RFC 7905 : ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 juillet 2016

Date de publication du RFC : Juin 2016

https://www.bortzmeyer.org/7905.html

Ce court RFC ajoute les algorithmes de cryptographie ChaCha20 et Poly1305 à la liste de ceux utilisables dans le protocole TLS.

ChaCha20 (dérivé de l'ancien Salsa20) est un algorithme de chiffrement symétrique et Poly1305 un authentificateur. Tous les deux ont été conçus par Dan Bernstein et sont décrits dans le RFC 8439 [1]. (Ce nouveau RFC ne fait que de les adapter à leur usage dans le cas spécifique de TLS.) ChaCha a été utilisé dans BLAKE, la version de ce RFC, ChaCha20 doit son nom au fait qu'il exécute 20 tours ("*rounds*"). Quant à Poly1305, c'est un authentificateur de Wegman-Carter. Que fait un authentificateur ? Il prend une clé, un message et fabrique une étiquette. Un attaquant n'a qu'une probabilité infime de produire une étiquette valide.

Les deux algorithmes ont été conçus en pensant surtout à une mise en œuvre en logiciel (AES restant sans doute plus rapide quand on peut utiliser du matériel spécialisé <https://calomel.org/aesni_ssl_performance.html>. On trouve des mesures de performance dans cet article de Google <https://security.googleblog.com/2014/04/speeding-up-and-strengthening-https.html> ou cet article de Cloudflare <https://blog.cloudflare.com/do-the-chacha-better-mobile-perform>.)

Les algorithmes potentiellement concurrents ont des faiblesses : risques de sécurité pour AES-CBC ou RC4 (cf. RFC 7465), problèmes de performance pour les autres algorithmes AEAD comme AES-GCM. Comme RC4, ChaCha20 est un algorithme à flot continu, mais il n'a pas ses failles de sécurité.

Pour le cas de TLS (section 2 du RFC), ChaCha20 et Poly1305 sont utilisés ensemble, pour former un algorithme AEAD (RFC 5116). Son identifiant TLS est `AEAD_CHACHA20_POLY1305` et il peut s'utiliser avec les différents algorithmes d'authentification utilisés dans TLS. Par exemple, on peut avoir une session TLS dont la "*cipher suite*" est `TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256` ce qui veut dire :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc8439.txt

— Échange de clés Diffie-Hellman,
— Algorithme de cryptographie asymétrique (pour l'authentification) ECDSA,
— ChaCha20 et Poly1305 pour chiffrer les données,
— SHA-256 pour condenser.

Ces identifiants ont été ajoutés dans le registre IANA pour TLS <https://www.iana.org/assignments/tls-parameters/tls-parameters.xml#tls-parameters-4>.

Quel est le niveau de sécurité du nouvel algorithme ? Son prédécesseur Salsa20 a bénéficié d'analyses de sécurité sérieuses ("*Salsa20 security*" <http://cr.yp.to/snuffle/security.pdf> et "*The eSTREAM Portfolio*" <http://www.ecrypt.eu.org/stream/finallist.html>). ChaCha20 traite les failles connues de Salsa <http://eprint.iacr.org/2007/472.pdf>. Et il était utilisé dans un des finalistes du concours SHA-3, ce qui lui a valu d'autres examens de près.

Si, en plus de ChaCha20 et Poly1305, on utilise Curve25519 pour la cryptographie asymétrique, on aura une cryptographie tout-Bernstein, ce qui peut aussi amener à se poser des questions <http://www.metzdowd.com/pipermail/cryptography/2016-March/028824.html>.

Et les mises en œuvre ? ChaCha20 est dans OpenSSL depuis la version 1.1.0 <https://www.openssl.org/news/openssl-1.1.0-notes.html> (pas encore officiellement publiée, et qui semble encore peu répandue) et dans GnuTLS depuis la 3.4.0 <http://lists.gnutls.org/pipermail/gnutls-devel/2015-April/007535.html>. Il existe une liste apparemment à jour <https://ianix.com/pub/chacha-deployment.html> des mises en œuvre.