Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 7696
BCP 201
Catégorie : Bonnes pratiques actuelles
ISSN : 2070-1721

R. Housley, Vigil Security

novembre 2015
Traduction Claude Brière de L'Isle

Lignes directrices pour la gestion des algorithmes cryptographiques d'application obligatoire

Résumé
De nombreux protocoles de l'IETF utilisent des algorithmes de chiffrement pour assurer la confidentialité, l'intégrité, l'authentification, ou la signature numérique. Les homologues communicants doivent prendre en charge un ensemble commun d'algorithmes de chiffrement pour que ces mécanismes fonctionnent correctement. Le présent mémoire donne des lignes directrices pour s'assurer que les protocoles ont la capacité de migrer d'une suite d'algorithmes de mise en œuvre obligatoire à une autre au fil du temps.

Statut de ce mémoire
Le présent mémoire 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/rfc7696

Table des matières

1. Introduction....2
1.1 Terminologie....2
2. Lignes directrices pour la souplesse des algorithmes....2
2.1 Identifiants d'algorithmes....2
2.2. Algorithmes de mise en œuvre obligatoire....3
2.3 Transition à partir d'algorithmes faibles....4
2.4 Mécanismes de transition d'algorithme....5
2.5 Établissement des clés de chiffrement....5
2.6 Préserver l'interopérabilité....5
2.7 Compromis sur la force de la sécurité....6
2.8. Compromis sur la complexité du protocole....6
2.9 Sécurité opportuniste....6
3. Spécifications des algorithmes de chiffrement....6
3.1 Choisir des algorithmes de mise en œuvre obligatoire....7
3.2 Trop de choix peut être dommageable....7
3.3 Prendre une vraie suite de chiffrement peut être dommageable....8
3.4 Suites de chiffrement nationales....8
4. Considérations sur la sécurité....8
5. Considérations relatives à l'IANA....9
6. Références normatives....9
7. Références pour information....10
Remerciements....11
Adresse de l'auteur....11

1. Introduction

De nombreux protocoles de l'IETF utilisent des algorithmes de chiffrement pour assurer la protection de la confidentialité, de l'intégrité, l'authentification, ou une signature numérique. Pour l'interopérabilité, les homologues communicants doivent prendre en charge un ensemble commun d'algorithmes de chiffrement. Dans la plupart des cas, une combinaison d'algorithmes de chiffrement compatibles sera utilisée pour fournir les services de sécurité désirés. L'ensemble d'algorithmes de chiffrement utilisé à un moment donné est souvent appelé une suite d'algorithmes de chiffrement ou suite de chiffrement. Dans un protocole, des identifiants d'algorithme peuvent désigner un seul algorithme de chiffrement ou une suite complète d'algorithmes.

Les algorithmes de chiffrement vieillissent ; ils deviennent plus faibles au fil du temps. Avec le développement de nouvelles techniques de cryptanalyse et l'amélioration des capacités de calcul, le travail nécessaire pour casser un algorithme de chiffrement particulier se réduit, rendant une attaque sur l'algorithme plus faisable pour plus d'attaquants. Bien qu'on ne sache pas comment vont évoluer les attaques cryptoanalytiques, il est certain qu'elles vont s'améliorer. On ne sait pas dans quelle mesure elles vont s'améliorer, ni quand les avancées vont se faire. Les concepteurs de protocoles doivent supposer que les avancées dans la puissance de calcul ou les avancées des techniques de cryptoanalytise vont finalement rendre obsolète tout algorithme. Pour cette raison, les protocoles ont besoin de mécanismes pour migrer d'une suite d'algorithmes à une autre au fil du temps.

L'agilité d'algorithme est réalisée quand un protocole peut facilement migrer d'une suite d'algorithme à une autre qui avec le temps est devenue plus désirable. Pour celui qui met en œuvre le protocole, cela signifie que les mises en œuvre devraient être modulaires pour s'accommoder facilement de l'insertion de nouveaux algorithmes ou suites d'algorithmes. Idéalement, les mises en œuvre vont aussi fournir un moyen pour mesurer quand les mises en œuvre déployées ont glissé des anciens algorithmes à de meilleurs. Pour le concepteur de protocole, agilité d'algorithme signifie qu'un ou plusieurs identifiants d'algorithme ou de suite d'algorithmes doivent être pris en charge, que l'ensemble des algorithmes de mise en œuvre obligatoire va changer avec le temps, et qu'un registre IANA des identifiants d'algorithme va être nécessaire.

Les identifiants d'algorithme par eux-même ne sont pas suffisants pour assurer une migration facile. Une action est nécessaire de la part des gens qui maintiennent les mises en œuvre et font fonctionner les services pou développer, déployer, et ajuster les réglages de configuration pour permettre les nouveaux algorithmes plus désirables et déconseiller ou désactiver les plus vieux, moins désirables. Pour diverses raisons, surtout des soucis d'interopérabilité, l'expérience a montré qu'il s'est révélé difficile pour ceux qui mettent en œuvre et les administrateurs de retirer ou désactiver les algorithmes faibles. De plus, l'incapacité des systèmes traditionnels et des appareils à ressource restreintes à prendre en charge les nouveaux algorithmes ajoute encore des soucis. Par suite, les gens vivent avec des algorithmes plus faibles, parfois sérieusement fautifs, bien après que les experts aient recommandé la migration.

1.1 Terminologie

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].

2. Lignes directrices pour la souplesse des algorithmes

Ces lignes directrices sont à l'usage des groupes de travail de l'IETF et des auteurs de protocoles de l'IETF qui font usage d'algorithmes de chiffrement. Les tentatives passées d'agilité d'algorithme n'ont pas été complètement couronnées de succès, et cette section donne quelques détails sur ces expériences.

2.1 Identifiants d'algorithmes

Les protocoles de l'IETF qui font usage des algorithmes de chiffrement DOIVENT prendre en charge un ou plusieurs algorithmes ou suites d'algorithmes. Le protocole DOIT inclure un mécanisme pour identifier l'algorithme ou suite utilisé. Un identifiant d'algorithme peut être explicitement porté dans le protocole. Autrement, un mécanisme de gestion peut être utilisé pour identifier l'algorithme. Par exemple, une entrée dans un tableau de clés qui inclut une valeur de clé et un identifiant d'algorithme pourrait être suffisant.

Si un protocole ne porte pas d'identifiant d'algorithme, alors le numéro de version de protocole ou quelque autre changement majeur est nécessaire pour la transition d'un algorithme à un autre. L'inclusion d'un identifiant d'algorithme est une étape minimale vers l'agilité d'algorithme de chiffrement.

Parfois, une combinaison de numéros de version de protocole et d'identifiantsexplicites d'algorithme ou suite est appropriée. Par exemple, le numéro de version de sécurité de couche transport (TLS, *Transport Layer Security*) [RFC5246] désigne la fonction de déduction de clé par défaut, et l'identifiant de suite de chiffrement désigne le reste des algorithmes nécessaires.

Certaines approches portent un identifiant pour chaque algorithme utilisé. D'autres approches portent un identifiant pour une suite complète d'algorithmes. Les deux approches sont utilisées dans les protocoles de l'IETF. Les concepteurs sont invités à prendre une de ces approches et à l'utiliser de façon cohérente dans tout le protocole ou famille de protocoles. Les identifiants de suite rendent plus facile au concepteur de protocole de s'assurer que le choix des algorithmes est complet et compatible pour de futures allocations. Cependant, les identifiants de suite font par nature face à une explosion combinatoire lorsque de nouveaux algorithmes sont définis. Les identifiants d'algorithme imposent par ailleurs une charge aux mises en œuvre en forçant une détermination au moment du démarrage pour savoir quelles combinaisons d'algorithmes sont acceptables.

Sans considération de l'approche utilisée, les protocoles négocient historiquement ensemble le chiffrement symétrique et le mode de chiffrement pour s'assurer qu'ils sont compatibles.

Dans la suite de protocoles IPsec, le protocole d'échange de clé Internet version 2 (IKEv2, *Internet Key Exchange Protocol version 2*) [RFC7296] porte les identifiants d'algorithmes pour l'en-tête d'authentification (AH, *Authentication Header*) [RFC4302] et l'encapsulation de charge utile de sécurité (ESP, *Encapsulating Security Payload*) [RFC4303]. Une telle séparation est un choix de conception très fin. À l'opposé, TLS [RFC5246] porte des identifiants de suite de chiffrement, qui sont aussi un choix de conceplion très fin.

Un registre de l'IANA DEVRAIT être utilisé pour ces identifiants d'algorithme ou de suite d'algorithmes. Une fois qu'un identifiant d'algorithme est ajouté au registre, il ne devrait pas être changé ou supprimé. Cependant, il est souhaitable de marquer une entrée de registre comme déconseillée quand la mise en œuvre n'est plus conseillée.

2.2. Algorithmes de mise en œuvre obligatoire

Pour une interopérabilité sûre, le BCP 61 [RFC3365] reconnaît que les homologues communicants qui utilisent des mécanismes cryptographiques doivent prendre en charge un ensemble commun d'algorithmes de chiffrement forts. Pour cette raison, les protocoles de l'IETF qui emploient le chiffrement DOIVENT spécifier un ou plusieurs algorithmes ou suites d'algorithmes forts de mise en œuvre obligatoire. Ceci n'exige pas de tous les déploiements qu'ils utilisent cet algorithme ou suite, mais cela exige que cela soit disponible pour tous les déploiements.

L'IETF doit être capable de changer les algorithmes de mise en œuvre obligatoire au fil du temps. Il est très souhaitable de faire ce changement sans mettre à jour la spécification du protocole de base. Pour réaliser cet objectif, il est RECOMMANDÉ que la spécification du protocole de base inclue une référence à un document d'accompagnement des algorithmes, permettant de mettre à jour un document sans exiger nécessairement une mise à jour de l'autre. Cette division facilite aussi l'avancement de la spécification du protocole de base sur l'échelle de maturité des normes même si le document d'algorithme change fréquemment.

L'IETF DEVRAIT garder petit l'ensemble des algorithmes de mise en œuvre obligatoire. Pour ce faire, l'ensemble des algorithmes devra nécessairement changer dans le temps, et la transition DEVRAIT arriver avant que les algorithmes de l'ensemble en cours n'arrivent au point de rupture de l'affaiblissement.

2.2.1 Spécifications de plateformes

Noter que les algorithmes ou suites de mise en œuvre obligatoire ne sont pas spécifiés pour les protocoles qui sont incorporés dans d'autres protocoles ; dans ce cas, la spécification du protocole de niveau système identifie l'algorithme ou suite de mise en œuvre obligatoire. Par exemple, S/MIME [RFC5751] utilise la syntaxe de message cryptographique (CMS, *cryptographic message Syntax*) [RFC5652], et S/MIME spécifie les algorithmes de mise en œuvre obligatoire, mais pas la CMS. Cette approche permet aux autres protocoles de faire usage de la CMS et de faire des choix différents d'algorithme de mise en œuvre obligatoire.

2.2.2 Taille de clé de chiffrement

Certains algorithmes de chiffrement sont liés de façon inhérente à une taille de clé spécifique, mais d'autres permettent de nombreuses tailles de clé différentes. De même, certains algorithmes supportent des paramètres de différentes tailles, comme des valeurs de vérification d'intégrité ou de noms occasionnels. La spécification de l'algorithme DOIT identifier les tailles de

clé et tailles de paramètres spécifiques qui doivent être prises en charge. Quand plus d'une taille de clé est disponible, on s'attendra à ce que la taille de clé de mise en œuvre obligatoire augmente avec le temps.

On trouvera des lignes directrices sur la taille de clé de chiffrement pour les clés asymétriques dans le BCP 86 [RFC3766].

On trouvera des lignes directrices sur la taille de clé de chiffrement pour les clés symétriques dans le BCP 195 [RFC7525].

2.2.3 Fournir l'explication des changements attencus

Heureusement, les échecs d'algorithme sans avertissement sont rares. Le plus souvent, la transition d'algorithme est le résultat de l'age. Par exemple, la transition de DES à Triple-DES puis à AES a eu lieu sur des décennies, causant un glissement de la force du chiffrement de bloc symétrique de 56 bits à 112 bits puis à 128 bits. Lorsque possible, les auteurs DEVRAIENT donner un avertissement aux mises en œuvre sur les transitions d'algorithme prévues. Une approche qui a d'abord été utilisée dans la [RFC4307] est d'utiliser DEVRAIT+, DEVRAIT-, et DOIT- dans la spécification des algorithmes. Les définitions ci-dessous sont légèrement modifiées par rapport à celles de la RFC 4307.

DEVRAIT+ : ce terme signifie la même chose que DEVRAIT. Cependant, il est probable qu'un algorithme marqué DEVRAIT+ va être promu à un DOIT à l'avenir.

DEVRAIT- : ce terme signifie la même chose que DEVRAIT. Cependant, il est probable qu'un algorithme marqué DEVRAIT- va être déconseillé en un PEUT ou pire à l'avenir.

DOIT- : ce terme signifie la même chose que DOIT. Cependant, il est prévu qu'un algorithme marqué DOIT- va être dégradé à l'avenir. Bien que le statut de l'algorithme soit déterminé ultérieurement, il est raisonnable de s'attendre à ce que le statut d'un algorithme DOIT- va rester au moins un DEVRAIT ou DEVRAIT-.

2.3 Transition à partir d'algorithmes faibles

La transition à partir d'un vieil algorithme qui se trouve être faible peut être difficile. Il est bien sûr normal de spécifier l'utilisation d'un nouvel algorithme, meilleur. Et ensuite, quand le nouvel algorithme est largement déployé, le vieil algorithme ne devrait plus être utilisé. Cependant, la connaissance de la mise en œuvre et du déploiement du nouvel algorithme va toujours être imparfaite, de sorte qu'on ne peut pas être complètement assuré de l'interopérabilité avec le nouvel algorithme.

Une transition d'algorithme est naturellement facilitée par un mécanisme de choix ou de négociation d'algorithme. Les protocoles choisissent traditionnellement le meilleur algorithme ou suite qui est pris en charge par tous les homologues communicants et acceptable par leurs politiques. De plus, un mécanisme est nécessaire pour déterminer si le nouvel algorithme a été déployé. Par exemple, SMIMECapabilities [RFC5751] permet aux agent d'utilisateur de la messagerie S/MIME de partager la liste des algorithmes qu'ils veulent utiliser dans l'ordre de préférence. Dans un autre exemple, l'option DNSSEC EDNS0 [RFC6975] mesure l'acceptation et l'utilisation de nouveaux algorithmes de signature numérique.

Dans l'infrastructure de clé publique de ressource (RPKI, *Resource Public Key Infrastructure*) une signature numérique mondialement reconnue est nécessaire. Le BCP 182 [RFC6916] donne une approche de transition, où un second algorithme de signature est introduit, puis celui d'origine est éliminé.

Dans le pire des cas, le vieil algorithme peut se trouver être tragiquement fautif, permettant à un attaquant occasionnel de télécharger un simple script pour le casser. Malheureusement, cela s'est produit quand un algorithme sûr est utilisé de façon incorrecte ou est utilisé avec une mauvaise gestion de clés, résultant en une suite faible d'algorithmes de chiffrement. Dans une telle situation, la protection offerte par l'algorithme est sévèrement compromise, peut-être au point qu'on veuille arrêter d'utiliser la suite faible, rejectant les offres d'utilisation de la suite faible bien avant que la nouvelle suite soit largement déployée.

Dans tous les cas, il vient un moment où les gens refusent d'utiliser le vieil algorithme ou suite faible. Cela peut arriver à un jour désigné, ou chaque installation peut choisir une date de son choix.

2.4 Mécanismes de transition d'algorithme

Le choix ou la négociation de l'algorithme de chiffrement DEVRAIT être protégé en intégrité. Si le choix n'est pas protégé en intégrité, le protocole va alors être soumis à une attaque en dégradation. Sans protection de l'intégrité du choix de l'algorithme ou suite d'algorithmes, la tentative de transition à un nouvel algorithme ou suite d'algorithmes peut introduire de nouvelles opportunités d'attaques en dégradation.

Les mécanismes de transition ont besoin de considérer l'algorithme utilisé pour fournir la protection de l'intégrité pour la négociation de l'algorithme lui-même.

Si un protocole spécifie un seul algorithme d'intégrité de mise en œuvre obligatoire, cet algorithme va finir par être trouvé faible.

Une attention supplémentaire est nécessaire quand un algorithme de mise en œuvre obligatoire est utilisé pour fournir la protection de l'intégrité pour la négociation d'autres algorithmes de chiffrement. Dans cette situation, une faute dans l'algorithme de mise en œuvre obligatoire peut permettre à un attaquant d'influencer le choix des autres algorithmes.

2.5 Établissement des clés de chiffrement

Traditionnellement, les concepteurs de protocoles ont évité d'avoir plus d'une approche pour les échanges qui établissent les clés de chiffrement parce que cela rend l'analyse de la sécurité du protocole global plus difficile. Quand des cadres comme le protocole d'authentification extensible (EAP, *Extensible Authentication Protocol*) [RFC3748] et l'authentification simple et couche de sécurité (SASL, *Simple Authentication et Security Layer*) [RFC4422] sont employés, l'établissement de clé est très souple, cachant souvent de nombreux détails de l'application. Il en résulte des protocoles qui prennent en charge plusieurs approches d'établissement de clés. En fait, l'approche d'établissement de clé elle-même est négociable, ce qui pose un défi au concepteur pour protéger la négociation de l'approche de l'établissement de clé avant qu'elle soit utilisée pour produire les clés de chiffrement.

Les protocoles peuvent négocier une approche d'établissement de clé, déduire une clé de chiffrement initiale, et ensuite authentifier la négociation. Cependant, si l'authentification échoue, le seul recours est de recommencer la négociation depuis le début.

Certains environnements vont restreindre les approches d'établissement de clé par politique. De telles politiques tendent à améliorer l'interopérabilité dans un environnement particulier, mais elles causent des problèmes à ceux qui ont besoin de travailler dans plusieurs environnements incompatibles.

2.6 Préserver l'interopérabilité

Déconseiller un algorithme de chiffrement est très difficile. Les gens n'aiment pas introduire des problèmes d'interopérabilité, même pour préserver la sécurité. Il en résulte que des algorithmes fautifs sont acceptés pendant beaucoup trop longgtemps. L'impact des logiciels traditionnels et des longues queues de prise en charge sur la sécurité peut être réduit en rendant facile la transition des vieux algorithmes et suites à des nouveaux. Une pression sociale est souvent nécessaire pour faire arriver la transition.

Les développeurs ont été réticents à retirer les algorithmes ou suites déconseillés du logiciel de serveur, et les administrateurs de serveurs ont été réticents à les désactiver au motif que certains ne seraient plus capables de se connecter à leur serveur. Les utilisateurs et les administrateurs veulent améliorer la sécurité en utilisant les algorithmes les mieux pris en charge, mais leurs actions sont tempérées par le désir de préserver la connectivité. Récemment, des fabricants de navigateurs ont commencé à fournir des avertissements visuels quand un algorithme ou suite d'algorithmes déconseillé est utilisé. Ces avertissements visuels fournissent une nouvelle incitation à la transition des algorithmes et suites d'algorithmes déconseillés, invitant les consommateurs à demander une sécurité améliorée.

La transition dans l'infrastructure de l'Internet est particulièrement difficilé. La signature numérique sur le certificate pour une autorité de certification (CA, *certification authority*) [RFC5280] intermédiaire est souvent prévue pour durer des décennies, ce qui fait obstacle à la transition d'un algorithme de signature faible ou d'une courte longueur de clé. Une fois qu'un certificat à longue durée de vie est produit avec un algorithme de signature particulier, cet algorithme va être utilisé par de nombreux consommateurs d'assertions, et aucun d'eux ne peut arrêter de le prendre en charge sans invalider tous les certificats subordonnés. Dans un système hiérarchique, de nombreux certificats subordonnés pourraient être impactés par la décision d'abandonner la prise en charge d'un algorithme de signature faible ou d'une fonction de hachage associée.

Les organisations qui ont une influence significative peuvent aider en coordonnant leur abandon d'une suite d'algorithmes, rendant la transition plus facile pour leurs propres utilisateurs ainsi que pour les autres.

2.7 Compromis sur la force de la sécurité

Lors du choix ou de la négociation d'une suite d'algorithmes de chiffrement, la force de chaque algorithme DEVRAIT être considérée. Les algorithmes dans une suite DEVRAIENT être en gros égaux pour fournir des facteurs de travail comparables pour les attaques bien connues. Cependant, le service de sécurité fourni par chaque algorithme dans un contexte particulier doit

être considéré lors du choix. La force de l'algorithme doit être considérée au moment de la conception d'un protocole. Elle doit aussi être considérée au moment où la mise en œuvre de protocole est déployée et configurée. L'avis des experts est utile, mais, en réalité, un tel avis est souvent indisponible pour les administrateurs de système qui déploient une mise en œuvre de protocole. Pour cette raison, les concepteurs de protocoles DEVRAIT fournir des lignes directrices claires pour la mise en œuvre, pour évaluer les options disponibles au moment du déploiement.

Les performances sont toujours un facteur de choix des algorithmes de chiffrement. Les performances et la sécurité doivent être mises en balance. Certains algorithmes offrent de la souplesse dans leur force en ajustant la taille de clé, le nombre de tours, la taille de l'étiquette d'authentification, la taille du groupe premier, et ainsi de suite. Par exemple, les suites de chiffrement TLS incluent Diffie-Hellman ou RSA sans spécifier une longueur de clé publique particulière. Si l'identifiant d'algorithme ou l'identifiant de suite d'algorithmes désigne une longueur de clé publique particulière, la migration à de plus longues va être plus difficile. Par ailleurs, l'inclusion d'une longueur de clé publique rendrait plus facile de migrer sur de plus longues quand les ressources de calcul disponibles pour un attaquant imposent de le faire. La souplesse sur une longueur de clé asymétrique a conduit à des problèmes d'interopérabilité, et pour éviter ces problèmes à l'avenir, tout aspect de l'algorithme non spécifié par les identifiants d'algorithme doit être négocié, incluant la taille de clé et les paramètres.

Dans la CMS [RFC5652], une clé de chiffrement de clé précédemment distribuée peut être utilisée pour chiffrer une clé de chiffremenr de contenu, qui est utilisée à son tour pour chiffrer le contenu. Les algorithmes de chiffrement de clé et de chiffrement de contenu sont souvent différents. Si, par exemple, un contenu de message est chiffré avec une clé AES de 128 bits et si la clé de chiffrement de contenu est enveloppée avec une clé AES de 256 bits, alors au plus 128 bits de protection sont fournis. Dans cette situation, les choix de l'algorithmr et de la taille de clé devraient assurer que le chiffrement de la clé est au moins aussi fort que le chiffrement du contenu. En général, envelopper une clé dans une autre clé de taille différente donne la force de sécurité de la plus courte clé.

2.8. Compromis sur la complexité du protocole

Les concepteurs de protocole doivent anticiper les changements de l'ensemble d'algorithmes de chiffrement pris en charge au cours du temps. Il y a différentes façons de permettre la transition, et la Section 3 discute certains des problèmes qui s”y rapportent.

Garder les mises en œuvre aussi simples que possible. Une négociation de protocole complexe offre des opportunités d'attaque, comme des attaques en dégradation. La prise en charge de nombreuses solution de remplacement pour les algorithmes est aussi dommageable. Ces deux situations peuvent conduire à ce que des portions de la mise en œuvre qui sont rarement utilisées augmentent les opportunités de fautes non découvertes exploitables de la mise en œuvre.

2.9 Sécurité opportuniste

En dépit des lignes directrices du paragraphe 2.4, une sécurité opportuniste [RFC7435] mérite aussi la considération, en particulier au moment où le la mise en œuvre d'un protocole est déployée et configurée. La sécurité opportuniste, comme les autres raisons de chiffrement du trafic, doit faire usage des plus forts algorithmes de chiffrement qui sont mis en œuvre et permis par la politique. Quand les parties communicantes n'ont pas des algorithmes forts en commun, utiliser des algorithmes qui sont faibles contre des attaquants évolués mais suffisants contre d'autres est une façon de rendre une surveillance invasive significativement plus difficile. Par suite, quand les parties communicantes n'ont pas d'algorithmes forts en commun, des algorithmes qui ne seraient pas acceptables dans de nombreuses situations négociées sont acceptables pour une sécurité opportuniste quand des systèmes traditionnels sont utilisés pour des session chiffrées non authentifiées (comme discuté à la Section 3 de la [RFC7435]) pour autant que leur utilisation ne facilite pas des attaques en dégradation. De même, des algorithmes plus faibles et des tailles de clé plus courtes sont aussi acceptables pour une sécurité opportuniste avec les mêmes contraintes.

Ceci dit, l'utilisation d'algorithmes forts est toujours préférable.

3. Spécifications des algorithmes de chiffrement

Il y a des compromis à faire entre le nombre d'algorithmes de chiffrement pris en charge et le temps pour déployer un nouvel algorithme. Cette section donne des indications sur le compromis qui s'offre aux concepteurs de protocoles.

Idéalement, deux ensembles indépendants d'algorithmes de mise en œuvre obligatoire vont être spécifiés, permettant une suite principale et une suite secondaire. Cette approche assure que la suite secondaire est largement déployée si une faute se trouve dans la suite principale.

3.1 Choisir des algorithmes de mise en œuvre obligatoire

Il peut sembler que si la capacité d'utiliser un algorithme de son propre choix est très désirable, cependant, le choix est souvent plutôt laissé aux experts. Quand il y a un choix, l'utilisateur final peut le faire entre des profils de configuration qui ont été définis par des experts. De plus, les experts n'ont pas besoin de spécifier toutes les solutions d'algorithme de chiffrement qui se présentent. Spécifier tous les choix possibles ne va pas les rendre disponibles à toutes les mises en œuvre. Les algorithmes de mise en œuvre obligatoire DOIVENT avoir une spécification publique et une documentation publique stables qui ont été bien étudiées, source d'une confiance significative. L'IETF a toujours eu une préférence pour les algorithmes simples. Il y a des avantages significatifs à choisir des algorithmes et suites largement déployés. Les algorithmes choisis doivent être résistant aux attaques de canal latéral et aussi satisfaire aux exigences de performances, puissance, et taille de code sur une large variété de plateformes. De plus, l'inclusion de trop de solutions de remplacement peut ajouter de la complexité au choix ou négociation d'algorithme. La spécification de trop de solutions de remplacement va probablement amoindrir l'interopérabilité et peut amoindrir aussi la sécurité. Lorsque ils spécifient de nouveaux algorithmes ou suites, les concepteurs de protocoles devraient avoir la prudence de considérer si ceux existants peuvent être déconseillés.

Il y a un avantage significatif à choisir les mêmes algorithmes et suites pour différents protocoles. Utiliser les mêmes algorithmes peut simplifier la mise en œuvre quand plus d'un des protocoles est utilisé dans le même appareil ou système

Parfois plus d'un algorithme de mise en œuvre obligatoire est nécessaire pour augmenter la probabilité d'interopérabilité parmi une population diverse. Par exemple, le chiffrement authentifié est fourni par AES-CCM [RFC3610] et AES-GCM [GCM]. Ces deux algorithmes sont considérés comme sûrs. AES-CCM est disponible dans le matériel utilisé par de nombreux petits appareils, et AES-GCM est mis en parallèle et convient bien pour les appareils à haut débit. Donc, une application qui a besoin de chiffrement authentifié peut spécifier un de ces algorithmes ou les deux, selon la population.

3.2 Trop de choix peut être dommageable

Il est très facile de spécifier l'utilisation de tout algorithme de chiffrement arbitraire, et une fois la spécification disponible, l'algorithme est mis en œuvre et déployé. Certains disent que la liberté de spécifier des algorithmes indépendamment du reste du protocole a conduit à la spécification de trop d'algorithmes de chiffrement. Une fois déployés, même avec un succès modéré, il est assez difficile de retirer des algorithmes parce que l'interopérabilité avec certaines parties va être impactée. Par suite, des chiffrements faibles restent en vigueur beaucoup trop longtemps. Les mises en œuvre sont parfois forcées de conserver un algorithme de chiffrement bien au delà de sa durée de vie utile.

Afin de gérer la prolifération des choix d'algorithme et donner un espoir d'interopérabilité, de nombreux protocoles spécifient des algorithmes ou suites de mise en œuvre obligatoire. Toutes les mises en œuvre sont supposées prendre en charge l'algorithme de chiffrement de mise en œuvre obligatoire, et elles peuvent inclure tous les autres algorithmes qu'elles désirent. Les algorithmes de mise en œuvre obligatoire sont choisis pour être très sûrs et suivent les lignes directrices de la [RFC1984]. Bien sûr, de nombreux autres facteurs, incluant les droits de propriété intellectuelle, ont un impact sur les algorithmes de chiffrement qui sont retenus par la communauté. Généralement, les algorithmes de mise en œuvre obligatoire devraient être préférés, et les autres algorithmes devraient n'être choisis que dans des situations particulières. Cependant, il peut être très difficile pour un administrateur de système expérimenté de déterminer la configuration appropriée pour réaliser ces préférences.

Dans certains cas, plus d'un algorithme de chiffrement de mise en œuvre obligatoire a été spécifié. C'est destiné à assurer qu'au moins un algorithme de chiffrement sûr va être disponible, même si d'autres algorithmes de mise en œuvre obligatoire sont cassés. Pour atteindre cet objectif, les algorithmes choisis doivent être divers, afin que les avancées en cryptanalyse contre un des algorithmes n'impacte pas aussi les autres algorithmes choisis. L'idée est d'avoir un algorithme mis en œuvre et déployé comme solution de repli. Cependant, tous les algorithmes choisis doivent être régulièrement vérifiés pour s'assurer de la qualité de la mise en œuvre. Ce n'est pas toujours facile à faire, en particulier si les divers algorithms choisis exigent des accréditifs différents. Obtenir plusieurs accréditifs pour la même installation est une charge inacceptable pour les administrateurs de système. Aussi, la façon dont les administrateurs de système sont avertis de changer les algorithmes ou suites est, au mieux, ad hoc et, au pire, entièrement absente.

3.3 Prendre une vraie suite de chiffrement peut être dommageable

Dans le passé, les concepteurs de protocoles ont choisi un algorithme ou suite de chiffrement, et ensuite lié de nombreux détails du protocole à ce choix. Il faut prévoir la transition d'algorithme, soit à cause d'une faute dans le choix initial, soit parce que le protocole est utilisé avec succès pendant une longue période et que l'algorithme devient faible avec l'age. De l'une et l'autre façon, la conception devrait permettre la transition.

Les concepteurs de protocole sont parfois trompés par la simplicité qui résulte du choix d'un vrai algorithme ou suite. Comme les algorithmes vieillissent, le choix ne peut pas être stable pour toujours. Même le plus simple protocole a besoin d'un numéro

de version pour signaler quel algorithme est utilisé. Cette approche a au moins deux conséquences souhaitables. D'abord, le protocole est plus simple parce qu'il n'est pas besoin de négociation d'algorithme. Ensuite, les administrateurs de système n'ont pas besoin de prendre de décisions de configuration en relation avec l'algorithme. Cependant, la seule façon de répondre à la nouvelle que l'algorithme qui fait partie de la suite de chiffrement indispensable a été cassé, est de mettre à jour la spécification du protocole avec la prochaine version, de mettre en œuvre la nouvelle spécification, et de la déployer.

La première spécification de IEEE 802.11 [WiFi] incluait la confidentialité équivalente à celle des résaux câblés (WEP, *Wired Equivalent Privacy*) comme seule technique de chiffrement. De nombreux détails du protocole découlaient de l'algorithme choisi. WEP a été trouvé assez faible [WEP], et un très gros effort a été nécessaire pour spécifier, mettre en œuvre, et déployer les techniques de chiffrement de remplacement. Cet effort a été rendu encore plus dur par les choix de conception du protocole qui étaient liés au choix de l'algorithme initial et au désir de rétro compatibilité.

L'expérience de la transition de SHA-1 à SHA-256 indique que le temps entre la spécification du protocole et l'utilisation la plus large prend plus de cinq années. Dans ce cas, les spécifications et la mise en œuvre du protocole étaient directes et très rapides. Dans de nombreux produits logiciels, le nouvel algorithme n'est pas considéré comme une mise à jour de la livraison existante, de sorte que la sortie de la prochaine livraison, le déploiement suivant, et finalement l'ajustement de la configuration par les administrateurs de système prend plusieurs années. Dans de nombreux produits matiériels de grande consommation, le logiciel pour mettre en œuvre le nouvel algorithme est difficile à localiser et installer, ou n'est simplement pas disponible. De plus, les fournisseurs d'infrastructure ne souhaitent pas faire la transition tant que leurs clients potentiels ne sont pas capables d'utiliser le nouvel algorithme.

3.4 Suites de chiffrement nationales

Certains pays spécifient des algorithmes de chiffrement, et exigent ensuite leur utilisation par la loi ou le règlement. Ces algorithmes peuvent n'avoir pas eu une très large révision publique, et peuvent avoir une portée géographique limitée dans leur déploiement. Ainsi, la loi ou le réglement crée un marché captif. Par suite, de tels algorithmes vont être spécifiés, mis en œuvre, et déployés. Le serveur par défaut ou la configuration qui répond DEVRAIT désactiver de tels algorithmes ; de cette façon, une action explicite de l'administrateur de système est nécessaire pour les activer lorsque ils sont effectivement nécessaires. Pour les petits appareils sans interface d'utilisateur, une action de l'administrateur n'est possible qu'au moment de l'achat de l'appareil.

Les algorithmes nationaux peuvent forcer une mise en œuvre à produire plusieurs livraisons incompatibles d'un produit pour les différents pays ou régions ; ceci a un coût significativement supérieur sur le développement d'un produit utilisant un algorithme acceptable mondialement. Cette situation pourrait être encore pire si les divers algorithmes nationaux imposent des exigences différentes au protocole, sa gestion de clés, ou son utilisation de valeurs aléatoires.

4. Considérations sur la sécurité

Le présent document donne des lignes directrices pour les groupes de travail et les concepteurs de protocole. La sécurité de l'Internet est améliorée quand des algorithmes de chiffrement cassés ou faibles peuvent être facilement remplacés par de plus forts.

Du point de vue du développement et de la maintenance du logiciel, des algorithmes de chiffrement peuvent souvent être ajoutés et supprimés sans faire de changement à la structure des données environnante, aux sous programmes d'analyse du protocole, ou aux automates à états. Cette approche sépare la mise en œuvre de l'algorithme de chiffrement du reste du code, ce qui rend plus facile de prendre en compte des soucis de sécurité particuliers comme l'exposition de clé et l'exécution en temps constant.

Parfois les protocoles de couche d'application peuvent faire usage de protocoles de sécurité de couche transport, comme TLS [RFC5246] ou Datagram TLS (DTLS) [RFC6347]. Cela isole le protocole de couche d'application des détails de cryptographie, mais il est probable qu'il sera toujours nécessaire de traiter la transition de trafic non protégé à trafic protégé dans le protocole de couche d'application. De plus, le protocole de couche application peut avoir besoin de traiter la dégradation de communication chiffrée à communication en clair.

Le matériel pose des défis dans la transition d'algorithmes, à la fois pour les petits appareils et les équipements de centre de données à très haut débit. De nombreux petits appareils n'incluent pas du tout de capacité de mettre à jour les logiciels. Même si le logiciel peut être mis à jour, les petits appareils sont souvent déployés dans des endroits où il est très peu pratique de le faire. Un équipement de centre de données à haut débit peut utiliser des puces spéciales pour réaliser de très hautes performances, ce qui signifie que le remplacement au niveau du bureau peut être nécessaire pour changer l'algorithme. Les coûts et les délais sont tous deux des facteurs d'une telle mise à niveau.

Dans la plupart des cas, l'algorithme de chiffrement reste fort, mais une attaque a été trouvée contre la façon dont l'algorithme fort est utilisé dans un protocole particulier. Dans ce cas, un changement de protocole va probablement être nécessaire. Par exemple, l'ordre des opérations de chiffrement dans le protocole TLS a évolué parce que diverses attaques ont été découvertes. À l'origine, TLS effectuait le chiffrement après le calcul du code d'authentification de message (MAC, *message authentication code*). Cet ordre des opérations est appelé MAC-puis-chiffrement, qui implique en fait le calcul du MAC, le bourrage, et ensuite le chiffrement. Ceci n'est plus considéré comme sûr [BN], [K]. Par suite, un mécanisme a été spécifié pour utiliser à la place chiffrement-puis-MAC [RFC7366]. De futures versions de TLS sont prévues qui utiliseront exclusivement des algorithmes de chiffrement authentifiés [RFC5116], ce qui devrait résoudre aussi la discussion sur l'ordre. Après la découverte de ces attaques, la mise à jour des algorithmes de chiffrement ne sera probablement pas suffisante pour déjouer de nouvelles attaques. Il peut être nécessaire de faire des changements significatifs au protocole.

Certains protocoles sont utilisés pour protéger les données mémorisées. Par exemple, S/MIME [RFC5751] peut protéger un message conservé dans une boîte aux letres. Pour récupérer les données mémorisées protégées, les mises en œuvre de protocole doivent prendre en charge les anciens algorithmes, même quand elles n'utilisent plus les anciens algorithmes pour la protection des nouvelles données mémorisées.

La prise en charge de trop d'algorithmes peut conduire à des vulnérabilités de la mise en œuvre. Quand de nombreux algorithmes sont pris en charge, certains d'entre eux vont être rarement utilisés. Tout code qui est rarement utilisé peut contenir des fautes non détectées, et les mises en œuvre d'algorithme ne sont pas différentes. Des mesures DEVRAIENT être utilisées pour déterminer si des algorithmes mis en œuvre sont réellement utilisés, et si ils ne le sont pas, de futures livraisons devraient les supprimer. De plus, les algorithmes ou suites non utilisés DEVRAIENT être marqués comme déconseillés dans le registre IANA. En bref, éliminer le gras.

Le paragraphe 2.3 parle de la transition d'algorithme sans considérer d'autres aspects de la conception du protocole. En pratique, il y a des interdépendances entre l'algorithme de chiffrement et les autres aspects du protocole. Par exemple, l'attaque BEAST [BEAST] contre TLS [RFC5246] a été cause que de nombreux sites ont supprimé des algorithmes de chiffrement modernes en faveur d'algorithmes plus anciens et plus faibles.

Des mécanismes pour mettre à jour en temps utile les appareils sont nécessaires pour déployer un algorithme ou suite de remplacement. Cela prend longtemps pour spécifier, mettre en œuvre, et déployer un remplacement ; donc, le processus de transition doit commencer lorsque des fautes pratiquement exploitables sont connues. Le processus de mise à jour sur certains appareils implique la certification, ce qui augmente encore le délai pour déployer un remplacement. Par exemple, des appareils qui font partie des systèmes de santé ou de sécurité exigent souvenr une certification avant le déploiement. Les systèmes incorporés et les systèmes de contrôle de supervision et d'acquisition de données (SCADA, *supervisory control et data acquisition*) ont souvent des cycles de mise à niveau qui s'étendent sur de nombreuses années, conduisant à des problèmes similaires de temps de déploiement. Une action prompte est nécessaire si un remplacement a quelque espoir d'être déployé avant que les techniques d'exploitation deviennent largement disponibles.

5. Considérations relatives à l'IANA

Le présent document n'établit aucun nouveau registre IANA, ni n'ajoute d'entrée à un registre existant.

Le présent document RECOMMANDE une convention pour les nouveaux registres d'identifiants d'algorithmes ou suite d'algorithmes de chiffrement. Une fois qu'un identifiant d'algorithme ou de suite est ajouté au registre, il NE DEVRAIT PAS être changé ou supprimé. Cependant, il est souhaitable d'inclure un moyen de marquer une entrée de registre comme déconseillée lorsque sa mise en œuvre n'est plus conseillée.

6. Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997, DOI 10.17487/RFC2119.

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

7. Références pour information

[BEAST] Wikipedia, "BEAST attack" under "Transport Layer Security", novembre 2015, <https://en.wikipedia.org/w/index.php?title=Transport_Layer_Security&oldid=689441642#BEAST_attack >.

[BN] Bellare, M. and C. Namprempre, "Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm", Proceedings of AsiaCrypt '00, Springer-Verlag LNCS No. 1976, p. 531, DOI 10.1007/3-540-44448-3_41, décembre 2000.

[GCM] Dworkin, M, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) et GMAC", NIST Special Publication 800-30D, novembre 2007.

[K] Krawczyk, H., "The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?)", Proceedings of Crypto '01, Springer-Verlag LNCS No. 2139, p. 310, DOI 10.1007/3-540-44647-8_19, août 2001.

[RFC1984] IAB, IESG, "Déclaration IAB IESG sur la technologie cryptographique dans l'Internet", août 1996. *(Info.)* DOI 10.17487/RFC1984.

[RFC3365] J. Schiller, "Exigence d'une sécurité forte dans les protocoles standard de l'IETF", août 2002. (BCP0061), DOI 10.17487/RFC3365.

[RFC3610] D. Whiting, R. Housley, N. Ferguson, "Compteur avec CBC-MAC (CCM)", septembre 2003. *(Information)* DOI 10.17487/RFC3610.

[RFC3748] B. Aboba et autres, "Protocole extensible d'authentification (EAP)", juin 2004. *(P.S., MàJ par* RFC5247), DOI 10.17487/RFC3748,

[RFC4302] S. Kent, "En-tête d'authentification IP", décembre 2005. *(P.S.)*, DOI 10.17487/RFC43020

[RFC4303] S. Kent, "Encapsulation de charge utile de sécurité dans IP (ESP)", décembre 2005. (*Remplace* RFC2406) *(P.S.)* DOI 10.17487/RFC4303.

[RFC4307] J. Schiller, "Algorithmes cryptographiques à utiliser avec la version 2 de l'échange de clés sur Internet (IKEv2)", décembre 2005. *(P.S. ; rendue obsolète par* RFC8247), DOI 10.17487/RFC4307.

[RFC4442] S. Fries, H. Tschofenig, "Amorçage de l'authentification tolérante aux pertes de flux à synchronisation efficace (TESLA)", mars 2006. *(P.S.)*, DOI 10.17487/RFC4422.

[RFC5116] D. McGrew, "Interface et algorithmes pour le chiffrement authentifié", janvier 2008. *(P.S.)*, DOI 10.17487/RFC5116.

[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), DOI 10.17487/RFC5246.

[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), DOI 10.17487/RFC5280.

[RFC5652] R. Housley, "Syntaxe de message cryptographique (CMS)", STD70, septembre 2009. *(Remplace* RFC3852), DOI 10.17487/RFC5652.

[RFC5751] B. Ramsdell, S. Turner, "Spécification du message d'extensions de messagerie Internet multi objets sécurisée (S/MIME) version 3.2", janvier 2010. (*Remplace* RFC3851). *(P. S. ; Remplacée par* RFC8551*)*, DOI 10.17487/RFC5751.

[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), DOI 10.17487/RFC6347.

[RFC6916] R. Gagliano, S. Kent, S. Turner, "Procédure d'agilité d'algorithme pour l'infrastructure de clé publique de ressource (RPKI)", BCP0182, avril 2013, DOI 10.17487/RFC6916.

[RFC6975] S. Crocker, S. Rose, "Compréhension de la signalisation des algorithmes de chiffrement dans les extensions de

sécurité du DNS (DNSSEC)", juillet 2013. *(P.S.)*, DOI 10.17487/RFC6975.

[RFC7296] C. Kaufman, et autres, "Protocole d'échange de clé Internet version 2 (IKEv2)", octobre 2014. STD 79. *(MàJ par* RFC7670, RFC8247), DOI 10.17487/RFC7296.

[RFC7366] R. Gutmann, "Négociation du mécanisme Encrypt-then-Mac pour TLS et DTLS", septembre 2014. *(P.S.)*, DOI 10.17487/RFC7366.

[RFC7435] V. Dukhovni, "Sécurité opportuniste : une protection la plupart du temps", décembre 2014. *(Information)*, DOI 10.17487/RFC7435.

[RFC7525] Y. Scheffer, R. Holz, P. Saint-André, "Recommandations pour l'utilisation sûre de TLS et DTLS", mai 2015. BCP195, DOI 10.17487/RFC7525.

[WEP] Wikipedia, "Wired Equivalent Privacy", novembre 2015, <https://en.wikipedia.org/w/index.php?title=Wired_Equivalent_Privacy&oldid=688848497>.

[WiFi] IEEE Std 802.11-1997, "Wireless LAN Medium Access Control (MAC) And Physical Layer (PHY) Specifications".

Remerciements

Merci à Bernard Aboba, Derek Atkins, David Black, Randy Bush, Jon Callas, Andrew Chi, Steve Crocker, Viktor Dukhovni, Stephen Farrell, Tony Finch, Ian Grigg, Peter Gutmann, Phillip Hallam-Baker, Wes Hardaker, Joe Hildebrand, Paul Hoffman, Christian Huitema, Leif Johansson, Suresh Krishnan, Watson Ladd, Paul Lambert, Ben Laurie, Eliot Lear, Nikos Mavrogiannopoulos, Kathleen Moriarty, Yoav Nir, Kenny Paterson, Rich Salz, Wendy Seltzer, Joel Sing, Rene Struik, Kristof Teichel, Martin Thompson, Jeffrey Walton, Nico Williams, et Peter Yee de leur relecture et de leur commentaires enrichissants. Bien que certaines de ces personnes ne soient pas d'accord avec certains aspects de ce document, la discussion qui a résulté de leurs commentaires a certainement contribué à améliorer le document.

Adresse de l'auteur

Russ Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170
United States
mél : housley@vigilsec.com