RFC 10073 : Composite Module-Lattice-Based Digital Signature Algorithm (ML-DSA) for use in X.509 Public Key Infrastructure

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 7 novembre 2026

Date de publication du RFC : Novembre 2026

https://www.bortzmeyer.org/10073.html

C'est bien joli d'utiliser des algorithmes post-quantiques pour que ses signatures ne soient pas attaquées par les futurs calculateurs quantiques. Mais si l'algorithme post-quantique cède devant une attaque cryptanalytique classique, n'aura t-on pas échangé la proie pour l'ombre ? La solution **hybride** est de mettre dans ses certificats X.509 deux clés et deux signatures, traditionnelle et post-quantique, et d'exiger que les deux signatures soient valides.

Ce RFC normalise donc les clés et signatures pour une configuration hybride, un algorithme traditionnel, qui peut être RSA, ECDSA ou EdDSA, et un algorithme post-quantique, actuellement uniquement ML-DSA. Le RFC utilise le terme de **composé** plutôt qu'hybride pour insister sur le fait que les deux algorithmes sont toujours utilisés ensemble.

Si vous voulez un arrière-plan sur ce post-quantique et ses motivations, voyez le RFC 9958 [1]. Tout le projet de transition vers le post-quantique est soumis à deux incertitudes, la date à laquelle on aura des CRQC ("*Cryptographically Relevant Quantum Computers*") et la sécurité des algorithmes post-quantiques, plus récents et moins testés. N'utiliser que l'algorithme traditionnel est imprudent, car des CRQC pourraient arriver un jour, n'utiliser que l'algorithme post-quantique est imprudent car on ne peut pas encore être parfaitement sûr de sa sécurité, à la fois face à la cryptanalyse, et face à une éventuelle erreur de programmation. D'où l'hybride, tel que décrit dans le RFC 9794. Certaines régulations, par exemple en France <https://cyber.gouv.fr/sites/default/files/document/Quantum_Key_Distribution_Position_Paper.pdf> ou en Allemagne <https://www.bsi.bund.de/SharedDocs/Downloads/

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc9958.txt

EN/BSI/Publications/Brochure/quantum-safe-cryptography.pdf> exigent ces systèmes hybrides.

Notre RFC 10073 pousse le concept d'hybride un peu plus loin avec un système **composé**, c'est-à-dire que les deux clés et les deux signatures seront toujours vues comme une (RFC 9794, section 2). Cela permet l'utilisation dans des protocoles qui ne connaissent pas le concept d'hybride et qui ne s'attendent pas à devoir faire deux vérifications. (L'idée originelle vient de cet article <https://link.springer.com/chapter/10.1007/978-3-319-59879-6_22>, ce qui n'a pas empêché des brevets comme US20220353061A1 ou US11968316B1 d'être déposés après <https://datatracker.ietf.org/ipr/4761/>.) Et les composés ont aussi l'avantage de moins changer les logiciels, facilitant une éventuelle re-certification.

Donc, résumons : ce RFC normalise un schéma hybride de signature qui inclus un algorithme post-quantique, ML-DSA (normalisé dans FIPS 204 <https://csrc.nist.gov/pubs/fips/204/final> et aussi un peu dans le RFC 9881), avec, au choix, RSA tel que décrit dans le RFC 8017, ECDSA tel que normalisé dans FIPS 186.5 <https://csrc.nist.gov/pubs/fips/186-5/final> ou encore Ed25519 (voir le RFC 8410). Les signatures, pour ce RFC, sont toujours combinées en une seule. Une vérification réussie implique que les deux signatures soient valides.

Je ne vais pas reprendre les parties purement cryptographiques du RFC (trop dur pour moi), regardez par exemple la section 4 si vous voulez voir l'encodage exact des clés et signatures. Point amusant : on met toujours clé et signature ML-DSA en premier car, contrairement à RSA, leurs tailles sont fixes et cette règle permet donc facilement de retrouver les deux clés ou les deux signatures. (Sinon, l'annexe A détaille les tailles de différents objets, vous y verrez que l'augmentation de taille due à l'utilisation d'une clé ou d'une signature composées est négligeable par rapport à ML-DSA seul.)

Le but de ce RFC est l'utilisation de ce schéma composé dans X.509/PKIX donc la section 5 décrit comment représenter clés et signatures dans un certificat. (Notez que le cas de CMS est repoussé à une future norme.) Comme ML-DSA ne permet que la signature, pas le chiffrement, le certificat peut avoir les utilisations `digitalSignature` ou `keyCertSign` mais pas `keyEncipherment` ou `encipherOnly` (même si le composant traditionnel l'aurait permis ; rappelez-vous qu'un schéma composé doit utiliser les deux algorithmes, le post-quantique et le traditionnel, sinon il ne fournit pas la sécurité attendue). Des identifiants sont réservés <https://www.iana.org/assignments/smi-numbers/smi-numbers.xml#smi-numbers-1.3.6.1.5.5.7.6> pour toutes les variantes possibles, par exemple `id-MLDSA65-Ed25519-SHA512` identifie un composé ML-DSA 65 avec Ed25519. Le module ASN.1 complet est en section 7 du RFC, avec, pour reprendre cet exemple :

```
id-MLDSA65-Ed25519-SHA512 OBJECT IDENTIFIER ::= {
     iso(1) identified-organization(3) dod(6) internet(1) security(5)
     mechanisms(5) pkix(7) alg(6) 48 }
```

Et comme la sécurité est compliquée, la section 9 précise un certain nombre de points pour faire proprement du composé avec ML-DSA. Elle rappelle qu'un schéma hybride de cryptographie peut servir à augmenter la sécurité (imposant à un attaquant de devoir casser l'algorithme post-quantique **et** le traditionnel) ou bien à permettre une transition en douceur au fur et à mesure de la disponibilité de logiciel qui connait la cryptographie post-quantique. Dans le premier cas, celui de notre RFC, on impose que les deux signatures soient valides. Dans le second, qui n'est **pas** le choix du RFC, qu'une seule le soit. (Je crois me souvenir qu'au début de ce RFC, le projet était d'avoir les deux possibilités. Quelqu'un a mieux suivi les débats que moi ?)

Certains experts n'aiment pas l'idée de l'hybride en disant que, si un attaquant ne sait attaquer qu'un seul des deux algorithmes, il ne va pas s'embêter, il va simplement génerer un faux document avec une

seule signature, celle de l'algorithme qu'il sait casser. La réponse à cet argument est que le vérificateur choisit sa politique : il peut décider de n'accepter que les hybrides.

Si vous voulez une analyse de sécurité plus détaillée, avec notamment la prise en compte des modèles de menace EUF-CMA et SUF-CMA, elle se trouve en section 9.2 (les systèmes décrits dans ce RFC résistent face à EUF-CMA mais pas face à SUF-DMA).

Autre question amusante (section 9.5), que se passe t-il lorsqu'un algorithme utilisé dans l'hybride est considéré comme dépassé ? Avec les systèmes à un seul algorithme, c'est clair : si on connait (ou soupçonne sérieusement) une attaque faisable contre l'algorithme, on arrête de l'utiliser. Mais avec un hybride, c'est plus compliqué de décider puisque, après tout, le deuxième algorithme tient toujours. Cela peut compliquer, par exemple, les analyses de sécurité puisqu'un audit trop sommaire verra l'algorithme vulnérable, sans tenir compte du fait qu'il n'est utilisé qu'au sein d'un système hybride.

La section 10 du RFC se penche ensuite sur quelques problèmes concrets. Par exemple, le bon sens nous dit que, dans un système hybride, il faut que les deux composants, le post-quantique et le traditionnel, aient à peu près la même « force », le même niveau de sécurité, sinon, le plus faible n'apportera pas grand'chose. Mais on ne sait pas quantifier cette « force » entre deux algorithmes aussi différents que, par exemple, ML-DSA et Ed25519. Et d'autant plus que ML-DSA, plus récent, est moins étudié ou testé « au feu ». Donc, le RFC ne prétend pas que tous les hybrides normalisés soient « équilibrés ». Le choix des combinaisons a plutôt été fait sur la base du réalisme (algorithmes courants). Lors de la discussion à l'IETF, plusieurs personnes ont estimé que ce RFC créait trop de composés et qu'il aurait fallu réduire le choix à quelques combinaisons bien choisies, mais le groupe de travail a finalement décidé de « donner » à chacun sa combinaison favorite.

Sinon, vous savez que les décideurs, en matière de sécurité, se préoccupent souvent davantage de conformité que de sécurité. Il y a des régulations diverses et il est plus important de les respecter que de choisir le « meilleur » système. La section 10.2 vient à leur aide en expliquant comment utiliser les systèmes composés de ce RFC si on doit respecter FIPS. Si un seul des deux algorithmes du composé est conforme à FIPS, il faudra présenter le système comme ayant la sécurité de ce composant, pas celle du composé complet. Si on n'a pas ce problème de conformité, et qu'on n'est pas expert en cryptographie soi-même, le composé recommandé par ce RFC est `id-MLDSA65-ECDSA-P256-SHA512`.

Enfin, l'annexe E contient les vecteurs de test, qui forment, en nombre de pages, la grande majorité du RFC. Plutôt que d'essayer de les extraire laborieusement du RFC, il vaut mieux aller les chercher en ligne <`https://github.com/lamps-wg/draft-composite-sigs/tree/main/src`>.