RFC 8709 : Ed25519 and Ed448 Public Key Algorithms for the Secure Shell (SSH) Protocol

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 février 2020

Date de publication du RFC : Février 2020

https://www.bortzmeyer.org/8709.html

Un très court RFC, juste pour ajouter au protocole SSH les algorithmes de signature Ed25519 et Ed448. Ces algorithmes sont déjà disponibles dans OpenSSH.

Ce protocole SSH est normalisé dans le RFC 4251 [1], et a de nombreuses mises en œuvre, par exemple dans le logiciel libre OpenSSH. Pour authentifier le serveur, SSH dispose de plusieurs algorithmes de signature. Ce nouveau RFC en ajoute deux, dont Ed25519, qui avait été normalisé dans le RFC 8032. (En toute rigueur, l'algorithme se nomme EdDSA et Ed25519 est une des courbes elliptiques possibles avec cet algorithme. Mais je reprends la terminologie du RFC.) À noter que les courbes elliptiques sous-jacentes peuvent également être utilisées pour l'échange de clés de chiffrement, ce que décrit le RFC 8731.

La section 3 de notre RFC donne les détails techniques, suivant le RFC 4253. L'algorithme se nomme `ssh-ed25519`. Son copain avec la courbe elliptique Ed448 est `ssh-ed448`. Ils sont tous les deux enregistrés à l'IANA <https://www.iana.org/assignments/ssh-parameters/ssh-parameters.xml#ssh-parameters-19>.

Le format de la clé publique est la chaîne "ssh-ed25519" suivie de la clé, telle que décrite dans le RFC 8032, section 5.1.5 (et 5.2.5 pour Ed448). Avec OpenSSH, vous pouvez la voir dans `/.ssh/id_ed25519.pub`. Les signatures sont faites selon la technique du RFC 8032, sections 5.1.6 et 5.2.6. Leur format est décrit en section 6, et la vérification de ces signatures en section 7, en suivant la procédure des sections 5.1.7 et 5.2.7 du RFC 8032.

La façon la plus courante de vérifier la clé publique du serveur SSH auquel on se connecte est le TOFU. Si on préfère une vérification plus sérieuse, on peut utiliser les clés SSH publiées dans le DNS, méthode décrite dans le RFC 4255, utilisant des enregistrements de type SSHFP. Cela fait longtemps que ces enregistrements peuvent utiliser Ed25519 (cf. RFC 7479) et notre RFC ajoute le cas de Ed448, par exemple :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc4251.txt`

```
example.net.  IN SSHFP 6 2 ( a87f1b687ac0e57d2a081a2f282672334d90ed316d2b818ca9580ea384d924 01 )
```

(Il est enregistré à l'IANA <https://www.iana.org/assignments/dns-sshfp-rr-parameters/dns-sshfp-rr-parameters.xml#dns-sshfp-rr-parameters-1>.)

Ed25519 a été ajouté à OpenSSH en janvier 2014 <http://www.openssh.com/txt/release-6.5> (donc bien avant la publication de ce RFC.) C'est l'option `-t` de `ssh-keygen` qui permet de sélectionner cet algorithme :

```
% ssh-keygen -t ed25519  -f /tmp/ed25519
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /tmp/ed25519.
Your public key has been saved in /tmp/ed25519.pub.
The key fingerprint is:
SHA256:VEN6HVM0CXq+TIflAHWCOQ88tfR35WXQZ675mLIhIIs stephane@godin
The key's randomart image is:
+--[ED25519 256]--+
|         o==O+*++|
|         oB* B.+*|
|        o o== oo=|
|       . . o.= .o|
|      . S   + oo |
|     . o . o oo  |
|    E .   . +  + |
|           ...o .|
|            .o   |
+----[SHA256]-----+
```

À noter que OpenSSH 7.6 n'a pas ed448. D'une manière générale, ed25519 a été beaucoup plus souvent mise en œuvre dans les clients et serveurs SSH.