Améliorations de DTLS : connexion rapide et augmentation de la charge utile pour les réseaux contraints

Philippe Pittoli, Pierre David, Thomas Noël

Améliorations de DTLS : connexion rapide et augmentation de la charge utile pour les réseaux contraints

Philippe Pittoli[1] and Pierre David[1] and Thomas Noël[1]

[1]*Laboratoire ICube – Université de Strasbourg*

Transport Layer Security est un protocole défini par l'IETF pour sécuriser les communications sur Internet, et Datagram TLS est sa version utilisée dans l'Internet des Objets. Cependant, le protocole n'a pas été conçu pour des appareils contraints en mémoire, en taille de code et en vitesse de calcul comme nous pouvons retrouver dans ce domaine. Ce papier propose une version optimisée de DTLS avec pour objectif la réduction du délai de connexion et du surplus protocolaire de DTLS durant l'échange de données. Ces travaux fournissent un protocole de communication sécurisé basé sur une connexion plus efficace pour le domaine des environnements contraints.

Keywords: DTLS, Sécurité, IoT

1 Introduction

Les environnements contraints mis en œuvre par exemple dans l'Internet des Objets, nécessitent des communications sécurisées. Les capteurs ont des capacités en mémoire, en calcul et en communication limitées (RFC 7228) qui rendent inadaptés les protocoles existants sur Internet.

Transport Layer Security (TLS, RFC 5246) est un protocole standard de sécurisation des échanges sur Internet. Il est composé d'une négociation d'algorithmes et de matériel cryptographique, et d'échanges de données qui assurent la confidentialité, l'authenticité et l'intégrité des messages. TLS repose sur la couche transport TCP, ce qui le rend inadapté à certaines applications telles que la visio-conférence par exemple. Afin de permettre la sécurisation de ces applications, l'IETF a conçu Datagram Transport Layer Security (DTLS, RFC 6347).

DTLS est une version de TLS qui en reprend toutes les notions mais en étant basée sur UDP pour avoir un meilleur contrôle sur la transmission des messages. La figure 1 montre les différents paquets de signalisation échangés lors de l'établissement de la connexion, ces messages sont expliqués dans la RFC 6347. Toutefois, ce protocole reste coûteux pour des environnements contraints : de nombreux messages sont transmis lors de la négociation et la taille des messages est limitée par une surcharge protocolaire importante.

Nos travaux ont pour but de fournir une version optimisée du protocole de sécurité DTLS, avec d'une part une diminution des messages de signalisation et d'autre part une surcharge protocolaire amoindrie. Ces optimisations sont basées sur les recommandations de l'IETF sans par ailleurs dépendre d'autres couches de communication, et garantissent le même niveau de sécurité et de fonctionnement que la version originale du protocole.

Une comparaison est fournie entre la version originale de DTLS et l'amélioration apportée dans ce document, ainsi qu'une comparaison entre un chiffrement logiciel ou via du matériel dédié plus efficace. Nous nous concentrerons sur la vitesse de connexion et d'échange de messages, ainsi que sur l'occupation mémoire utilisée.

2 Objectifs, hypothèses et améliorations du protocole

Objectifs. Un des buts à atteindre dans les environnements contraints (drone, voiture, etc) est de pouvoir se connecter le plus rapidement possible. Nous cherchons à atteindre cet objectif en nous situant sous la

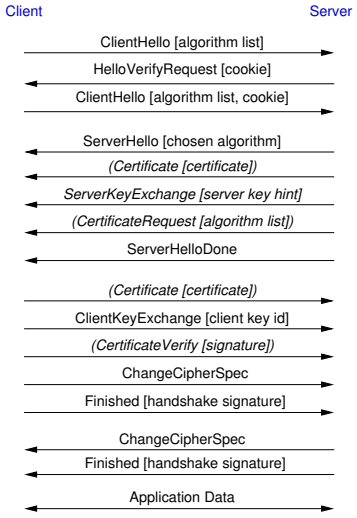

FIGURE 1: Négociation DTLS standard.

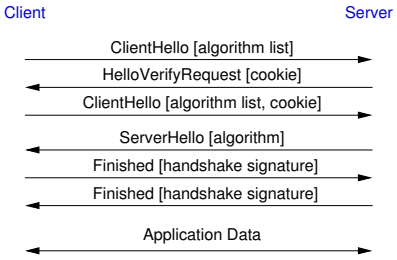

FIGURE 2: Négociation DTLS optimisée.

barre des 200 ms, même sur des architectures reposant sur des processeurs à une fréquence aussi basse que 16 MHz. Un autre objectif est d'augmenter la charge utile des messages en diminuant l'en-tête protocolaire de DTLS, tout en conservant les fonctionnalités de celui-ci et en se basant sur les recommandations de l'IETF. Enfin, nous souhaitons nous conformer aux algorithmes préconisés par l'IETF concernant le chiffrement et le partage de clés.

Le groupe de travail DICE de l'IETF a défini un profil d'usage de DTLS adapté à de nombreux appareils contraints [TF15] et qui conseille l'usage de la méthode de chiffrement Counter with CBC-MAC (CCM) et de l'algorithme de chiffrement symétrique AES 128 bits. Le groupe recommande également l'usage de clés pré-chargées, ou une négociation avec une gestion de certificats pour l'authentification si possible. Toutefois, il est important de rappeler que la négociation de clés de chiffrement via un mécanisme de certificats est coûteuse, même avec l'usage de courbes elliptiques : l'initialisation, la signature puis la vérification d'un certificat dépassent une seconde sur du matériel avec un processeur à 13 MHz [LN08].

Hypothèses. Puisque l'usage de certificat est trop coûteux pour atteindre nos objectifs, nous faisons l'hypothèse que les clés seront pré-chargées dans les appareils avant déploiement, ce qui est acceptable par exemple dans un réseau de capteurs, mais l'usage d'un canal hors bande pour cet échange initial est également envisageable. Une autre hypothèse est que tous les acteurs sont connus dans l'environnement : leur identité est déduite par l'adresse MAC (ou IP), soit elle a été partagée via le protocole applicatif. La dernière hypothèse est qu'une seule clé cryptographique est active à un instant donné par couple de nœuds.

Suppression de messages de négociation. DTLS est un protocole verbeux (voir la figure 1) et non conçu pour les environnements contraints. Compte tenu de nos hypothèses, un certain nombre de messages peuvent être supprimés pour atteindre les objectifs fixés.

Tout d'abord, nos objectifs proscrivent l'usage de certificats (première hypothèse), par conséquent nous retirons les messages correspondants. Ensuite, nous retirons les messages indiquant l'identité d'un pair, puisqu'elle est connue (seconde et troisième hypothèses). Enfin, suite à ces changements, les messages restants sont obligatoires (seuls les messages liés aux certificats et à l'annonce d'une identité sont optionnels) et leur ordre est fixe. Les messages de signalisation de DTLS qui annoncent les prochains messages envoyés ou la fin d'une séquence de messages sont donc rendus obsolètes et nous les retirons également.

La figure 2 montre une version optimisée de la négociation, ce qui simplifie l'impémentation (moindre consommation de mémoire Flash) et en améliore la lisibilité et la maintenance.

Diminution de la surcharge protocolaire. La figure 3 montre deux en-têtes utilisées dans DTLS (Record et Application), lorsqu'un algorithme de chiffrement de type Authenticated Encryption with Authenticated Data (AEAD) est utilisé pour protéger la connexion (c'est le cas avec l'algorithme CCM). L'entête Record contient le champ Epoch et un numéro de séquence (pertes, duplication, ordonnancement), ce qui est

FIGURE 3: Redondance dans les couches DTLS lors de l'usage d'algorithme AEAD.

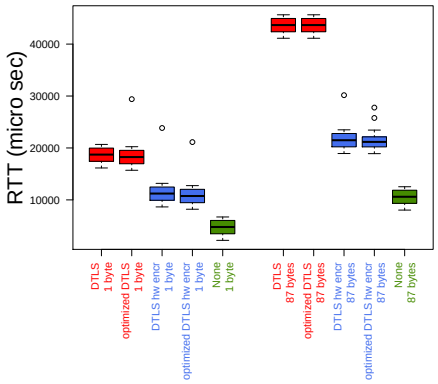

FIGURE 4: Temps d'aller-retour mesuré sur 3000 allers retours, message de 1 ou 87 octets via DTLS, DTLS optimisé, leur version avec chiffrement matériel puis sans protection.

également le cas dans l'entête Application qui utilise par convention ces champs comme vecteur d'initialisation (IV) pour l'algorithme AEAD.

Avec une couche liaison IEEE 802.15.4, ces 8 octets dupliqués représentent 6% de charge utile, leur suppression fait passer la surcharge protocolaire de DTLS de 29 à 21 octets. Le gain est avant tout la charge utile maximum disponible dans le paquet, qui passe de 89 à 97 octets. Cela permet aux applications d'envoyer plus de données en un seul message, donc de diminuer la fragmentation et la durée de transmission.

Impact des hypothèses et des modifications. Nous ne changeons pas la sémantique des messages DTLS échangés, la sécurité est donc identique. L'usage d'une clé pré-chargée sans Diffie-Hellmann implique deux restrictions (RFC 4279). Tout d'abord, il n'y a pas de *Perfect Forward Secrecy* donc un attaquant peut lire les anciens messages s'il obtient la clé. Aussi, la clé doit être générée aléatoirement puisqu'elle est sensible aux attaques par dictionnaire. Enfin, la modification de DTLS telle que présentée implique une incompatibilité avec les implémentations existantes, ce qui est acceptable dans un réseau où tous les acteurs sont connus.

3 Expérimentation et résultats

Afin de démontrer l'intérêt de notre solution, nous avons procédé à diverses mesures d'occupation mémoire, de durée de connexion et d'échange de données. Nous avons également mesuré l'impact du chiffrement matériel sur ces durées. Le matériel dédié au chiffrement est souvent intégré à la puce de communication 802.15.4 et peut être utilisé pour diminuer le temps de chiffrement s'il n'y a pas de contrainte énergétique. Ensuite nous avons mesuré l'influence de la taille des messages sur le temps d'échange, en utilisant des messages courts (1 octet) et longs (87 octets, taille maximale possible dans notre environnement). Enfin, dans le but d'isoler l'impact de DTLS sur une connexion, la couche protocolaire sera réduite au minimum pour nos expérimentations : pas d'adressage IP, de couche de transport ou de protocole applicatif.

Le code utilisé est basé sur TinyDTLS (DTLS 1.2) avec les algorithmes souhaités (CCM et AES). Le matériel utilisé correspond à l'environnement contraint visé : un micro contrôleur ATMega128RFA1 à 16 MHz doté d'une connectivité IEEE 802.15.4. La connectivité IEEE 802.15.4 est en mode pair-à-pair, avec la radio allumée sans interruption pour ne pas avoir de délai au niveau 2. Deux appareils sont utilisés, le premier agit comme un serveur DTLS, l'autre comme un client, et ils sont connectés en simple saut pour cibler les performances de DTLS.

L'expérience a été faite avec plusieurs configurations, dont une version sans DTLS ni chiffrement pour mettre en perspective le coût de la sécurité. Le tableau 1 récapitule les résultats obtenus.

Premièrement, le coût de la sécurité en mémoire vive est de plus de 10 Ko, et de presque 41 Ko en mémoire Flash. La version optimisée est moins consommatrice de ces deux ressources (7.3 % et 5 % d'usage en moins), ce qui s'explique par la suppression importante de code et de messages d'erreurs. La plus grande différence en occupation mémoire est lorsqu'on active le chiffrement matériel puisque la bibliothèque de chiffrement est retirée du code.

TABLE 1: Comparaison en occupation mémoire, puis en durée de connexion et d'échange de message entre DTLS et DTLS optimisé en chiffrement logiciel, puis avec chiffrement matériel et enfin sans sécurité (ni DTLS ni chiffrement).

Occupation mémoire (octets)	**DTLS standard chiffrement logiciel**	**DTLS optimisé chiffrement logiciel**	**DTLS standard chiffrement matériel**	**DTLS optimisé chiffrement matériel**	**Sans sécurité**
Mémoire vive	11 216	10 388	7 120	6 292	897
Mémoire non volatile	48 966	46 610	42 208	39 852	8 000
Durée médiane (μs)					
Connexion	**233 214**	**186 660**	220 840	173 628	
Échange messages courts	18 720	18 248	11 220	10 748	4 780
Échange messages longs	43 664	43 664	21 488	21 168	10 600

La durée de connexion est améliorée de 19 % grâce à nos optimisations et passe sous la barre des 200 ms : la suppression de messages réduit le nombre d'allers et retours ainsi que la durée totale d'attente pour le mécanisme CSMA-CA. Le chiffrement matériel apporte un gain significatif de 12 ms lors de la connexion, diminuant encore le temps nécessaire avant l'envoi du premier message de données.

Enfin, la figure 4 montre graphiquement les résultats des durées d'échanges obtenues pour chaque configuration. Cela montre que la durée d'échange est étroitement liée à la taille du message échangé (chiffrement coûteux des messages), et que le chiffrement matériel permet un gain de 66 jusqu'à 103 %. Pour conclure, il est également important de noter que la protection du message implique un temps d'aller-retour 3.9 fois supérieur à sa version sans sécurité, et ce malgré l'usage des algorithmes standards les plus rapides pour le chiffrement.

4 Conclusion

Dans cet article, nous avons optimisé le protocole DTLS et obtenu une diminution significative du temps de connexion (19 %) et de l'occupation mémoire (7.3 % de mémoire vive et 5 % de mémoire Flash). La diminution de la durée de connexion aide des nœuds très mobiles à envoyer leurs données rapidement avant d'être hors de portée de leur pair. La suppression du vecteur d'initialisation pour les algorithmes AEAD permet de transmettre un plus grand message, ce qui contribue à diminuer la fragmentation et à transmettre plus rapidement les données. Enfin, le chiffrement matériel apporte une diminution drastique de la durée d'échange des données en allégeant grandement le temps de chiffrement.

Références

[LN08] A. Liu and P. Ning. TinyECC : A configurable library for elliptic curve cryptography in wireless sensor networks. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, IPSN '08, pages 245–256, Washington, DC, USA, 2008. IEEE Computer Society.

[RSH+13] Shahid Raza, Hossein Shafagh, Kasun Hewage, Hummen Rene, and Thiemo Voigt. Lithe : Lightweight Secure CoAP for the Internet of Things. *IEEE Sensors Journal*, 13(10) :3711–3720, 2013.

[TF15] Hannes Tschofenig and Thomas Fossati. TLS/DTLS Profiles for the Internet of Things. Internet-Draft draft-ietf-dice-profile-17.txt, IETF Secretariat, October 2015.

[VTW+15] Malisa Vucinic, Bernard Tourancheau, Thomas Watteyne, Franck Rousseau, Andrzej Duda, Roberto Guizzetti, and Laurent Damon. DTLS Performance in Duty-Cycled Networks. In *International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC - 2015)*, Hong-Kong, China, aug 2015. IEEE.