

Anonymat dans les communautés de confiance

Olivier Heen (olivier.heen@irisa.fr)*
Gilles Guette (gilles.guette@irisa.fr)*
Thomas Genet (thomas.genet@irisa.fr)*

Résumé :

De plus en plus de terminaux mobiles utilisent des communications sans fil afin de s'annoncer ou d'envoyer et de récupérer de l'information. L'échange de ces données permet aux terminaux de s'organiser en réseau ad hoc mobile, c'est-à-dire de créer un ensemble de terminaux communiquant et œuvrant ensemble pour maintenir les communications. Il est parfois nécessaire de créer au sein de cet ensemble des communautés sécurisées de terminaux permettant ainsi d'échanger des données de manière plus sûre.

De nombreuses solutions existent alors pour permettre à un nœud de joindre une communauté sécurisée. Dans ce cas, l'authentification des nœuds n'est généralement pas indispensable : il suffit de mettre en œuvre des relations de confiance, pour qu'un nœud puisse détecter en qui il a confiance dans la communauté et communiquer en conséquence.

Une des question qui se pose alors est : peut-on mettre en œuvre cette opération de détection de manière totalement anonyme (nul ne peut savoir quels nœuds sont impliqués) ou même de manière non-observable (nul ne peut savoir si l'opération a lieu) ? En effet, une des contraintes majeures à prendre en compte dans ce genre de contexte est le respect de la vie privée.

Dans ce papier, nous exhibons une solution simple autorisant la détection de la confiance de manière anonyme vis-à-vis de tout type d'attaquant et de manière non-observable vis-à-vis de certains attaquants passifs. La non-observabilité n'est pas garantie cependant contre des attaquants actifs ou ayant de fortes capacités de caractérisation de trafic.

Ces travaux trouvent naturellement des applications dans les réseaux de véhicules (VANET) pour lesquels l'anonymat et la non-observabilité sont des propriétés essentielles [FFBA07, GFL⁺07] et pour lesquels les attaques actives sont difficilement réalisables.

Mots Clés : Non-observabilité, Anonymat, communautés de confiance, MANET.

1 Introduction

L'usage d'équipements individuels communicants se généralise : téléphones mobiles, assistants personnels, navigateurs GPS dans les véhicules, *etc.* De tels équipements peuvent de plus en plus facilement s'organiser en réseau ad hoc mobiles (aussi appelés MANET pour *Mobile Ad hoc NETWORK*). Dans un MANET, les équipements sont appelés les nœuds du réseau et peuvent joindre ou quitter le réseau en fonction de leurs capacités de communication. Dans un modèle de pure communication ad hoc, les nœuds n'ont pas d'accès à une infrastructure fixe du type borne, point d'accès ou serveur Internet et cela pas même de manière opportuniste. Les nœuds sont alors dans l'obligation de collaborer entre eux pour participer au maintien du réseau.

* IRISA, Campus Universitaire de Beaulieu, 35042 Rennes, France

La notion de *communauté de confiance* apparaît naturellement dans les MANET : certains nœuds peuvent établir des relations de confiance à long terme afin d'améliorer leurs communications à venir. Un nœud peut ainsi accepter d'échanger des informations publiques avec tous les autres nœuds mais réserver ses informations sensibles aux seuls nœuds en lesquels il a confiance, c'est-à-dire les nœuds appartenant à sa communauté de confiance.

Plusieurs solutions existent pour établir et maintenir des communautés de confiance. La première et la plus connue est le *Resurrecting Duckling* [SA00]. D'autres solutions ont vu le jour par la suite comme [PBAH03], particulièrement adaptée aux réseaux ad hoc domestiques. Dans la plupart des solutions, l'initialisation et la gestion des relations de confiance sont effectuées par l'utilisateur, tandis que la détection du statut de confiance entre les nœuds est réalisée automatiquement : dès que deux nœuds entrent en communication ils détectent leurs relations de confiance et adaptent leurs réactions en conséquence.

Parallèlement à l'apparition de ces solutions, le respect de la vie privée est devenu une préoccupation grandissante pour les utilisateurs. Cette préoccupation apparaît très naturellement dans de nombreux domaines d'application des MANETS : réseaux domestiques, réseaux personnels, les réseaux de véhicules (VANET pour *Vehicular Ad hoc NETWORKS*) comme le souligne par exemple [Toh08, F105].

Le respect de la vie privée va bien au-delà de la seule confidentialité des communications. En particulier, plus les appareils disposent de capacité de communication plus ils peuvent révéler de l'information sur leur configuration, leur fonctionnement ou même leur simple présence.

Dans les réseaux de véhicules notamment, il apparaît de plus en plus nécessaire de se prémunir de ces fuites d'information : un observateur extérieur ne devrait pas être en mesure de suivre les déplacements d'un véhicule pendant une longue période, simplement en se basant sur l'observation des communications entre véhicules. Concernant les relations de confiance entre véhicules, il serait souhaitable que leur existence même reste inobservable.

Les préoccupations liées au respect de la vie privée n'étaient probablement pas aussi importantes lors de la publication des premières solutions de gestion de la confiance dans les MANET ; ces solutions offrent peu de propriétés d'anonymat et encore moins de non-observabilité.

Nous nous intéressons, dans ce papier, aux propriétés facilitant le respect de la vie privée dans les MANET. Nous recherchons notamment le chiffrement et l'anonymat des communications, ainsi que la discrétion de l'opération de détection automatique de la confiance entre deux nœuds.

Nous montrons notamment comment réaliser cette opération en garantissant l'anonymat des nœuds qui y participent. Puis nous montrons comment rendre cette opération non-observable pour certains types d'attaquants.

La section 2 donne les définitions utilisées dans la suite du papier ainsi que les contraintes que nous avons respectées dans la conception de notre solution. La section 3 donne les notations utilisées dans le papier ainsi qu'une description complète de notre solution. La section 4 présente une vérification des propriétés liées à la sécurité et à l'anonymat. Une partie des vérifications est effectuée à l'aide de l'outil de vérification automatique AVISPA [ABB⁺05].

2 Préliminaires

Dans la section 2.1, nous indiquons les définitions des concepts utilisés et nous les utilisons ensuite dans la section 2.2, pour poser le problème à résoudre. Nous donnons des précisions sur le modèle de confiance et sur le modèle de communication dans les sections 2.4 et 2.5.

2.1 Définitions

Nous indiquons tout d'abord les définitions relatives aux nœuds et à la relation de confiance.

Définition (Nœud)

Un nœud A est une entité communicante munie d'une paire de clé asymétriques notée (K_A, K_A^{-1}) . N.B. : aucune des deux clés n'est publique. L'ensemble de tous les nœuds est noté \mathcal{N} .

Définition (Confiance)

Un nœud A fait confiance au nœud B , noté $A \top B$, si A connaît la clé asymétrique K_B . On remarque que $A \top B$ n'implique pas $B \top A$. On remarque également que le nœud A peut donner la clé K_B à un autre nœud X pour que X fasse aussi confiance à B .

Définition (Communauté de confiance)

La communauté de confiance d'un nœud A , notée \mathcal{T}_A est l'ensemble des nœuds auxquels A fait confiance. $\mathcal{T}_A = \{B \in \mathcal{N}, A \top B\}$

Nous rappelons maintenant les définitions les plus communément admises de l'anonymat et de ses généralisations. Ces définitions sont toutes tirées de [PK00, PH08].

Définition (Ensemble d'anonymat)

Un ensemble d'anonymat est un ensemble de nœuds ayant les mêmes attributs et susceptibles d'effectuer les même actions.

Définition (Anonymat)

Un nœud est anonyme s'il n'est pas identifiable parmi en ensemble d'anonymat. Il y a anonymat de l'émetteur (resp. du destinataire) lorsque l'analyse d'un message ne permet pas de déterminer son émetteur (resp. son destinataire) dans l'ensemble d'anonymat.

Définition (Non-observabilité)

La non-observabilité est l'impossibilité de déduire l'existence d'un événement. Il y a non-observabilité de l'émetteur (resp. du récepteur) lorsqu'il est impossible de déterminer s'il a émit (resp. reçu) un message.

Nous terminons par les définitions des différents types d'attaquants.

Définition (Attaquant passif)

Un attaquant passif peut écouter tous les messages envoyés sur le réseau. Il ne connaît pas de clés cryptographiques *a priori*.

Définition (Attaquant actif)

Un attaquant actif peut écouter, ajouter, enlever et modifier tous les messages du réseau. On dit que "l'attaquant *est* le réseau".

Définition (Caractérisation de trafic)

Une caractérisation de trafic est possible lorsque l'observation suffisamment longue et complète du réseau permet de déduire de l'information sur les flux de données (par exemple différencier un flux de données multimédia d'un flux de signalisation).

N.B. : les attaquants que nous considérons par la suite n'ont pas la possibilité de caractériser le trafic. Typiquement, ils ne peuvent écouter qu'un petit nombre de communications réseaux. Cette limitation est importante mais se justifie en pratique par les arguments suivants :

- Dans les MANET, les attaques par capture d'une grande quantité de trafic réseau sont généralement difficiles à réussir. L'attaquant doit se déplacer en même temps que le MANET et prend le risque d'être détecté.
- Le trafic entre nœuds se faisant confiance peut être beaucoup plus intense que le trafic standard. Par exemple, ils peuvent échanger des contenus multimédia, alors qu'il ne le font pas sans relation de confiance. Des techniques complémentaires peuvent être mise en œuvre *a posteriori* pour masquer ces différences.

2.2 Le problème à résoudre

Étant donné un ensemble de nœuds et leurs relations de confiance, est-il possible simultanément :

1. de permettre des communications anonymes entre chaque nœud.
2. de rendre la détection des relations de confiance inobservables.

En particulier est il possible d'empêcher un attaquant de comprendre qu'il existe une relation de confiance entre deux nœuds ?

2.3 Les contraintes à respecter

Le but que nous poursuivons est de reconstruire des relations de confiance sans pour autant divulguer des informations sur notre propre identité ou sur le fait que nous appartenons à une communautés de confiance donnée. Notre protocole doit donc garantir les propriétés d'anonymat et de non-observabilité.

Un nœud doit envoyer certaines données pour prouver sa présence aux autres nœuds de la communauté de confiance qui pourraient se trouver dans son voisinage. Ces données doivent permettre aux membres de la communauté de le reconnaître comme un nœud de confiance. Les nœuds n'appartenant pas à la communauté doivent seulement percevoir les données envoyées comme des données génériques et ils ne doivent pas pouvoir en tirer d'informations particulières.

Dans la suite, nous appelons ces données un *message de signalisation*. Ce message de signalisation peut changer à chaque envoi et obéit aux propriétés suivantes :

Proposition 1 *Un nœud recevant un message de signalisation doit être capable de dire si celui-ci est bien formé.*

Cette propriété assure qu'un attaquant ne peut pas forger un message de signalisation pour tenter d'entrer dans une communauté de confiance.

Proposition 2 *Un nœud recevant un message de signalisation doit être capable de lier ce message à un secret partagé.*

Cette propriété assure que si un nœud reçoit un message de signalisation d'un autre nœud appartenant à sa communauté de confiance, il est capable de retrouver le secret qu'ils partagent.

2.4 Le modèle de confiance

De nombreux modèles de confiance existent, du simple partage de clé secrète aux chaînes de certificats. Pour plus d'information sur les différents modèles de confiance, consulter [BSSW02] ou [PBAH03]. Le modèle que nous utilisons ici est particulièrement adapté aux réseaux mobiles ad hoc. Nous en donnons les principales caractéristiques et quelques exemples d'application.

Rôle de la confiance. Elle sert typiquement à enrichir les échanges; un nœud peut décider d'envoyer certaines informations uniquement à des nœuds en lesquels il a confiance. Dans les réseaux de véhicules, un camion peut par exemple envoyer à tous les autres des informations relatives au trafic, mais il n'envoie des informations relatives à son trajet qu'aux camions en lesquels il a confiance (par exemple ceux de la même flotte). Dans des réseaux d'échange pair-à-pair, un pair peut envoyer des fragments de bande-annonces à tous les autres pairs et des fragments de films uniquement aux pairs en lesquels il a confiance.

Établissement de la confiance. Comme dans beaucoup de modèles, l'opération d'établissement de la confiance se fait au préalable et nécessite un canal sécurisé. Pour établir la confiance $A \top B$ il suffit de transmettre une fois K_B à A au travers du canal sécurisé. Si par la suite B change de clés, ou si A perd K_B , il faudra recréer la relation de confiance. Si A ne fait plus confiance à B , il lui suffit d'oublier K_B .

Extension de la confiance. Dans la situation où $A \top B$, A peut décider de recommander B à un d'autre nœud X ; pour cela, il lui suffit de transmettre une fois K_B à X au travers d'un canal sécurisé. Dans l'exemple des réseaux de véhicules, un camion A peut par exemple faire confiance à un équipement routier B et transmettre cette confiance à un autre camion X , même en l'absence de B .

Détection de la confiance. Lorsqu'ils entrent en communication, par exemple quand ils se rapprochent, deux nœuds A et B doivent pouvoir détecter rapidement leurs éventuelles relations de confiance; tous les cas peuvent se produire $A \top B$ ou $B \top A$ ou les deux ou ni l'un ni l'autre. À titre d'exemple, l'échange suivant permet de détecter de manière simple (mais observable) la relation $A \top B$:

$B \rightarrow A : \{j\}_{K_B^{-1}}$ où j est un nombre aléatoire

$A \rightarrow B : \{i, j\}_{K_A^{-1}}$ où i est un nombre aléatoire

$B \rightarrow A : \{message\}_{f(i, j)}$ où $f(i, j)$ est une clé que A et B peuvent calculer connaissant i et j .

2.5 Le modèle de communication

Notre solution pour la détection de la confiance utilise des messages envoyés régulièrement et en *broadcast*. Dans le même temps les nœuds sont libres d'envoyer et de recevoir des messages quelconques, à leurs risques et périls : certains messages peuvent en effet révéler leur identité ou la nature de leurs communications.

Nous nous concentrons uniquement sur les garanties d'anonymat et de non-observabilité pour la solution de détection de la confiance. Dans ce cadre, chaque nœud envoie régulièrement des messages, par exemple un par seconde. Ces messages sont tous de la forme : $P_{from}.P_{to}.N.Payload$ où P_{from} est un pseudonyme de l'émetteur, P_{to} est un pseudonyme du destinataire, N est une information cryptographique et $Payload$ est la partie utile du message.

L'émission des messages de manière régulière est critique du point de vue de la non-observabilité. Intuitivement, un observateur n'obtiendra aucune information du simple fait qu'un message est émis, puisque cela arrive régulièrement. Dans le reste du papier dire que " A envoie m à B " signifie que le message $P_A.P_B.N.m$ est mis dans une file d'attente. Sur une base régulière, le premier message en attente sera extrait de la file et émis en *broadcast*.

3 Description de la solution

3.1 Notations

La figure 1 indique toutes les notations utilisées dans ce papier.

Notation	Definition
$A, B, C \dots$	Les nœuds du réseau
K_A, K_A^{-1}	La paire de clés du nœud A
P_A^i	Un pseudonyme du nœud A
$A \top B$	A fait confiance à B
\mathcal{T}_A	La communauté de confiance de A
$i, j, k, l, m, n \dots$ et R	Des nombres aléatoires frais (non utilisés auparavant)
db	Un nombre public, par exemple $0xDEADBEEF$
g	Un générateur Diffie-Hellman public
$secret$	Un message à protéger
$m_1, m_2 \dots$	Des messages aléatoires
$\{ \}_K$	Un algorithme de chiffrement / déchiffrement utilisant la clé K
$\{ \{ M \}_K \}_{K^{-1}}$	le message M chiffré par K et déchiffré par K^{-1}

FIG. 1: Notations

3.2 Description notre solution

Notre solution associe trois techniques essentielles :

- L'utilisation de pseudonymes pouvant changer au cours du temps.
- Le chiffrement systématique des messages.
- L'envoi régulier de messages.

Les messages ont pour forme générale $P_{from}.P_{to}.N.Payload$, comme indiqué en section 2.5. Les composantes P_{from} et P_{to} servent à échanger les pseudonymes. Elles servent également à détecter les relations de confiance lorsqu'elles existent. La troisième composante sert à établir une clé Diffie-Hellman. Cette clé sera authentifiée si des relations de confiance existent, et non authentifiée sinon. Pour plus d'information sur les échanges Diffie-Hellman authentifiés, consulter [BWM98].

N.B. : nous recherchons des propriétés de non-observabilité pour lesquelles il est critique que les messages échangés se ressemblent tous vis-à-vis d'un observateur extérieur. Dans les descriptions ci-après, il est donc normal que certains messages semblent répétés. En particulier, le message débutant chaque échange est toujours le même. Sans perte de généralité, nous posons qu'il est envoyé par le nœud appelé A .

Cas $B \top A$

Nous décrivons tout d'abord la séquence de messages échangés entre deux nœuds A et B tels que $B \top A$ (i.e. B connaît K_A) et tels que A commence l'échange.

1. A choisit un pseudonyme $P_A^i = \{i.0.db\}_{K_A^{-1}}$ et mémorise i .
2. $A \rightarrow Tous : P_A^i.R.g^i.m_1$
3. B vérifie que $\{P_A^i\}_{K_A} = i'.0.db$ et que $i' = i$.
4. $B \rightarrow Tous : \{i.j.db\}_{K_A}.P_A^i.g^j.m_2$
5. A vérifie que $\{\{i.j.db\}_{K_A}\}_{K_A^{-1}} = i'.j'.db$ et que $i' = i$ et $j' = j$
6. $A \rightarrow Tous : P_A^i.\{i.j.db\}_{K_A}.g^k.\{secret\}_{g^{ij}}$

Dès l'étape 3, B détecte que P_A^i est un pseudonyme de A , en qui il a confiance. Les modalités de cette détection peuvent varier. Une manière simple (mais peu efficace) de procéder est d'essayer une à une toutes les clés des nœuds de \mathcal{T}_B jusqu'à trouver la clé K_A qui déchiffre le pseudonyme de A . Si B fait confiance à beaucoup de nœuds l'ensemble \mathcal{T}_B est grand et la détection peut être longue. Il existe plusieurs astuces pour rendre la détection plus efficace comme par exemple essayer en premier les clés qui réussissent souvent (principe de localité).

Dès l'étape 5, A détecte que $\{i.j.db\}_{K_A}$ est un pseudonyme d'un nœud en lequel il a confiance (par contre A ne sait pas que le nœud en question est B).

Aussi bien A que B peuvent conserver leur pseudonyme pendant la suite de leur conversation. Ils s'échangent alors autant de messages chiffrés qu'ils le veulent sous la forme :

$$A \rightarrow Tous : P_A^i.\{i.j.db\}_{K_A}.g^l.\{secret\}_{g^{ij}}$$

$$B \rightarrow Tous : \{i.j.db\}_{K_A}.P_A^i.g^m.\{secret\}_{g^{ij}}$$

Si l'un des deux participants décide de changer son pseudonyme, la communication en cours sera rompue. Elle pourra bien évidemment être recrée en redémarrant une séquence du type 1 à 6.

Cas ATB

Nous décrivons maintenant la séquence de messages échangés lorsque ATB (et toujours avec A qui commence l'échange).

1. A choisit un pseudonyme $P_A^i = \{i.0.db\}_{K_A^{-1}}$ et mémorise i .
2. $A \rightarrow Tous : P_A^i.R.g^i.m_1$
3. B choisit un pseudonyme $P_B^j = \{j.0.db\}_{K_B^{-1}}$ et mémorise j .
4. $B \rightarrow Tous : P_B^j.P_A^i.g^j.m_2$
5. A vérifie que $\{P_B^j\}_{K_B} = j'.0.db$ et que $j' = j$.
6. $A \rightarrow Tous : \{j.k.db\}_{K_B}.P_B^j.g^k.m_3$
7. B vérifie que $\{\{j.k.db\}_{K_B}\}_{K_B^{-1}} = j'.k'.db$ et que $j' = j$ et $k' = k$
8. $B \rightarrow Tous : P_B^j.\{j.k.db\}_{K_B}.g^l.\{secret\}_{g^{jk}}$

Les étapes 3 à 8 sont en fait les équivalent des étapes 1 à 6 de l'échange précédent. Dès l'étape 5, A détecte que P_B^j est un pseudonyme de B , en qui il a confiance. Dès l'étape 7, B détecte que $\{j.k.db\}_{K_B}$ est un pseudonyme d'un nœud en lequel il a confiance (par contre B ne sait pas que le nœud en question est A).

Cas sans confiance

Nous décrivons maintenant la séquence de messages échangés entre deux nœuds A et C qui ne partage aucune relation de confiance.

1. A choisit un pseudonyme $P_A^i = \{i.0.db\}_{K_A^{-1}}$ et mémorise i .
2. $A \rightarrow Tous : P_A^i.R.g^i.m_1$.
3. B choisit un pseudonyme $P_B^j = \{j.0.db\}_{K_B^{-1}}$ et mémorise j .
4. $B \rightarrow Tous : P_B^j.P_A^i.g^j.m_2$.
5. $A \rightarrow Tous : P_A^i.P_B^j.g^k.\{secret\}_{g^{ij}}$.

Ici, A et B ne partagent aucune clé et ne peuvent donc pas collaborer pour vérifier des coefficients Diffie-Hellman. La clé utilisée pour chiffrer leurs communications ultérieures est simplement la clé Diffie-Hellman g^{ij} élaborée dans la troisième composante des messages 2 et 4.

Cas ATB et BTA

Ce cas n'est pas géré de manière particulière dans notre solution. Selon le nœud qui débute la communication, il peut être traité comme un cas ATB ou comme un cas BTA . Néanmoins, une fois le canal anonyme sécurisé établi entre A et B rien n'empêche l'un des deux de proposer à l'autre de s'authentifier, par exemple en signant une variable aléatoire. Dans la cas BTA par exemple A peut tenter de s'authentifier auprès de B . Cela donne :
 $A \rightarrow Tous : P_A^i.\{i.j.db\}_{K_A}.g^l.\{n.\{n\}_{K_A^{-1}}\}_{g^{ij}}$ B vérifie que $\{n\}_{K_A^{-1}} = n'$ et que $n' = n$

4 Analyse de la solutions

4.1 Propriétés de sécurité

La solution doit permettre à deux nœuds A et B partageant (au moins) une relation de confiance de communiquer secrètement. En particulier, le contenu des messages échangés ne doit pas être accessible à un attaquant.

Pour vérifier cette propriété, nous avons utilisé AVISPA [ABB⁺05]. La modélisation de la solution en HLPSL est indiquée en annexe. Nous avons utilisé deux outils inclus dans AVISPA (ATSE et OFMC) pour vérifier le secret du payload dans les cas ATB et BTA . Aucun outil n'a révélé d'attaque pour un attaquant actif.

Par contre dans le cas où il n'existe aucune confiance entre A et B , le secret repose seulement sur un échange Diffie-Hellman. Le secret reste vérifié contre un attaquant passif, mais pas contre un attaquant actif. En théorie l'attaquant actif peut réaliser une attaque par interposition (*man-in-the-middle*). En pratique, cette attaque est difficile à réaliser en contexte MANET ou VANET. Malheureusement, il existe une variante plus facile à réaliser : l'attaquant actif (D) envoie fréquemment des messages de la forme $B \rightarrow Tous$: $P_D^i . P_A^i . g^1 . m$ en réponse au message de signalisation envoyé par A . On note que la troisième composante est la valeur publique g^1 , de manière à ce que la clé Diffie-Hellman calculée par A soit égale à $(g^1)^i$ qui n'est autre que g^i , valeur apprise par l'attaquant lorsqu'il a capté le message de signalisation de A .

Cette attaque n'est pas gênante *en soi* puisqu'elle ne révèle que des données que les nœuds sont prêts à échanger sans relation de confiance. En revanche, cette attaque a des conséquences sur la non-observabilité de la relation de confiance; un attaquant actif qui parvient à déchiffrer un échange entre deux nœuds comprend alors que les nœuds en questions n'ont pas de relation de confiance. À l'inverse, si son attaque échoue, c'est que les nœuds ont au moins une relation de confiance.

4.2 Propriété d'anonymat

Tout d'abord, nous remarquons que tous les participants X à une communication utilisent des pseudonymes de la forme $\{i.0.db\}_{K_X^{-1}}$. Pour ne rien révéler du secret permanent K_X^{-1} de X il faut donc que l'algorithme cryptographique $\{\}_{K_X^{-1}}$ ne révèle rien sur sa clé. C'est l'une des propriétés de base des algorithmes de chiffrement asymétriques, tels RSA.

Nous remarquons également que les nœuds changent régulièrement leur pseudonymes. En particulier, ils choisissent un nouveau pseudonyme à chaque fois qu'ils souhaitent établir une connexion avec d'autres nœuds. Selon qu'il y a ou non des relations de confiance le nouveau pseudonyme n'est pas élaboré de la même manière, mais en aucun cas il ne révèle d'information permanente.

4.3 Propriété de non-observabilité

Nous avons indiqué que la non-observabilité ne tenait pas contre un attaquant actif. Nous donnons des justifications de la non-observabilité contre un attaquant passif ayant peu de capacité d'analyse de trafic. Notre solution possède les propriétés suivantes :

1. Tous les messages sont envoyés en broadcast.
2. Tous les nœuds envoient des messages régulièrement.
3. Tous les composants des messages sont chiffrés ou de la forme g^i .

Selon [PH08] §8, les propriétés 1 et 3 impliquent la non-observabilité du destinataire et les propriétés 2 et 3 impliquent l'anonymat de l'émetteur.

5 Conclusion

Nous constatons que les questions relatives à la protection de la vie privée deviennent de plus en plus importantes. C'est particulièrement vrai dans les réseaux MANET ou VANET qui sont par nature très liés aux utilisateurs et aux activités individuelles. Dans ces réseaux, nous avons exhibé une solution permettant de détecter des relation de confiance prééxistantes tout en garantissant l'anonymat et dans une moindre mesure la non-observabilité. Nous travaillons maintenant à l'amélioration de cette solution sur plusieurs paramètres : la résistance à des attaquants plus puissant, la diminution de la complexité, l'assouplissement du modèle de confiance.

Nous avons évoqué deux champs d'application possibles de notre solution : les VANET et les réseaux pair-à-pair (moyennant quelques adaptations). Nous pensons que le champ d'application peut s'élargir, en particulier avec l'apparition de composants communicants capables d'effectuer des opérations cryptographiques.

Références

- [F105] Florian Dotzer. Privacy Issues in Vehicular Ad Hoc Network. In *Workshop on Privacy Enhancing Technologies*, pages 197–209, 2005.
- [ABB⁺05] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santos Santiago, M. Turuani, L. Viganò, and L. Vigneron. The AVISPA Tool for the automated validation of internet security protocols and applications. In K. Etessami and S. Rajamani, editors, *17th International Conference on Computer Aided Verification, CAV'2005*, volume 3576 of *Lecture Notes in Computer Science*, pages 281–285, Edinburgh, Scotland, 2005. Springer.
- [BSSW02] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to strangers : Authentication in adhoc wireless networks, February 2002. In Symposium on Network and Distributed Systems Security (NDSS '02), San Diego, California.
- [BWM98] Simon Blake-Wilson and Alfred Menezes. Authenticated Diffie–Hellman key agreement protocols. In *Selected Areas in Cryptography*, pages 339–361, 1998.
- [FFBA07] E. Fonseca, A. Festag, R. Baldessari, and R. Aguiar. Support of Anonymity in VANETs - Putting Pseudonymity into Practice. In *IEEE Wireless Communications and Networking Conference*, 2007.
- [GFL⁺07] Matthias Gerlach, Andreas Festag, Tim Leinmuller, Gabriele Goldacker, and Charles Harsch. Security Architecture for Vehicular Communication. In *Workshop on Intelligent Transportation*, 2007.
- [PBAH03] N. Prigent, C. Bidan, J.P. Andreaux, and O. Heen. Secure long term communities in ad hoc networks. In *SASN '03 : Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 115–124, New York, NY, USA, 2003. ACM.

- [PH08] Andreas Pfitzmann and Marit Hansen. Anonymity, unobservability, and pseudonymity : A consolidated proposal for terminology. Draft, July 2008.
- [PK00] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In *Workshop on Design Issues in Anonymity and Unobservability*, pages 1–9, 2000.
- [SA00] Frank Stajano and Ross J. Anderson. The resurrecting duckling : Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Workshop on Security Protocols*, pages 172–194, London, UK, 2000. Springer-Verlag.
- [Toh08] C.K. Toh. Research challenges in intelligent transportation networks. keynote speech at ifip networking 2008, singapore, May 2008.

Annexe

```

role nodeA (A: agent, Ka: public_key, SND, RCV: channel(dy)) played_by A def=
  local State : nat,
    Nai, Naj, Nj2, Na0: text,
  Ma, Ma2, R, Dh, Secret: message,
  K, Kgen: message,
  Kx: public_key
  init State := 0
  transition
0. State = 0 /\ RCV(start) =|>
  State' := 1 /\ Nai' := new() /\ Na0' := new() /\ R' := new() /\ Ma' := new()
  /\ SND({Nai'.Na0'.db}_inv(Ka).R'.exp(g,Nai').Ma')
a1. State = 1 /\ RCV({Nai.Naj'.db}_Ka.{Nai.Na0.db}_inv(Ka).exp(g,Naj').Ma')
  =|> % Naj correspond entre Dh et Pb
  State' := 2 /\ K' := exp(exp(g,Naj'),Nai) /\ Secret' := new() /\
  SND({Secret'}_K')
% Role A si B ne connait pas A
b1. State = 1 /\ RCV({Ma2'}_inv(Kx').{Nai.Na0.db}_inv(Ka).Dh'.Ma') /\ not(Kx'=Ka)
  =|> State' := 3 /\ Kgen' := exp(Dh',Nai) /\ Ma' := new() /\ SND({Ma'}_Kgen')
% Role A si B ne respecte pas le jeu (tentative de man in the middle)
c1. State = 1 /\ RCV({Nai.Naj'.db}_Ka.{Nai.Na0.db}_inv(Ka).exp(g,Nj2').Ma')
  /\ not(Nj2'=Naj') =|> % Nj ne correspond pas on envoie n'importe quoi
  State' := 4 /\ Kgen' := exp(exp(g,Nj2'),Nai) /\ Ma' := new() /\
  SND({Ma'}_Kgen')
end role

role nodeB (B: agent, Kb: public_key, KeyRing: (agent.public_key) set,
  SND, RCV: channel(dy))
played_by B def=
  local State : nat,
    Nbi, Nbj, Nj2, Nb0, Nb2: text,
  Mb, Mb2, R, Dh, Secret: message,
  K, Kgen: message,
  Kx: public_key,
  X: agent
  init State := 0
  transition
a3. State = 0 /\ RCV({Nbi'.Nb0'.db}_inv(Kx').R'.Dh'.Mb')
  /\ in(X'.Kx',KeyRing) =|>
  State' := 5 /\ Nbj' := new() /\ Mb' := new() /\ K' := exp(Dh',Nbj')

```

```

    /\ SND({Nbi'.Nb2'.db}_Kx.{Nbi'.Nb0'.db}_inv(Kx')).exp(g,Nbj').Mb')
a4. State = 5 /\ RCV({Secret'}_K) =|> State':= 6 /\ secret(Secret',sec,{B,X})
% Role B dans lequel A n'est pas connu
b3. State = 0 /\ RCV({Nbi'.Nb2'.db}_inv(Kx')).R'.Dh'.Mb')
    /\ not(in(X'.Kx',KeyRing)) =|>
    State':= 7 /\ Nb0':= new() /\ Nbj':= new() /\ Mb':= new() /\ Kgen':= exp(Dh',Nbj')
    /\ SND({Nbj'.Nb0'.db}_inv(Kb').{Nbi'.Nb2'.db}_inv(Kx')).exp(g,Nbj').Mb')
b4. State = 7 /\ RCV({Mb'}_Kgen) =|> State':= 8 /\ secret(Mb',sec2,{a,c})
end role

role environment() def=
  local KeyMapA, KeyMapB, KeyMapC, KeyMapD: (agent.public_key) set,
        SND, RCV: channel(dy)
  const a,b,c,d,i: agent,
        ka, kb, kc, kd, ki: public_key,
  g,db: message,
        sec, sec2, alice_bob_nb, bob_alice_na: protocol_id
  init KeyMapA := {} /\ KeyMapB := {a.ka} /\ KeyMapC:= {} /\ KeyMapD := {a.ka,b.kb}
  intruder_knowledge = {a, b, c, d, g, db, ki, inv(ki)}
  composition
    nodeA(a, ka, SND, RCV) /\ nodeB(b, kb, KeyMapB, SND, RCV)
end role

goal
  secrecy_of sec
  secrecy_of sec2
end goal

environment()
```