RFC 6955 : Diffie-Hellman Proof-of-Possession Algorithms

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 30 mai 2013

Date de publication du RFC : Mai 2013

https://www.bortzmeyer.org/6955.html

Lorsqu'on détient une clé cryptographique privée et qu'on veut le prouver à un correspondant, la méthode la plus simple et la plus standard est de signer un message avec cette clé et de l'envoyer audit correspondant, qui, en vérifiant la signature, s'assurera qu'on était bien en possession de la clé privée. Mais dans certains cas, signer n'est pas possible. Ce RFC décrit trois algorithmes qui permettent de prouver qu'on connait une clé, sans signer. Il remplace le RFC 2875 [1].

Ce test de POP ("*Proof Of Possession*") est un de ceux que doit faire, par exemple, une autorité de certification, avant de signer un certificat. (Le RFC 4211, annexe C, explique pourquoi c'est important.) Ainsi, une CSR (demande de signature d'un certificat) faite par OpenSSL est signée et cela peut se vérifier :

```
% openssl req  -verify -in server.csr -noout
verify OK
```

Le format CRMF du RFC 4211 et le format PKCS#10 permettent tous les deux d'envoyer une demande de signature de certificat à une AC mais seul CRMF a un moyen d'inclure une POP pour les algorithmes ne permettant pas de signature. PKCS#10 (RFC 2986) n'a rien. Cela existe, de tels algorithmes, ne permettant pas de signature ? Oui, c'est le cas de Diffie-Hellman et de ECDH.

Les trois algorithmes de POP de notre RFC sont le Diffie-Hellman statique en section 4, le logarithme discret en section 6 et l'ECDH statique en section 6. Ne me demandez pas de vous les expliquer, cela dépasse largement mes compétences en cryptographie. Lisez le RFC pour cela. (Et révisez le RFC 6090 pour le troisième, qui utilise les courbes elliptiques.) Notre RFC 6955 fournit les algorithmes et les modules ASN.1 qui les décrivent.

La section 1.1 décrit les changements depuis le RFC 2875. Les deux algorithmes du précédent RFC ont été réécrits pour permettra leur paramétrisation par rapport à la fonction de condensation (obligatoirement SHA-1 dans l'ancien RFC, alors que plusieurs fonctions sont désormais possibles, notamment la famille SHA-2). Et un troisième algorithme a été ajouté, pour ECDH.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc2875.txt