Fusion distribuée évidentielle pour la détection d'attaques sybil dans un réseau de véhicules

Distributed evidential fusion for detecting sybil attacks in VANETs

N. El Zoghby V. Cherfaoui B. Ducourthial T. Denoeux
Heudiasyc UMR CNRS 7253, Université de Technologie de Compiègne, France

nicole.el-zoghby@hds.utc.fr
veronique.cherfaoui@hds.utc.fr
bertrand.ducourthial@hds.utc.fr
thierry.denoeux@hds.utc.fr

Résumé :
L'attaque "Sybil" peut avoir lieu dans un réseau de véhicules et peut affecter le fonctionnement de ce réseau. Nous présentons une méthode de fusion de données distribuées appliquée à la détection d'attaques "Sybil" dans un réseau de véhicules. Il s'agit de quantifier, dans le cadre des fonctions de croyance, la confiance dans un nœud du réseau par échange des messages. Chaque nœud a un avis sur l'ensemble de son voisinage. Le but est de fusionner les informations transmises afin d'arriver à un état de connaissance globale dans le réseau. Pour valider cette approche, des simulations ont été faites sur différentes configurations du réseau.

Mots-clés :
Fusion de données, attaque sybil, réseau de véhicules, fonctions de croyance

Abstract:
Sybil attacks can occur in a Vehicular Ad hoc Network "VANET" and can affect its functionnality for the benefit of the attacker. This paper presents a method based on distributed data fusion in order to detect such attacks in VANETs. By exchanging messages, nodes can quantify confidence over the network in the context of belief functions. Each node have an opinion on its neighborhood. The aim is to combine the transmitted data in order to build distributed confidence over the network. In order to validate this approach, simulations were performed on different network configurations.

Keywords:
Data fusion, Sybil attack, VANET, belief functions

1 Introduction

Les véhicules seront bientôt capables de communiquer entre eux afin d'échanger des informations importantes pour la sécurité et le confort du conducteur. On se place dans le contexte des VANETs "Vehicular Ad hoc NETworks" où les véhicules sont considérés comme les nœuds d'un réseau ad hoc sans fil. Ces réseaux sont vulnérables aux différentes attaques comme l'intrusion. Échanger des données au sein d'un tel réseau implique l'introduction de la notion de confiance. Ainsi chaque nœud doit avoir confiance dans les autres nœuds ou dans les données reçues avant d'utiliser les informations échangées dans d'autres applications. En diffusant les messages, les nœuds découvrent leur voisinage. Leurs voisins peuvent être des vrais ou faux nœuds, et ils peuvent aussi être des attaquants. Des travaux de recherche récents visent à trouver des solutions pour de tels problèmes. Certains travaux portent sur les mécanismes de réputation ([21],[1],[10]) et l'évaluation de la confiance ([17],[18]) dans le but de prendre en compte la confiance dans les sources de l'information (les nœuds). D'autres s'intéressent à l'agrégation des données sans prendre en compte la confiance dans les sources [2][3][11][14].
On propose dans cet article une méthode de fusion de données dans un système distribué afin d'établir la confiance au sein du réseau. Les nœuds diffusent leurs avis qui sont réutilisés à la réception pour évaluer d'autres nœuds. Comme l'avis local de chaque nœud est incertain et incomplet, l'utilisation des fonctions de croyance pour évaluer les messages reçus semble appropriée. La fusion de la connaissance locale

du nœud avec les messages reçus est faite en utilisant la règle de Dempster. Des cycles de disséminations des données pouvant avoir lieu au sein du réseau, la même information ne doit pas être combinée plusieurs fois comme si elle provenait de sources indépendantes ([15],[12]). Dans une telle situation, la règle prudente est utilisée [5].

On cherche à quantifier la confiance dans un nœud du réseau afin de détecter l'attaque sybil dans un réseau de véhicules. L'attaque "Sybil" est le cas où un nœud *malveillant* est capable de revendiquer des entités multiples appelées *nœuds sybil* ou *faux nœuds* [6]. Ainsi, en se faisant passer pour ces différentes identités, le nœud malveillant pourra compromettre plus facilement le fonctionnement général du réseau de véhicules. Différentes techniques ont été développées pour détecter les faux nœuds dans les VANETs. Gole et al. [8] ont proposé une méthode basée sur un principe de parcimonie qui consiste à trouver la meilleure explication pour les données corrompues. Les véhicules distinguent leurs voisins en utilisant des caméras ou en échangeant des messages dans le spectre infrarouge. La technique décrite par Xiao et al. [19] pour détecter les nœuds sybils est basée sur l'analyse de la puissance du signal en utilisant comme support les infrastructures routières. Yan et al. [20] utilisent le radar pour détecter les voisins et vérifier les positions annoncées. Piro et al. [13] présentent une détection passive de l'attaque sybil en utilisant des observateurs (unique ou multiples). Étant données la dynamique du réseau de véhicules, le nombre des véhicules et la difficulté d'avoir un accès permanent aux infrastructures, les outils classiques comme le PKI (Public key infrastructure) ne sont pas adaptés. Comme il a été montré dans [9], par une simple comparaison de la puissance du signal reçu, la moitié des véhicules peut détecter les faux nœuds et il est prévu que les techniques coopératives permettent de diminuer le nombre des véhicules malicieux. Un algorithme coopératif entre les véhicules permettrait d'éviter les méthodes cryptographiques.

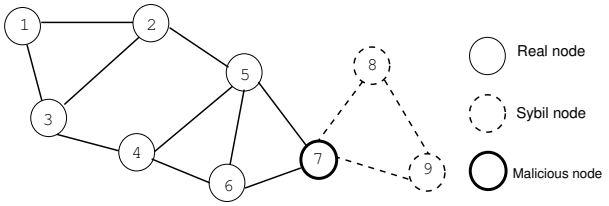

Figure 1 – La configuration du réseau

Dans cet article, nous développons une technique de fusion distribuée basée sur la théorie de fonctions de croyance. On décrit le système et la représentation de la confiance par des fonctions de masse. On présente l'approche de la fusion distribuée et l'algorithme proposé. La validation de cette approche est faite par simulations.

2 Fusion de données distribuées

On considère un réseau constitué de nœuds échangeant des messages. Il peut être représenté par un graphe direct $G = (V, E)$ où V est l'ensemble des nœuds $V = \{v_1, v_2, ..., v_n\}$ et E est l'ensemble d'arrêtes. Les voisins de chaque nœud sont représentés par $\Gamma(v) = \{v_j \in V, \{v_i, v_j\} \in E\}$. Pour simplifier, on suppose que chaque nœud connaît $n = |V|$. La figure 1 montre un exemple de configuration d'un tel réseau. Chaque nœud envoie périodiquement des *messages réguliers* contenant son identité et sa position géographique. Le nœud malveillant envoie des *messages réguliers* et des *faux messages* qui contiennent une fausse identité et une fausse position. En recevant les faux messages, les autres nœuds sont leurrés et considèrent des nœuds non existants, appelé *nœuds sybil* ou *faux nœuds*. On suppose ici qu'un seul nœud malveillant crée différents nœuds sybil. Tous les nœuds utilisent le même système de transmission (même antenne et même puissance). La topologie du réseau est donnée par la gamme de transmission radio des nœuds (unit disk graph). Le message contient aussi la confiance de l'émetteur dans les nœuds du réseau. On propose une méthodologie de fusion de données

pour combiner les données échangées dans un réseau ad hoc mobile dans le but de quantifier la confiance dans un nœud du réseau.

2.1 Représentation de la confiance par une fonction de masse

Chaque nœud est capable d'attribuer une confiance en chacun des autres nœuds du réseau. Cette confiance est représentée par une masse noté m, répartie sur le cadre de discernement $\Omega = \{0, 1\}$ où 0 représente le Faux Nœud et 1 représente le Vrai Nœud. Soit m_{ij} la masse correspondante qui représente l'avis d'un nœud v_i sur le nœud v_j. m_{ij} est définie sur Ω de la façon suivante :

$$\begin{array}{rcl} m_{ij}(\emptyset) & = & 0 \\ m_{ij}(0) & = & p_{ij} \\ m_{ij}(1) & = & q_{ij} \\ m_{ij}(\Omega) & = & 1 - p_{ij} - q_{ij}. \end{array} \quad (1)$$

2.2 Principe de l'approche

Le nœud v_k envoie au nœud v_i un message contenant son identité, ses coordonnées et son avis sur l'ensemble du réseau. A la réception, le nœud v_i établit, après avoir analysé la puissance du signal, une confiance directe de i sur k. Celle-ci est représentée par un vecteur de masse noté $m_{d_{ik}}$. Cette confiance est conservée dans une mémoire locale appelée *connaissance privée* ou *locale*.
Il faut noter que chaque nœud possède deux types de connaissance : *locale* et *publique*. La connaissance locale provient des mesures directes du nœud dans son voisinage. Elle est combinée avec la connaissance publique qui provient des autres nœuds pour mettre à jour la connaissance publique rediffusée à travers le réseau. Ce principe correspond à un système distribué [7]. La connaissance locale dépend seulement de la puissance du signal des messages et pas de leurs contenus : par conséquence, elle ne peut pas être influencée. En revanche, la connaissance publique est le résultat de la combinaison des contenus des messages et peut être influencée par l'avis des faux nœuds. Cette distinction permet de séparer ce qui provient des mesures directes de ce qui est calculé dans le réseau. La mémoire interne de chaque nœud est représentée par deux vecteurs de masse (tableau de $|V|$ cases initialisé à $m(\Omega)$ si $i \neq j$ et $m(1)$ si i=j) :

$$\begin{array}{rcl} C_{locale_i}(t) & = & [m_{l_{ij}}^{(t)}] \\ C_{publique_i}(t) & = & [m_{p_{ij}}^{(t)}]. \end{array} \quad (2)$$

2.3 Algorithme de la fusion distribuée

L'algorithme 1 présente les différentes étapes du traitement effectué à la réception du message. Cet algorithme est expliqué dans les sections suivantes.

Algorithme 1 : Traitement du message reçu par le nœud v_i

Données : message de v_k à v_i, Puissance du signal P, message contient $m_{p_{kj}} \forall j$
Résultat : C_{locale_i}=[$m_{l_{ij}}^{(t)}$] et $C_{publique_i}$=[$m_{p_{ij}}^{(t)}$]$\forall j \in V$
$m_{d_{ik}}^{(t)} \leftarrow ConfianceDirecte(message, P)$
$m_{l_{ik}}^{(t)} \leftarrow MiseajourClocale(m_{l_{ik}}^{(t-1)}, m_{d_{ik}}^{(t)})$
$m_{p_{ik}}^{(t)} \leftarrow MiseajourCpublique(m_{p_{ik}}^{(t-1)}, m_{l_{ik}}^{(t)})$
$\alpha \leftarrow FacteurAffaiblissement(m_{l_{ik}})$
for *chaque nœud j ∈ V tel que j≠ i , j≠ k* **do**
  ${}^{\alpha}m_{p_{kj}}^{(t)} \leftarrow AffaibliCemetteur(\alpha, {m_{pkj}}^{(t)}, m_{\Omega}^{(t)})$
  $m_{p_{ij}}^{(t)} \leftarrow MiseajourCpublique(m_{p_{ij}}^{(t-1)}, {}^{\alpha}m_{p_{kj}}^{(t)})$

2.4 Fusion distribuée

A la réception du message, le nœud v_i calcule la confiance directe $m_{d_{ik}}$. Cette confiance est indépendante des messages précédents et n'est pas le résultat d'autres combinaisons. On l'utilise pour mettre à jour la connaissance locale du récepteur sur l'émetteur avec la règle de Dempster [4]. La fonction MiseajourCLocale($m_{l_{ik}}^{(t-1)}, m_{d_{ik}}^{(t)}$) est calculée de la manière suivante :

$$m_{l_{ik}}^{(t)} = m_{l_{ik}}^{(t-1)} \oplus m_{d_{ik}}^{(t)}, \quad (3)$$

où $\oplus$ dénote la règle de Dempster. L'avis des autres nœuds est nécessaire sachant que les faux

nœuds peuvent falsifier l'avis de chaque nœud. En ce qui concerne l'avis de l'émetteur, étant donné que ce nœud n'est pas nécessairement fiable, on a choisi d'affaiblir ses connaissances avant de les combiner avec la connaissance interne du nœud. Le coefficient d'affaiblissement α est calculé en fonction de la connaissance locale $m_{l_{ik}}$ du nœud récepteur v_i sur le nœud émetteur v_k. Ce coefficient est égal à la plausibilité que le nœud émetteur soit non fiable :

$$\alpha = 1 - m_{l_{ik}}(1). \tag{4}$$

La connaissance de l'émetteur est affaiblie avec la fonction AffaibliCemetteur$(\alpha, m_{kj}^{(t)}, m_{\Omega}^{(t)})$ dans l'équation 5 :

$$^{\alpha}m_{p_{kj}}^{(t)} = (1-\alpha).m_{p_{kj}}^{(t)} + \alpha.m_{\Omega}^{(t)}. \tag{5}$$

Pour mettre à jour la connaissance publique du récepteur, on utilise la règle prudente [5]. Dans un système distribué, la même information peut être reçue et traitée différentes fois. En combinant l'information, il est utile d'utiliser une règle idempotente pour éviter de prendre en considération la même information plusieurs fois (data incest) comme si elle provenait de différentes sources indépendantes. La fonction MiseajourCpublique$(m_{p_{ij}}^{(t-1)}, ^{\alpha}m_{p_{kj}}^{(t)})$ permet de combiner la connaissance publique du récepteur avec celle de l'émetteur affaiblie comme suit :

$$m_{p_{ij}}^{(t)} = m_{p_{ij}}^{(t-1)} \textcircled{\wedge}^{\alpha} m_{p_{kj}}^{(t)}. \tag{6}$$

2.5 Confiance Directe

Différentes méthodes peuvent être utilisées pour calculer la confiance directe $m_{d_{ik}}$. On propose une méthode qui permet de convertir une mesure réelle en une fonction de masse. A chaque réception de message provenant d'un émetteur, nous supposons que le récepteur peut analyser le signal reçu et détecter les incohérences physiques. Il s'agit donc de mesurer la puissance du signal reçu et de calculer la puissance théorique à partir des coordonnées du nœud voisin. La puissance estimée μ est calculée selon la formule de Friis de la façon suivante :

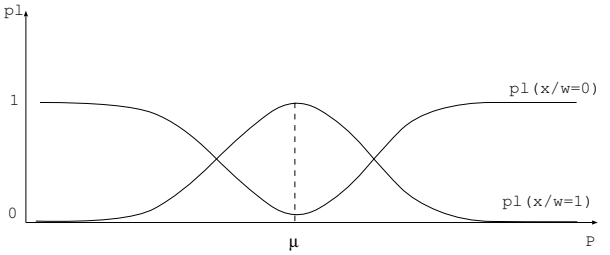

Figure 2 – Valeurs de la plausibilité de la puissance reçue pour les vrais ($\omega = 1$) et faux ($\omega = 0$) nœuds

$$\mu = P_e.G_{SR}.\frac{1}{d_{ik}^2} \tag{7}$$

où

- P_e est la puissance du signal émis. Cette puissance est une puissance standard définie selon les antennes utilisées ;
- $G_{SR} = \frac{G_e.G_r.\lambda^2}{16.\pi^2}$ est le gain, G_e et G_r étant les gains linéaires de l'antenne d'émission et de réception et λ la longueur d'ondes ;
- d_{ik} est la distance entre le nœud émetteur v_k et le nœud récepteur v_i.

La comparaison entre la puissance mesurée et la puissance estimée nous permet de détecter d'éventuels faux nœuds. On propose de calculer la plausibilité que la puissance du signal reçu P soit égale à x, sachant que le nœud émetteur est un vrai nœud $(w = 1)$ de la manière suivante :

$$pl(P = x/\omega = 1) = \frac{f(x/\omega = 1)}{\sup_{x' \in \mathbb{R}}(f(x'/\omega = 1)}\ , \tag{8}$$

où $f(x/\omega = 1)$ est une fonction de densité normale de moyenne μ et de variance σ dépendant de l'antenne du récepteur.

La plausibilité $pl(P = x/\omega = 0)$ est définie comme illustré sur la figure 2 : si les puissances estimée et théorique sont égales, on conserve la possibilité que l'émetteur soit un faux nœud. En effet, si l'émetteur est un faux nœud mais si sa position est proche de celle du nœud malveillant, la position estimée est approximativement égale à la puissance mesurée. Ce résultat peut influencer la détection des faux nœuds.

La confiance directe est calculée en utilisant le théorème de Bayes Généralisé [16]. Elle s'obtient par la formule suivante :

$$\begin{aligned} m_{d_{ik}}^{(t)} &= m^{\Omega}(./x) \\ &= m_0^{\Omega} \textcircled{\cap} \{0\}^{pl(x/w=1)} \textcircled{\cap} \{1\}^{pl(x/w=0)}, \end{aligned} \tag{9}$$

où $\textcircled{\cap}$ représente la règle de Dempster non normalisée et la notation $\{w\}^c$ représente la fonction de masse simple affectant la masse c à Ω et $1-c$ à $\{w\}$.

3 Résultats

Pour valider notre approche, l'algorithme 1 a été implémenté en Matlab. Les simulations ont été faites sur des réseaux statiques et dynamiques. Pour simplifier l'analyse, on a d'abord supposé que les nœuds du réseau sont statiques. On a effectué des simulations sur différentes configurations aléatoires du réseau. L'algorithme a ensuite été testé sur un réseau dynamique, où les nœuds sont en mouvement dans la même direction comme des véhicules se déplaçant sur une autoroute.

3.1 Implémentation

Pour bien comprendre le fonctionnement de l'algorithme, un exemple de réseau composé de six vrais nœuds, parmi lesquels un nœud malveillant crée trois faux nœuds est présenté. La puissance du signal de transmission P_e est égale à 600 mW et la portée de l'antenne est de l'ordre de 400m. On suppose que chaque émetteur envoie son *identité*, sa *position* et sa *connaissance publique*. Le récepteur utilise ces informations pour faire tous les calculs et vérifier si le nœud est vrai ou faux. Les simulations sont effectuées jusqu'à la convergence de l'algorithme. On considère que l'algorithme converge quand $|m_{ij}^{(t-1)} - m_{ij}^{(t)}| < \epsilon$, où ϵ est un seuil prédéfini. Les résultats de la simulation sont représentés par des matrices à niveau de gris. La figure 3 représente une initialisation de ces matrices. Les matrices de la première ligne

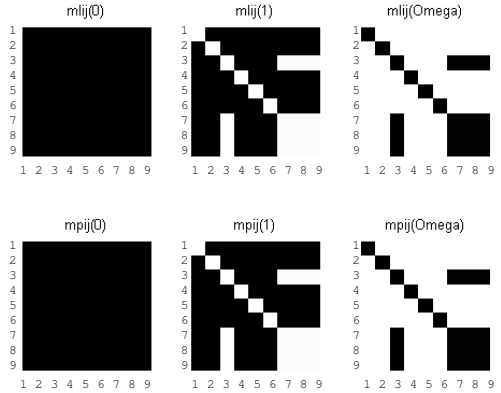

Figure 3 – Initialisation des matrices.

sont la connaissance locale et celles de la seconde ligne la connaissance publique. Les matrices de gauche représentent $A = 0$, du milieu $A = 1$ et de droite $A = \Omega$. Chaque case représente $m_{l_{ij}}(A)$ (respectivement, $m_{p_{ij}}(A)$). La couleur blanche correspond à une masse égale à 1 et la noire correspond à une masse égale à 0.

3.2 Réseau statique

La figure 4 présente un exemple de configuration du réseau où les nœuds sont statiques ; la figure 5 montre l'avancement de la simulation ainsi que le changement du niveau de gris à l'itération 25 ; la figure 6 montre le résultat de la simulation après 98 itérations. Le nœud malveillant 3 tente de convaincre les autres nœuds que les faux nœuds (7,8,9) sont des vrais nœuds. Les faux nœuds ont le même avis que le nœud malveillant. La première partie de la figure 5 représente la connaissance privée. Chaque nœud dispose seulement d'informations sur ses voisins. La seconde partie représente la connaissance publique. On voit que $m_{p_{ij}}(\{1\}) = 0$ pour $i = \{1,2,4,5,6\}$ et $j = \{7,8,9\}$, ce qui signifie que les vrais nœuds ont détecté que les nœuds $\{7,8,9\}$ sont des nœuds sybil. Pour vérifier la convergence de cet algorithme, des simulations ont été faites sur différentes configurations aléatoires du réseau en changeant

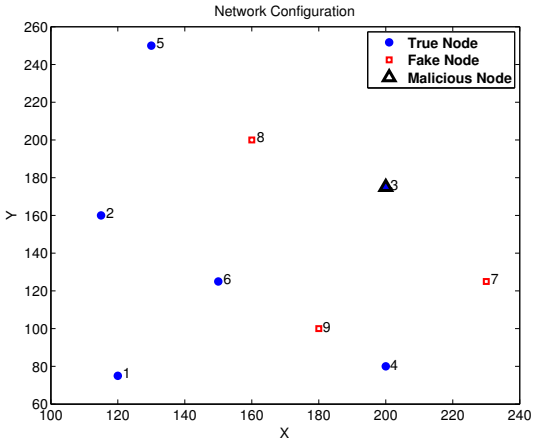

Figure 4 – Exemple de configuration du réseau.

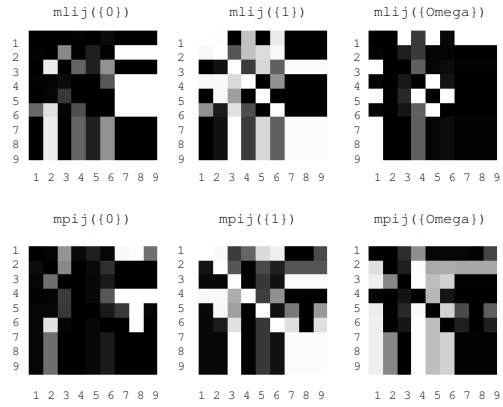

Figure 5 – Avancement d'une simulation pour la configuration de la Figure 4.

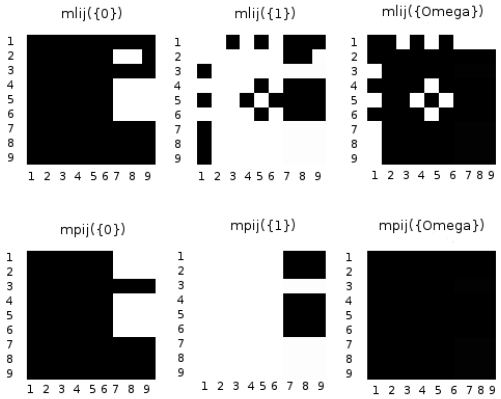

Figure 6 – Résultats d'une simulation pour la configuration de la Figure 4.

le nombre de faux nœuds. Le tableau 1 montre les résultats avec différentes proportions de faux nœuds. Une itération représente le traitement d'un message. L'algorithme prend plus de temps pour converger quand le nombre de faux nœuds augmente. Notre approche permet de détecter les nœuds sybil dans différentes configurations statiques.

Tableau 1 – Résultats dans différentes configurations de réseau.

Nœuds	Moyenne des nombres d'itérations[a]	Variance
VN=6 FN=3 [b]	207.05	7.86
VN=6 FN=4	227.55	6.89
VN=6 FN=5	255.8	6.33
VN=6 FN=6	304.7	7.55

[a] Ces résultats représentent la moyenne de 20 simulations.
[b] VN (Vrais nœuds) et FN (Faux nœuds).

3.3 Réseau dynamique

Les configurations statiques sont limitées, surtout dans le cas où le nœud malveillant n'est pas dans le voisinage des faux nœuds : dans

cette situation, les faux nœuds ne peuvent pas être détectés. Pour cela, nous avons simulé un scénario dynamique plus réaliste où les nœuds évoluent dans la même direction comme sur une autoroute. En se déplaçant, le voisinage de chaque nœud change. Ceci influence la connaissance privée car celle-ci dépend du voisinage. Grâce à la connaissance publique, chaque nœud a accès à l'information sur tout le réseau et peut quantifier la confiance. Le tableau 2 montre les résultats dans différentes configurations d'un réseau dynamique. Le nombre d'itérations jusqu'à la convergence change à chaque simulation, ce qui est dû au déplacement des nœuds et au changement du voisinage. Ces résultats préliminaires montrent que les vrais nœuds peuvent détecter les faux nœuds en se déplaçant sur une autoroute.

Tableau 2 – Résultats d'un réseau dynamique dans différentes configurations.

Nœuds	Moyenne des nombres d'itérations[a]	Variance
VN=6 FN=3 [b]	119.3	45.88
VN=6 FN=4	274.4	40.96
VN=6 FN=5	361.1	54.23
VN=6 FN=6	376.3	32.05

[a] Ces résultats représentent la moyenne de 10 simulations.
[b] VN (Vrais nœuds) et FN (Faux nœuds).

4 Conclusion

Les travaux décrits dans cet article développent une approche de fusion de données distribuées basée sur la théorie des fonctions de croyance dans le but de détecter l'attaque sybil dans un réseau de véhicules. La méthode utilise la règle de Dempster et la règle prudente pour combiner les informations et quantifier une confiance distribuée au sein du réseau. Les résultats sont prometteurs et démontrent la possibilité de détecter des faux nœuds dans le réseau. Nous travaillons actuellement sur des scénarios plus réalistes en utilisant un simulateur de réseaux ad hoc et sur la preuve de convergence de l'algorithme vu comme un opérateur auto stabilisant [7].
La méthode présentée dans cet article calcule la confiance dans un nœud sans prendre en considération le contenu des messages échangés dans le réseau. Une nouvelle approche qui prend en considération la confiance dans un nœud et dans l'information est également en cours de développement. Les résultats seront présentés dans des futures publications.

Références

[1] M.P. Singh B. Yu. An evidential model of distributed reputation management. In *First international Joint Conference on Autonomous Agents and Multi-Agents Systems, ACM Press*, pages 294–301, Bologna, Italy, 2002.

[2] T. M. Chen and V. Venkataramanan. Dempster-shafer theory for intrusion detection in ad hoc networks. In *IEEE Internet Computing*, volume 9, pages 35–41, 2005.

[3] V. Cherfaoui, T. Denoeux, and Z. L. Cherfi. Distributed data fusion : application to confidence management in vehicular networks. In *11th Int. Conf. on Information Fusion*, pages 846–853, Germany, 2008.

[4] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38 :325–339, 1967.

[5] T. Denoeux. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artificial Intelligence*, 172 :234–264, 2008.

[6] J.R Douceur. The sybil attack. In *the International Workshop on Peer to Peer Systems*, pages 251–260, Cambridge, MA, USA, 2002.

[7] B. Ducourthial. r-semi-groups : A generic approach for designing stabilizing silent

tasks. In *Self-Stabilizing Systems*, pages 281–295, 2007.

[8] P. Golle, D. Greene, and J. Staddon. Detecting and correcting malicious data in vanets. In *1st ACM Workshop on Vehicular Ad hoc Networks (VANET)*, pages 29–37, New York, NY, USA, 2004.

[9] G. Guette and B. Ducourthial. On the sybil attack detection in vanet. In *International Workshop on Mobile Vehicular Networks (MoveNet 2007), co-located with IEEE MASS 2007*, Pisa, October 2007.

[10] J. Liu and V. Issarny. Enhanced reputation mechanism for mobile ad hoc networks. In *2nd International Conference on Trust Management*, pages 48–62, Oxford, UK, 2004.

[11] C. Lochert, B. Scheuermann, and M. Mauve. Probabilistic aggregation for data dissemination in vanets. In *4th ACM international Workshop on Vehicular Ad Hoc Networks*, pages 1–8, Montréal, QC, Canada, 2007.

[12] H.B. Mitchell. *Multisensor Data Fusion : An introduction*. Springer, 2007.

[13] C. Piro, C. Shields, and B.N Levine. Detecting the sybil attack in mobile ad hoc networks. In *IEEE/ACM Intl Conf on Security and privacy in Communication Networks (SecureComm)*, pages 1–11, August 2006.

[14] M. Raya, P. Papadimitratos, V. D. Gligor, and J p. Hubaux. On data-centric trust establishment in ephemeral ad hoc networks. In *the 28th IEEE conference on Computer Communications (INFOCOM)*, pages 1238–1246, Phoenix, AZ., USA, April 2008.

[15] R. J. Evans S. Mclaughlin, V. Krishnamurthy. Bayesian network model for data incest in a distributed sensor network. In *the 7 th International Conference on Information Fusion*, volume 1, Stockholm, Sweden, 2004.

[16] Ph. Smets. Belief functions : the disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of Approximate Reasoning*, 9 :1–35, 1993.

[17] G. Theodorakopoulos and J. S. Baras. Trust evaluation in ad-hoc networks. In *ACM Workshop Wireless Security*, pages 1–10, Philadelphia, PA, USA, 2004.

[18] J. Wang and H j. Sun. A new evidential trust model for open communities. *Computer Standards & Interfaces*, 31 :994–1001, 2009.

[19] B. Xiao, B.Yu, and C.Gao. Detection and localization of sybil nodes in vanets. In *the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks*, pages 1–8, Los Angeles, CA,USA, 2006.

[20] G. Yan, G. Choudhary, M. Weigle, and S. Olariu. Providing vanet security through active position detection. *Computer Communications : Special Issue on Mobility Protocols for ITS/ VANET*, 31(12) :2883–2897, 2008.

[21] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14 :881–907, 2000.