RFC 7321 : Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 19 septembre 2014

Date de publication du RFC : Août 2014

https://www.bortzmeyer.org/7321.html

Le protocole de cryptographie IPsec vient avec une liste d'obligations concernant les algorithmes cryptographiques qu'il **faut** inclure. Autrefois dans le RFC 4835[1], cette liste a été mise à jour dans ce RFC 7321, remplacé depuis par le RFC 8221. Ainsi, les différentes mises en œuvre d'IPsec sont sûres d'avoir un jeu d'algorithmes corrects en commun, assurant ainsi l'interopérabilité.

Plus précisément, ce nouveau RFC concerne les deux services d'IPsec, ESP ("*Encapsulating Security Payload*", RFC 4303) et AH ("*Authentication Header*", RFC 4302). Les RFC normatifs sur IPsec se veulent stables, alors que la cryptographie évolue. D'où le choix de mettre les algorithmes dans un RFC à part. Par exemple, la section 3.2 du RFC 4303 note « "*The mandatory-to-implement algorithms for use with ESP are described in a separate RFC, to facilitate updating the algorithm requirements independently from the protocol per se*" » (c'était à l'époque le RFC 4305, remplacé depuis par le RFC 4835, puis par notre RFC 7321, sept ans après son prédécesseur).

Ce RFC « extérieur » à IPsec spécifie les algorithmes obligatoires, ceux sur lesquels on peut toujours compter que le pair IPsec les comprenne, ceux qui ne sont pas encore obligatoires mais qu'il vaut mieux mettre en œuvre car ils vont sans doute le devenir dans le futur, et ceux qui sont au contraire déconseillés, en général suite aux progrès de la cryptanalyse, qui nécessitent de réviser régulièrement ce RFC (voir section 1). Cette subtilité (différence entre « obligatoire aujourd'hui » et « sans doute obligatoire demain ») mène à une légère adaptation des termes officiels du RFC 2119 : "*MUST-*" (avec le signe moins à la fin) est utilisé pour un algorithme obligatoire aujourd'hui mais qui ne le sera sans doute plus demain, en raison des avancées cryptanalytiques, et "*SHOULD+*" est pour un algorithme qui n'est pas obligatoire maintenant mais le deviendra sans doute.

La section 2 donne la liste des algorithmes. Je ne la répète pas intégralement ici. Parmi les points à noter :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4835.txt

— ESP a un mode de chiffrement intègre ("*authenticated encryption*" qu'on peut aussi traduire par chiffrement vérifié ou chiffrement authentifié, que je n'aime pas trop parce qu'on peut confondre avec l'authentification, cf. RFC 5116). Ce mode n'a pas d'algorithme obligatoire mais un "*SHOULD+*" qui sera peut-être donc obligatoire dans la prochaine version, AES-GCM (il était "*MAY*" dans le RFC 4835).
— Le mode le plus connu d'ESP, celui de chiffrement, a deux algorithmes obligatoires, AES-CBC (RFC 3602) et le surprenant `NULL`, c'est-à-dire l'absence de chiffrement (RFC 2410 ; on peut utiliser ESP pour l'authentification seule, d'où cet algorithme). Il y a aussi un algorithme noté "*MUST NOT*", DES-CBC (RFC 2405) qui ne doit pas être mis en œuvre, afin d'être sûr qu'on ne s'en serve pas (il était seulement "*SHOULD NOT*" dans le RFC 4835).
— Le mode d'authentification (enfin, intégrité serait peut-être un meilleur mot mais c'est subtil) d'ESP a un "*MUST*", HMAC-SHA1 (RFC 2404) mais aussi un "*SHOULD+*" qui pourra le rejoindre, AES-GMAC, GMAC étant une variante de GCM (et qui était en "*MAY*" dans le vieux RFC).
— Et AH, lui, a les mêmes algorithmes que ce mode d'authentification d'ESP.

La section 3 donne des conseils sur l'utilisation d'ESP et AH. AH ne fournit que l'authentification, alors qu'ESP peut fournir également le chiffrement. Bien sûr, le chiffrement sans l'authentification ne sert pas à grand'chose, puisqu'on risque alors de parler à l'homme du milieu sans le savoir (voir l'article de Bellovin, S. « "*Problem areas for the IP security protocols*" » dans les "*Proceedings of the Sixth Usenix Unix Security Symposium*" en 1996). Certaines combinaisons d'algorithmes ne sont pas sûres, par exemple, évidemment, ESP avec les algorithmes de chiffrement et d'authentification tous les deux à `NULL` (voir par exemple l'article de Paterson, K. et J. Degabriele, « "*On the (in)security of IPsec in MAC-then-encrypt configurations*" » à l'"*ACM Conference on Computer and Communications Security*" en 2010). Si on veut de l'authentification/intégrité sans chiffrement, le RFC recommande d'utiliser ESP avec le chiffrement `NULL`, plutôt que AH. En fait, AH est rarement utile, puisque ESP en est presque un sur-ensemble, et il y a même eu des propositions de le supprimer <`http://www.schneier.com/paper-ipsec.html`>. AH avait été prévu pour une époque où le chiffrement était interdit d'utilisation ou d'exportation dans certains pays et un logiciel n'ayant que AH posait donc moins de problèmes légaux. Aujourd'hui, la seule raison d'utiliser encore AH est si on veut protéger certains champs de l'en-tête IP, qu'ESP ne défend pas.

La section 4 de notre RFC donne quelques explications à certains des choix d'algorithmes effectués. Le chiffrement intègre/authentifié d'un algorithme comme AES-GCM (RFC 5116 et RFC 4106) est la solution recommandée dans la plupart des cas <`http://blog.cryptographyengineering.com/2012/05/how-to-choose-authenticated-encryption.html`>. L'intégration du chiffrement et de la vérification d'intégrité est probablement la meilleure façon d'obtenir une forte sécurité. L'algorithme de chiffrement AES-CTR (auquel on doit ajouter un contrôle d'intégrité) n'a pas de faiblesses cryptographiques, mais il ne fournit aucun avantage par rapport à AES-GCM (ne tapez pas sur le messager : c'est ce que dit le RFC, je sais que tous les cryptographes ne sont pas d'accord, par exemple parce qu'ils trouvent GCM beaucoup plus complexe).

Par contre, Triple DES et DES, eux, ont des défauts connus et ne doivent plus être utilisés. Triple DES a une taille de bloc trop faible et, au-delà d'un gigaoctet de données chiffrées avec la même clé, il laisse fuiter des informations à un écoutant, qui peuvent l'aider dans son travail de décryptage. Comme, en prime, il est plus lent qu'AES, il n'y a vraiment aucune raison de l'utiliser. (DES est encore pire, avec sa clé bien trop courte. Il a été cassé avec du matériel dont les plans sont publics.)

Pour l'authentification/intégrité, on sait que MD5 a des vulnérabilités connues (RFC 6151), question résistance aux collisions. Mais cela ne gêne pas son utilisation dans HMAC-MD5 donc cet algorithme, quoique non listé pour IPsec, n'est pas forcément ridicule aujourd'hui. SHA-1 a des vulnérabilités analogues (quoique beaucoup moins sérieuses) mais qui ne concernent pas non plus son utilisation dans HMAC-SHA1, qui est donc toujours en "*MUST*". Bien que les membres de la famille SHA-2 n'aient pas ces défauts, ils ne sont pas cités dans ce RFC, SHA-1 étant très répandu et largement suffisant.

Dans le précédent RFC, Triple DES était encore noté comme une alternative possible à AES. Ce n'est plus le cas aujourd'hui, où les vulnérabilités de Triple DES sont bien connues (sans compter ses performances bien inférieures). Triple DES est maintenu dans IPsec (il est en "*MAY*") mais uniquement pour des raisons de compatibilité avec la base installée. Le problème est qu'il n'y a donc plus de solution de remplacement si un gros problème est découvert dans AES (section 5, sur la diversité des algorithmes). Il n'y a aucune indication qu'une telle vulnérabilité existe mais, si elle était découverte, l'absence d'alternative rendrait le problème très sérieux.

Voilà, c'est fini, la section 8 sur la sécurité rappelle juste quelques règles bien connues, notamment que la sécurité d'un système cryptographique dépend certes des algorithmes utilisés mais aussi de la qualité des clés, et de tout l'environnement (logiciels, humains).

Ce RFC se conclut en rappelant que, de même qu'il a remplacé ses prédécesseurs comme le RFC 4835, il sera probablement à son tour remplacé par d'autres RFC, au fur et à mesure des progrès de la recherche en cryptographie. (Ce fut fait avec le RFC 8221.)

Si vous voulez comparer avec un autre document sur les algorithmes cryptographiques à choisir, vous pouvez par exemple regarder l'annexe B1 du RGS, disponible en ligne <https://www.ssi.gouv.fr/entreprise/reglementation/confiance-numerique/le-referentiel-general-de-securite->.

Merci à Florian Maury pour sa relecture acharnée. Naturellement, comme c'est moi qui tiens le clavier, les erreurs et horreurs restantes viennent de ma seule décision. De toute façon, vous n'alliez pas vous lancer dans la programmation IPsec sur la base de ce seul article, non ?