Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 7525
BCP 195
Catégorie : Bonnes pratiques actuelles
ISSN : 2070-1721

Y. Sheffer, Intuit
R. Holz, NICTA
P. Saint-Andre, &yet

mai 2015
Traduction Claude Brière de L'Isle

Recommandations pour l'utilisation sûre de la sécurité de la couche transport (TLS) et de la sécurité de la couche transport de datagrammes (DTLS)

Résumé
La sécurité de la couche transport (TLS, *Transport Layer Security*) et la sécurité de la couche transport de datagrammes (DTLS, *Datagram Transport Layer Security*) sont largement utilisées pour protéger les données échangées sur des protocoles d'application comme HTTP, SMTP, IMAP, POP, SIP, et XMPP. Au cours de ces dernières années, plusieurs attaques sérieuses sur TLS sont apparues, incluant des attaques sur ses suites de chiffrement les plus couramment utilisées et leur mode de fonctionnement. Le présent document fournit des recommandations pour améliorer la sécurité des services déployés qui utilisent TLS et DTLS. Les recommandations sont applicables à la majorité des cas d'utilisation.

Statut de ce mémoire
Le présent document documente les bonnes pratiques actuelles de l'Internet.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Plus d'informations sur les normes de l'Internet sont disponibles à la Section 2 de la [RFC5741].

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc7525

Table des Matières

1. Introduction

La sécurité de la couche transport (TLS, *Transport Layer Security*) [RFC5246] et la sécurité de la couche transport de datagrammes (DTLS, *Datagram Transport Layer Security*) [RFC6347] sont largement utilisées pour protéger les données échangées sur des protocoles d'application comme HTTP, SMTP, IMAP, POP, SIP, et XMPP. Au cours de ces dernières années, plusieurs attaques sérieuses sur TLS sont apparues, incluant des attaques sur ses suites de chiffrement les plus couramment utilisées et leur mode de fonctionnement. Par exemple, les deux algorithmes de chiffrement AES-CBC [RFC3602] et RC4 [RFC7465], qui sont les deux chiffrements les plus largement déployés, ont été attaqués dans le contexte de TLS. Un document d'accompagnement [RFC7457] donne des informations détaillées sur ces attaques et aidera le lecteur à comprendre les raisons des recommandations fournies ici.

À cause de ces attaques, ceux qui mettent en œuvre et déploient TLS et DTLS ont besoin de lignes directrices à jour sur la façon dont TLS peut être utilisé en toute sécurité. Le présent document fournit des lignes directrices pour la mise en œuvre des services déployés ainsi que des logiciels, en supposant que l'utilisateur s'attend à ce que son code soit déployé dans les environnements définis à la Section 5. En fait, le présent document appelle au déploiement d'algorithmes largement mis en œuvre mais pas encore largement déployés. Concernant le déploiement, le présent document vise une large audience – à savoir tous ceux qui souhaitent ajouter à leurs communications l'authentification (qu'elle soit unidirectionnelle ou mutuelle) la confidentialité, et la protection de l'intégrité des données.

Les recommandations faites ici prennent en considération la sécurité de divers mécanismes, leur maturité et interopérabilité technique, et leur prévalence dans les mises en œuvre au moment de la rédaction. Sauf si il est explicitement mentionné que une recommandation s'applique à TLS seul ou à DTLS seul, chaque recommandation s'applique à la fois à TLS et DTLS.

Il est prévu que la spécification de TLS 1.3 résolve beaucoup des vulnérabilités énumérées dans le présent document. Un système qui déploie TLS 1.3 devrait avoir moins de vulnérabilités que TLS 1.2 ou antérieur. Le présent document sera probablement mis à jour lorsque TLS 1.3 aura connu un déploiement notable.

Ces recommandations sont un minimum pour l'utilisation de TLS dans la vaste majorité des scénarios de mise en œuvre et déploiement, à l' exception de TLS non authentifié (voir la Section 5). D'autres spécifications qui font référence au présent document peuvent avoir des exigences plus strictes sur un ou plusieurs aspects du protocole, sur la base de leurs circonstances particulières (par exemple, pour l'utilisation d'un protocole d'application particulier) ; lorsque c'est le cas, il est conseillé aux utilisateurs de respecter ces exigences plus strictes. De plus, le présent document donner un plancher, pas un plafond, de sorte que des options plus fortes sont toujours permises (par exemple, en fonction des différences d'évaluation de l'importance de la force cryptographique par rapport à la charge de calcul).

Les connaissances de la communauté sur la force des divers algorithmes et des attaques faisables peuvent changer rapidement, et l'expérience montre qu'un document des bonnes pratiques actuelles (BCP, *Best Current Practice*) sur la sécurité est une déclaration à un moment donné. Il est conseillé de rechercher tous les errata ou mises à jour qui s'appliquent au présent document.

2. Terminologie

Un certain nombre de termes relatifs à la sécurité dans le présent document sont utilisés dans le sens défini dans la [RFC4949].

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

3. Recommandations générales

Cette section fournit des recommandations générales sur l'utilisation sûre de TLS. Les recommandations relatives aux suites de chiffrement sont discutées dans la section suivante.

3.1 Versions de protocole

3.1.1 Versions de protocole SSL/TLS

Il est important à la fois d'arrêter d'utiliser les anciennes versions moins sûres de SSL/TLS et de commencer à utiliser les versions modernes, plus sûres ; donc, ce qui suit sont des recommandations concernant les versions de protocole TLS/SSL :

o Les mises en œuvre NE DOIVENT PAS négocier SSL version 2.
Raison : aujourd'hui, SSLv2 est considéré comme non sûr [RFC6176].

o Les mises en œuvre NE DOIVENT PAS négocier SSL version 3.
Raison : SSLv3 [RFC6101] était une amélioration de SSLv2 et comblait des trous de sécurité significatifs mais ne prenait pas en charge de fortes suites de chiffrement. SSLv3 ne prend pas en charge les extensions à TLS, dont certaines (par exemple, renegotiation_info [RFC5746]) sont critiques pour la sécurité. De plus, avec l'émergence de l'attaque POODLE [POODLE], SSLv3 est maintenant largement reconnu comme fondamentalement non sûr. Voir les détails dans la [RFC7568].

o Les mises en œuvre NE DEVRAIENT PAS négocier TLS version 1.0 [RFC2246] ; la seule exception est lorsque aucune version supérieure n'est disponible dans la négociation.
Raison : TLS 1.0 (publiée en 1999) ne prend pas en charge de nombreuses suites de chiffrement modernes plus fortes. De plus, TLS 1.0 n'a pas de valeur d'initialisation (IV) par enregistrement pour les suites de chiffrement fondées sur CBC et n'avertit pas contre les erreurs de bourrage courantes.

o Les mises en œuvre NE DEVRAIENT PAS négocier TLS version 1.1 [RFC4346] ; la seule exception est lorsque aucune version supérieure n'est disponible dans la négociation.
Raison : TLS 1.1 (publiée en 2006) est une amélioration de la sécurité par rapport à TLS 1.0 mais ne prend pas en charge certaines suites de chiffrement plus fortes.

o Les mises en œuvre DOIVENT prendre en charge TLS 1.2 [RFC5246] et DOIVENT préférer négocier TLS version 1.2 plutôt que les versions antérieures de TLS.
Raison : plusieurs suites de chiffrement plus fortes ne sont disponibles qu'avec TLS 1.2 (publiée en 2008). En fait, les suites de chiffrement recommandées par le présent document (paragraphe 4.2) ne sont disponible que dans TLS 1.2.

Ces BCP s'appliquent à TLS 1.2 et aussi aux versions antérieures. Il n'est pas sûr pour les lecteurs de supposer que les recommandations de ces BCP s'appliquent à une future version de TLS.

3.1.2 Versions de protocole DTLS

DTLS, une adaptation de TLS pour les datagrammes UDP, a été introduit lorsque TLS 1.1 a été publié. Voici les recommandations à l'égard de DTLS :

o Les mises en œuvre NE DEVRAIENT PAS négocier DTLS version 1.0 [RFC4347].
La version 1.0 de DTLS est corrélée à la version 1.1 de TLS (voir ci-dessus).

o Les mises en œuvre DOIVENT prendre en charge et DOIVENT préférer négocier DTLS version 1.2 [RFC6347].
La version 1.2 de DTLS est corrélée à la version 1.2 de TLS (voir ci-dessus). (Il n'y a pas de version 1.1 de DTLS.)

3.1.3 Repli sur des versions inférieures

Les clients qui "se replient" sur des versions antérieures du protocole après le rejet par le serveur des versions de numéro plus élevé du protocole NE DOIVENT PAS revenir à SSLv3 ou antérieure.

Raison : certaines mises en œuvre de clients reviennent à des versions antérieures de TLS ou même à SSLv3 si le serveur a rejeté des versions supérieures du protocole. Ce repli peut être forcé par un attaquant interposé (MITM, *man-in-the-middle*). TLS 1.0 et SSLv3 sont significativement moins sûrs que TLS 1.2, la version recommandée par le présent

document. Alors que les serveurs TLS 1.0 seul sont encore assez courants, les analyses d'IP montrent que les serveurs SSLv3 seul ne représentent qu'environ 3 % de la population actuelle de serveurs de la Toile. (Au moment de la rédaction du présent document, une méthode explicite pour empêcher les attaques en dégradation a été définie dans la [RFC7507].)

3.2 TLS strict

Les recommandations suivantes sont fournies pour aider à empêcher le déshabillage SSL (une attaque qui est résumée au paragraphe 2.1 de la [RFC7457]):

- Dans les cas où un protocole d'application permet la mise en œuvre ou le déploiement d'un choix entre une configuration TLS stricte et une mise à niveau dynamique de trafic non chiffré à du trafic protégé par TLS (comme STARTTLS) clients et serveurs DEVRAIENT préférer la configuration TLS stricte.

- Les protocoles d'application fournissent normalement un moyen pour que le serveur offre TLS durant un échange initial de protocole, et parfois aussi fournissent un moyen pour que le serveur annonce la prise en charge de TLS (par exemple, avec un fanion qui indique que TLS est exigé) ; malheureusement, ces indications sont envoyées avant que le canal de communication soit chiffré. Un client DEVRAIT tenter de négocier TLS même si ces indications ne sont pas communiquées par le serveur.

- Les mises en œuvre de client et serveur HTTP DOIVENT prendre en charge l'en-tête de sécurité de transport HTTP strict (HSTS, *HTTP Strict Transport Security*) [RFC6797], afin de permettre aux serveurs de la Toile d'annoncer qu'ils acceptent les clients TLS seul.

- Les serveurs de la Toile DEVRAIENT utiliser HSTS pour indiquer qu'ils acceptent les clients TLS seul, sauf si ils sont déployés de façon telle que l'utilisation de HSTS affaiblirait en fait la sécurité globale (par exemple, il peut être problématique d'utiliser HSTS avec des certificats auto signés, comme décrit au paragraphe 11.3 de la [RFC6797]).

Raison : Combiner des communications non protégées et des communications protégées par TLS ouvre la voie au déshabillage SSL et à des attaques similaires, car l'intégrité de la partie initiale de la communication n'est pas protégée et donc peut être manipulée par un attaquant dont le but est de garder la communication en clair.

3.3. Compression

Afin d'aider à empêcher les attaques en rapport avec la compression (résumées au paragraphe 2.6 de la [RFC7457]), les mises en œuvre et déploiements DEVRAIENT désactiver la compression au niveau TLS (paragraphe 6.2.2 de la [RFC5246]), sauf si le protocole d'application en question a été prouvé ne pas être sujet à de telles attaques.

Raison : la compression TLS a été l'objet d'attaques contre la sécurité, comme l'attaque CRIME.

Les mises en œuvre devraient noter que la compression au niveau des protocoles supérieurs peut permettre à un attaquant actif d'extraire des informations en clair de la connexion. L'attaque BREACH est un de ces cas. Ces problèmes ne peuvent être atténués qu'en dehors de TLS et sortent du domaine d'application du présent document. Voir les détails au paragraphe 2.6 de la [RFC7457].

3.4 Reprise de session TLS

Si la reprise de session TLS est utilisée, il faut veiller à la faire de façon sûre. En particulier, lorsque on utilise des tickets de session [RFC5077], les informations de reprise DOIVENT être authentifiées et chiffrées pour empêcher leur modification ou leur observation par un attaquant. D'autres recommandations s'appliquent aux tickets de session :

- Une suite de chiffrement forte DOIT être utilisée lors du chiffrement du ticket (au moins aussi fort que la suite de chiffrement TLS principale).

- Les clés de ticket DOIVENT être changées régulièrement, par exemple, une fois par semaine, de façon à ne pas anéantir les avantages du secret vers l'avant (voir au paragraphe 6.3 les détails sur le secret vers l'avant).

- Pour des raisons similaires, la validité de ticket de session DEVRAIT être limitée à un durée raisonnable (par exemple, la moitié de la validité d'une clé de ticket).

Raison : la reprise de session est une autre forme de prise de contact TLS, et doit donc être aussi sûre que la prise de contact initiale. Le présent document (Section 4) recommande l'utilisation de suites de chiffrement qui assurent le secret vers

l'avant, c'est-à-dire qui empêchent un attaquant qui obtient un accès momentané au point d'extrémité TLS (client ou serveur) et à ses secrets de lire les communications passées ou futures. Les tickets doivent être gérés de telle sorte qu'ils ne contreviennent pas à cette propriété de sécurité.

3.5 Renégociation TLS

Lorsque on met en œuvre la renégociation TLS, les clients et les serveurs DOIVENT mettre en œuvre l'extension renegotiation_info, comme défini dans la [RFC5746].

L'option la plus sûre pour contrer l'attaque de la triple prise de contact est de refuser tout changement de certificat durant la renégociation. De plus, les clients TLS DEVRAIENT appliquer la même politique de validation pour tous les certificats reçus sur une connexion. Le document [triple-handshake] suggère plusieurs autres contre mesures possibles, comme de lier le secret maître à la prise de contact complète (voir la [RFC7627]) et de lier la prise de contact abrégée de reprise de session à la prise de contact originale complète. Bien que ces deux dernières techniques soient encore en cours de développement et donc non qualifiées comme pratiques courantes, il est conseillé à ceux qui mettent en œuvre et déploient TLS d'être à l'affût des développement à venir de contre mesures appropriées.

3.6 Indication de nom de serveur

Les mises en œuvre de TLS DOIVENT prendre en charge l'extension Indication de nom de serveur (SNI, *Server Name Indication*) définie à la Section 3 de la [RFC6066] pour les protocoles de niveau supérieur qui vont en bénéficier, incluant HTTPS. Cependant, l'utilisation réelle de SNI dans des circonstances particulières est une affaire de politique locale.

Raison : SNI prend en charge le déploiement de plusieurs serveurs virtuels protégés par TLS sur une seule adresse, et donc permet une sécurité de granularité fine pour ces serveurs virtuels, en permettant à chacun d'avoir son propre certificat.

4. Recommandations : suites de chiffrement

TLS et ses mises en œuvre donnent une souplesse considérable dans le choix des suites de chiffrement. Malheureusement, certaines suites de chiffrement disponibles ne sont pas sûres, certaines ne fournissent pas les services de sécurité ciblés, et certaines ne fournissent plus assez de sécurité. Configurer un serveur de façon incorrecte conduit à une sécurité réduite ou à pas de sécurité du tout. La présente section inclut des recommandations sur le choix et la négociation des suites de chiffrement.

4.1 Lignes directrices générales

Les algorithmes de chiffrement perdent de la force au fil du temps avec l'amélioration de la cryptanalyse : les algorithmes qui ont naguère été considérés comme forts deviennent faibles. Ces algorithmes doivent être éliminés un jour et remplacés par des suites de chiffrement plus sûres. Cela contribue à assurer que les propriétés de sécurité désirées tiennent toujours. SSL/TLS existe depuis presque 20 ans et beaucoup des suites de chiffrement qui ont été recommandées dans diverses versions de SSL/TLS sont maintenant considérées comme faibles ou tout au moins pas aussi fortes que désiré. Donc, cette section modernise les recommandations concernant le choix des suites de chiffrement.

- Les mises en œuvre NE DOIVENT PAS négocier des suites de chiffrement avec le chiffrement NULL.

Raison : Les suites de chiffrement NULL ne chiffrent pas le trafic et donc ne fournissent aucun service de confidentialité. Toute entité du réseau qui a accès à la connexion peut voir le texte en clair du contenu échangé entre le client et le serveur. (Néanmoins, le présent document ne déconseille pas que le logiciel mettre en œuvre des suites de chiffrement NULL, car elles peuvent être utiles pour les essais et le débogage.)

- Les mises en œuvre NE DOIVENT PAS négocier des suites de chiffrement RC4.

Raison : Le chiffrement de flux RC4 a diverses faiblesses cryptographiques, comme documenté dans la [RFC7465]. Noter que DTLS interdit déjà spécifiquement l'utilisation de RC4.

- Les mises en œuvre NE DOIVENT PAS négocier des suites de chiffrement offrant moins de 112 bits de sécurité, incluant ce qu'on appelle le chiffrement de "niveau export" (qui fournit 40 ou 56 bits de sécurité).

Raison : Sur la base de la [RFC3766], au moins 112 bits de sécurité sont nécessaires. La sécurité à 40 bits et 56 bits est considérée comme non sûre aujourd'hui. TLS 1.1 et 1.2 ne négocient jamais de chiffrement d'export à 40 bits ou 56 bits.

- Les mises en œuvre NE DEVRAIENT PAS négocier de suites de chiffrement qui utilisent des algorithmes offrant moins de 128 bits de sécurité.

Raison : Les suites de chiffrement qui offrent entre 112 bits et 128 bits de sécurité ne sont pas considérées comme faibles pour l'instant ; cependant, on s'attend à ce que leur durée de vie utile soit assez courte pour justifier la prise en charge de suites de chiffrement plus forte dès maintenant. Les chiffrements à 128 bits sont supposés rester sûrs pour au moins plusieurs années, et ceux à 256 bits jusqu'à la prochaine rupture technologique fondamentale. Noter que, à cause de ce qu'on appelle les attaques de "rencontre en chemin" [Multiple-Encryption], certains suites de chiffrement traditionnelles (par exemple, 3DES à 168 bits) ont une longueur de clé efficace qui est plus petite que leur longueur de clé nominale (112 bits dans le cas de 3DES). De telles suites de chiffrement devraient être évaluées conformément à leur longueur de clé efficace.

- Les mises en œuvre NE DEVRAIENT PAS négocier de suites de chiffrement sur la base du transport de clé RSA, autrement dit "RSA statique".

Raison : Ces suites de chiffrement, qui ont des valeurs allouées qui commencent par la chaîne "TLS_RSA_WITH_*", ont plusieurs inconvénients, en particulier le fait qu'elles ne prennent pas en charge le secret vers l'avant.

- Les mises en œuvre DOIVENT prendre en charge et préférer négocier des suites de chiffrement offrant le secret vers l'avant, comme celles des familles Diffie-Hellman éphémère et courbe elliptique Diffie-Hellman éphémère ("DHE" et "ECDHE").

Raison : Le secret vers l'avant (parfois appelé "secret parfait vers l'avant") empêche la récupération des informations qui ont été chiffrées avec de plus anciennes clés de session, limitant donc la durée pendant laquelle des attaques peuvent réussir. Voir une discussion détaillée au paragraphe 6.3.

4.2 Suites de chiffrement recommandées

Étant données les considérations précédentes, la mise en œuvre et le déploiement des suites de chiffrement ci après est RECOMMANDÉ :

- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

Ces suites de chiffrement ne sont prises en charge que dans TLS 1.2 parce que ce sont des algorithmes de chiffrement authentifiés (AEAD, *Authenticated Encryption with Associated Data*) [RFC5116].

Normalement, afin de préférer ces suites, l'ordre des suites doit être explicitement configuré dans le logiciel du serveur. (Voir d'utiles lignes directrices dans [BETTERCRYPTO], mais noter que ses recommandations diffèrent du présent document par certains détails.) Il serait idéal que les mises en œuvre de logiciel de serveur préfèrent ces suites par défaut.

Certains appareils ont une prise en charge incorporée de AES-CCM mais pas de AES-GCM, de sorte qu'ils sont incapables de suivre les recommandations précédentes concernant les suites de chiffrement. Il y a même des appareils qui ne prennent pas en charge du tout la cryptographie à clé publique, mais ils sont entièrement hors de notre domaine d'application.

4.2.1 Détails de mise en œuvre

Les clients DEVRAIENT inclure TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 comme première proposition à tout serveur, sauf si ils savent par avance que le serveur ne peut pas répondre à un message client_hello TLS 1.2.

Les serveurs DOIVENT préférer cette suite de chiffrement à des suites de chiffrement plus faibles chaque fois qu'elle est proposée, même si ce n'est pas la première proposée.

Les clients ont bien sûr toute liberté d'offrir de plus fortes suites de chiffrement, par exemple, en utilisant AES-256 ; lorsque ils le font, le serveur DEVRAIT préférer la plus forte suite de chiffrement sauf si il y a des raisons impérieuses de faire autrement (par exemple, des performances sérieusement dégradées).

Le présent document ne change pas les suites de chiffrement TLS de mise en œuvre obligatoire prescrites par TLS. Pour maximiser l'interopérabilité, la RFC 5246 rend obligatoire la mise en œuvre de la suite de chiffrement TLS_RSA_WITH_AES_128_CBC_SHA, qui est significativement plus faible que les suites de chiffrement recommandées ici. (Le mode GCM ne souffre pas de la même faiblesse, causée par l'ordre de MAC puis chiffrement dans TLS [Krawczyk2001], car il utilise un mode de fonctionnement AEAD.) Les mises en œuvre devraient considérer le gain d'interopérabilité par rapport à la perte de sécurité lors du déploiement de la suite de chiffrement TLS_RSA_WITH_AES_128_CBC_SHA. D'autres protocoles d'application spécifient d'autres suites de chiffrement comme étant de mise en œuvre obligatoire (MTI, *Mandatory To Implement*).

Noter que certains profils de TLS 1.2 utilisent des suites de chiffrement différentes. Par exemple, la [RFC6460] définit un

profil qui utilise les suites de chiffrement TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 et TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384.

La [RFC4492] permet aux clients et serveurs de négocier les paramètres (courbes) ECDH. Clients et serveurs DEVRAIENT tous deux inclure l'extension "Courbes elliptiques acceptées" [RFC4492]. Pour l'interopérabilité, les clients et serveurs DEVRAIENT prendre en charge la courbe NIST P-256 (secp256r1) [RFC4492]. De plus, les clients DEVRAIENT envoyer une extension ec_point_formats avec un seul élément, "uncompressed".

4.3 Longueur de clé publique

Lorsque on utilise les suites de chiffrement recommandées dans le présent document, deux clés publiques sont normalement utilisées dans la prise de contact TLS : une pour l'accord de clé Diffie-Hellman et une pour l'authentification du serveur. Lorsque un certificat de client est utilisé, on ajoute une troisième clé publique.

Avec un échange de clés fondé sur les groupes d'exponentiation modulo un nombre premier (MODP, *Modular Prime exponential*) Diffie-Hellman (suites de chiffrement "DHE") des longueurs de clé DH d'au moins 2048 bits sont RECOMMANDÉES.

Raison : Pour diverses raisons, en pratique, les clés DH sont normalement générées dans des longueurs qui sont des puissances de deux (par exemple, 2^10 = 1024 bits, 2^11 = 2048 bits, 2^12 = 4096 bits). Parce que une clé DH de 1228 bits serait en gros équivalente à seulement une clé symétrique de 80 bits [RFC3766], il vaut mieux utiliser des clés plus longues que pour la famille "DHE" de suites de chiffrement. Une clé DH de 1926 bits serait en gros équivalente à une clé symétrique de 100 bits [RFC3766] et une clé DH de 2048 bits pourrait être suffisante pour au moins les dix prochaines années [NIST.SP.800-56A]. Voir au paragraphe 4.4 des informations supplémentaires sur l'utilisation de Diffie-Hellman MODP dans TLS.

Comme noté dans la [RFC3766], la correction tenant compte de l'émergence d'une machine TWIRL impliquerait que des clés DH de 1024 bits donnent environ 65 bits de force équivalente et qu'une clé DH de 2048 bits donnerait environ 92 bits de force équivalente.

À l'égard des clés ECDH, le registre IANA "EC Named Curve" (au sein du registre "Paramètre TLS" [IANA-TLS]) contient des courbes elliptiques de 160 bits qui sont considérées comme en gros équivalentes à seulement une clé symétrique de 80 bits [ECRYPT-II]. Des courbes de moins de 192 bits NE DEVRAIENT PAS être utilisées.

Lorsque on utilise RSA, les serveurs DEVRAIENT s'authentifier en utilisant des certificats avec au moins un module de 2048 bits pour la clé publique. De plus, l'utilisation de l'algorithme de hachage SHA-256 est RECOMMANDÉ (voir plus de détails dans [CAB-Baseline]). Les clients DEVRAIENT indiquer aux serveurs qu'ils demandent SHA-256, en utilisant l'extension "Signature Algorithms" définie dans TLS 1.2.

4.4 Suites de chiffrement modulaires exponentielles ou DH à courbe elliptique

Toutes les mises en œuvre de TLS ne prennent pas en charge à la fois les groupes Diffie-Hellman d'exponentiation modulo un nombre premier (MODP) et les groupes à courbe elliptique (EC) comme exigé au paragraphe 4.2. Certaines mises en œuvre sont sévèrement limitées pour la longueur des valeurs DH. Lorsque de telles mises en œuvre doivent être traitées, on RECOMMANDE ce qui suit (dans l'ordre de priorité) :

1. Courbe elliptique DHE avec les paramètres négociés appropriés (par exemple, la courbe à utiliser) et un algorithme de code d'authentification de message (MAC, *Message Authentication Code*) plus fort que HMAC-SHA1 [RFC5289].

2. TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 [RFC5288], avec des paramètres Diffie-Hellman de 2048 bits.

3. TLS_DHE_RSA_WITH_AES_128_GCM_SHA256, avec des paramètres de 1024 bits.

Raison : Bien que la cryptographie à courbe elliptique soit largement répandue, il y a certaines communautés où son adoption a été limitée pour plusieurs raisons, incluant sa complexité comparée à l'arithmétique modulaire et des problèmes de droits de propriété intellectuelle qui ont traîné pendant longtemps (mais ont été pour la plupart résolus maintenant [RFC6090]). Noter que les suites de chiffrement ECDHE existent pour les deux certificats RSA et ECDSA, de sorte que passer aux suites de chiffrement ECDHE n'exige pas d'abandonner les certificats fondés sur RSA. D'un autre côté, il y a deux problèmes à ce sujet qui entravent l'utilisation effective des suites de chiffrement Diffie-Hellman MODP dans TLS :

- Il n'y a pas de mécanisme de protocole standard largement mis en œuvre pour négocier les groupes DH ou les longueurs de paramètres acceptés par client et serveur.

- De nombreux serveurs choisissent des paramètres DH de 1024 bits ou moins.

- Il y a des mises en œuvre de client largement répandues qui rejettent les paramètres DH reçus si ils font plus de 1024 bits. De plus, plusieurs mises en œuvre n'effectuent pas la validation appropriée des paramètres de groupe et sont vulnérables aux attaques référencées au paragraphe 2.9 de la [RFC7457].

Noter qu'avec les suites de chiffrement DHE et ECDHE, la clé maîtresse TLS ne dépend que des paramètres Diffie-Hellman et non de la force du certificat RSA ; de plus, les paramètres DH MODP de 1024 bits sont généralement considérés comme insuffisants dès maintenant.

Avec DH MODP éphémère, les déployeurs devraient évaluer attentivement les considérations d'interopérabilité par rapport à celles de sécurité lorsque ils configurent leurs points d'extrémité TLS.

4.5 HMAC tronqué

Les mises en œuvre NE DOIVENT PAS utiliser l'extension HMAC tronquée , définie à la Section 7 de la [RFC6066].

Raison : l'extension ne s'applique pas aux suites de chiffrement AEAD recommandées ci-dessus. Cependant, elle s'applique à la plupart des autres suites de chiffrement TLS. Son utilisation a été montrée non sûre par [PatersonRS11].

5. Déclaration d'applicabilité

Les recommandations du présent document s'appliquent principalement à la mise en œuvre et au déploiement de protocoles d'application qui sont le plus couramment utilisés aujourd'hui avec TLS et DTLS sur l'Internet. Des exemples incluent, mais sans s'y limiter :
- des logiciels et services de la Toile qui souhaitent protéger le trafic HTTP avec TLS,
- des logiciels et services de messagerie qui souhaitent protéger le trafic IMAP, POP3, ou SMTP avec TLS,
- des logiciels et services de messagerie instantanée qui souhaitent protéger le trafic du protocole de messagerie et de présence extensibles (XMPP, *Extensible Messaging et Presence Protocol*) ou du relais d'échanges de l'Internet (IRC, *Internet Relay Chat*) avec TLS.
- des logiciels et services de supports en temps réel qui souhaitent protéger le trafic du protocole de transport sûr en temps réel (SRTP, *Secure Realtime Transport Protocol*) avec DTLS.

Le présent document ne modifie pas les recommandations de mise en œuvre et déploiement (par exemple, les suites de chiffrement de mise en œuvre obligatoire) prescrites dans les protocoles d'application existants qui emploient TLS ou DTLS. Si la communauté qui utilise un tel protocole d'application souhaite moderniser l'usage de TLS ou DTLS pour être cohérent avec les bonnes pratiques recommandées ici, elle doit explicitement mettre à jour la définition existante du protocole d'application (un exemple est celui de la [RFC7590], qui met à jour la [RFC6120]).

Les concepteurs de nouveaux protocoles d'application développés par le processus de normalisation de l'Internet [RFC2026] sont supposés au minimum se conformer aux bonnes pratiques recommandées ici, sauf à produire une documentation de raisons impérieuses qui empêchent une telle conformité (par exemple, le large déploiement sur des appareils contraints qui ne prennent pas en charge les algorithmes nécessaires).

5.1 Services de sécurité

Le présent document fait des recommandations pour un public qui souhaite sécuriser ses communications avec TLS pour réaliser ce qui suit :
- Confidentialité : toutes les communications de couche application sont chiffrées dans le but qu'aucun tiers ne soit capable de le déchiffrer à part son destinataire.
- Intégrité des données : tout changement fait à la communication dans le transit est détectable par le receveur.
- Authentification : un point d'extrémité de communication TLS est authentifié comme entité de destination avec laquelle communiquer.

À l'égard de l'authentification, TLS permet l'authentification d'un ou des deux points d'extrémité de la communication. Dans le contexte d'une sécurité opportuniste [RFC7435], TLS est parfois utilisé sans authentification. Comme discuté au paragraphe 5.2, les considérations de sécurité opportuniste ne relèvent pas du présent document.

Si les déployeurs dévient des recommandations du présent document, ils doivent savoir qu'ils risquent de perdre l'accès à l'un

des services de sécurité ci-dessus.

Le présent document ne s'applique qu'aux environnements où la confidentialité est requise. Il recommande des algorithmes et des options de configuration qui appliquent le secret des données dans le transit.

Le présent document suppose aussi que la protection de l'intégrité des données est toujours un des buts d'un déploiement. Dans les cas où l'intégrité n'est pas exigée, il n'y a pas de raison d'employer TLS en premier. Il y a des attaques contre la protection de la seule confidentialité qui utilisent le manque de protection de l'intégrité pour casser aussi la confidentialité (voir, par exemple, [DegabrieleP07] dans le contexte de IPsec).

Le présent document s'adresse aux protocoles d'application qui sont le plus couramment utilisés sur l'Internet avec TLS et DTLS. Normalement, toutes les communication entre les clients TLS et les serveurs TLS exigent les trois services de sécurité ci-dessus. Ceci est particulièrement vrai lorsque les clients TLS sont des agents d'utilisateur comme des navigateurs de la Toile ou un logiciel de messagerie électronique.

Le présent document ne s'adresse pas aux plus rares scénarios de déploiement où une des trois propriétés ci-dessus n'est pas désirée, comme dans le cas d'utilisation décrit au paragraphe 5.2. Comme autre scénario où la confidentialité n'est pas nécessaire, considérons un réseau surveillé où les autorités en charge des domaines de trafic respectifs exigent un plein accès au trafic non chiffré (au texte source) et où les utilisateurs collaborent et envoient leur trafic en clair.

5.2 Sécurité opportuniste

Il y a plusieurs scénarios importants dans lesquels l'utilisation de TLS est facultative, c'est-à-dire, que le client décide de façon dynamique ("opportuniste") si il utilise TLS avec un serveur particulier ou si il se connecte en clair. Cette pratique, souvent appelée "sécurité opportuniste", est décrite dans la [RFC7435] et est souvent motivée par un désir de rétro compatibilité avec des déploiements traditionnels.

Dans ces scénarios, certaines des recommandations du présent document pourraient être trop strictes, car y adhérer pourrait causer le replis sur le texte en clair, résultat pire que d'utiliser TLS avec une version de protocole ou suite de chiffrement périmée.

Le présent document spécifie les bonnes pratiques pour TLS en général. Un document séparé contenant des recommandations pour l'utilisation de TLS avec la sécurité opportuniste sera réalisé à l'avenir.

6. Considérations sur la sécurité

Ce document entier discute des pratiques de sécurité qui affectent directement les applications qui utilisent le protocole TLS. Cette section contient de plus larges considérations de sécurité relatives aux technologies utilisées en conjonction avec ou par TLS.

6.1 Validation de nom d'hôte

Les auteurs d'applications devraient noter que certaines mises en œuvre de TLS ne valident pas les noms d'hôtes. Si la mise en œuvre de TLS qu'ils utilisent ne valide pas les noms d'hôte, ils pourraient avoir besoin d'écrire leur propre code de validation ou envisager d'utiliser une mise en œuvre différente de TLS.

On note que les exigences concernant la validation de nom d'hôte (et, en général, le lien entre la couche TLS et le protocole qui fonctionne par dessus elle) varient selon les protocoles. Pour HTTPS, ces exigences sont définies à la Section 3 de la [RFC2818].

Le lecteur se reportera à la [RFC6125] pour les détails de la validation générique de nom d'hôte dans le contexte de TLS. De plus, cette RFC contient une longue liste d'exemples de protocoles, dont certains mettent en œuvre une politique très différente de HTTPS.

Si le nom d'hôte est découvert indirectement et d'une manière non sûre (par exemple, par une interrogation non sécurisée au DNS pour un enregistrement MX ou SRV) il NE DEVRAIT PAS être utilisé comme identifiant de référence [RFC6125] même quand il correspond au certificat présenté. Cette disposition ne s'applique pas si le nom d'hôte est découvert de façon sûre (pour en savoir plus, voir les [RFC7672] et [RFC7673]).

La validation de nom d'hôte ne s'applique normalement qu'au certificat de l'entité d'extrémité "feuille". Naturellement, afin de

s'assurer d'une authentification appropriée dans le contexte de PKI, les clients d'application doivent vérifier le chemin de certification entier conformément à la [RFC5280] (voir aussi la [RFC6125]).

6.2 AES-GCM

Le paragraphe 4.2 recommande l'utilisation de l'algorithme de chiffrement authentifié AES-GCM. Se reporter à la Section 11 de la [RFC5246] sur les considérations générales de sécurité pour l'utilisation de TLS 1.2, et à la Section 6 de la [RFC5288] pour les considérations de sécurité qui s'appliquent spécifiquement à AES-GCM utilisé avec TLS.

6.3 Secret vers l'avant

Le secret vers l'avant (aussi appelé "secret parfait vers l'avant" ou "PFS" et défini dans la [RFC4949]) est une défense contre un attaquant qui enregistre des conversations chiffrées où les clé de session ne sont chiffrées qu'avec les clés à long terme des parties communicantes. Si l'attaquant est capable d'obtenir ces clé à long terme à un certain moment, les clé de session et donc la conversation toute entière pourrait être déchiffrée. Dans le contexte de TLS et DTLS, une telle compromission des clés à long terme n'est pas entièrement improbable. Cela peut arriver, par exemple, parce que :

- un client ou serveur est attaqué par un autre vecteur d'attaque, et la clé privée est trouvée ;
- une clé à long terme est restituée d'un appareil qui a été vendu ou désactivé sans effacement préalable des données ;
- une clé à long terme a été utilisée sur un appareil comme clé par défaut [Heninger2012] ;
- une clé générée par un tiers de confiance comme une CA, et restituée ultérieurement de celle-ci par extorsion ou compromission [Soghoian2011] ;
- un raccourci cryptographique, ou l'utilisation de clés asymétriques d'une longueur insuffisante [Kleinjung2010] ;
- des attaques d'ingénierie sociale contre les administrateurs du système ;
- une collection de clés privées provenant de sauvegardes inadéquatement protégées.

Le secret vers l'avant assure que dans de tels cas il n'est pas faisable qu'un attaquant détermine les clés de session même si l'attaquant a obtenu les clés à long terme un certain temps après la conversation. Il protège aussi contre un attaquant qui serait en possession des clés à long terme mais resterait passif durant la conversation.

Le secret vers l'avant est généralement réalisé en utilisant le schéma Diffie-Hellman pour déduire les clés de session. Le schéma Diffie-Hellman a les deux parties qui conservent leurs secrets privés et envoient des paramètres sur le réseau comme des puissances modulaires sur certains groupes cycliques. Les propriétés de ce qu'on appelle le problème du logarithme discret (DLP, *Discrete Logarithm Problem*) permettent aux parties de déduire les clés de session sans qu'un espion soit capable de le faire. Il n'y a actuellement aucune attaque connue contre DLP si des paramètres suffisamment grands sont choisis. Une variante du schéma Diffie-Hellman utilise des courbes elliptiques au lieu de l'arithmétique modulaire proposée à l'origine.

Malheureusement, de nombreuses suites de chiffrement ont été définies sans la caractéristique de secret vers l'avant, par exemple, TLS_RSA_WITH_AES_256_CBC_SHA256. Le présent document plaide donc en faveur d'une utilisation stricte de chiffrement avec secret vers l'avant.

6.4 Réutilisation d'exposant Diffie-Hellman

Pour des raisons de performances, de nombreuses mises en œuvre de TLS réutilisent les exposants Diffie-Hellman et les courbes elliptiques Diffie-Hellman sur plusieurs connexions. Une telle réutilisation peut résulter en des problèmes de sécurité majeurs :

- Si les exposants sont réutilisés pendant trop longtemps (par exemple, même plus que quelques heures) un attaquant qui obtient l'accès à l'hôte peut décrypter les connexions précédentes. En d'autres termes, la réutilisation d'exposant détruit les effets du secret vers l'avant.
- Les mises en œuvre de TLS qui réutilisent les exposants ne devraient pas essayer la clé publique DH qu'ils reçoivent pour l'adhésion de groupe, afin d'éviter certaines attaques connues. Ces essais ne sont pas normalisés dans TLS au moment de la rédaction du présent document. Voir dans la [RFC6989] les essais de réception exigés par les mises en œuvre de IKEv2 qui réutilisent les exposants DH.

6.5 Révocation de certificat

Les considérations et recommandations suivantes représentent l'état de l'art actuel concernant la révocation de certificat, bien qu'aucune solution complète et efficace n'existe pour le problème de la vérification de l'état de révocation des certificats courants de clé publique [RFC5280] :

- Bien que les listes de révocation de certificat (CRL, *Certificate Revocation List*) soient le mécanisme le plus largement pris en charge pour distribuer les informations de révocation, il y a des défis d'adaptabilité connus qui limitent leur utilité (en

dépit d'astuces comme les partitions de CRL et les CRL delta).

- Des mécanismes propriétaires qui incorporent les listes de révocation dans la base de données de configuration des navigateurs de la Toile ne peuvent pas couvrir plus qu'un petit nombre des serveurs de la Toile les plus utilisés.
- Le protocole d'état de certification en ligne (OCSP, *On-Line Certification Status Protocol*) [RFC6960] présente des problèmes d'adaptabilité et de confidentialité. De plus, les clients font normalement un "échec en douceur" *(soft-fail)* ce qui signifie qu'ils n'interrompent pas la connexion TLS si le serveur OCSP ne répond pas. (Cependant, ceci peut être une astuce pour éviter les attaques de déni de service si un répondant OCSP est pris hors ligne.)
- L'extension TLS Demande d'état de certificat (Section 8 de la [RFC6066]), couramment appelée "agrafage OCSP", résout les problèmes opérationnels avec OCSP. Cependant, elle est encore inefficace en présence d'une attaque MITM parce que l'attaquant peut simplement ignorer la demande du client d'une réponse d'agrafage OCSP.
- L'agrafage OCSP comme défini dans la [RFC6066] ne s'étend pas aux certificats intermédiaires utilisés dans une chaîne de certificats. Bien que l'extension État de multiples certificats [RFC6961] traite ce problème, c'est un ajout récent sans beaucoup de déploiements.
- Les CRL et OCSP dépendent tous deux d'une connexité relativement fiable à l'Internet, qui peut n'être pas disponible à certaines sortes de nœuds (comme des appareils nouvellement provisionnés qui ont besoin d'établir une connexion sûre afin de s'amorcer pour la première fois).

À l'égard des certificats de clé publique courants, les serveurs DEVRAIENT prendre en charge ce qui suit comme bonnes pratiques, étant donné l'état actuel de l'art et comme fondement pour une future solution possible :

1. OCSP [RFC6960]

2. L'extension status_request définie dans la [RFC6066] et l'extension status_request_v2 définie dans la [RFC6961] (Cela peut permettre l'interopérabilité avec la plus large gamme de clients.)

3. L'extension d'agrafage OCSP définie dans la [RFC6961]

Les considérations de cette section ne s'appliquent pas aux scénarios où l'enregistrement de ressource DANE-TLSA [RFC6698] est utilisé pour signaler à un client quel certificat un serveur considère comme valide et bon à utiliser pour les connexions TLS.

7. Références

7.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC2818] E. Rescorla, "HTTP sur TLS", mai 2000. *(Information)*

[RFC3766] H. Orman, P. Hoffman, "Détermination de la force des clés publiques utilisées pour l'échange de clés symétriques", avril 2004. (BCP0086)

[RFC4492] S. Blake-Wilson et autres, "Suites de chiffrement de cryptographie à courbe elliptique (ECC) pour la sécurité de la couche Transport (TLS)", mai 2006. (*MàJ par* RFC5246, 7919), *(Information ; rendue obsolète par* RFC8422)

[RFC4949] R. Shirey, "Version 2 du glossaire de la sécurité sur Internet", août 2007. (*Remplace* RFC2828) (FYI0036) *(Info.)*

[RFC5246] T. Dierks, E. Rescorla, "Version 1.2 du protocole de sécurité de la couche Transport (TLS)", août 2008. *(P.S.* ; *remplace* RFC3268, 4346, 4366 ; *MàJ* RFC4492 ; *rendue obsolète par la* RFC8446)

[RFC5288] J. Salowey et autres, "Suites de chiffrement pour TLS en AES avec mode de compteur de Galois (GCM)", août 2008. (*P.S.)*

[RFC5289] E. Rescorla, "Suites de chiffrement à courbe elliptique pour TLS avec SHA-256/384 et AES avec mode de compteur de Galois (GCM)", août 2008. *(Information)*

[RFC5746] E. Rescorla, M. Ray, S. Dispensa, N. Oskov, "Sécurité de la couche Transport (TLS) : Extension Indication de renégociation", février 2010. (*MàJ* RFC5246, RFC4366, RFC4347, RFC4346, RFC2246). *(P. S.)*

[RFC6066] D. Eastlake 3rd, " Extensions à la sécurité de la couche Transport (TLS) : Définitions d'extension", janvier 2011.

(*Remplace la* RFC4366 *; MàJ par* RFC8446, RFC8449)

[RFC6125] P. Saint-André, J. Hodges, "Représentation et vérification d'identité de service d'application fondé sur le domaine au sein de l'infrastructure Internet de clé publique utilisant les certificats X.509 (PKIX) dans le contexte de la sécurité de la couche Transport (TLS)", mars 2011. *(P.S.)*

[RFC6176] S. Turner, T. Polk. "Interdiction de l'utilisation de la couche de prises sécurisées (SSL) version 2.0", mars 2011. (*MàJ* RFC2246, RFC4346, RFC5246) *(P. S.)*

[RFC6347] E. Rescorla, N. Modadugu, "Sécurité de la couche transport de datagrammes, version 1.2", janvier 2012. *(Remplace la* RFC4347) *(P.S. ; MàJ par* RFC7905)

[RFC7465] A. Popov, "Interdiction des suites de chiffrement RC4", février 2015. *(P.S. ; MàJ RFC* 5246*,* 4346*,* 2246*)*

7.2 Références pour information

[BETTERCRYPTO] bettercrypto.org, "Applied Crypto Hardening", avril 2015, <https://bettercrypto.org/static/applied-crypto-hardening.pdf >.

[CAB-Baseline] CA/Browser Forum, "Baseline Requirements for the Issuance et Management of Publicly-Trusted Certificates Version 1.1.6", 2013, <https://www.cabforum.org/documents.html >.

[DegabrieleP07] Degabriele, J. and K. Paterson, "Attacking the IPsec Standards in Encryption-only Configurations", IEEE Symposium on Security and Privacy (SP '07), 2007, < http://dx.doi.org/10.1109/SP.2007.8 >.

[ECRYPT-II] Smart, N., "ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012)", 2012, <http://www.ecrypt.eu.org/ecrypt2/ >.

[Heninger2012] Heninger, N., Durumeric, Z., Wustrow, E., and J. Halderman, "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices", Usenix Security Symposium 2012, 2012.

[IANA-TLS] IANA, "Transport Layer Security (TLS) Parameters", < http://www.iana.org/assignments/tls-parameters >.

[Kleinjung2010] Kleinjung, T., "Factorization of a 768-Bit RSA modulus", CRYPTO 10, 2010, <https://eprint.iacr.org/2010/006.pdf >.

[Krawczyk2001] Krawczyk, H., "The Order of Encryption and Authentication for Protecting Communications (Or: How Secure is SSL?)", CRYPTO 01, 2001, <https://www.iacr.org/archive/crypto2001/21390309.pdf >.

[Multiple-Encryption] Merkle, R. and M. Hellman, "On the security of multiple encryption", Communications of the ACM, Vol. 24, 1981, < http://dl.acm.org/citation.cfm?id=358718 >.

[NIST.SP.800-56A] Barker, E., Chen, L., Roginsky, A., and M. Smid, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography", NIST Special Publication 800-56A, 2013, <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf >.

[POODLE] US-CERT, "SSL 3.0 Protocol Vulnerability and POODLE Attack", Alert TA14-290A, octobre 2014, <https://www.us-cert.gov/ncas/alerts/TA14-290A >.

[PatersonRS11] Paterson, K., Ristenpart, T., and T. Shrimpton, "Tag size does matter: attacks and proofs for the TLS record protocol", 2011, <http://dx.doi.org/10.1007/978-3-642-25385-0_20 >.

[RFC2026] S. Bradner, "Le processus de normalisation de l'Internet -- Révision 3", (BCP0009) octobre 1996. *(Remplace* RFC1602, RFC1871) (*MàJ par* RFC3667, 3668, 3932, 3979, 3978, 5378, 6410, 8179)

[RFC2246] T. Dierks et C. Allen, "Protocole TLS version 1.0", janvier 1999. *(P.S. ; MàJ par* RFC7919)

[RFC3602] S. Frankel, R. Glenn, S. Kelly, "Algorithme de chiffrement AES-CBC et utilisation avec IPsec", septembre 2003. *(P.S.)*

[RFC4346] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", avril 2006. *(Remplace*

RFC2246 ; *Remplacée par* RFC5246 ; *MàJ par* RFC4366, 4680, 4681, 5746, 6176, 7465, 7507, 7919)

[RFC4347] E. Rescorla, N. Modadugu, "Sécurité de la couche de transport de datagrammes ", avril 2006. *(P.S.)*

[RFC5077] J. Salowey et autres, "Reprise de session de sécurité de la couche Transport (TLS) sans état côté serveur", janvier 2008. (*Remplace* RFC4507) *(P.S. ; rendue obsolète par la* RFC8446)

[RFC5116] D. McGrew, "Interface et algorithmes pour le chiffrement authentifié", janvier 2008. *(P.S.)*

[RFC5280] D. Cooper et autres, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", mai 2008. (*Remplace les* RFC3280, RFC4325, RFC4630) *(P.S. ; MàJ par* RFC8398, 8399)

[RFC6090] D. McGrew, K. Igoe, M. Salter, "Algorithmes fondamentaux de cryptographie par courbe élliptique", février 2011. *(Info.)*

[RFC6101] A. Freier, P. Karlton, P. Kocher "Protocole de couche de connexion sécurisée (SSL) version 3.0", août 2011. *(Historique)*

[RFC6120] P. Saint-André "Protocole de messagerie et de présence extensibles (XMPP) : cœur", mars 2011. *(Remplace* RFC3920) *(P.S. ; MàJ par* RFC7590)

[RFC6460] M. Salter, R. Housley, "Profil de suite B pour la sécurité de la couche Transport (TLS)", janvier 2012. (*Remplace la* RFC5430) *(Historique)*

[RFC6698] P. Hoffman, J. Schlyter, "Authentification fondée sur les entités désignées du DNS (DANE) du protocole de sécurité de la couche Transport (TLS) : TLSA", août 2012. *(P.S.) (MàJ par* RFC7219, RFC7671)

[RFC6797] J. Hodges, C. Jackson, A. Barth, "Sécurité stricte du transport HTTP (HSTS)", novembre 2012. *(P.S.)*

[RFC6960] S. Santesson et autres, "Protocole d'état de certificat en ligne (OCSP) pour l'infrastructure de clé publique Internet X.509", juin 2013. (*Remplace* RFC2560, 6277) (*MàJ* RFC5912) *(P.S.)*

[RFC6961] Y. Pettersen, "Extension Demande d'état de certificat multiple pour la sécurité de la couche Transport (TLS)", juin 2013. *(P.S. ; rendue obsolète par la* RFC8446)

[RFC6989] Y. Sheffer, S. Fluhrer, "Essais Diffie-Hellman supplémentaires pour le protocole d'échange de clés Internet version 2 (IKEv2)", juillet 2013. (*MàJ* RFC5996) *(P.S.)*

[RFC7435] V. Dukhovni, "Sécurité opportuniste : une protection la plupart du temps", décembre 2014. *(Information)*

[RFC7457] Y. Sheffer, et autres, "Résumé des attaques connues contre la sécurité de la couche transport (TLS) et DTLS", février 2015. *(Information)*

[RFC7507] B. Moeller, A. Langley, "Valeur de suite de chiffrement de signalisation de repli pour TLS pour empêcher les attaques de dégradation de protocole", avril 2015. *(P.S. ; MàJ* 2246, 4346, 5246, 6347*)*

[RFC7568] R. Barnes, et autres, "La couche de connexions sécurisées version 3.0 (SSLv3) est déconseillée", juin 2015. *(P.S. ; MàJ* RFC5246*)*

[RFC7590] P. Saint-André, "Utilisation de TLS dans XMPP", juin 2015. *(P.S. ; MàJ* RFC6120*)*

[RFC7627] K. Bhargavan, et autres, "Extension au secret maître étendu et au hachage de session de la sécurité de couche transport (TLS)", septembre 2015.*(P.S. ; MàJ* RFC5246*)*

[RFC7672] V. Dukhovni, W. Hardaker, "Sécurité SMTP via l'utiliation par la sécurité de couche transport de l'authentification fondée sur le DNS des entités désignées (DANE)", octobre 2015. *(P.S.)*

[RFC7673] T. Finch, M. Miller, P. Saint-André, "Utilisation des enregistrements TLSA d'authentification fondée sur le DNS des entités désignées (DANE) avec les enregistrements SRV", octobre 2015. *(P.S.)*

[Smith2013] Smith, B., "Proposal to Change the Default TLS Ciphersuites Offered by Browsers.", 2013, <https://briansmith.org/browser-ciphersuites-01.html >.

[Soghoian2011] Soghoian, C. and S. Stamm, "Certified lies: Detecting and defeating government interception attacks against SSL", Proc. 15th Int. Conf. Financial Cryptography and Data Security, 2011.

[triple-handshake] Delignat-Lavaud, A., Bhargavan, K., and A. Pironti, "Triple Handshakes Considered Harmful: Breaking and Fixing Authentication over TLS", 2014, < https://secure-resumption.com/ >.

Remerciements

Merci à RJ Atkinson, Uri Blumenthal, Viktor Dukhovni, Stephen Farrell, Daniel Kahn Gillmor, Paul Hoffman, Simon Josefsson, Watson Ladd, Orit Levin, Ilari Liusvaara, Johannes Merkle, Bodo Moeller, Yoav Nir, Massimiliano Pala, Kenny Paterson, Patrick Pelletier, Tom Ritter, Joe St. Sauver, Joe Salowey, Rich Salz, Brian Smith, Sean Turner, et Aaron Zauner de leurs retours et des améliorations suggérées. Merci aussi à Brian Smith, qui a fourni beaucoup d'idées pour le présent document dans sa "Proposal to Change the Default TLS Ciphersuites Offered by Browsers" [Smith2013]. Finalement, merci à tous ceux qui ont fourni des commentaires sur les listes de diffusion TLS, UTA, et autres mais ne sont pas mentionnés ici par leur nom.

Robert Sparks et Dave Waltermire ont fourni d'utiles relectures au nom respectivement de l'équipe de révision de la zone générale et de la Direction de la sécurité.

Durant la revue par l'IESG, Richard Barnes, Alissa Cooper, Spencer Dawkins, Stephen Farrell, Barry Leiba, Kathleen Moriarty, et Pete Resnick ont fourni des commentaires qui ont conduit à d'autres améliorations.

Ralph Holz remercie de son soutien la Technische Universitaet Muenchen. Les auteurs témoignent de leur reconnaissance de leur assistance à Leif Johansson et Orit Levin comme présidents de groupe de travail et à Pete Resnick comme Directeur de la zone qui a parrainé ces travaux.

Adresse des auteurs

Yaron Sheffer
Intuit
4 HaHarash St.
Hod HaSharon 4524075
Israel
mél : yaronf.ietf@gmail.com

Ralph Holz
NICTA
13 Garden St.
Eveleigh 2015 NSW
Australia
mél : ralph.ietf@gmail.com

Peter Saint-Andre
&yet
mél : peter@andyet.com
URI : https://andyet.com/