suite de ce document.

2.1 Le protocole Handshake de TLS 1.3

Les échanges du protocole Handshake de TLS 1.3 sont rappelés dans la Figure 2. Les messages munis d'un cadenas sont protégés en confidentialité et intégrité dans le protocole Record (tous les messages de TLS 1.3 passent par le protocole Record avant d'être envoyés, y compris les messages du Handshake). Cela concerne donc tous les messages sauf ClientHello et ServerHello. Les messages entre crochets sont des messages optionnels en fonction du contexte.

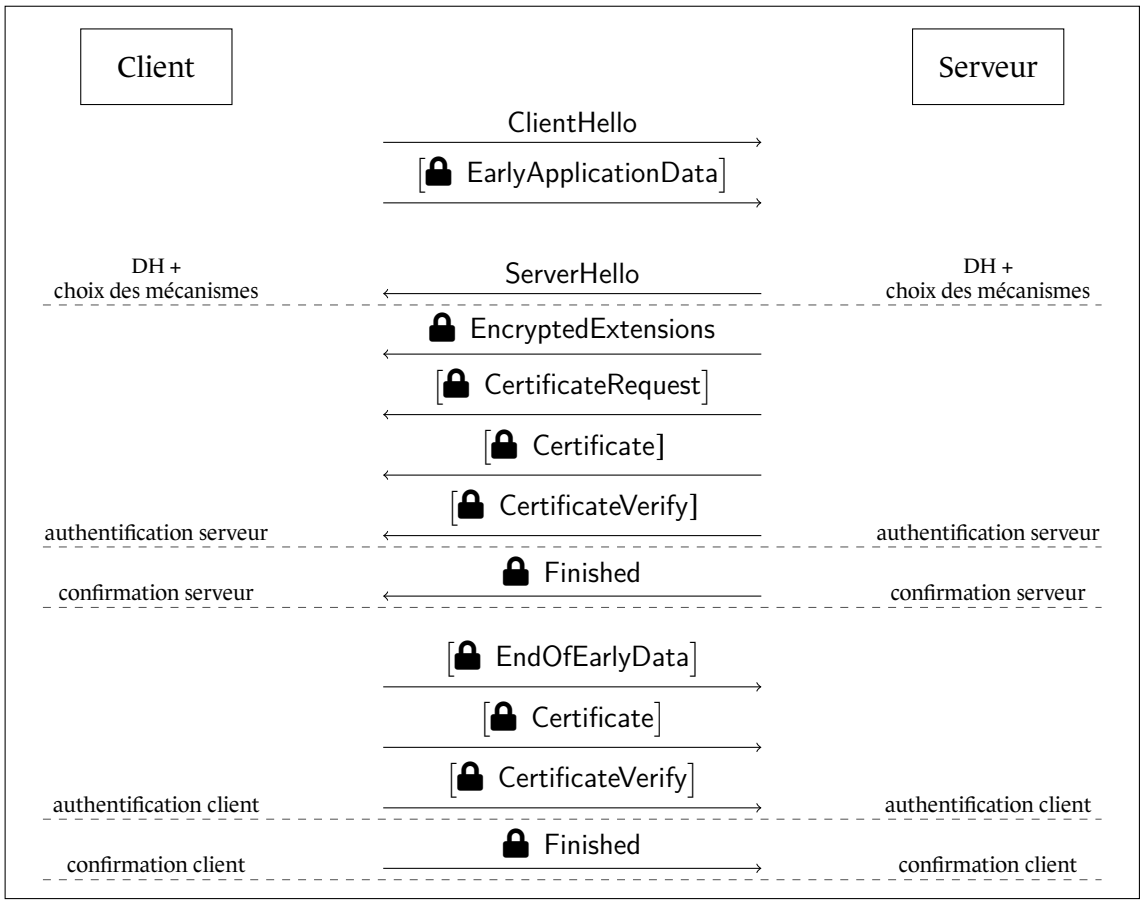


FIGURE 2 – Handshake de TLS 1.3

- Les messages ClientHello et ServerHello servent principalement à négocier les suites cryptographiques et effectuer l'échange de clés. Dans le message ClientHello, le client envoie notamment :
  - une ou plusieurs propositions de suites cryptographiques pour les mécanismes symétriques assurant la confidentialité et l'intégrité des données applicatives et des messages TLS 1.3, ainsi que la fonction de hachage pour la fonction de dérivation des clés symétriques,
  - l'ensemble des groupes supportés par le client pour l'échange de clés basé sur le schéma DH. Le client envoie également sa clé publique correspondante à chacun des groupes,

- l'ensemble des mécanismes asymétriques de signature supportés par le client pour l'authentification du serveur,
- l'identifiant d'une clé pré-partagée avec le serveur si le client le souhaite.

Information

TLS 1.3 permet l'utilisation de deux types de clés pré-partagées : *"resumption psk"* qui est une clé pré-partagée générée pendant une autre session TLS 1.3 entre les mêmes client et serveur, ou *"external psk"* qui est une clé pré-partagée générée dans un échange indépendant de TLS 1.3, défini par le client et le serveur.

Ensuite, le serveur répond dans son message ServerHello avec ses choix pour les suites cryptographiques. Il envoie aussi le groupe DH négocié avec sa clé publique correspondante au groupe. Dans TLS 1.3, le client et le serveur peuvent également utiliser une clé pré-partagée pour dériver les clés symétriques de chiffrement et d'intégrité. Dans ce cas, les messages ClientHello et ServerHello servent aussi à négocier l'identifiant de la clé pré-partagée entre le client et le serveur. Dans TLS 1.3, il est possible de négocier l'identifiant d'une clé pré-partagée et ne pas effectuer un échange de clés basé sur le schéma DH. Dans ce cas, la sécurité des clés symétriques finales repose uniquement sur cette clé pré-partagée.

- Les messages EarlyApplicationData et EndOfEarlyData sont envoyés dans le cadre de la fonctionnalité 0-RTT de TLS 1.3 [1].
- Le message EncryptedExtensions contient des extensions liées au fonctionnement du protocole.
- Le serveur envoie les messages Certificate et CertificateVerify pour s'authentifier auprès du client. Le serveur peut demander l'authentification du client avec son message CertificateRequest. Dans ce dernier, il précise l'ensemble des mécanismes de signature qu'il supporte. Dans ce cas, le client répond aussi avec Certificate et CertificateVerify. Pour s'authentifier :
  - le message Certificate contient le certificat liant l'identité du serveur ou du client et sa clé publique,
  - le message CertificateVerify contient une signature calculée sur les messages échangés, avec la clé privée correspondante au certificat.

  Les messages Certificate et CertificateVerify sont optionnels des deux côtés si l'utilisation d'une clé pré-partagée est négociée dans les messages ClientHello et ServerHello. En effet, la clé pré-partagée est utilisée dans la dérivation des clés symétriques, mais sert aussi à authentifier implicitement le client et le serveur.
- Les messages Finished du client et du serveur servent à confirmer les secrets partagés et finaliser l'authentification des échanges passés et des clés symétriques de chiffrement et d'intégrité.

1. La fonctionnalité 0-RTT de TLS 1.3 permet au client d'envoyer des données applicatives dès l'envoi du message ClientHello, en utilisant la clé pré-partagée pour la dérivation des clés symétriques de chiffrement et d'intégrité. Ces données applicatives sont protégées en confidentialité et intégrité, mais ne bénéficient pas de confidentialité persistante (ou *Perfert Forward Secrecy* (PFS)).

3
Transition post-quantique de TLS 1.3

Comme mentionné précédemment, le protocole Handshake de TLS 1.3 met en œuvre des mécanismes asymétriques pour l'échange de clés et l'authentification par signature. Ces derniers sont vulnérables à la menace quantique. La transition post-quantique du protocole Handshake de TLS 1.3 peut être réalisée en utilisant des clés pré-partagées (*c.f.*, Section 3.1), ou en utilisant une méthode d'hybridation (*c.f.*, Section 3.2) qui est la solution privilégiée par l'ANSSI [5, 6]. Notons que la transition post-quantique pour la confidentialité des échanges est plus urgente que celle pour l'authentification, notamment à cause des attaques *"store-now decrypt-later"*.

3.1 Sécurité post-quantique par clé pré-partagée

Comme précisé dans l'avis sur la transition post-quantique de l'ANSSI [5, 6], une clé pré-partagée peut être utilisée dans le cadre de la transition post-quantique, notamment si la confidentialité et l'intégrité (classiques et post-quantiques) de la clé pré-partagée sont assurées. De plus, une attention particulière doit être portée à la gestion et la distribution de clés pré-partagées. Enfin, la mise en œuvre de clé pré-partagée peut être une mesure temporaire dans la transition post-quantique. La mise en œuvre de l'hybridation et l'utilisation de mécanismes post-quantiques pour assurer une sécurité post-quantique devront être effectuées à terme.

Confidentialité post-quantique par clé pré-partagée. TLS 1.3 offre la possibilité d'utiliser une clé pré-partagée pour dériver les clés symétriques de chiffrement et d'intégrité (*c.f.*, Section 2.1). Ceci peut assurer la confidentialité post-quantique des échanges chiffrés avec ces clés (dans le protocole Record). Ainsi, l'utilisation de cette solution est possible dans TLS 1.3 pour assurer une confidentialité post-quantique.

Attention

La compromission de la clé pré-partagée est rétroactive, elle implique la compromission de toutes les sessions où elle a été utilisée. Ceci a des conséquences sur la propriété de la confidentialié persistante (ou *Perfert Forward Secrecy* (PFS)) des messages TLS 1.3 et des données applicatives chiffrés. La propriété PFS signifie que la compromission des clés d'une session ne doit pas compromettre des clés de sessions passées. Par exemple, si la dérivation des clés de chaque session utilise une même clé pré-partagée et un nouveau secret DH, la compromission de la clé pré-partagée signifie qu'un attaquant quantique pourra réaliser l'attaque *"store-now decrypt-later"* sur toutes les sessions passées utilisant cette clé, puisque les secrets DH peuvent être compromis avec un ordinateur quantique. Dans ce cas, la propriété PFS est assurée en classique avec le secret DH, mais pas en post-quantique.

Authentification post-quantique par clé pré-partagée. L'utilisation de la clé pré-partagée dans TLS 1.3 assure également une authentification mutuelle du client et du serveur. Ainsi, la même solution est envisageable pour les propriétés de confidentialité et d'authentification post-quantiques.

Implémentations. La librairie OpenSSL [2] implémente le protocole TLS 1.3 et l'utilisation d'une clé pré-partagée dans le protocole telle que décrite dans la Section 2.1.

3.2 Sécurité post-quantique par hybridation

L'hybridation du protocole Handshake de TLS 1.3 implique la modification de plusieurs messages. En particulier, le client et le serveur effectuent un échange de clés basé sur le schéma DH dans les messages ClientHello et ServerHello. Ensuite, le client et le serveur s'authentifient avec les messages Certificate et CertificateRequest.

3.2.1 Hybridation de l'échange de clés

L'échange de clés a lieu dans le protocole Handshake de TLS 1.3. L'hybridation consiste à exécuter deux échanges de clés : un échange de clés avec un mécanisme classique (par exemple le schéma DH) et un échange de clés avec un mécanisme post-quantique (par exemple ML-KEM [12]). Ensuite, les deux secrets partagés sont combinés pour dériver le secret final ayant une sécurité classique et post-quantique.

Attention

Le mécanisme de combinaison des secrets partagés issus des deux échanges de clés (classique et post-quantique) pour dériver le secret partagé final doit assurer une sécurité classique et post-quantique. Des manières de combiner les deux secrets sont citées dans l'avis sur la transition post-quantique de l'ANSSI [5, 6].

L'hybridation de l'échange de clés du protocole Handshake de TLS 1.3 implique les modifications suivantes :

- dans le message ClientHello, le client envoie une liste de groupes Diffie-Hellman avec sa clé publique pour chacun des groupes. Dans le cas d'un échange de clés hybride, cette liste présente des noms de mécanismes pour des échanges de clés hybrides (groupe DH et KEM post-quantique). Pour chacun des échanges de clés hybrides proposés, le client envoie également sa clé publique DH et sa clé publique du KEM post-quantique, qui peuvent être concaténées.

  #### Attention

  En proposant plusieurs choix d'échanges de clés hybrides, le message ClientHello devient de taille conséquente, à cause des tailles des clés des KEM post-quantiques.

- Dans le message ServerHello, le serveur répond avec son groupe DH négocié et sa clé publique correspondante. Dans le cas d'un échange de clés hybride, le serveur répond avec les noms des mécanismes pour l'échange de clés hybride négocié. Pour ce dernier, il envoie également sa clé

publique DH et le chiffré résultant de l'encapsulation sur la clé publique du KEM post-quantique du client, qui peuvent être concaténés.

- Dans le message EncryptedExtensions, le serveur peut aussi envoyer une liste de groupes Diffie-Hellman supportés pour des futures connexions, ou dans un message HelloRetryRequest dans le cas d'une demande de renvoi. Dans le cas d'un échange de clés hybride, la liste contient des noms de mécanismes pour des échanges de clés hybrides.

Documents. Un draft internet [15] définit la manière de faire un échange de clés hybride dans TLS 1.3, en précisant les noms et les formats des différents champs des messages. La solution proposée dans le draft est similaire à la description dans cette section. Un autre draft internet [10] instancie l'échange défini dans [15] avec le schéma DH basé sur les courbes elliptiques et ML-KEM [12]. Enfin, un draft internet [8] définit l'utilisation de ML-KEM [12] (de façon non-hybride) pour l'échange de clés dans TLS 1.3.

Implémentations. La version 3.5.0 d'OpenSSL [1] implémente un échange de clés hybride utilisant le schéma DH avec ML-KEM [12]. De plus, plusieurs entreprises intègrent des échanges de clés hybrides dans leurs implémentations de TLS 1.3. Google intègre un échange de clés hybride avec DH et ML-KEM dans le navigateur Chrome [4], de façon similaire à [15], fonctionnel uniquement sur les navigateurs de poste bureautique. Meta intègre également un échange de clés hybride dans la librairie TLS Fizz [3], en utilisant la librairie post-quantique liboqs [16].

3.2.2 Hybridation de l'authentification par signature

L'hybridation de l'authentification par signature nécessite l'hybridation des signatures et des certificats utilisés dans TLS 1.3. Il existe différents formats pour hybrider les certificats (authentifiant les clés de signature classique et post-quantique), par exemple avec deux certificats distincts (un pour la clé de signature classique, et un autre pour la clé de signature post-quantique), ou un seul certificat authentifiant la concaténation des clés de signature, etc. L'hybridation concerne aussi les chaînes de certification. Pour l'instant, il n'existe pas de standards pour l'authentification hybride.

L'hybridation de l'authentification par signature du protocole Handshake de TLS 1.3 implique les modifications suivantes :

- dans le message ClientHello, le client envoie les mécanismes de signatures qu'il supporte. Dans le cas de l'hybridation, ces mécanismes correspondent à des mécanismes de signature hybrides (composés d'une signature classique et d'une signature post-quantique).
- Dans le message CertificateRequest, le serveur envoie les mécanismes de signature qu'il supporte. Dans le cas de l'hybridation, ces mécanismes correspondent à des mécanismes de signature hybrides.
- Dans les messages Certificate et CertificateVerify, le serveur et optionnellement le client s'authentifient en envoyant un certificat et une signature. Dans le cas de l'hybridation, les certificats dans les messages Certificate et les signatures dans les messages CertificateVerify deviennent hybrides.

Documents. Un draft internet [9] en cours définit l'utilisation de l'algorithme de signature post-quantique ML-DSA [11] dans TLS 1.3.

Implémentations. La version 3.5.0 d'OpenSSL [1] implémente les algorithmes de signature post-quantiques ML-DSA [11] et SLH-DSA [13], mais leur utilisation dans le protocole TLS 1.3 n'est pas encore possible.

Information

L'utilisation des mécanismes de cryptographie post-quantique augmente la taille des messages échangés lors des phases de l'échange de clés et de l'authentification par signature. Outre les impacts sur les performances, l'augmentation de la taille des messages oblige les protocoles à gérer la fragmentation de ces messages. Ceci est géré nativement dans le protocole TLS 1.3.

4
Conclusion

La transition post-quantique du protocole TLS 1.3 concerne le protocole Handshake, où le client et le serveur négocient les suites cryptographiques, effectuent un échange de clés basé sur le schéma DH et s'authentifient mutuellement. L'ANSSI recommande d'utiliser l'hybridation pour les échanges de clés et l'authentification par signature. Le protocole TLS 1.3 propose l'utilisation d'une clé pré-partagée pour assurer la confidentialité et l'authentification entre le client et le serveur. La mise en œuvre de cette solution pour assurer une confidentialité et une authentification résistantes à la menace quantique peut être envisageable mais temporaire, l'hybridation doit être implémentée à terme. L'utilisation d'une clé pré-partagée est implémentée dans OpenSSL [2].

L'hybridation de l'échange de clés dans le protocole Handshake nécessite un échange de clés basé sur un KEM post-quantique, en plus de l'échange de clés basé sur le schéma DH. Ces deux échanges de clés ont lieu dans les premiers messages du protocole Handshake. Il existe un draft internet [15] en cours qui décrit cet échange de clés hybride, et un draft [10] qui l'instancie avec le schéma DH basé sur les courbes elliptiques et ML-KEM [12]. Un autre draft internet [8] définit l'utilisation de ML-KEM [12] pour l'échange de clés dans TLS 1.3. De plus, plusieurs projets intègrent l'échange de clés hybride dans leurs implémentations. OpenSSL 3.5.0 utilise DH avec ML-KEM [12]. Google utilise également DH avec ML-KEM [12] pour l'hybridation de l'échange de clés dans le navigateur Chrome de poste bureautique [4]. Enfin, Meta intègre l'échange de clés hybride dans la librairie Fizz [3] en utilisant la librairie liboqs [16].

L'hybridation de l'authentification par signature dans le protocole Handshake nécessite l'hybridation des certificats utilisés pour authentifier le client et le serveur, ainsi que les signatures envoyées dans les messages d'authentification. L'hybridation de l'authentification nécessite plus de travaux. Il n'existe pas pour l'instant de standards officiels permettant l'authentification par signature et certificat hybrides dans TLS 1.3. Un draft internet [9] définit l'utilisation de l'algorithme de signature post-quantique ML-DSA [11] dans TLS 1.3. De plus, la version 3.5.0 d'OpenSSL [1] implémente les algorithmes de signature post-quantiques ML-DSA [11] et SLH-DSA [13], mais leur utilisation dans le protocole TLS 1.3 n'est pas encore possible.

La transition post-quantique du protocole TLS 1.3 étant un sujet très étudié, il est nécessaire de suivre l'évolution des travaux et des implémentations.

Bibliographie

[1] The features of 3.5 : Hybrid ml-kem. https://openssl-foundation.org/post/2025-04-29-ml-kem/index.html.

[2] Openssl. https://github.com/openssl/openssl.

[3] Post-quantum readiness for tls at meta. https://engineering.fb.com/2024/05/22/security/post-quantum-readiness-tls-pqr-meta/#:~:text=To%20address%20this%20issue,%20the%20cryptography%20community%20has%20been.

[4] Protecting chrome traffic with hybrid kyber kem. https://blog.chromium.org/2023/08/protecting-chrome-traffic-with-hybrid.html.

[5] ANSSI. Avis de l'anssi sur la migration vers la cryptographie post-quantique. https://cyber.gouv.fr/publications/avis-de-lanssi-sur-la-migration-vers-la-cryptographie-post-quantique.

[6] ANSSI. Avis de l'anssi sur la migration vers la cryptographie post-quantique. https://cyber.gouv.fr/publications/avis-de-lanssi-sur-la-migration-vers-la-cryptographie-post-quantique-0.

[7] ANSSI. Recommandations de sécurité relatives à tls. https://cyber.gouv.fr/publications/recommandations-de-securite-relatives-tls.

[8] Deirdre Connolly. ML-KEM Post-Quantum Key Agreement for TLS 1.3. Internet-Draft draft-ietf-tls-mlkem-00, Internet Engineering Task Force, April 2025. Work in Progress.

[9] Tim Hollebeek, Sophie Schmieg, and Bas Westerbaan. Use of ML-DSA in TLS 1.3. Internet-Draft draft-ietf-tls-mldsa-01, Internet Engineering Task Force, September 2025. Work in Progress.

[10] Kris Kwiatkowski, Panos Kampanakis, Bas Westerbaan, and Douglas Stebila. Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3. Internet-Draft draft-ietf-tls-ecdhe-mlkem-01, Internet Engineering Task Force, September 2025. Work in Progress.

[11] NIST. Module-lattice-based digital signature standard. https://csrc.nist.gov/pubs/fips/204/final.

[12] NIST. Module-lattice-based key-encapsulation mechanism standard. https://csrc.nist.gov/pubs/fips/203/final.

[13] NIST. Stateless hash-based digital signature standard. https://csrc.nist.gov/pubs/fips/205/final.

[14] Eric Rescorla and Tim Dierks. Rfc 8446 : The transport layer security (tls) protocol version 1.3, 2014.

[15] Douglas Stebila, Scott Fluhrer, and Shay Gueron. Hybrid key exchange in TLS 1.3. Internet-Draft draft-ietf-tls-hybrid-design-12, Internet Engineering Task Force, January 2025. Work in Progress.

[16] Douglas Stebila and Michele Mosca. Post-quantum key exchange for the internet and the open quantum safe project. In *International Conference on Selected Areas in Cryptography*, pages 14–37. Springer, 2016.

Version 1.0 - 02/02/2026 - ANSSI-FT-115

AGENCE NATIONALE DE LA SÉCURITÉ DES SYSTÈMES D'INFORMATION

ANSSI - 51 boulevard de La Tour-Maubourg 75000 PARIS 07 SP

cyber.gouv.fr / conseil.technique@ssi.gouv.fr

RÉPUBLIQUE FRANÇAISE

Liberté
Égalité
Fraternité