RFC 9934 : Privacy-Enhanced Mail (PEM) File Format for Encrypted ClientHello (ECH)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 mars 2026

Date de publication du RFC : Mars 2026

https://www.bortzmeyer.org/9934.html

La technique ECH ("*Encrypted Client Hello*") permet de boucher une faille de TLS, la transmission en clair du nom du serveur demandé. Elle est normalisée dans le RFC 9849[1] mais il y manquait la description d'un format standard pour envoyer les données nécessaires à ECH. C'est désormais fait (et vous reconnaitrez le classique format PEM, cf. RFC 7468).

Le principe d'ECH est de chiffrer le nom du serveur demandé, avec la clé publique du serveur TLS. Pour configurer le serveur, il faut la clé privée correspondante et différents détails, la `ECHConfig` (RFC 9849, section 4). C'est pour `ECHConfig` que notre RFC normalise un format.

Le fichier PEM (RFC 7468) contient une ou plusieurs clés privées et des informations de configuration (pour un ou plusieurs serveurs), le tout en Base64 (section 4 du RFC 4648). Les informations publiques sont typiquement mises dans le DNS (RFC 9848), la clé privée, cela va sans dire (mais le RFC le dit quand même, on ne sait jamais) n'est pas publiée. Un fichier PEM contenant des clés privées doit évidemment être bien protégé.

Voici un exemple, tiré du RFC (la clé privée a donc déjà été publiée [Caractère Unicode non montré [2]]) :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc9849.txt`
2. Car trop difficile à faire afficher par LaTeX

```
-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VuBCIEICjd4yGRdsoP9gU7YT7My8DHx1Tjme8GYDXrOMCi8v1V
-----END PRIVATE KEY-----
-----BEGIN ECHCONFIG-----
AD7+DQA65wAgACA8wVN2BtscOl3vQheUzHeIkVmKIiydUhDCliA4iyQRCwAEAAEA
AQALZXhhbXBsZS5jb20AAA==
-----END ECHCONFIG-----
```

Dans le DNS, la publication de la clé publique et de la configuration donnerait quelque chose du genre :

```
% dig +short HTTPS foo.example.com
1 . ech=AD7+DQA65wAgACA8wVN2BtscOl3vQheUzHeIkVmKIiydUhDCliA4iyQRwAEAAEAAQALZXhhbXBsZS5jb20AAA==
```

Bien des logiciels reconnaissent cette syntaxe. Je ne les ai pas testé mais, apparemment :

— Dans une branche d'OpenSSL <https://github.com/openssl/openssl/tree/feature/ech> (les tests sont dans `test/ech_test.c`, avec des exemples du format de ce RFC. Apache ou HAProxy peuvent l'utiliser.
— Pour BoringSSL <https://boringssl.googlesource.com/boringssl/>, il y a ce script <https://github.com/defo-project/ech-dev-utils/blob/nginx-pr/scripts/bssl2pem.sh>.