RFC 7518 : JSON Web Algorithms (JWA)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 mai 2015

Date de publication du RFC : Mai 2015

https://www.bortzmeyer.org/7518.html

Ce nouveau RFC fait partie de la série des RFC JOSE <https://www.bortzmeyer.org/jose.html> (cryptographie pour JSON). Il normalise les algorithmes cryptographiques utilisables pour signer ou chiffrer des textes JSON et crée des registres IANA pour les conserver.

Comme le fait en général l'IETF en matière de cryptographie, les algorithmes cryptographiques ne sont pas inscrits « en dur » dans les RFC (rappelez-vous qu'un RFC, une fois publié, n'est jamais modifié) mais mis dans un registre IANA. En effet, la cryptanalyse progresse régulièrement et des algorithmes qui semblaient sûrs à un moment ne le sont plus par la suite. Il est plus facile et plus rapide de modifier un registre IANA que de créer un nouveau RFC.

Pour chaque registre, les algorithmes cités sont marqués Obligatoire, Recommandé ou Facultatif (cf. RFC 2119 [1]). En n'utilisant que des algorithmes marqués Obligatoire, on augmente l'interopérabilité.

Commençons par les algorithmes utilisés pour les signatures et MAC (section 3). C'est le membre `alg` de JWS ("*JSON Web Signature*", cf. RFC 7515). La table complète est dans son registre IANA <https://www.iana.org/assignments/jose/jose.xhtml#web-signature-encryption-algorithms>. Comme JOSE privilégie la concision, les termes utilisés sont plus courts que, par exemple, dans les registres prévus pour TLS. Ainsi, `HS512` veut dire « HMAC avec SHA-512 ». (Pour l'algorithme HMAC, voir le RFC 2104.) Et `ES256` désignera un algorithme de signature couplant ECDSA avec une courbe P-256 et SHA-256 (on peut aussi signer avec RSA mais les signatures sont plus courtes avec les courbes elliptiques).

Il y a même un algorithme `none` dans le registre, dont le nom indique clairement que le texte JSON n'est pas authentifié, ni protégé en intégrité. A priori, les applications qui veulent vérifier le JSON reçu

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc2119.txt

vont rejeter des textes ainsi marqués (sections 3.6 et aussi 8.5), sauf cas très spécifiques, par exemple si on est absolument certain que le texte JSON est arrivé sur une connexion sécurisée (mais relire la section 8.5, sur les pièges que cela pose).

Maintenant, le chiffrement. Il faut pouvoir indiquer comment ont été chiffrées les clés ayant servi à chiffrer le contenu. La section 4 décrit les algorithmes prévus pour cela (qui sont aussi dans le même registre IANA <https://www.iana.org/assignments/jose/jose.xhtml#web-signature-encryption-a
et qu'on met dans le membre `alg` d'un JWE ("*JSON Web Encryption*", cf. RFC 7516).

Ainsi, un cas simple est celui où les deux parties chiffrent avec une clé statique symétrique, prééchangée entre eux. On met alors `dir` (pour "*direct use of a key*") comme algorithme. Si au contraire on chiffre la clé de chiffrement avec RSA, on mettra par exemple l'algorithme `RSA1_5` qui veut dire « RSA avec PKCS#1 ». (On a droit aussi, bien sûr, aux échanges de clés Diffie-Hellman.)

Et le contenu lui-même, on le chiffre avec quoi ? La section 5 décrit les algorithmes de chiffrement symétriques utilisés (encore le même registre IANA). Ils se mettent dans le membre `enc` du JWE. Par exemple, `A128CBC-HS256` est AES en mode CBC avec un HMAC sur SHA-256, alors que `A256GCM` est AES mais en mode GCM (qui, fournissant du chiffrement intègre, dispense d'utiliser HMAC).

À noter que le petit nouveau à l'IETF, le ChaCha20 du RFC 8439, n'est pas encore arrivé dans JOSE et ne figure pas dans ce registre, qui ne compte qu'AES.

Passons maintenant aux clés (section 6), représentés par un JWK ("*JSON Web Key*", RFC 7517. Dans le membre `kty` ("*Key TYpe*") d'un JWK, on peut trouver `RSA`, `EC` ("*Elliptic Curve*") ou `oct` ("*octet*"), ce dernier désignant une clé symétrique. Les paramètres dépendent ensuite du type de clés. Par exemple, pour les courbes elliptiques, `crv` désigne la courbe utilisée (P-256, P-384, etc, les valeurs possibles ayant leur propre registre <https://www.iana.org/assignments/jose/jose.xhtml#web-key-elliptic-curve>), pour RSA, `n` sera le modulo et `e` l'exposant. Les valeurs possibles pour ces paramètres de clés sont dans un registre IANA <https://www.iana.org/assignments/jose/jose.xhtml#web-key-parameters>.

Je l'ai dit au début, les progrès de la cryptanalyse nécessitent de changer les algorithmes de temps en temps (section 8.1 sur l'« agilité cryptographique »). La section 7 fixe les règles de modification des registres. En gros, il faut avoir une spécification écrite (pas forcément un RFC) pour voir son algorithme entrer dans le registre ("*Specification Required*", RFC 5226, section 4.1).