RFC 7925 : Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 juillet 2016

Date de publication du RFC : Juillet 2016

https://www.bortzmeyer.org/7925.html

Dans le futur, nous répètent les consultants, il y aura partout dans nos maisons, bureaux et usines, des petits objets électroniques connectés à l'Internet. C'est le fameux Internet des Objets. Ces choses serviront à mesurer le monde et à le modifier et poseront donc des problèmes de sécurité. Par exemple, sans précautions particulières, un capteur dans une usine qui envoie en Wi-Fi les informations qu'il récolte permettrait à un écoutant indiscret, même situé en dehors de l'usine, de surveiller l'activité de celle-ci. Il va donc de soi qu'il faut utiliser de la cryptographie dans ces objets. Mais celle-ci est parfois coûteuse en ressources machine, et ce nouveau RFC propose donc des **profils** du protocole TLS, limitant les options possibles de façon à économiser les ressources des objets.

Ces objets sont en effet **contraints** au sens du RFC 7228 [1]. Un TLS complet serait trop pour eux. Bien sûr, on pourrait concevoir des protocoles cryptographiques spécialement conçus à leur intention mais il faudrait développer, valider et déboguer ces protocoles et l'expérience de la cryptographie sur l'Internet montre que c'est beaucoup plus difficile que ça n'en a l'air. D'où le choix d'utiliser TLS (RFC 5246), protocole connu et éprouvé.

Notons au passage que, contrairement à ce que dit le RFC, le principal danger que pose l'Internet des Objets ne vient pas forcément d'un tiers inconnu : presque toute la domotique actuelle fonctionne avec le "*cloud*", toutes les données récoltées étant envoyées sur les serveurs du vendeur, qui peut en faire ce qu'il veut <https://www.bortzmeyer.org/objets-esclaves-federez.html>. Chiffrer le trafic entre l'objet et ces serveurs ne comble pas cette énorme faille de sécurité.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7228.txt

Les profils de TLS (et de DTLS, son équivalent pour UDP, cf. RFC 6347) spécifiés dans ce RFC peuvent s'appliquer aux communications utilisant CoAP (RFC 7252) comme à celles utilisant d'autres protocoles applicatifs. Ce sont des profils, des **restrictions** de TLS, et ils n'introduisent donc pas un nouveau protocole, ni même de nouvelles extensions à ces protocoles.

Par exemple (section 4.4), ce RFC impose que, si on authentifie avec un certificat, les certificats X.509 utilisent ECDSA uniquement. (Gérer tous les algorithmes possibles dans un objet contraint serait trop coûteux, par exemple en occupation de la flash.) Au revoir, RSA, donc.

Toujours sur les certificats, le profil abandonne OCSP (RFC 6960) et les CRL (qui ne marchent guère, en pratique) : la révocation des certificats devra se faire uniquement par le biais d'une mise à jour du logiciel de l'objet.

Toujours concernant la cryptographie, ce RFC impose (section 13) de n'utiliser que des suites de chiffrement intègres (AEAD) comme `TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8` (AES avec CCM, cf. RFC 7251).

Comme l'utilisation de TLS dans ces objets contraints est récente, il n'y a pas trop à se préoccuper de la base installée. On peut donc décider de ne jamais gérer des trucs trop anciens. C'est ainsi que la version minimale de TLS acceptée dans les profils de ce RFC est la 1.2 (section 18).

Il y a quelques points où ce RFC rend obligatoire des fonctions qui étaient optionnelles dans TLS, parce qu'elles sont cruciales pour des objets contraints. C'est le cas de la reprise de session TLS précédente, qui permet d'éviter de refaire des opérations cryptographiques coûteuses à chaque démarrage. Par contre, certaines fonctions sont déconseillées, comme la compression, coûteuse et sans doute inutile dans l'Internet des Objets où les protocoles sont déjà optimisés pour réduire la taille des données échangées. (En outre, il existe de bonnes raisons de sécurité de couper la compression TLS, RFC 7525, section 3.3.) Idem pour la renégotiation du RFC 5746, qui est exclue.

Un problème courant en cryptographie est celui de la génération de nombres aléatoires (TLS en utilise plusieurs). Il est encore plus aigu dans l'Internet des Objets car ces objets n'ont souvent pas beaucoup de sources d'entropie (section 12 du RFC). Pas de disque dur mobile, pas de clavier, pas de souris dont les mouvements sont partiellement aléatoires. Il y a donc un risque élevé que tous les objets identiques génèrent les mêmes nombres « aléatoires ». Bref, il faut que les développeurs fassent très attention aux conseils du RFC 4086.

Autre manque des objets contraints, celui d'une horloge fiable. Pour des opérations comme la validation de la non-expiration d'un certificat, cela peut être très gênant.

Les objets contraints sont... contraints de bien d'autre façon, comme en puissance du processeur. Cela peut rendre certaines opérations cryptographiques irréalistes en logiciel. La section 19 du RFC donne des conseils sur la mise en œuvre matérielle de ces opérations :

— Les rendre accessibles au programmeur, pas uniquement au système de base, de manière à ce que la mise en œuvre de TLS puisse bénéficier, par exemple, de l'AES présent dans le matériel, au lieu de devoir le réécrire en moins efficace.
— Mais être souple, car un même algorithme cryptographique peut être utilisé de plusieurs façons. Ainsi, certaines mises en œuvre matérielles d'AES-CCM sont conçues pour Bluetooth avec un numnique <https://www.bortzmeyer.org/nonce.html> de taille fixe, qui n'est pas celle utilisée par TLS, ce qui interdit leur réutilisation.

— Penser à l'agilité cryptographique (la possibilité de changer d'algorithme cryptographique, pour suivre les progrès de la cryptanalyse). Par exemple, il faut se demander comment permettre à ChaCha20 (RFC 8439 et RFC 7905) de remplacer AES dans le futur.

Toujours en cryptographie, la section 20 donne des recommendations sur la longueur des clés. Un point important, et spécifique aux objets, est que les engins déployés restent souvent sur le terrain très longtemps, par exemple dix ans. On ne remplace pas les capteurs, ou l'ordinateur embarqué dans une machine, comme on change d'iPhone ! La longueur des clés doit donc être prévue pour les attaques du futur, pas pour celles d'aujourd'hui.

Les objets dont on parle dans ce RFC sont souvent déployés dans des environnements où les contraintes de vie privée sont fortes. Cela peut être une usine qui ne veut pas que ses processus soient espionnés, ou bien une maison dont les habitants ne veulent pas être surveillés (section 22 du RFC). TLS ne protège pas contre tout et certaines fuites d'information persistent même quand il est utilisé. Elles peuvent venir du protocole TLS lui-même (la reprise de session utilise des informations qui permettent de suivre une machine à la trace) ou bien provenir de métadonnées diverses. Par exemple, s'il existe un capteur de présence qui envoie un message lorsque quelqu'un rentre dans l'appartement, chiffrer ce message ne fait pas gagner grand'chose : une fois qu'un observateur a identifié le capteur de présence, le seul envoi du message suffit à l'informer d'une arrivée. Une vraie protection de la vie privée peut donc nécessiter des méthodes additionnelles comme l'envoi de trafic bidon.

Le RFC se termine en rappelant qu'il est crucial de fournir un mécanisme de mise à jour simple des engins. Les objets de l'Internet des Objets sont souvent de type « je pose et puis j'oublie », sans aucun moyen de mettre à jour leur logiciel. Or, la sécurité peut nécessiter une telle mise à jour, soit pour corriger une bogue dans la bibliothèque TLS, soit pour révoquer des clés et en mettre d'autres. Un exemple d'un mécanisme de mise à jour adapté est LWM2M <http://openmobilealliance.org/about-oma/work-program/m2m-enablers/>.

Si vous voulez rire un peu, l'annexe A du RFC précise comment faire tourner DTLS sur... SMS.