Contrôle d'accès fédéré dans les réseaux véhiculaires

Ramzi Debab, Yacine Challal

Contrôle d'accès fédéré dans les réseaux véhiculaires

Ramzi Debab[1], Yacine Challal[2]

[1] Ecole nationale Supérieure d'Informatique, Algérie
[2] Université de Technologie de Compiègne
Heudiasyc, UMR CNRS 6599
BP. 20529, Compiègne Cedex, France

[1] r_debab@esi.dz, [2] ychallal@hdc.utc.fr

Abstract. L'émergence des réseaux véhiculaires a conduit les constructeurs automobiles et chercheurs à développer des applications embarquées communicantes afin d'échanger des messages dans le cadre de la sécurité routière ou de divertissement. D'autre part, la sécurité est devenue un des pré-requis vitaux pour le déploiement de tels réseaux. En termes de contrôle d'accès aux ressources, plusieurs solutions hétérogènes sont proposées à base de certificats X.509, de tickets Kerberos, du couple Username/Password, etc. Cette hétérogénéité technologique nous pousse à trouver une solution fédérant les techniques existantes et même futures. WS-Federation est l'une des clés visant à résoudre une telle problématique mais s'appliquant uniquement à une architecture SOA à base de Web Services. En conséquence, notre démarche a consisté à intégrer les applications véhiculaires réparties dans une architecture SOA puis à appliquer les spécifications WS-* liées à la sécurité des Web Services tout en vérifiant la fédération du contrôle d'accès aux ressources grâce à WS-Federation et à la notion des revendications. Le résultat étant une architecture communicationnelle à base du pattern ESB que nous baptisions VSB (Vehicular Servie Bus).

Keywords: Réseaux Véhiculaires, Fédération, Revendications, SOA, Web Services.

1 Introduction

Contexte:

Grâce à la technologie des ITS (Intelligent Traffic Systems), nos véhicules vont embarquer des applications ayant pour but l'échange de messages vitaux avec les unités de l'infrastructure routière afin d'assurer la sécurité des usagers de la route. Comme il sera possible, aussi et à partir de son véhicule, de consommer des services sur Internet ou sur le Cloud (L. SICHITIU et KIHL 2008).
Aujourd'hui, nous connaissons des écoles diverses (américaines, européennes et niponnes) dans le domaines des ITS. Ainsi, beaucoup de projets émergent des laboratoires de recherche de plusieurs constructeurs automobiles et universités. Néanmoins, l'hétérogénéité technologique sera, sans aucun doute, une des problématiques épineuses à résoudre afin de construire un réseau d'applications réparties se communiquant en toute transparence.
D'autre part, la sécurité est un pré requis crucial (Hubaux, Raya et Papadimitratos 2006) pour le déploiement des réseaux véhiculaires et plusieurs solutions ont été proposées. Le contrôle d'accès aux ressources est l'un des aspects de sécurité ayant attiré l'attention des chercheurs. Ainsi, des implémentations variées et hétérogènes ont été développées à base de certificats X.509 (Lu, et al. 2008)(Hubaux, Raya et Papadimitratos 2006), de tickets Kerberos, du couple Username/Password (Guo, Ngoh et Teo 2009), etc.

Problématique:

Avant de présenter notre problématique, imaginons le cas d'un véhicule dans un pays "A" accédant aux différents services en utilisant la technologie des certificats X.509. Ce même véhicule voyage dans un autre pays "B" et veut consommer les mêmes services. Or, dans le pays "B", la technologie d'authentifcation est plutôt à base de tickets Kerberos. Ainsi, notre problématique consiste à trouver une solution pour fédérer les différentes techniques de contrôle d'accès aux ressources des réseaux véhiculaires. La solution doit être ouverte et agile afin d'intégrer les futures techniques non encore existantes.
Les Web Services sont une technologie à la mode afin de construire des architecture SOA. Ce succès est dû à la richesse des spécifications WS-* définies et touchant à plusiueurs aspects: orchestration, transactions, sécurité, etc.
WS-Federation est l'une des spécifications WS-* liées à la sécurité ayant pour objectif de fédérer les protocoles de contrôles d'accès aux ressources. D'où l'idée d'appliquer cette spécification dans les réseaux véhiculaires.
Néanmoins, une deuxième problématique se fait montrer: comment intégrer des applications véhiculaires hétérogènes technologiquement dans une architecture SOA et à base de Web Service afin de pouvoir appliquer WS-Federation?

Démarches:

Dans un premier temps notre objectif est de concevoir une technique d'intégration des applications véhiculaires dans une architecture SOA en utilisant le pattern du bus. Notre source d'inspiration est le bus d'entreprise (ESB) (Erl, SOA design Patterns 2009); ce qui a engendré une solution adaptée aux réseaux véhiculaires baptisée VSB (Vehicular Service Bus).
Grâce à VSB, il nous est possible de considérer un réseau véhiculaire comme un ensemble de noeuds mobiles et fixes consommant et fournissant à la fois des services.
La seconde étape consiste à appliquer la spécification WS-Federation sur notre architecture du VSB.

Structure du papier:

Ce papier présente la solution développée en trois phases. La première phase sera consacrée à l'architecture communicationnelle à base du VSB. La deuxième phase décortiquera plutôt le module de la sécurité fédérée. Et on clôturera le document par une présentation de quelques résultats de notre travail.

2 Architecture Communicationnelle

Dans cette partie, nous présenterons notre solution VSB qui est à base du pattern de l'ESB. Le VSB est conçu à base de la technologie WCF[1] (Nadeem et Hamayun 2010) qui grâce à son modèle, permet d'intégrer des applications LOB (Line Of Business) dans une architecture orientée services.

2.1 Architecture de VSB

Afin d'intégrer des applications hétérogènes et de les exposer et consommer comme services, nous avons conçu un bus de communication embarqué au niveau de chaque noeud du réseau véhiculaire. Le VSB se base sur le pattern ESB et s'inspire dans son architecture de la plateforme Biztalk[2] de Microsoft (Dunphy, et al. 2009).
Le schéma (cf. Fig. 1) présente les différents composants du VSB (ainsi qu'un exemple du cycle de vie d'un message d'alerte dans le bus) en définissant les éléments suivants:

i. Le moteur d'exécution VSB: il s'agit de l'environnement d'exécution du bus contenant une couche moteurs exposés comme services WCF (transformation des formats de message [à base de XSLT par exemple], routage des messages, règles métiers, orchestration [à base de Workflows], moteur Pub/Sub pour les notifications et événements, résolution des

[1] Windows Communication Foundation.
[2] http://www.microsoft.com/france/serveur/biztalk/default.aspx

endpoints (comme résoudre un URN UDDI 3.0 en une URI). Le runtime VSB est aussi doté d'un environnement d'hébergement pour les différents services WCF, ainsi que des annuaires pour référencer les services tels les annuaires UDDI 3.0.

ii. Les adaptateurs : qui sont sous formes d'applications façades du VSB et exposées comme Web Services en entrée (ports d'entrée) et en sortie (ports de sortie), et qui ont pour rôle d'adapter les protocoles de transports, les encodages et les formats en adéquation avec le runtime en entrée et vice versa en sortie avec le système externe.

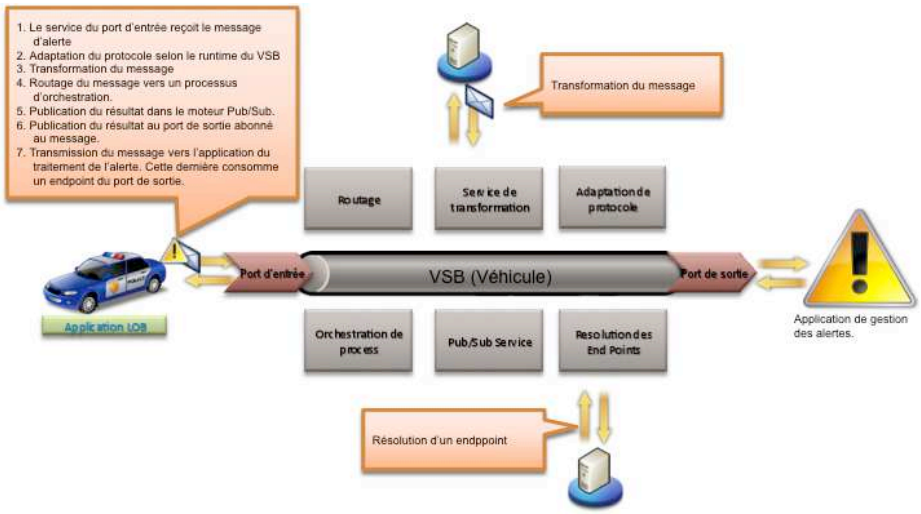

Fig. 1. Composants du VSB.

2.2 Architecture décentralicée du VSB

La figure (cf. Fig. 2) décrit un exemple d'un cycle de vie d'un message d'alerte correspondant à l'état de la chaussée glissante généré par une application LOB d'un véhicule (1). Ce message d'alerte passé par le bus du service routier (2) est retransmis par la suite à tous les véhicules concernés (3). Le retransmission se fait sur la base du contenu du message après un certain traitement dans le bus du service routier. Enfin, ce message passé par le bus de chaque véhicules pour d'éventuels traitements afin qu'il puisse être géré par l'application adéquate. Le message peut être même routé vers un dispositif mobile du conducteur pour d'avantages d'interactions.

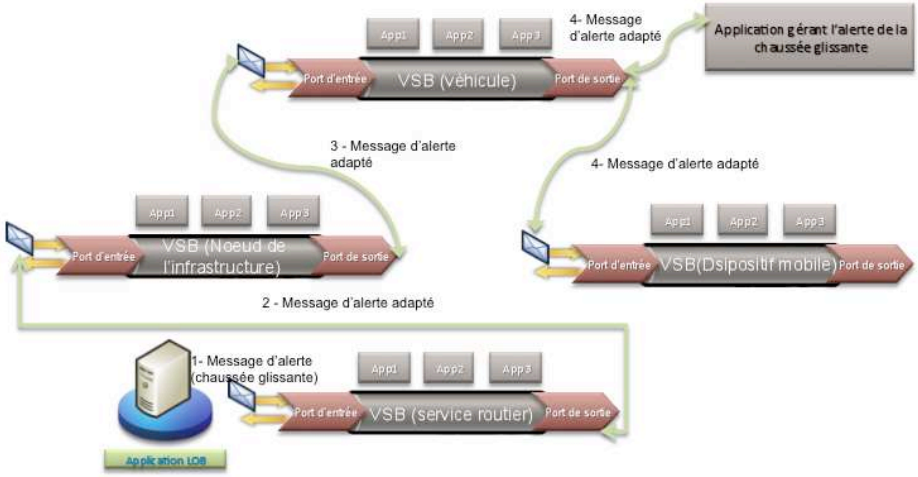

Fig. 2. Architecture décentralisée du VSB.

3 Architecture de la sécurité fédérée

Dans cette partie, nous présenterons l'architecture de notre module de sécurité fédérée regroupant authentification et autorisation à base de revendications formant ainsi les jetons de sécurité.

3.1 Intérêts du contrôle d'accès à base de revendications

Dans ce type d'autorisation, le contrôle d'accès aux opérations du service appelé se fait sur la base du contenu (les revendications) des jetons de sécurité fournis par les appelants d'un service. Ainsi, les permissions d'accès sont définies dynamiquement dans les politiques de sécurité.

Ce modèle offre plus de flexibilité que les autres modèles (orientés rôles et ressources) pour les qualités suivantes (Erl, SOA Design Patterns 2009):

i. Découplage du mécanisme d'authentification des applications et services. En effet, si un client consomme des services où chacun implémente une technique d'authentification différente, il devra les implémenter toutes. Néanmoins, le modèle orienté jetons est abstrait et supporte les mécanismes d'authentifications existants et futurs.
ii. Support de tous les type de crédentiels fournis. En effet, il est possible de sérialiser tous les types de crédentiels.
iii. Possibilité de transformer les jetons d'un format vers un autre.
iv. Orientation politique à base de revendications. En effet, le modèle des jetons se base sur les revendications définies dans la politique de sécurité du service en remplacement des rôles.

3.2 Architecture du contrôle d'accès fédéré

Dans cette partie, nous proposons l'architecture du contrôle d'accès fédéré appliquée à notre plateforme VSB. Cette architecture est basée sur le scénario actif de fédération de la spécification WS-Federation (WS-Federation Active Requestor Profile). Le choix de ce profil est étayé par la souplesse de l'hébergement des services dans des processus au lieu des serveurs Web. De plus, le profil actif s'applique sur les clients dits « intelligents » qui correspondent au cas des applications véhiculaires et qui ne sont pas en conséquence passifs comme les navigateurs.

Une architecture fédérée est composée généralement de trois entités :

i. Le client : dans notre cas, il s'agit de tout nœud du réseau véhiculaire voulant consommer un service exposé dans le même réseau, sur Internet ou sur le Cloud.
ii. Le service : il s'agit du service exposant des fonctionnalités à consommer. Imaginons le cas d'un service de régulation des feux de circulation.
iii. Le service de génération des jetons de sécurité (STS(Security Token Service)) : ce service fournit des jetons de sécurité requis par le service à consommer grâce à WS-Trust. Une relation de confiance lie le STS aux services à consommer.

La figure suivante (cf. Fig.3) schématise l'architecture fédérée d'un exemple de deux véhicules (oridinaire et de police) qui tentent de modifier l'état du service d'un feu de circulation.

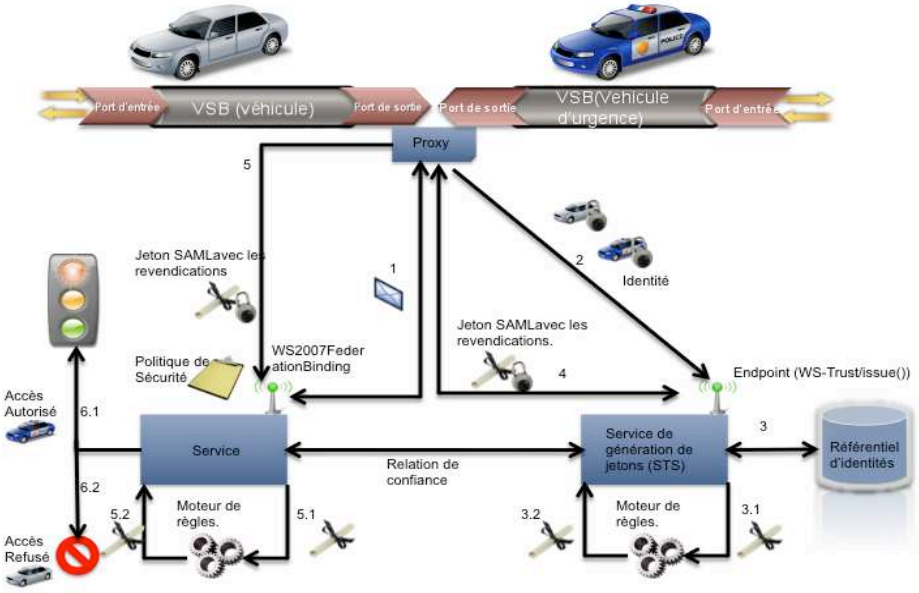

Fig. 3. Architecture du contrôle d'accès fédéré.

Un véhicule étant équipé d'un VSB est doté d'une application en sortie (par exemple) et développe un proxy qui se charge de consommer le service en question. Ainsi, le profil actif de la fédération suit les étapes suivantes : le proxy du véhicule découvre le service grâce à WS-Discovery et récupère la politique de sécurité exposée sur un endpoint fédéré grâce à WS-MetadataExchange et WS-SecurityPolicy. Cette politique exprime les revendications à inclure dans le jeton de sécurité qui doit être délivré par un ou des STS déterminés (1). Ensuite, l'identité du véhicule qui ne correspond pas aux revendications est envoyée au service de génération de jetons de sécurité (STS) ainsi qu'un message de requête de jeton de sécurité conforme aux revendications (RST(Request Security Token)) (2). Il est à noter que le service de génération des jetons de sécurité implémente la spécification WS-Trust. Le STS authentifie ensuite l'identité du véhicule et fournit le jeton requis (3) et génère, en conséquence, la réponse à la requête RST sous forme d'une structure de données nommée RSTR (Request Security Token Response) (4) et contenant les éléments suivants:

i. Le jeton sérialisé, signé par le STS et contenant les evendications requises ainsi que la clé de preuve chiffrée avec la clé publique du service à consommer. Le jeton est aussi chiffré avec la clé publique du service à consommer.
ii. La clé de preuve pour l'application consommatrice.

L'envoi de la RSTR est protégé par la politique exprimée dans le Binding du endpoint du service de génération des jetons. L'application consommatrice du véhicule récupère la clé de preuve de la RSTR et signe le message envoyé vers le service à consommer avec la clé de prevue (5). Ainsi, le service à consommer authentifie le jeton envoyé avec la clé publique du STS, récupère la clé de preuve du jeton avec sa clé privée, et authentifie en conséquence l'application consommatrice comme source de la demande du jeton. Enfin, le service à consommer prend une décision d'accès en termes d'autorisations fournies par l'application consommatrice (6).

4 Evaluation des performances

Afin de valider notre travail, nous avons développé un POC[3] applicatif résolvant le problème de gestion des intersections. Dans notre POC, on permet juste aux véhicules d'urgence de modifier l'état des feux de circulation en toute transparence par rapport aux technologies d'authentification des véhicules.
L'architecture de notre application en termes de composants est décrite par la figure suivante:

[3] Proof Of Concept

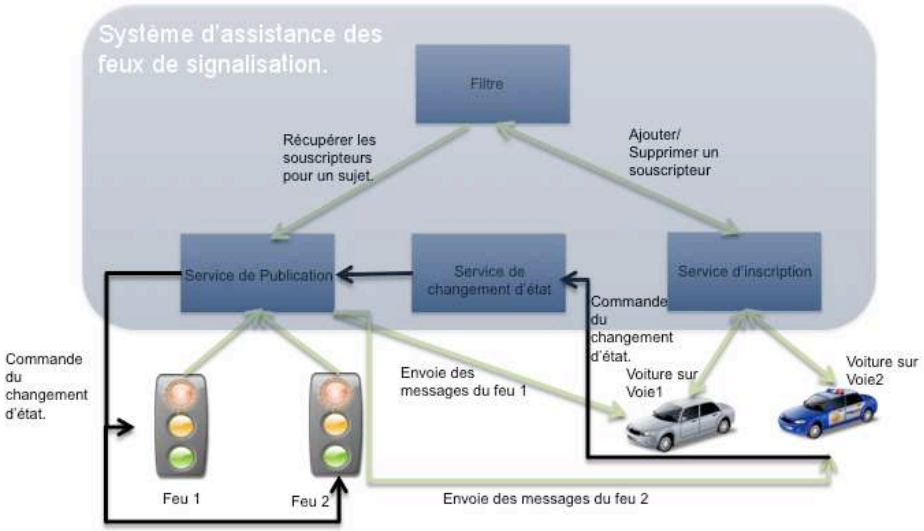

Fig. 4. Composants du POC.

Il est à noter que nous avons supposé que l'architecture est déjà orientée services. Ainsi, notre POC avait pour objectif de prouver la faisabilité de l'architecture orientée services dans le cas des réseaux véhiculaires ainsi que tester quelques paramètres importants comme le temps consommé dans les échanges des messages.
Nous avons utilisé l'outil Visual Studio 2010 avec le langage C#, les Frameworks WCF et WIF[4] afin de développer notre POC.
Les différents processus de l'application (processus des véhicules, des feux et services du système d'assistance des feux) ont été testés sur une machine ayant les caractéristiques suivantes :

i. Modèle : MacBook de Apple.
ii. OS : Windows 7.
iii. RAM : 4Go DDR2 cadencée à 667 MHz.
iv. Processeur : Intel Core 2 Duo T7250 cadencé à 2.16GHz.

Nous avons effectué des tests de performance sur une application véhiculaire consommant le service d'assistance des feux pour un scénario d'authentification directe et mutuelle à base de certificats[5]. A chaque scénario, nous modifions le binding correspondant au service et à l'application (protocole de transport, mode de sécurité, algorithmes de cryptographie, etc.). Après authentifications, nous enregistrons le temps moyen entre la date d'envoi du message et celle de son interprétation au niveau applicatif pendant tout un cycle du feu (vert, orange, rouge).

[4] Windows Identity Framework.
[5] Le choix des certificats X.509 est justifié par le fait qu'il soit le mode d'authentification le plus répandu dans la littérature de la sécurité des réseaux véhiculaires.

Les protocoles de transport sujets de nos tests sont TCP qui a été utilisé dans (L. SICHITIU et KIHL 2008) et HTTP similaire au protocole défini dans (Nadeem, et al. 2007).

Nous avons enregistré les temps suivants :

Tableau 1 Temps de Traitement des Messages.

Binding	Temps moyen
HTTP sans sécurité	6.73 ms.
HTTP avec sécurité Mode: Message. Algorithme: AES 256	10.73 ms.
HTTP avec sécurité. Mode: Message. Algorithme: AES 256 SHA 256 RSA 15 [6]	10.85 ms.
TCP sans sécurité	1.84 ms.
TCP avec sécurité Mode: Transport.	3.59 ms.
TCP avec sécurité. Mode: Message. Algorithme: AES 256	5.59 ms.
TCP avec sécurité. Mode: Message. Algorithme: AES 256 SHA 256 RSA 15	6.24 ms.
TCP avec sécurité. Mode: Message. Algorithme: 3 DES	7.65 ms.
TCP avec sécurité. Mode: Message. Algorithme: 3DES SHA 256 RSA 15	8.07 ms.

Discussions. Le temps réel est, peut être, l'élément le plus important dans les communications véhiculaires. D'après (C. L., et al. 2007), la latence autorisée pour le signal d'alerte en cas de violation du feu rouge est de 100ms.
D'après nos tests, le temps le plus défavorable atteint juste le dixième de la latence. Ce qui est fort encourageant. Néanmoins, il faut s'attendre à ce que les temps enregistrés dans des conditions réelles soient plus grands que ceux enregistrés en simulation.
Tandis que pour les durées des authentifications fédérées, le premier temps est évalué à près de 1.4 s. Néanmoins, grâce à WS-SecureConversation, une session sécurisée est maintenue entre les entités communicantes ce qui évite des ré authentifications.

[6] L'algorithme RSA (asymétrique) pour générer les clés symétriques utilisées avec AES pour le chiffrement tandis que la signature est à base de SHA 256.

5 Conclusion

Notre travail entre dans le cadre de la sécurité des IVC et a essayé de répondre à une double problématique : comment intégrer les applications véhiculaires dans une SOA? Et comment fédérer différentes technologies de contrôle d'accès ou d'authentification?
Afin de répondre à la première problématique, nous avons développé une solution d'intégration des applications véhiculaires en suivant le pattern du bus. Notre source d'inspiration était la technologie du bus d'entreprise (ESB); ce qui a donné naissance à un bus embarqué dans tout nœud du réseau véhiculaire et que nous avons baptisé VSB (Vehicular Service Bus).
La réponse à la deuxième problématique était l'application de la spécification WS-Federation et bien d'autres spécifications liées à la sécurité comme essentiellement WS-Security. Cette solution est très faisable grâce à l'architecture du VSB à base de Web Services supportant ainsi toutes spécifications WS-*.

Références

- C. L., Robinson, Caveney D., Baliga G., Laberteaux K., et Kumar P. R. «Efficient Message Composition and Coding for Cooperative Vehicular Safety Applications.» *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY* 56 (2007): 3244-3255.
- Dunphy, George, Sergei Moukhnitski, Stephen Kaufman, Peter Kelcey, David Peterson, et Harold Campos. *Pro Biztalk 2009*. aPress, 2009.
- Erl, Thomas. *SOA design Patterns*. PRENTICE HALL, 2009.
- Guo, Huaqun, Lek Heng Ngoh, et Josef Chee Ming Teo. *An Anonymous DoS-Resistant Password-based Authentication, Key Exchange and Pseudonym Delivery Protocol for Vehicular Networks*. Singapore: International Conference on Advanced Information Networking and Applications, 2009.
- Hubaux, Jean Pierre, Maxim Raya, et Panos Papadimitratos. «SECURING VEHICULAR COMMUNICATIONS.» *Journal of Computer Security* 15, n° 1 (Janvier 2007): 39-68.
- L. Sichitiu, Mihail, et Maria Kihl. «Inter-Vehicle Communication Systems: A Survey.» *IEEE Communications Surveys and Tutorials* (IEEE Communications) 10 (2008): 88-105.
- Lu, Rongxing, Xiadong Lin, Chenxi Zhang, Haojin Zhu, Pin-Han Ho, et Xuemin Shen. *Security in Vehicular Ad Hoc Networks*. IEEE, 2008.
- Nadeem, Ahmed, et Muhammed Hamayun. *Performance Evaluation of Windows Communication Foundation's Interoperability*. Thèse de Master, Ronnby, Suède: Blekinge Institute of Technology, 2010.
- Raya, Maxim, et Jean-Pierre Hubaux. «Securing vehicular ad-hoc networks.» *J. Comput. Secur.* (IOS Press) 15 (2007): 39--68.