UNIVERSITÉ DE LIMOGES
ÉCOLE DOCTORALE « Science, Technologie, Santé »
FACULTÉ DES SCIENCES ET TECHNIQUES

Thèse N° 35-2009

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Discipline / Spécialité : Informatique

présentée et soutenue par

Céline BURGOD

le 12 octobre 2009

Contribution à la sécurisation du routage dans les réseaux ad hoc

Thèse dirigée par Jean-Pierre Borel,
Co-encadrée par Carlos Aguilar Melchor et Julien Iguchi-Cartigny

JURY

Rapporteurs :	
Mme Maryline LAURENT-MAKNAVICIUS	Professeur à TELECOM SudParis
M. Ludovic MÉ	Professeur à Supélec Rennes
Examinateurs :	
M. Carlos AGUILAR MELCHOR	Maître de conférences à l'Université de Limoges
M. Christophe BIDAN	Maître de Conférences à Supélec Rennes
M. Jean-Pierre BOREL	Professeur des Universités à l'Université de Limoges
M. Yves DESWARTE	Directeur de Recherche CNRS au LAAS-CNRS
M. Julien IGUCHI-CARTIGNY	Maître de Conférences à l'Université de Limoges

Remerciements

Je tiens en premier lieu à remercier mon directeur de thèse Jean-Pierre Borel pour la confiance qu'il m'a accordée, ainsi que pour la liberté octroyée relativement à mes orientations de recherche.

Je ne saurai jamais assez remercier Julien Iguchi-Cartigny pour avoir accepté de co-encadrer ma thèse et qui m'a apporté l'aide et les encouragements nécessaires dans les moments les plus difficiles. Cet encadrement a été largement accompagné par Carlos Aguilar Melchor, que je tiens également à remercier ici.

J'adresse mes sincères remerciements aux professeurs Maryline Laurent-Maknavicius et Ludovic Mé pour avoir accepté la charge de rapporteur malgré leur emploi du temps chargé, ainsi qu'à Yves Deswarte et Christophe Bidan pour leur rôle d'examinateur.

Merci à tous les doctorants et amis de Jidé sans qui ces années de travail auraient été bien ternes. Je pense tout particulièrement à Alex, Agnès, Ahmadou, Boucle d'Or, Boussad, Chinois, Emma, Jeff, Kevin, Mickael, Nadir, Salman, Stan, Yann. Un grand merci à Kosso pour m'avoir donné l'occasion de goûter aux bienfaits du « beignet au chocolat ».

Merci à Hubert et Passe-Partout pour leur amitié.

Merci à tous les Kichouilles de m'avoir offert de purs moments de détente sur et en dehors des terrains de foot.

Ma pensée se tourne également vers mon compagnon, Damien, qui a toujours su trouver les moyens pour me réconforter, et sans qui ce travail ne serait pas ce qu'il est aujourd'hui. Dans la dernière ligne droite, il a engagé une chasse sans merci aux coquilles, même s'il faut avouer que certaines sont tenaces.

Enfin, je terminerai ces remerciements par une pensée pour ma famille, et tout particulièrement mes parents qui m'ont constamment soutenue dans mes projets d'études, sans trop savoir jusqu'où tout cela pourrait me mener. Rassurez-vous, cette fois c'est vraiment la fin !

Résumé

Les réseaux ad hoc sont composés d'entités autonomes sans fil qui organisent par elles-mêmes leurs communications. Ces réseaux sont plus vulnérables aux attaques que les réseaux filaires ou sans fil conventionnels, en raison de l'absence d'infrastructure de gestion, de l'utilisation de canaux de communication sans fil, et de la protection limitée des entités. Par exemple, on peut s'attendre à des dégradations des communications à partir du moment où des entités malveillantes corrompent les opérations de base dont elles sont responsables.

Les travaux réalisés dans le cadre de cette thèse portent sur la sécurité du routage. Notre approche consiste d'une part à empêcher les attaques sur les messages de contrôle, et d'autre part à fournir un support fiable pour la détection des comportements malveillants. Afin de pallier les limites des approches existantes, nous étudions l'utilisation d'un équipement matériel résistant à la manipulation. Nous décrivons un schéma de contrôle, situé entre la couche liaison de données et la couche réseau, permettant de détecter avec précision les entités corrompues dans les opérations élémentaires requises par tous les protocoles de routage.

Dans la seconde partie de ce travail, nous proposons un cadre d'analyse systématique de la sécurité du protocole de routage OLSR (*Optimized Link State Routing*). Nous décrivons : (1) les différents éléments qui définissent le fonctionnement d'OLSR, (2) les attaques possibles sous la forme d'actions élémentaires non conformes, puis (3) les relations de causalité entre les différentes actions non conformes et l'étendue des perturbations. Cette représentation permet de construire une base de référence substantielle à partir de laquelle il est possible de comparer plusieurs versions renforcées d'OLSR.

Mots-clés : réseau ad hoc, routage, sécurité, détection de pertes, équipement résistant à la manipulation, carte à puce, OLSR.

Abstract

Ad hoc networks are composed of wireless autonomous entities which organize their communications by themselves. These networks are more vulnerable to attacks than conventional wired or wireless networks because of the lack of central management to carry out basic networking functions, the use of wireless channels, and the limited physical protection of the entities. For example, one can expect communication damage if malicious entities corrupt the basic operations for which they are responsible.

In the scope of this thesis, we have focused on the security of the routing functions. Our approach consists on one hand in preventing attacks against the control messages, and on the other hand in providing a reliable support for the detection of malicious behaviors. To mitigate the limits of the existing approaches, we study the use of a tamper resistant hardware. We describe a control scheme, located between the data link layer and the network layer, allowing to accurately detect and locate corrupted entities with regards to the elementary operations required by any routing protocols.

In the second part of this work, we propose a systematic security analysis for the OLSR (*Optimized Link State Routing*) routing protocol. Within our framework, we describe : (1) the various elements which define the OLSR protocol, (2) the possible attacks in the form of elementary illegal actions, then (3) the causal relationships between various illegal actions and the extent of disruptions. This representation allows to build a substantial reference base from which it is possible to compare several reinforced versions of OLSR.

Key-words : ad hoc network, routing, security, packet dropping detection, tamper resistant hardware, smart card, OLSR.

Table des matières

Table des figures

Liste des tableaux

Introduction

L'émergence de nouvelles technologies de communication sans fil (WiMAX[1] [wim04], Wi-Fi[2] [IEE07], Bluetooth [blu], ...) et la prolifération des équipements mobiles évolués tels que les assistants personnels, les téléphones ou les ordinateurs portables, font que nous assistons depuis quelques années à des modifications importantes dans le domaine de l'information et de la communication. En effet, le déploiement croissant de réseaux sans fil offre la perspective d'un réseau omniprésent, permettant aux utilisateurs en déplacement d'être rapidement connectés et d'avoir de plus accès à tout instant et en tous lieux à des réseaux et à leurs services associés.

Les réseaux sans fil que nous utilisons actuellement (tels que par exemple le GSM[3], le GPRS[4], l'UMTS[5] ou la Wi-Fi) sont ce que nous appelons des réseaux sans fil à station de base : ils sont assez similaires aux réseaux filaires dans le sens où ils fonctionnent tous deux de manière hiérarchique et centralisée. C'est la présence d'une infrastructure de communication fixe et interconnectée de manière filaire qui permet de structurer l'ensemble des communications entre les entités. Par exemple, dans un réseau GSM, un téléphone portable se connecte à une station de base du réseau au moyen d'un dispositif sans fil, puis le reste de la connexion est géré par l'infrastructure filaire du réseau. Ce type d'architecture impose par conséquent que le téléphone portable soit à portée de communication d'une station de base - appelée aussi zone de couverture - pour pouvoir être raccordé au réseau téléphonique. Pour offrir une connexion sur une zone géographique étendue, la mise en place d'une infrastructure fixe avec le déploiement de nombreuses stations de base est donc nécessaire. Or cette mise en œuvre est souvent considérée comme coûteuse en temps, en argent, en matériel et en configuration, voire même impossible dans certaines zones géographiques.

Pour pallier ces limitations, une nouvelle approche est envisagée. Dans cette approche, les entités équipées d'un dispositif de communication sans fil s'auto-organisent pour fournir les services de support des communications. Ce type de réseau est caractérisé par une absence totale d'infrastructure fixe ou dédiée. C'est la portée de transmission des entités le composant qui sert de fondement à la formation d'une infrastructure de communication. Ainsi, selon l'hypothèse qu'il existe assez d'entités pour un espace donné, chacune peut joindre n'importe quelle autre entité, soit directement si elle est à portée radio, soit en

[1] WiMAX : *Worldwide Interoperability for Microwave Access.*
[2] Wi-Fi : *Wireless Fidelity.*
[3] GSM : *Global System for Mobile Communications.*
[4] GPRS : *General Packet Radio Service.*
[5] UMTS : *Universal Mobile Telecommunications System.*

utilisant des entités situées entre elles pour relayer leurs messages. Ce mode de fonctionnement est connu sous le nom de réseau *ad hoc*. Du fait de leur facilité de déploiement et de leur faible coût associé, le champ d'application de ces réseaux est vaste. Initialement, ils ont été conçus pour des applications militaires afin par exemple de permettre la mise en relation de fantassins et de véhicules mobiles sur un champ de bataille. Récemment, leur utilisation a été envisagée dans d'autres contextes tels que la mise en communication de personnes durant un colloque, la coordination de secours sur une zone sinistrée, l'interconnexion de véhicules civils en déplacement, l'extension de la couverture des réseaux (pour par exemple faciliter l'accès à Internet), etc. . ..

Les technologies sans fil n'ont pas été spécialement conçues pour les réseaux ad hoc. La zone de couverture d'un dispositif sans fil est limitée, allant en théorie jusqu'à 500 mètres en extérieur pour la Wi-Fi (802.11 norme b à 1Mbit/s). Dans un tel système, la structuration de l'ensemble des communications est alors assurée grâce à une coopération des entités dans le processus de routage. Ce processus permet d'établir et de maintenir des chemins logiques, de telle sorte que toute entité puisse communiquer avec n'importe quelle autre entité, même si elles sont hors de portée de transmission. Il fonctionne selon deux phases distinctes : (1) une phase de signalisation et (2) une phase d'acheminement des paquets de données. La première assure que chaque entité soit en mesure de collecter et de maintenir suffisamment d'informations sur la topologie du réseau pour construire des chemins vers n'importe quelle autre entité. Quant à la seconde, elle assure l'acheminement des paquets de données entre une entité source et destination, grâce aux entités intermédiaires appartenant au chemin déterminé.

Cadre de la thèse

Les travaux réalisés dans le cadre de cette thèse portent sur la sécurité du routage dans les réseaux ad hoc. La sécurité des opérations de routage est de première importance, car elles servent de briques de base au support des communications - et donc l'accès aux services des couches supérieures. Un manque de sécurité au niveau d'une infrastructure de communication est un des facteurs qui peut freiner leur acceptation ou leur déploiement, surtout pour les applications considérées comme sensibles telles que les applications militaires ou l'organisation d'unités de secours.

Tandis que la nature des infrastructures a évolué, allant de machines isolées, aux réseaux filaires puis aux réseaux sans fil, l'objectif de sécurité reste généralement le même, à savoir préserver la confidentialité et l'intégrité opérationnelle des ressources et du réseau contre les dégradations involontaires ou les attaques malveillantes. Cependant :

- l'absence de toute entité centrale,
- l'utilisation de canaux de communication sans fil,
- les ressources limitées des entités en termes de puissance de calcul, d'espace de stockage et d'énergie,
- l'intégrité physique limitée des entités,
- la nécessité de la coopération entre les entités,

sont autant de caractéristiques qui font que les réseaux ad hoc sont souvent considérés comme plus difficiles à sécuriser que les réseaux traditionnels.

Problématique

Dans les réseaux ad hoc, les fonctions de routage et de sécurité sont confiées aux entités qui y participent. Etant donné que le fonctionnement du réseau repose entièrement sur une coopération entre les entités, on peut s'attendre à des dégradations importantes à partir du moment où des entités tentent de détourner les opérations de bases qui sont à leur charge.

Parmi ces attaques, qu'elles soient motivées par l'égoïsme ou par la malveillance, figurent la dissémination d'informations de routage erronées, et plus simplement la non-retransmission du trafic. Par exemple, une entité prenant part à un réseau ad hoc pourrait, dans un souci de préservation des ressources énergétiques, participer uniquement aux opérations de routage la concernant et détruire tous les autres messages. Or un tel comportement porte atteinte au fonctionnement global du réseau, car il peut provoquer la formation de chemins plus longs, des pertes de connectivité entre les entités (rendant les communications à travers de multiples sauts impossibles), ou une moins bonne répartition du trafic (réduisant potentiellement la durée de vie du réseau).

Alors que les attaques contre les opérations de routage sont faciles à mettre en œuvre, elles sont en revanche particulièrement difficiles à contenir. Les systèmes de réputation font partie des solutions élaborées ces dernières années pour traiter le problème de la non-retransmission des messages dans les réseaux ad hoc [MGLB00, BB02, MM02, BB03]. En règle générale, ils sont formés de trois composants : un mécanisme de contrôle pour identifier les entités à l'origine de comportements malveillants, un mécanisme de calcul de réputation, et un mécanisme de réponse. Le principal avantage de ces solutions est qu'elles sont entièrement distribuées. En effet, ces trois composants sont mis en place au niveau de chaque entité, ce qui lui permet d'apprendre les comportements d'autres entités voisines, puis de se protéger contre ces dernières lorsque des actions de malveillance sont identifiées. Néanmoins, une hypothèse communément prise dans ces travaux est que les entités fonctionnent correctement durant la phase de signalisation. En d'autres termes, il est supposé que les attaques par suppression surviennent uniquement dans la phase d'acheminement des paquets de données. Une telle hypothèse est discutable, car à partir du moment où une entité ne participe pas pleinement à la phase de signalisation (en supprimant certains messages de contrôle), elle peut parvenir à s'exclure de la construction des chemins de retransmission. Dès lors, elle n'appartiendra pas aux chemins formés et il ne lui sera plus nécessaire de supprimer des paquets de données dans la phase d'acheminement. Malgré les effets néfastes d'un tel comportement sur la connectivité du réseau, elle ne sera donc pas identifiée comme malveillante. Par ailleurs, les techniques utilisées pour identifier de telles entités malveillantes présentent certaines limitations : elles sont souvent imprécises, coûteuses en nombre de messages ou en temps (de détection), ou encore dépendantes du protocole de routage sous-jacent. Un dernier point est que ces solutions ne traitent pas complètement les situations dans lesquelles une entité malveillante tente de détourner les mécanismes de sécurité, soit individuellement soit de manière plus subtile par une collusion avec d'autres entités.

Dans une première partie de nos travaux, nous traitons le problème de l'identification des entités malveillantes qui détruisent intentionnellement les messages d'autres entités

présentes dans le réseau. Nous traitons en particulier deux questions : comment détecter de telles suppressions d'une part, et comment exploiter cette information dans le but de renforcer le fonctionnement du réseau malgré cette présence d'autre part. Pour aborder ces questions, nous sommes partis du principe que ces suppressions peuvent survenir aussi bien dans la phase de signalisation que dans la phase d'acheminement des protocoles de routage. Nous nous sommes également donnés comme objectif de limiter au maximum la dépendance de la solution au regard du protocole de routage sous-jacent.

Une classe de solutions permettant de contrer les attaques par dissémination de fausses informations de routage fait appel à des mécanismes cryptographiques [SDL+02, ZA02, HPJ02, BV04, RACM04, CXL06]. Dans ce contexte, un des défis à relever est l'analyse de la sécurité de ces solutions. Cette analyse est en effet de première nécessité pour les concepteurs, car elle leur permet d'avoir connaissance des propriétés de sécurité atteintes pour une solution donnée, et ce, dans un environnement de déploiement adverse donné. Cependant, nous pouvons constater que les protocoles de routage sont essentiellement analysés par (1) des méthodes d'évaluation non structurées (et souvent incomplètes), ou plus rarement par (2) des méthodes de vérification formelle. Dans le premier cas, comme les hypothèses émises sur le contexte d'utilisation du protocole évalué ou sur les capacités de l'attaquant sont généralement trop restrictives, les résultats obtenus ne sont valides que dans les seuls cas définis par leurs auteurs. Or les protocoles ont souvent des usages inattendus et sont exposés à des classes d'attaquants plus larges que prévu, et il n'est pas rare de trouver des situations dans lesquelles ces protocoles peuvent être mis à défaut. Quant aux méthodes de vérification formelle, elles fournissent une structure complète d'analyse, mais sont complexes à appliquer aux protocoles de routage ad hoc à cause des concessions requises pour modéliser le système.

Pour traiter ces problèmes, il nous semble qu'un cadre d'analyse systématique de la sécurité du routage des réseaux ad hoc est fondamental.

Apports de nos travaux

Dans le cadre de l'analyse de la sécurité du routage dans les réseaux ad hoc, nos contributions s'articulent autour de deux thèmes : (1) la détection et l'identification des entités à l'origine de suppressions de messages, et (2) l'évaluation de la sécurité des solutions basées sur des mécanismes cryptographiques.

Premièrement, nous avons étudié une solution pour traiter les problèmes de sécurité du routage dans les réseaux ad hoc. Notre approche consiste d'une part à sécuriser la signalisation, et d'autre part à fournir un support fiable de détection des comportements malveillants pour atténuer les effets nuisibles des attaques par suppression. Afin de pallier les limites des approches existantes, nous proposons l'utilisation d'un équipement matériel résistant à la manipulation. Dans un premier temps, nous avons défini les fonctionnalités gérées par cet équipement. Ensuite, par une analyse des différentes interactions possibles entre les entités intervenant dans le routage, nous montrons qu'il reste à traiter le problème de la suppression malveillante de messages. Ensuite, afin de permettre la détection précise de ces entités malveillantes, nous avons analysé les opérations élémentaires communes à

tous les protocoles de routage, à savoir l'émission et la réception des messages (et par extension, la retransmission). Finalement, nous avons défini une couche de contrôle, située entre la couche liaison de données et la couche réseau qui permet, contrairement aux solutions existantes, de détecter et d'identifier les nœuds à l'origine de suppressions de messages aussi bien dans la phase de signalisation que dans la phase d'acheminement. Elle présente également l'avantage d'être indépendante du protocole de routage sous-jacent.

Dans une deuxième partie, nos travaux ont porté sur la définition d'un cadre d'analyse systématique de la sécurité des versions renforcées du protocole de routage OLSR (*Optimized Link State Routing*) [CJ03]. Ce cadre a pour but de fournir une meilleure compréhension des environnements dans lesquels un protocole est sécurisé, et sur les opportunités d'attaques restantes. Notre approche se fonde sur une analyse décomposée en trois étapes.

1. En premier lieu, nous proposons un examen des interactions entre les différents éléments intervenants dans le protocole OLSR de référence. Dans cette analyse, nous montrons comment les entités réagissent aux différentes informations de routage qu'elles reçoivent d'une part, et quelle est la portée de chacune de ces informations dans le réseau, d'autre part.
2. Après avoir identifié les points critiques du protocole, nous étudions en détail les effets néfastes des attaques menées sur les messages de contrôle. Dans notre approche, nous partons du principe qu'il est impossible d'identifier tous les scénarios d'attaques. C'est pourquoi nous proposons de raisonner sur un ensemble d'**actions non conformes**. Ce sont des actions unitaires portant sur un champ des messages de contrôle du protocole et qui, lorsqu'elles sont combinées, servent à la construction d'attaques complexes.
3. Nous nous basons alors sur cet ensemble d'actions non conformes et leurs effets associés afin de déterminer les apports et les limitations des versions renforcées d'OLSR.

Globalement, notre approche ne permet pas d'affirmer qu'une version renforcée d'OLSR est exempte d'attaques, mais met en évidence la potentialité de failles en se basant sur la possibilité ou non de réaliser certaines actions non conformes.

Organisation du mémoire

L'ensemble de ce document est organisé en trois chapitres.

Dans le premier chapitre, nous présentons un état de l'art des problèmes de sécurité rencontrés dans les réseaux ad hoc. Nous commençons par une présentation des principaux protocoles de routage pour ces réseaux, à savoir ceux en voie de standardisation par l'IETF [IETa]. Ensuite, nous présentons les attaques auxquelles ils sont confrontés et nous étudions les deux principales classes de solutions de sécurité proposées pour y faire face :

– La première se rapporte à la sécurité de la phase de signalisation. Nous y présentons succinctement les principales caractéristiques des différentes techniques existantes, puis nous les comparons selon des critères de coûts qu'elles introduisent.

– La seconde porte sur la sécurité de la retransmission des messages. Comme pour les méthodes portant sur la sécurité de la phase de signalisation, nous y comparons les principales techniques existantes.

Le chapitre 2 traite des travaux que nous avons effectués sur l'utilisation d'un matériel résistant à la manipulation (par exemple une carte à puce) pour sécuriser le fonctionnement des protocoles de routage. Dans le but de détecter les suppressions de messages, volontaires ou involontaires, nous présentons une couche de contrôle protocolaire basée sur des acquittements. Afin d'en analyser la sécurité, nous étudions comment cette couche réagit à différents scénarios d'attaques. Ensuite, nous analysons de manière théorique quels sont les coûts induits par cette approche.

Dans le chapitre 3, nous présentons nos travaux portant sur l'analyse de la sécurité du protocole de routage OLSR et de deux versions renforcées : la première repose sur des mécanismes cryptographiques avancés (AdvSig [RACM04]), et la seconde utilise un équipement matériel résistant à la manipulation. Ces deux méthodes sont notamment comparées entre elles ainsi qu'avec le protocole OLSR de référence tel que décrit dans la RFC 3626.

Enfin, le dernier chapitre résume ces travaux de thèse, et présente les perspectives de recherche.

Chapitre 1

Etat de l'art

La démocratisation des terminaux mobiles évolués tels que les assistants numériques personnels, les ordinateurs et téléphones portables, combinée à l'expansion des réseaux sans-fil offrent de nouvelles perspectives dans le domaine des communications. Ils permettent à leurs utilisateurs en déplacement d'être rapidement connectés et d'accéder à tout instant et en tous lieux aux services de données offerts par le réseau. Pour réaliser une telle connexion, un exemple de réseau sans fil commercial que nous utilisons aujourd'hui est le GSM. Il s'agit d'un réseau où l'espace géographique est divisé en cellules et où chaque cellule est desservie par une station de base fixe. Le rôle de cette station est de permettre aux entités mobiles situées en son sein de communiquer entre elles ou avec des entités mobiles situées dans d'autres cellules (voir figure 1.1). En d'autres termes, le fonctionnement de ce type de réseau repose sur un ensemble d'équipements centraux dédiés, interconnectés de manière filaire, structurant l'infrastructure de transport et l'acheminement des données entre les entités mobiles distantes.

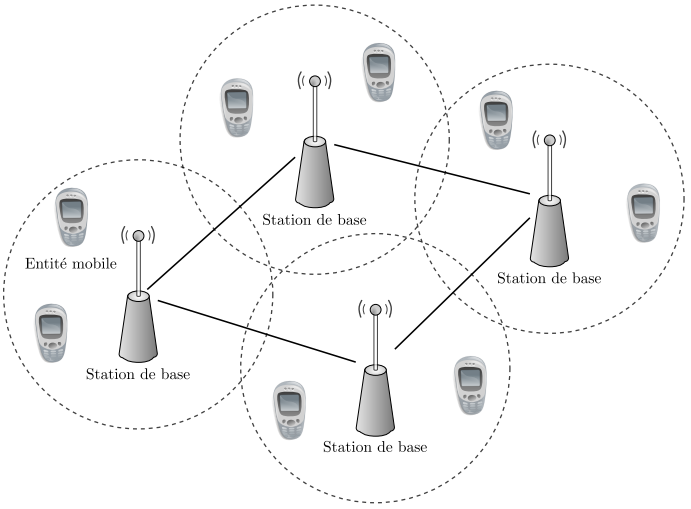


Fig. 1.1 – Architecture d'un réseau cellulaire.

Afin de réduire les coûts associés au déploiement d'une infrastructure, ou bien pour accroître la couverture des réseaux basés sur une infrastructure centralisée, de nouvelles

alternatives de réseaux totalement décentralisés ont récemment été proposées. Elles sont fondées sur une participation de chaque entité mobile dans les opérations de gestion de l'infrastructure de communication. Les entités qui forment ce réseau sont équipées d'un dispositif de communication sans fil. Ce dispositif permet naturellement à une entité de communiquer avec d'autres entités présentes dans son voisinage, c'est-à-dire avec toutes celles à portée directe de transmission. Afin de permettre des communications longues distances vers n'importe quelle entité du réseau, des entités intermédiaires (situées entre la source et la destination) sont sollicitées en tant que relais des paquets de données (voir figure 1.2). Le réseau formé est donc un réseau autonome, dans lequel les communications sont supportées par la collaboration de l'ensemble de ses participants. Ce contexte est plus connu sous le nom de réseau mobile ad hoc. Ces dernières années, ce domaine a suscité un fort intérêt de la part de la communauté réseau, ce qui s'est traduit par la formation d'un groupe de recherche spécifique de l'IETF[1] [IETa] baptisé MANet[2] [IETb].

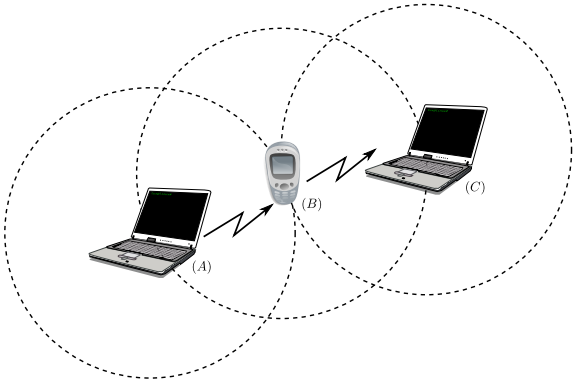


Fig. 1.2 – Exemple de transmission d'un message entre deux terminaux distants dans un réseau ad hoc : l'entité A veut communiquer avec C. Puisqu'elles sont hors de portée directe de transmission, A transmet son message vers B, qui à son tour le relaie vers C.

Dans la suite de ce chapitre, nous présentons la problématique de la sécurité dans le cadre des réseaux ad hoc. Dans une première partie, nous présentons les caractéristiques de ces réseaux et passons brièvement en revue les principales stratégies de routage multi-sauts. Ensuite, nous exposons les points de vulnérabilité qui pèsent sur le processus de routage et comment ils peuvent être exploités. Pour finir, nous discutons les différentes solutions de sécurité élaborées afin de prévenir certaines attaques et/ou de maintenir un niveau de fonctionnement acceptable.

1.1 Réseaux mobiles ad hoc

1.1.1 Définition

Ad hoc est un terme issu du latin, qui signifie littéralement « qui va vers ce vers quoi il doit aller ». Il est souvent utilisé pour décrire des solutions qui sont développées et/ou

[1]IETF : *Internet Engineering Task Force.*
[2]MANet : *Mobile Ad hoc Network.*

configurées à la volée pour répondre à un but spécifique. Dans le contexte des réseaux informatiques, un réseau ad hoc fait référence à un réseau sans fil auto-adaptatif. Il est capable de s'organiser spontanément et de manière autonome dans l'environnement dans lequel il est déployé, et ce, sans intervention humaine ou une quelconque infrastructure dédiée. Ainsi, afin de s'affranchir de la nécessité d'un câblage long et fastidieux, les entités qui le composent communiquent les unes les autres par voie aérienne au moyen d'une interface radio. De plus, contrairement aux réseaux filaires dans lesquels toutes les opérations fondamentales de découverte et de maintenance de l'infrastructure de communication sont assurées par des équipements dédiés tels que des routeurs ou des stations de base sans fil (*i.e.* des points d'accès), dans un réseau ad hoc, ces tâches sont réparties sur l'ensemble des entités. Cette décentralisation impose l'élaboration d'algorithmes totalement distribués. Ainsi, chaque entité gère ses communications à partir des informations dont elle dispose sur l'état du réseau et sur son état interne, ceci dans le but de faire émerger, à l'échelle du réseau, une structure de communication cohérente (*i.e.* la formation de chemins valides).

D'autres caractéristiques, spécifiques aux réseaux ad hoc, ajoutent des contraintes qui doivent être prises en considération lors de conception d'algorithmes et de protocoles pour ces réseaux. En particulier, la RFC 2501 [CM99] du groupe MANet de l'IETF définit les caractéristiques suivantes comme étant inhérentes aux réseaux mobiles ad hoc :

Topologie dynamique. Les entités du réseau sont potentiellement libres de se déplacer, indépendamment les unes des autres. La conséquence est que la topologie du réseau tend à changer rapidement, à n'importe quel moment, et de manière imprévisible. De plus, le fait qu'une entité quitte un groupe de communication est considéré comme un état normal qui ne doit pas perturber les autres participants.

Bande passante limitée. Actuellement, les liaisons sans fil offrent moins de capacité que les liaisons câblées. De plus, le débit obtenu sur une liaison de communication sans fil, après prise en considération des effets liés à la régulation des accès au média et les phénomènes d'atténuation du signal, de bruit, ou d'interférence, est sensiblement inférieur à son débit théorique maximal. De ce fait, la bande passante réservée à un nœud est faible, et la congestion des liaisons est courante dans ces réseaux.

Autonomie et puissance de calcul limitées. Les entités envisagées dans ces réseaux sont des terminaux légers et de taille réduite qui fonctionnent sur batterie. Par ailleurs, les capacités de calcul, de mémoire et de stockage sont plus restreintes que dans les réseaux filaires. Par conséquent, une préoccupation majeure pour ces entités est l'économie d'énergie, car elle permet d'augmenter leur durée de vie dans le réseau. D'autre part, les coûts en termes de calcul des opérations influent sur la réactivité du système et doivent de ce fait également être pris en considération lors de la conception de solutions.

Protection physique limitée. Dans un réseau filaire, les entités qui le structurent sont généralement disposées dans un local technique accessible seulement par des administrateurs autorisés. Une telle protection physique des entités n'existe pas dans un réseau ad

hoc. On peut par conséquent s'attendre à ce que leurs exigences de sécurité ne soient plus garanties ou leur fonctionnement soit altéré. En effet, à partir du moment où une entité est par exemple capturée, un attaquant peut exploiter les informations qu'elle maintient secrètes (par exemple les clés de chiffrement) afin d'accéder au contenu des messages en transit ou aux opérations du réseau. Quant au comportement global cohérent du réseau, il est obtenu grâce à la contribution de chaque entité dans des opérations spécifiques. L'altération du comportement d'une seule entité par rapport au comportement attendu s'avère alors particulièrement critique, car elle a pour effet de conduire à des perturbations dans le fonctionnement global du réseau.

1.1.2 Applications

L'origine des réseaux ad hoc remonte au début des années 1970 avec le projet ALOHA[3] [Kah75]. Commandité par l'université d'Hawaï, ce projet a été conduit dans le but de permettre aux ordinateurs des îles d'Hawaï d'être reliés entre eux par le biais d'ondes radio, et ce, dans un système de communication à un saut. Directement inspiré d'ALOHA, l'agence militaire états-unienne DARPA[4] commandita en 1973 le projet PRNet (*Packet Radio NETwork*) [Kah75] afin d'étudier les communications radio en mode paquet dans les réseaux autonomes. Le but est la construction d'un réseau capable de s'adapter dynamiquement aux changements topologiques afin de réagir à la mobilité des équipements d'une part, et de faire face à d'éventuelles pannes ou destructions, d'autre part. Dans PRNet, le protocole proposé permet le déploiement d'une infrastructure de communication multi-sauts entre différentes unités sur un champ de bataille, en passant par l'intermédiaire de véhicules communiquant par liaison radio.

Si les premiers travaux dans le contexte des réseaux ad hoc ont été menés dans une optique militaire, c'est en partie dû à leur mode de fonctionnement particulièrement bien adapté aux environnements hostiles et mobiles. Puisque le réseau est auto-organisé et qu'il ne requiert la présence d'aucune infrastructure, il peut être déployé rapidement et avec très peu d'intervention humaine dans n'importe qu'elle situation. Ainsi, les communications entre fantassins, véhicules et engins aérodynamiques deviennent autonomes et spontanées.

Le recours aux réseaux ad hoc se justifiant dès lors que l'installation d'une infrastructure s'avère inappropriée, que ce soit pour des raisons de temps, des raisons économiques ou pour des raisons physiques (par exemple pour des zones géographiques dévastées ou à accès difficiles), de nouveaux champs d'applications ont récemment été envisagés.

Ils peuvent être utilisés pour la mise en communication d'unités de secours, lorsqu'une catastrophe naturelle (telle qu'un tremblement de terre, une inondation) a détruit les infrastructures de télécommunications et que l'établissement d'une liaison satellite pour chaque entité en communication représente un coût trop élevé.

Un autre domaine d'application particulièrement bien adapté pour les réseaux ad hoc concerne les réseaux de capteurs [EGHK99]. De tels réseaux sont constitués d'équipements de taille réduite qui embarquent un système de communication sans fil et qui sont caractérisés par des ressources très limitées en termes de mémoire, de capacité calcul, de bande

[3] ALOHA : *Areal Locations of Hazardous Atmospheres.*
[4] DARPA : *Defense Advanced Research Agency.*

passante et de portée radio. Ces équipements, disséminés par centaines voire par milliers dans un environnement potentiellement hostile ou inaccessible, sont chargés de mesurer certaines propriétés physiques (telles que la température, la pression, la lumière, les sons, etc...) et de les transmettre de manière autonome. En s'affranchissant de toute limitation physique imposée par une infrastructure filaire, les réseaux ad hoc permettent la mise en communication de tous ces équipements et facilitent ainsi la transmission des informations mesurées vers un point de collecte.

Outre les applications scientifiques et militaires, des applications civiles ont commencé à tirer profit des caractéristiques des réseaux ad hoc. Ils peuvent être utilisés pour la mise en place instantanée d'un réseau reliant plusieurs ordinateurs entre eux. Ils s'avèrent particulièrement utiles lors de l'organisation d'évènements tels que des colloques, des salons afin de proposer un réseau de partage de l'information.

Un autre domaine de recherche concerne l'utilisation des réseaux ad hoc pour l'établissement de communications inter-véhicules [MJK+00]. Les objectifs envisagés par un tel usage sont, entre autres, de permettre aux véhicules d'obtenir des informations locales sur les conditions de circulation (alertes d'accidents, bulletins sur les risques d'encombrement), des informations touristiques, une téléphonie entre véhicules, et un accès à l'Internet. Le recours à un réseau ad hoc se justifie ici par ses faibles coûts de déploiement, car il permet d'éviter les investissements financiers importants que peut représenter la mise en place d'une infrastructure dédiée sur l'ensemble du réseau autoroutier.

L'informatique ambiante ou ubiquitaire prévoit, selon la définition donnée par Weiser [Wei93], une omniprésence de l'informatique dans notre quotidien à tel point qu'elle deviendrait invisible. Les réseaux ad hoc peuvent servir pour la mise en relation instantanée et transparente des équipements informatiques présents dans un environnement et rendent alors quasi réalité cette idée. Nous pouvons donner en exemple l'informatique vestimentaire [Man96] où chaque utilisateur transporte sur lui des ordinateurs qui ont la capacité de communiquer entre eux ou avec l'environnement.

Enfin, de plus en plus d'applications motivées par des aspects purement économiques ont recours aux réseaux ad hoc. Nous pouvons citer en exemple le cas des réseaux citoyens où le but est de permettre aux habitants d'une même ville de communiquer entre eux grâce à un réseau libre d'accès, et ce, sans obligation de contrôle ou de gestion par un quelconque opérateur. Ce type de réseau a suscité un fort engouement ces dernières années, ce qui s'est traduit par l'apparition d'associations dans de grandes agglomérations telles que Bruxelles [Res], Lille [LSF], Toulouse [TSF], etc...

Comme nous pouvons le constater, les réseaux ad hoc sont promis à un large spectre d'applications, qu'elles soient civiles ou militaires. Cependant, de nombreux défis se posent avant que les réseaux ad hoc puissent être réellement utilisés. Parmi ces défis, nous pouvons citer la conception de protocoles de routage et la mise en place d'architectures de sécurité adaptées à l'environnement ad hoc.

1.2 Fonction de routage

Afin de permettre les communications multi-sauts entre des nœuds hors de portée de transmission, une des fonctions fondamentales dans les réseaux ad hoc est le routage. C'est un mécanisme qui sert à trouver et maintenir des chemins, ceci dans le but de permettre, à n'importe quel moment, l'établissement d'une communication entre une paire de nœuds distants. Il fonctionne selon deux phases distinctes : une phase de signalisation assurée par des échanges de messages de contrôle afin de permettre la construction et le maintien de chemins, et une phase d'acheminement des paquets de données de bout en bout. Au regard de la phase d'acheminement, les paquets de données sont relayés par chaque nœud intermédiaire appartenant au chemin établi vers la destination. En l'absence d'équipement dédié, toutes ces opérations sont supportées par l'ensemble des nœuds qui forme le réseau ad hoc.

En raison des caractéristiques inhérentes aux réseaux ad hoc, les protocoles de routage conçus pour les réseaux filaires ne peuvent être directement utilisés. Pour fonctionner, ces protocoles doivent prendre en considération certains aspects liés à l'environnement dans lequel ils sont déployés tels que la volatilité de la topologie due à la mobilité des nœuds, l'absence d'une entité centrale de gestion, etc.... Dans les années 1990, l'amélioration des performances (réduction du nombre et de la taille des messages de contrôle, réduction des délais) des mécanismes de routage dans les réseaux ad hoc était l'une des principales problématiques. Ceci s'est traduit par l'apparition de centaines de protocoles dans la littérature, parmi lesquels seulement quelques-uns ont été soumis à normalisation par le groupe de travail MANet.

Selon la manière dont les nœuds établissent les chemins, nous pouvons distinguer trois grandes classes de protocoles de routage : les protocoles réactifs, proactifs, et hybrides. Pour cette dernière classe, il s'agit essentiellement d'une combinaison des protocoles proactifs et réactifs afin de tirer parti des avantages de chacun d'eux. Pour cette raison, nous ne discuterons pas plus en détail ce type d'approche. Dans les sections suivantes, nous donnons pour la catégorie des protocoles proactifs et réactifs un aperçu des caractéristiques de quelques-unes des principales propositions.

1.2.1 Protocoles proactifs

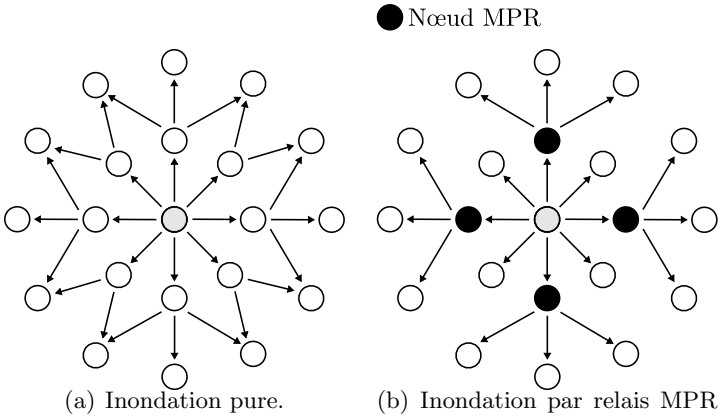
Le principe des protocoles proactifs est de maintenir à tout instant une vue globale et cohérente de la topologie du réseau, et de construire des chemins entre les nœuds avant qu'ils ne soient requis. Ces protocoles exigent que chaque nœud maintienne des informations pour chaque autre nœud du réseau, avec en particulier une table de routage indiquant par quel voisin passer pour atteindre un destinataire. Grâce à ces informations, chaque nœud dispose à tout instant d'un chemin vers n'importe quel autre nœud du réseau, et ce, sans qu'il y ait d'échange de messages de contrôle additionnels au moment de l'envoi d'un paquet de données. Afin de traiter les changements de topologie, les nœuds diffusent des messages de contrôle à travers le réseau soit périodiquement, soit en réponse à des évènements particuliers tels que la cassure d'un chemin.

Il existe plusieurs protocoles de routage, ceux-ci pouvant être distingués par le nombre

de tables maintenues par chaque nœud, et par la façon dont les mises à jour sont diffusées lorsque des changements topologiques sont détectés. Actuellement, seulement deux protocoles de routage proactifs ont été normalisés par l'IETF. Il s'agit des protocoles TBRPF (*Topology dissemination Based on Reverse-Path Forwarding*) [OTL04] et OLSR (*Optimized Link State Routing*) [JMC+01, CJ03]. Dans la suite de cette section, nous détaillerons le principe de fonctionnement du protocole OLSR.

1.2.1.1 OLSR

OLSR, proposé par Jacquet *et al.*, est un protocole de routage proactif à état de liens inspiré du protocole de routage filaire OSPF (*Open Shortest Path First*). Comme tout protocole à état de liens, OLSR opère sur la base d'échanges périodiques de message de contrôle afin d'accomplir les opérations de découverte et de maintenance de la topologie du réseau. Il possède essentiellement deux mécanismes d'optimisation qui le distingue d'OSPF. D'une part, il est fondé sur le concept de relais multi-point (MPR pour *Multi-Point Relay*) [QVL02, JLMV02] qui représente un sous-ensemble de nœuds spécifiquement sélectionnés pour assurer les opérations de génération et de retransmission des messages de contrôle pendant le processus d'inondation (voir figure 1.3). Ce mécanisme d'inondation contribue à réduire la charge réseau en nombre de diffusions de messages de contrôle. D'autre part, OLSR réduit la taille des messages de contrôle en ne diffusant que des informations partielles sur l'état des liens. Dans la suite de cette section, nous décrivons les éléments fondamentaux d'OLSR.



(a) Inondation pure.

(b) Inondation par relais MPR avec OLSR.

Fig. 1.3 – Inondation du réseau (d'après [JMC+01]).

Dans OLSR, deux types de messages de contrôle sont introduits : les messages HELLO et les messages TC (*Topology Control*). Les messages HELLO sont émis périodiquement par chaque nœud du réseau vers tous ses voisins à un saut et servent :

- A la découverte des identités des nœuds voisins ;
- A la découverte de l'état des liens ;
- Ainsi qu'à la signalisation de la sélection MPR.

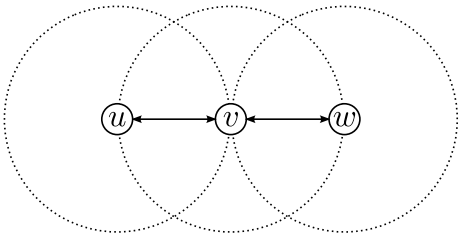
Ils encapsulent l'identité du nœud d'origine, la liste de ses liens connus, et pour chacun d'eux, l'état du lien correspondant à savoir : asymétrique (*asym*), symétrique (*sym*), MPR

(*mpr*) ou perdu. Ainsi, sur réception d'un message HELLO, un nœud extrait la liste des identités et l'information d'état de lien associée puis met à jour ses ensembles de liens (nommé *Link Set*), de nœuds voisins symétriques (nommé *Neighbor Set*) et de nœuds voisins qui ont déclaré le nœud MPR (nommé *MPR Selector Set*). Outre l'information sur le voisinage à un saut, les messages HELLO permettent à chaque nœud de prendre connaissance de leur voisinage à deux sauts. Seuls les nœuds affichant des liens symétriques avec le voisinage à un saut symétrique sont enregistrés dans l'ensemble de nœuds voisins à deux sauts (nommé *2-Hop Neighbor Set*). Sur la base des informations collectées pendant cette phase d'échange, chaque nœud est en mesure de sélectionner de manière indépendante le minimum de nœuds MPR parmi son ensemble de voisins symétriques à un saut, et à partir desquels il peut atteindre tous ses voisins symétriques à deux sauts.

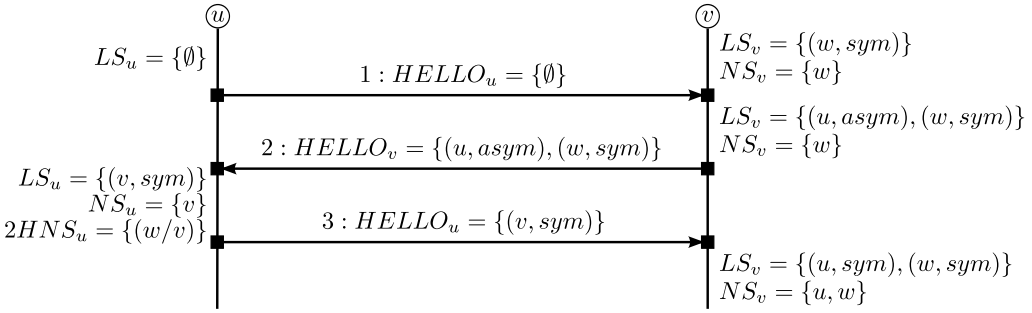
La figure 1.4 illustre la procédure de découverte des relations de voisinage. Selon la configuration réseau illustrée figure 1.4(a), au début de la procédure, le nœud u ne possède aucune information sur son voisinage, ce qui se traduit par un ensemble *Link Set* vide. En revanche, le nœud v ne connait pas u mais connait w comme étant un voisin symétrique, soit $LS_v = \{(w, sym)\}$. Premièrement, à la réception d'un message HELLO d'un nœud inconnu, tout nœud v à portée de transmission de u ajoute un état de lien asymétrique avec u dans son ensemble *Link Set*, soit $LS_v = \{(u, asym), (w, sym)\}$. Il génère en conséquence un message HELLO dans lequel il annonce tous ses états de liens courants, dont le lien avec u. Ensuite, à l'étape 2 de la figure 1.4(b), le nœud u reçoit le message HELLO de v et à partir de l'information d'état de lien $(u, asym)$ le concernant, il en déduit que v est un voisin symétrique : il l'ajoute comme tel dans ses ensembles *Link Set* et *Neighbor Set*, soit $LS_u = \{(v, sym)\}$ et $NS_u = \{v\}$. A partir du moment où v est connu comme étant un voisin symétrique par u, ce dernier extrait du message HELLO tous les voisins symétriques de v et les ajoute dans son ensemble de voisins à deux sauts, soit $2HNS_u = \{w/v\}$. Le nœud u génère ensuite un nouveau message HELLO dans lequel il annonce son ensemble courant de liens dont le lien avec v. A l'étape 3 de la figure 1.4(b), le nœud v reçoit ce message et extrait l'information de lien le concernant : il enregistre alors u comme étant un voisin symétrique dans ses ensembles *Link Set* et *Neighbor Set*.

En résumé, à partir des messages HELLO, chaque nœud acquiert une connaissance de la topologie locale du réseau, c'est-à-dire les nœuds situés à au plus deux sauts. Afin de construire une connaissance globale et récente du réseau, les nœuds sélectionnés MPR inondent périodiquement le réseau de messages TC (contenant les liens qu'ils entretiennent avec les nœuds voisins qui les ont sélectionnés MPR). Dans le but de réduire le nombre de messages durant ce processus d'inondation, seuls les nœuds MPR réémettent les messages TC. A la réception d'un message TC, chaque nœud prend connaissance d'un ensemble de nœuds atteignables ainsi que de l'identité du dernier nœud permettant de les atteindre. Ceci lui permet de maintenir à jour son ensemble topologique (*Topology Set* ou TS).

Enfin, à partir des informations partielles accumulées sur la topologie du réseau et des informations de voisinage, chaque nœud est capable de construire des chemins optimaux (en terme de distance) vers chaque destination du réseau. A chaque fois que les ensembles NS ou TS (voire les deux) sont modifiés, la table de routage est recalculée.



(a) Configuration réseau.



(b) Connaissance de la topologie locale du réseau des nœuds u et v.

Fig. 1.4 – Procédure de découverte des relations de voisinage dans OLSR.

1.2.1.2 TBRPF

A l'instar d'OLSR, le protocole TBRPF (*Topology Dissemination Based on Reverse-Path Forwarding*) [OTL04] est, lui aussi, un protocole à état de liens dans lequel chaque nœud maintient un arbre de routage des plus courts chemins. Les protocoles OLSR et TBRPF se distinguent principalement au niveau des techniques utilisées pour disséminer les informations de routage. Contrairement à OLSR pour lequel seuls les nœuds MPR diffusent l'état des liens, chaque nœud diffuse à ses voisins directs l'état des liens du réseau. Plus précisément, chaque nœud diffuse périodiquement à tous ses voisins directs, d'une part la liste de ses voisins directs et d'autre part l'arbre de routage qu'il a construit. A partir de ces informations, chaque nœud construit itérativement la topologie à une distance de deux sauts de lui ainsi qu'un arbre de plus courts chemins. En revanche, pour réduire l'utilisation de la bande passante, TBRPF possède un mécanisme d'optimisation. Ce mécanisme consiste à ne pas diffuser la totalité des informations topologiques dans chaque message de contrôle, mais seulement les différences par rapport au dernier message émis, réduisant ainsi le volume de trafic de contrôle.

1.2.1.3 DSDV

Il existe d'autres protocoles de routage proactifs tels que DSDV (*Destination Sequenced Distance Vector routing protocol*) [PB94]. DSDV est un protocole de routage à vecteur de distance basé sur l'algorithme Bellman-Ford. Dans ce protocole, chaque nœud dissémine à travers le réseau une copie de son vecteur de distance (ou table de routage), c'est-à-dire l'ensemble des nœuds qui lui est accessible et le nombre de sauts nécessaires pour les atteindre. Le principal apport de ce protocole est la résolution du problème des boucles de

routage. Grâce à l'ajout d'un numéro de séquence associé à l'identifiant de l'émetteur d'un message de contrôle, les nœuds connaissent l'information la plus récente à utiliser pour le calcul des tables de routage. Parmi les inconvénients qui font que DSDV est actuellement de moins en moins utilisé, nous pouvons citer sa lenteur pour les mises à jour des informations topologiques et ses surcoûts en communication ($O(n^2)$) pour un réseau composé de n nœuds). C'est la raison pour laquelle nous ne le détaillerons pas.

1.2.1.4 Discussion

Les protocoles proactifs offrent de bonnes performances en terme de temps de réponse, puisque lorsqu'un nœud souhaite communiquer, il dispose immédiatement des informations de routage nécessaires. En contrepartie, le principal inconvénient des protocoles proactifs est leur coût en terme d'utilisation de la bande passante. Ce coût, qui est dû à l'échange permanent de messages de contrôle nécessaires à la maintenance des tables de routage, est indépendant du nombre d'échanges de paquets de données entre les nœuds ou de la fréquence des changements de topologie. Par conséquent, il peut s'avérer excessif pour les applications dans lesquelles les nœuds échangent peu de paquets de données ou lorsque les changements de topologie sont peu fréquents.

1.2.2 Protocoles réactifs

Les protocoles réactifs optent pour une politique opposée. Plutôt que de maintenir constamment une vue globale de la topologie, ils tentent de construire des chemins uniquement lorsque ces derniers sont requis par un nœud source. On dit alors que la topologie du réseau est découverte à la demande (ou *source initiated on-demand*). Ce mode de fonctionnement a pour but d'économiser l'utilisation de la bande passante et la consommation d'énergie. Ainsi, lorsqu'un nœud cherche à joindre un correspondant pour lequel il ne connaît pas le chemin, il amorce un processus de découverte dans le réseau (généralement par inondation). Cette phase de découverte se termine dès lors que le chemin trouvé ou que toutes les possibilités sont examinées. Par ailleurs, ce chemin est maintenu jusqu'à ce que la destination devienne inaccessible ou jusqu'à ce que le chemin ne soit plus requis. Ensuite, en raison de la mobilité des nœuds, les chemins ainsi formés sont susceptibles d'être rompus. Les ruptures de liens sur les chemins sont alors traitées au moyen d'un mécanisme de maintenance, dont le but est de les identifier, puis si possible de les corriger.

1.2.2.1 AODV

Un des protocoles les plus connus dans cette catégorie est le protocole AODV (*Ad hoc On Demand Vector routing*), proposé par Perkin et Royer [PR99, PBRD03]. A l'instar de DSDV, il s'agit d'un protocole à vecteur de distance qui se distingue par une approche réactive pour la découverte des chemins. Premièrement, il utilise un numéro de séquence afin de maintenir la cohérence des informations de routage. En effet, la mobilité des nœuds entraîne des changements fréquents dans la topologie, conduisant potentiellement à l'invalidation de chemins maintenus par certains nœuds. Les numéros de séquence servent à

déterminer quels sont les chemins les plus à jour, le dernier connu étant associé aux informations les plus récentes. D'autre part, dans la phase de découverte de chemins, chaque nœud qui reçoit un message de diffusion conserve l'information sur l'identité du nœud prédécesseur. Ainsi, dans la phase de réponse initiée par le destinataire, chaque nœud valide le chemin pour lequel il est un des relais.

Si un nœud détecte un lien en aval comme étant rompu sur un chemin actif, alors dans une première étape, il tente une réparation locale par la diffusion d'une demande de chemin. En cas d'échec de cette tentative, il diffuse un message d'erreur dans lequel il signale le lien rompu. Un tel message est traité par tous les nœuds en amont qui ont connaissance d'un ou de plusieurs chemins passant par le lien rompu. Il sert, pour chaque nœud, à invalider les chemins concernés, et pour la source à relancer une recherche de chemin si aucun autre n'est connu.

1.2.2.2 DSR

A l'instar d'AODV, le protocole DSR (*Dynamic Source Routing*) proposé par Johnson et Maltz [JM96, JMH07] emploie un routage à la demande, mais il se distingue par une approche de routage à la source. La principale différence se situe au niveau du contenu des messages de contrôle. Contrairement à AODV où les informations sur les chemins sont réparties entre tous les nœuds, les messages de contrôle contiennent la liste de tous les nœuds formant les chemins à partir de la source.

1.2.2.3 Protocoles à stabilité des liens

Dans le but de permettre l'établissement de chemins plus fiables dans le temps, d'autres protocoles tiennent compte de la stabilité des liens. Dans cette optique, le protocole SSR (*Signal Stability Routing*) [DRWT97] capture des informations relatives à la puissance du signal en réception et le protocole ABR (*Associativity-Based long-lived Routing*) [Toh96] écoute et analyse la fréquence des changements dans le voisinage. A partir de ces informations, une mesure qualitative est associée à chaque lien. Les chemins sont ensuite construits de telle sorte que les liens avec un degré de stabilité élevé sont choisis en priorité.

1.2.2.4 Discussion

Le principal défaut des protocoles réactifs est qu'ils induisent un temps de réponse élevé à cause du mécanisme de découverte de chemin. En effet, le temps de réponse de cette découverte inclut la diffusion d'une requête de localisation de la destination sur le réseau, puis l'attente inhérente à l'obtention d'une réponse. Ce temps est d'autant plus important que la distance qui sépare un nœud source et d'un nœud de destination est grande.

1.2.3 Conclusion

De manière générale, les protocoles réactifs et proactifs présentent des performances différentes selon les caractéristiques du réseau. Dans le cas d'un réseau dense ou lorsque différentes paires de nœuds échangent fréquemment des données, un protocole réactif s'avère

plus coûteux qu'un protocole proactif puisque la diffusion excessive de demandes de recherche de chemin concourt à une inondation/saturation du réseau. En revanche, un protocole réactif affiche de meilleures performances qu'un protocole proactif dès lors que le trafic généré par les nœuds est faible, puisqu'il ne surcharge pas inutilement le réseau par des vérifications continuelles de la localisation des nœuds.

1.3 Sécurité du routage dans les réseaux ad hoc

Dans la plupart des travaux sur les protocoles de routage présentés aux paragraphes précédents, il est supposé que les entités sont altruistes, c'est-à-dire qu'elles sont pleinement coopératives et qu'elles participent honnêtement aux opérations de routage qui leur incombent. En particulier, les documents de spécification de ces protocoles ne décrivent actuellement aucune mesure de sécurité : il y est seulement fait état des différentes vulnérabilités auxquelles ils sont sujets. Or en raison de leurs caractéristiques, les réseaux ad hoc présentent tout autant de situations adverses que les réseaux filaires, voire même plus.

Concernant les réseaux filaires, l'infrastructure de communication, formée d'un ensemble d'entités physiques et logicielles dédiées, est administrée par une autorité légitime. Les ressources de communication offertes sont indépendantes des entités utilisatrices. C'est cette séparation franche entre les fonctionnalités de gestion et d'utilisation de l'infrastructure de communication qui permet d'assurer l'exécution correcte des opérations de gestion du réseau. Ainsi, dans le but de se protéger contre les entités malveillantes, les solutions retenues dans les réseaux filaires sont limitées à la définition et au maintien d'un périmètre de sécurité, par exemple sous la forme de mécanismes de chiffrement des communications, de pare-feu, de système de détection d'intrusion. Ces solutions visent à marquer la frontière entre le réseau administré et les réseaux voisins, c'est-à-dire à distinguer les entités situées à l'intérieur du réseau et qui doivent être protégées, de celles situées à l'extérieur et dont il faut se protéger.

Dans le contexte des réseaux ad hoc, la décentralisation est poussée à son extrême. Pour aboutir à une auto-organisation du réseau, les services gestion du réseau, dont le routage multi-sauts, sont assurés par les entités elles-mêmes. Ces entités, éventuellement indépendantes les unes des autres, font partie inhérente de l'infrastructure de communication, ce qui soulève de nombreux défis de sécurité. Les solutions basées uniquement sur la mise en place d'un périmètre de sécurité sont insuffisantes, car l'indépendance et le manque de protection physique des entités font que l'exécution correcte des opérations de gestion du réseau n'est plus garantie. En outre, l'absence d'une entité centrale entraîne la remise en question des architectures et des protocoles de sécurité retenus dans le contexte des réseaux filaires. Par ailleurs, l'utilisation de liaison sans fil introduit de nouvelles vulnérabilités puisque les communications sont exposées aux observations par n'importe quelle entité, dans la mesure où elle est munie d'un récepteur radio. Ici, le câblage physique ne représente plus un obstacle pour permettre les écoutes indésirables.

La flexibilité offerte grâce à l'auto-organisation et aux communications sans fil rend les réseaux ad hoc particulièrement vulnérables à de nouvelles attaques. Dans la suite de cette section, nous présentons une description de ces attaques.

1.3.1 Description des attaques

Une attaque contre un réseau vise essentiellement à compromettre la confidentialité et l'intégrité des informations en transit, ou de manière plus générale, à perturber son bon fonctionnement. Dans les réseaux ad hoc, selon le niveau d'intrusion des actions menées par un attaquant, on distingue généralement deux catégories d'attaques : les attaques passives et les attaques actives. Une attaque est passive lorsqu'un nœud non autorisé obtient un accès à des informations échangées sur le réseau, et ce, sans altérer les opérations du réseau. A contrario, une attaque est active lorsqu'un nœud non autorisé altère des informations en transit par des actions de modification, suppression, ou fabrication, ce qui conduit à des perturbations dans le fonctionnement du réseau.

En outre, selon le domaine d'appartenance d'un nœud, les attaques actives peuvent elles-mêmes être classées en deux catégories, à savoir les attaques externes et internes. Tandis que les attaques externes sont réalisées par des nœuds qui n'appartiennent pas au domaine du réseau, les attaques internes sont menées par des nœuds compromis qui sont autorisés à participer au fonctionnement du réseau. Étant donné que les attaquants font d'ores et déjà partie du réseau de nœuds autorisés, les attaques internes sont généralement plus pernicieuses et difficiles à détecter que les attaques externes.

Enfin, les attaques peuvent être de type individuelles ou par collusion. Les attaques individuelles sont menées par un seul nœud attaquant. Puisque les capacités de communication et de calcul de l'attaquant sont en général similaires à celles des autres nœuds du réseau, ces attaques demeurent relativement simples, et sont d'autant plus limitées que des mécanismes de sécurité sont mis en œuvre. En revanche, rien n'empêche à des nœuds attaquants de mutualiser leurs informations et leurs ressources, en exploitant les connexions qu'ils ont entre eux. Ces attaques par collusion, issues de plusieurs nœuds répartis à différents endroits dans le réseau, sont généralement plus évoluées et plus pernicieuses. Par ailleurs, en raison de l'intervention de plusieurs nœuds intermédiaires, leur détection et l'identification précise de leur origine sont rendues plus complexes.

1.3.1.1 Attaques passives

Dans cette catégorie, la technologie de communication sans fil sous-jacente constitue une vulnérabilité qui peut aisément être exploitée par un attaquant. Selon le contrôle d'accès au support défini par la norme IEEE 802.11 [IEE07], les nœuds communiquent par voie aérienne avec un accès partagé au média de type évitement de collisions ; plus connu sous le nom de CSMA/CA[5]. Une attaque consiste alors pour un attaquant à configurer son interface de communication en mode *promiscuous* afin de capturer et analyser toutes les transmissions réalisées sur le média. A partir des messages de contrôle capturés, un attaquant est alors en mesure d'extraire des informations stratégiques telles que la topologie du réseau, la localisation de certains nœuds, ou les identifiants des nœuds (adresse IP et MAC), ces informations servant de support à la réalisation d'attaques plus complexes.

La détection des attaques passives est particulièrement difficile, car le fonctionnement du réseau n'est en soi pas affecté : les ressources et les fonctions critiques du réseau de-

[5]CSMA/CA : *Carrier Sens Multiple Access with Collision Avoidance.*

Couche réseau	Attaques	Atteinte
Application	Ecoute indiscrète, corruption des données, vol d'identité	Confidentialité, intégrité
Transport	Vol de session, inondation de requêtes (SYN)	Disponibilité, intégrité
Réseau	Tunnels, détournement, modification et destruction des messages (de contrôle et de données), épuisement des ressources, inondation, etc...	Disponibilité des ressources de communication, intégrité du routage
Liaison	Analyse du trafic, détournement des protocoles d'accès au média, saturation de la bande passante	Disponibilité
Physique	Brouillage, interception	Disponibilité

Tab. 1.1 – Attaques contre les réseaux ad hoc par couche de la pile réseau.

meurent intactes. Elles représentent malgré tout une atteinte à la confidentialité et à l'anonymat. Des solutions de protection contre les observations peuvent reposer sur l'utilisation de protocoles de chiffrement tels que IPsec afin d'assurer la confidentialité des données transmises. Cependant, la solution IPsec n'est pas adaptée au contexte des réseaux ad hoc. La première raison provient du fait que les nœuds ne peuvent initialement pas avoir confiance les uns dans les autres. Cette caractéristique rend difficile la phase d'établissement d'une association de sécurité par un échange de clés qui est requise dans IPsec. En outre, même si IPsec est efficace pour protéger la confidentialité des données, il n'a pas été conçu dans l'optique de sécuriser les informations de signalisation.

1.3.1.2 Attaques actives

Les attaques actives peuvent être menées à différents niveaux de la couche d'abstraction du modèle OSI[6][Int94] (voir tableau 1.1). Comme tout autre type de réseau sans fil, un réseau ad hoc est vulnérable à des attaques au niveau de la couche physique. La raison est que les ondes radio sont très sensibles aux interférences. Ainsi, sans nécessairement prendre part au réseau ad hoc formé, un attaquant est susceptible de générer des signaux à une fréquence proche de celle utilisée dans le réseau afin de brouiller les transmissions [WTSL04]. En conséquence, toute communication devient impossible.

Au niveau de la couche liaisons de données, des protocoles ont été définis afin de maintenir la connectivité à un saut entre des nœuds voisins. Ils visent à offrir aux nœuds un accès équitable au média de communication. Pour ce faire, leur fonctionnement repose sur des échanges de trames de contrôle et suppose une coopération inconditionnelle entre les nœuds. De manière évidente, un nœud n'est pas contraint à suivre les spécifications de ces protocoles. Dès lors qu'un attaquant sature le média en émettant des trames de contrôle ou de données, les autres nœuds à proximité se trouvent dans l'incapacité de

[6]OSI : *Open Systems Interconnection.*

communiquer. En raison de l'indisponibilité des ressources de communication, on parle alors de déni de service. Des attaques spécifiques contre la couche de contrôle d'accès au support (MAC[7]) définie par la norme IEEE 802.11, et exploitant certaines caractéristiques du protocole, ont déjà été mises en évidence [GKF02, KV03].

Les protocoles définis au niveau de la couche réseau servent à étendre la connectivité à un saut à tous les nœuds dans le réseau. C'est donc à ce niveau que fonctionnent les protocoles de routage et le mécanisme de retransmission des paquets de données. Un simple détournement du fonctionnement normal de ces protocoles entraine une perturbation des communications, et l'ensemble du réseau peut être paralysé. La sécurité de la couche réseau est donc primordiale dans la mesure où le but du réseau est avant tout de mettre en relation des nœuds et d'acheminer leurs données.

Au niveau de la couche application, les attaques sont communes à tous les types de réseau et leur mise en œuvre dépend de l'application visée. Les mécanismes de contre-mesures envisagés ont pour la plupart recours à la cryptographie afin de protéger les échanges de bout en bout.

Nous pouvons constater qu'à la fois les attaques au niveau de la couche physique, liaison de données et réseau portent atteinte à la disponibilité des services de communication offerts par le réseau. Par conséquent, la seule protection des protocoles de la couche réseau n'est en soi pas suffisante pour en assurer le bon fonctionnement. Dans le meilleur des cas, une protection à ce niveau s'avère efficace contre les attaques présentes sur cette couche. Néanmoins, la protection des couches inférieures fait appel à des techniques d'accès au média pour la couche liaison de données et des techniques de transmission radio pour la couche physique. Etant donné que ces techniques étaient en dehors de notre domaine de compétences, nous avons orienté nos travaux vers des solutions de protection des protocoles de la couche réseau. Ainsi, dans la suite de cette section, nous détaillons les principales attaques contre les protocoles de routage.

Attaques sur la phase de signalisation La génération et le traitement des messages de contrôle sont entièrement sous la responsabilité des nœuds qui forment le réseau. Or ces derniers représentent un des éléments de base dans le processus de routage, car ils servent à établir et maintenir les relations de connectivité entre les nœuds. A partir du moment où un attaquant génère des messages de contrôle dont le contenu n'est pas conforme au regard de la logique du protocole, ou bien s'il ne participe pas correctement à la retransmission de ces messages, alors l'intégrité de la topologie peut être compromise. En raison de leur importance, ils constituent la principale cible des attaquants pour perturber le fonctionnement des réseaux ad hoc. Les attaques contre les messages de contrôle prennent de nombreuses formes telles que la fabrication, la modification, le rejeu, le refus de retransmission. Elles incluent par ailleurs tous les types d'attaquant décrits précédemment.

Dans une attaque par modification, l'attaquant met à jour certains champs des messages de contrôle qu'il reçoit et les retransmet, sans suivre les recommandations du protocole. La modification de nombreux champs est susceptible d'influer l'intégrité des données de routage. Figurent parmi ces champs : le numéro de séquence, le nombre de sauts, les

[7]MAC : *Medium Access Control.*

identités des nœuds (adresses IP), le descriptif du chemin en construction. Par exemple, en modifiant la métrique de routage ou le descriptif du chemin, un attaquant parvient à manipuler la construction des chemins en les faisant apparaître soit plus longs, soit plus courts que ce qu'ils sont réellement. Dans le cas de DSR, un attaquant est en mesure de modifier la séquence de nœuds formant un chemin dans un message de demande de chemin, en supprimant et/ou ajoutant un nouveau nœud ou en intervertissant l'ordre des nœuds.

Les messages de contrôle échangés entre les nœuds décrivent un état de la topologie du réseau à un instant donné. Une attaque qui exploite ces informations est le rejeu. Elle consiste en la réémission en un point du réseau de messages de contrôle caducs (car anciens). La conséquence est que des nœuds sont amenés à réaliser des mises à jour sur une base d'informations qui n'a plus lieu d'exister, entraînant alors des incohérences dans les chemins construits. Une particularité de cette attaque est qu'elle est effective même si les messages de contrôle sont protégés par une empreinte ou une signature numérique.

De manière générale, il en résulte un détournement du trafic de son cheminement normal. Par détournement de trafic, nous entendons le fait qu'un attaquant est capable d'attirer le trafic vers lui-même, de créer des boucles de routage, de créer des chemins sous optimaux, ou des créer des chemins non existants. En outre, des attaquants en collusion peuvent empêcher un nœud source de trouver un chemin vers une destination, conduisant alors à un partitionnement du réseau.

La non-retransmission est une attaque selon laquelle un nœud supprime les messages de contrôle qu'il reçoit au lieu de les retransmettre. Bien que triviale à mettre en œuvre, la réalisation de cette attaque, ne serait-ce que par un nombre limité d'attaquants, nuit sévèrement au fonctionnement du réseau. Elle entraine non seulement une réduction de la connectivité globale et du nombre de chemins de communication disponibles, mais aussi un raccourcissement de la durée de vie du réseau puisque le trafic sera moins bien répartis entre les nœuds.

Il existe également des attaques qui visent en particulier la phase de maintenance de chemin des protocoles réactifs. Si la destination ou un nœud intermédiaire le long d'un chemin actif se déplace, le nœud en amont de la rupture du lien diffuse un message d'erreur vers le nœud source. Tous les nœuds amenés à traiter ce message (et a fortiori la source) invalident le chemin vers cette destination dans leur table de routage. Un attaquant peut tirer avantage de ce mécanisme de deux façons :

- En l'absence d'événement de rupture, l'attaquant forge un faux message d'erreur. Il s'ensuit un déclenchement d'opérations coûteuses de maintenance et de reconstruction d'un chemin valide entre la source et la destination victimes de l'attaque.
- L'attaquant ne retransmet pas un message d'erreur pourtant valide, ce qui conduit à une augmentation du temps pour détecter et corriger l'erreur.

Attaques sur la phase d'acheminement des paquets de données En sus des attaques contre les messages de contrôle des protocoles de routage, un attaquant peut perturber l'opération d'acheminement des paquets de données. Dans ce type d'attaque, le but n'est pas de compromettre l'intégrité de la topologie construite. Ici, un attaquant participe aux phases de découverte et de maintenance des chemins. En revanche, il ne relaie pas correctement les paquets de données, même s'il appartient au chemin établi entre une

source et une destination. Il agit par des actions de suppression, de modification, de rejeu sur les paquets de données qu'il reçoit. Toujours dans un but de perturber l'opération d'acheminement, une autre attaque plus subtile consiste à ralentir la retransmission de paquets de données sensibles aux retards. Lorsqu'un attaquant ne fait que supprimer les paquets de données, cette attaque est assimilée à une attaque de non-coopération.

Autres attaques spécifiques Une attaque inhérente aux réseaux ad hoc et qui nuit au fonctionnement de tous les protocoles de routage est l'attaque du *wormhole*, terme en référence aux trous de ver en science-fiction qui sont des raccourcis entre deux points éloignés dans l'espace. Cette attaque, décrite par Hu *et al.* [HPJ03], se manifeste par un détournement de trafic : un paquet capturé en un point du réseau est rejoué en un autre point du réseau plus ou moins distant. Pour mener cette attaque, l'attaquant établit un tunnel privé entre deux points éloignés du réseau au moyen d'une liaison câblée ou d'une liaison sans fil longue portée. Ensuite, au cours de cette attaque, l'attaquant capture les transmissions sur une des extrémités, les envoie (éventuellement de manière sélective) à travers le tunnel vers l'autre extrémité, d'où elles sont rejouées dans le réseau. La conséquence directe de cette attaque est l'établissement d'une vision erronée des nœuds au regard de leur connectivité dans le réseau. En outre, de sorte à paraître virtuellement invisible, l'attaquant n'effectue aucune manipulation sur les transmissions relayées. La sévérité de l'attaque vient du fait qu'elle est difficilement détectable et que d'autre part, elle est effective même dans le cadre d'un réseau où l'authentification, l'intégrité et la confidentialité sont préservées.

En détournant le trafic vers lui-même, un attaquant se donne la possibilité d'analyser un maximum d'informations. A partir du moment où l'intégralité des paquets de données est retransmise vers la destination, l'attaque est quasiment transparente. Seuls des délais supplémentaires dans l'acheminement des données pourront apparaître. En revanche, si l'ensemble des paquets reçus par l'attaquant est supprimé, alors des dégradations sévères dans les communications de bout en bout seront occasionnées. Cette attaque, décrite par Hu *et al.* [HPJ02], est plus connue sous le nom de trou noir (ou *black hole*). Dans le but de rendre l'attaque moins intrusive (et donc plus difficilement détectable), la suppression des paquets peut être réalisée de manière sélective, c'est-à-dire par la suppression des paquets pour une source ou une destination spécifique, par la suppression d'un paquet toutes les t secondes, par la suppression aléatoire des paquets, etc. . ..

Enfin, les réseaux ad hoc se caractérisent par des ressources limitées en termes d'énergie et de bande passante. En effet, les nœuds fonctionnent sur batterie et même si ces dernières sont de plus en plus performantes en terme d'autonomie, elle n'en demeure pas moins une ressource capitale. Les réseaux ad hoc sont alors particulièrement vulnérables aux attaques brutales de type déni de service. Une des plus simples est l'attaque par harcèlement selon laquelle un attaquant inonde le réseau de paquets superflus (*i.e.* sans signification particulière), ceci dans le but d'augmenter la charge sur le réseau ainsi que d'épuiser les ressources des nœuds.

1.4 Mécanismes de protection existants

La littérature est riche de propositions de contre-mesures aux attaques discutées en section 1.3.1. Parmi celles-ci, nous pouvons distinguer deux catégories principales : celles qui gèrent les attaques contre les messages de contrôle, et celles qui traitent plus particulièrement les attaques contre l'acheminement des paquets de données. Dans la suite de cette section, nous passons en revue et discutons les principaux avantages et limitations pour chacune de ces catégories.

1.4.1 Protection de la signalisation

Pour traiter les actions illégitimes, différentes solutions de sécurité basées sur les classes de protocole décrites en section 1.2 ont été proposées. Tandis que certaines de ces solutions reposent sur des mécanismes cryptographiques et sur une infrastructure de gestion et de distribution de clés, d'autres s'appuient sur des techniques de détection d'intrusion. Dans cette section, nous présentons certaines de ces solutions, nous discutons les failles de sécurité les plus importantes qui subsistent malgré l'introduction de service de sécurité, et nous étudions les coûts qu'elles induisent.

1.4.1.1 Solutions utilisant la cryptographie asymétrique

Une première ligne de défense pour contrecarrer les attaques consiste à assurer les services d'authenticité et d'intégrité des informations qui sont échangées à l'aide de primitives cryptographiques. La différence entre ces solutions repose essentiellement sur le choix des mécanismes cryptographiques, et sur la façon dont est traité le caractère variant de certaines données. Ainsi, pour contrecarrer les attaques sur les données statiques au fil des retransmissions, ces dernières sont authentifiées par l'initiateur du message à l'aide de primitives cryptographiques classiques telles que la signature numérique ou le calcul de HMAC[8]. Cette authentification apporte la garantie qu'aucune modification n'est occasionnée par un nœud intermédiaire. En revanche, la protection des données variables fait l'objet de propositions plus originales, basées pour certaines sur des calculs récursifs.

ARAN Sanzgiri *et al.* ont proposé le protocole sécurisé ARAN (*Authenticated Routing for Ad hoc Networks*) [SDL+02] qui prévoit l'utilisation de la cryptographie à clé publique pour sécuriser la construction des chemins des protocoles réactifs tels que AODV. Il suppose l'existence d'un serveur d'authentification, dont le rôle est de gérer la distribution des certificats pour les nœuds autorisés dans le réseau. Ce certificat, signé par le serveur d'authentification, contient l'identité du nœud (*i.e.* l'adresse IP), sa clé publique, une date de création et une date d'expiration. Ainsi, avant de rejoindre le réseau, chaque nœud doit, au préalable, récupérer un certificat auprès du serveur qui lui servira à signer les messages de contrôle.

[8] HMAC : *keyed-Hash Message Authentification Code.* Il s'agit d'un type de code de message d'authentification calculé à partir d'une fonction de hachage en combinaison avec une clé secrète.

ARAN s'appuie sur deux mécanismes d'authentification. Le premier consiste en une authentification de bout en bout afin qu'un nœud destinataire puisse d'une part authentifier l'origine d'un message de contrôle, et d'autre part vérifier la non-modification des données statiques (*i.e.* l'adresse du nœud source et destinataire) pendant le transit. Le second est une authentification de saut en saut dans lequel chaque nœud sollicité dans un processus de recherche ou de maintenance de chemin utilise un certificat pour s'authentifier auprès d'autres nœuds voisins. En particulier, un nœud doit, pour chaque message de contrôle qu'il reçoit, vérifier le certificat fourni par le nœud précédent, puis s'il est valide, l'utiliser pour vérifier la signature. Ensuite, s'il n'est pas le destinataire du message, il supprime le certificat et la signature du nœud précédent, signe le message avec sa propre clé privée, et appose son certificat avant de le rediffuser.

ARAN offre des services d'authentification, d'intégrité et de non-répudiation. Une particularité d'ARAN concerne les informations incluses dans le message de demande de chemin. Contrairement aux spécifications d'AODV dont il s'inspire, ce message ne contient ni compteur de sauts devant être incrémenté au fur et à mesure des retransmissions, ni chemin spécifique associé à la source. Ainsi, ARAN ne garantit plus le chemin le plus court (exprimé en nombre de sauts), mais le plus rapide, c'est-à-dire celui qui donne lieu en premier à une réponse de chemin. En raison de cette approche, les messages de contrôle ne sont finalement formés d'aucun champ variant au fil de leur retransmission. L'authentification des messages effectuée de saut en saut assure alors qu'un attaquant indépendant, qu'il soit interne ou externe au réseau, ne peut ni créer des boucles de routage, ni rediriger le trafic en insérant par exemple des identités de nœuds illégitimes dans les messages de découverte de chemins.

Néanmoins, en termes de sécurité, une limitation vient du fait que dans la phase de maintenance, la véracité d'une information de rupture de lien n'est pas vérifiée. Cette absence de vérification offre l'opportunité à un attaquant interne d'inoculer de fausses annonces de ruptures, ceci dans le but d'invalider des liens (et des chemins) pourtant opérationnels. Un tel comportement est un déni de service qui peut entraîner une surconsommation des ressources, car des procédures de recherche de chemins devront être reconduites.

Une autre limitation vient du fait que les mécanismes proposés ne permettent pas de déterminer si les nœuds intermédiaires relaient les messages pour lesquels ils ont été sollicités. En d'autres termes, il ne permet pas de contrer les attaques par non-participation. Puisqu'aucun compteur de sauts n'est utilisé dans les messages de contrôle, cette caractéristique offre l'opportunité à un attaquant d'augmenter les délais lors de la formation d'un chemin, soit en retardant la propagation de la demande, soit dans le pire des cas, en supprimant la demande. Dans la phase de maintenance, la suppression d'une annonce de rupture de lien peut également s'avérer très néfaste pour le réseau, car des délais supplémentaires seront occasionnés pour d'une part détecter, et d'autre part corriger la rupture.

ARAN utilise la cryptographie à clé publique pour authentifier les nœuds de saut en saut, ce qui induit des coûts de calcul importants. Toutes les opérations de signature et de vérifications de signature peuvent s'avérer extrêmement coûteuses en puissance de calcul et énergie, d'autant plus si le nombre de messages devient important. Dans la mesure où un nœud n'arrive pas à effectuer ces traitements en temps réel, il peut en résulter des

pertes de certains messages valides (du fait de débordements de tampon).

SAODV Zapata et Asokan ont proposé une extension de sécurité pour le protocole AODV nommée Secure AODV [ZA02]. Contrairement à l'extension ARAN pour laquelle les données variables des messages de contrôle sont retirées, l'idée principale de SAODV consiste à faire usage d'une signature numérique (créée par cryptographie à clé publique) pour protéger les données statiques des messages de contrôle, puis de recourir à des chaînes de hachage pour protéger l'intégrité de la partie non statique qu'est le compteur de sauts. En particulier, étant donnée une fonction de hachage H, lorsqu'un nœud initie une demande de recherche de chemin, il choisit aléatoirement un nombre *seed* et inclut, en plus du compteur de sauts $hopCount = 0$, le nombre maximal de sauts que doit parcourir le message $maxHopCount = TTL$, la valeur $H^{maxHopCount}(seed)$ et la valeur $s = seed$. Avant de retransmettre un tel message, un nœud intermédiaire vérifie la signature numérique et si elle est valide, augmente la valeur du nombre de sauts $hopCount$ de un et remplace la valeur précédente de s par $H(s)$. Ensuite, pour vérifier l'authenticité du compteur de sauts courant $hopCount = k$, un nœud vérifie que la valeur $H^{maxHopCount}(seed)$ est égale à $H^{maxHopCount-k}(s)$. Ce procédé assure qu'à la réception d'un message de contrôle, la valeur du compteur de sauts est exacte, dans le sens où elle n'a pas été faussement décrémentée par un attaquant sur le chemin.

Comme pour ARAN, des services d'authentification, l'intégrité et la non-répudiation de bout en bout, entre le nœud source et destination, sont ainsi obtenues. Cependant, l'utilisation des chaînes de hachage pour contrer les manipulations illégales sur le compteur de sauts reste limitée, car même s'il est vrai que des chemins plus courts qu'ils ne le sont en réalité ne peuvent pas être annoncés, rien n'empêche un attaquant d'augmenter arbitrairement le nombre de sauts ou de le laisser inchangé. En outre, dans le cas où plusieurs attaquants sont en collusion, une attaque de type *wormhole* peut être menée. A travers cette attaque, l'attaquant parvient à manipuler le compteur de sauts et à raccourcir la longueur d'un chemin, ceci de manière transparente pour les autres nœuds.

1.4.1.2 Solutions utilisant la cryptographie symétrique

SRP Papadimitratos et Haas ont proposé SRP (*Secure Routing Protocol)* [PH02], une extension de sécurité pour les protocoles de routage réactif à la source, dont en particulier DSR. Cette extension se fonde sur l'ajout d'un en-tête de sécurité aux messages de recherche d'un chemin entre une source et une destination (laquelle contient un numéro de séquence de façon à garantir leur fraicheur), et sur l'existence d'une association de sécurité entre ces deux nœuds. Selon cette approche, le nombre d'opérations cryptographiques est considérablement réduit puisque seule une authentification mutuelle des nœuds de bout en bout, à savoir entre la source et la destination, est requise. Plus spécifiquement, ces nœuds utilisent leur association pour authentifier les messages de demande et de réponse à un chemin au moyen d'un code d'authentification de message (MAC). Ainsi, la destination est capable de détecter toute modification sur les données statiques d'un message de demande de chemin. Du point de vue de la source, le MAC lui permet de vérifier l'intégrité du chemin découvert inclus dans le message sur lequel il porte, et d'en authentifier son origine.

L'authentification mutuelle de bout en bout fait de SRP une approche relativement légère, puisqu'il requiert au total seulement quatre opérations cryptographiques pour sécuriser la phase de découverte de chemin, et ce, indépendamment de la longueur du chemin entre une paire de nœuds communicants.

En revanche, il présente certains défauts qui limitent son utilisation. Premièrement, SRP ne sécurise pas la phase de maintenance des chemins. Les auteurs suggèrent l'utilisation d'un protocole complémentaire. Ensuite, SRP ne permet pas de contrer les attaques par modification - menées aussi bien par des nœuds internes qu'externes - portant sur les informations de routage sujettes à changement lors des multiples retransmissions. Ceci est dû à l'absence d'authentification des messages de saut en saut. Par exemple, un attaquant peut, sans effort, altérer le contenu de la liste de nœuds inclue dans un message de demande de chemin, soit en supprimant soit en ajoutant des nœuds. De tels comportements ont des impacts directs sur le mécanisme de sélection des chemins par la source puisque soit les chemins découverts seront inutilisables soit ils auront moins de chance d'être sélectionnés.

Ariadne Hu et Johnson ont proposé Ariadne [HPJ02, HPJ05], un protocole de routage réactif sécurisé basé sur le protocole DSR. Dans le but de proposer un mécanisme adapté à la nature spontanée des réseaux ad hoc et aux capacités de calcul parfois restreintes des nœuds qui le composent, les auteurs proposent trois méthodes d'authentification :

- La première est basée sur le partage d'un secret entre chaque paire de nœuds et requiert par conséquent une configuration au préalable de $n * (n-1)/2$ clés, avec n le nombre de nœuds dans le réseau.
- La seconde est basée sur l'utilisation de la cryptographie asymétrique, ce qui rend particulièrement coûteuses les opérations de génération et de vérification de signatures.
- La dernière s'appuie sur le partage d'un secret entre chaque paire de nœuds communicants combiné à une authentification par diffusion qui est supportée par le protocole TESLA (*Time Efficient Stream Loss-tolerent Authentication*) [PTSC00]. Cette méthode permet à une destination d'authentifier la source d'un message diffusé sur le réseau, à la condition que les nœuds aient une synchronisation approximative de leurs horloges.

Contrairement à SRP, Ariadne vise à limiter les attaques par modification des données variables inclues dans les messages de contrôle. Pour ce faire, Ariadne permet à un nœud destinataire d'authentifier l'initiateur de la demande de chemin grâce à code d'authentification de message (MAC). Ensuite, il est requis que chaque nœud intermédiaire impliqué dans un processus de découverte de chemin signe la nouvelle information contenue dans le message reçu avant de le propager. Ce mécanisme permet à un nœud destinataire d'authentifier chaque nœud inclus dans la demande avant d'envoyer un message de réponse. Ariadne a recours à l'utilisation de chaînes de hachage à sens unique pour garantir l'intégrité de la liste des nœuds incluse dans la demande. Grâce à ce mécanisme, aucun nœud intermédiaire peut supprimer ou ajouter l'adresse d'un nœud présent dans la liste sans que cette modification passe inaperçue.

Dans la phase de maintenance, pour signaler une rupture de lien sur un chemin, un nœud envoie en direction de la source un message d'erreur signé dans lequel il annonce le

lien à l'origine de l'erreur. Tous les nœuds appartenant au chemin (et a fortiori la source), après authentification de son émetteur, réagissent à ce message en invalidant le chemin.

La prévention contre les attaques par modification offerte par Ariadne génère un surcoût : au fur et à mesure de la propagation d'un message de découverte, un MAC est apposé par chaque nœud intermédiaire ; il s'ensuit que la taille de ce message augmente linéairement avec la longueur du chemin. Concernant les différentes méthodes d'authentification d'Ariadne, il est à noter que l'utilisation du protocole TESLA présente l'avantage d'être plus souple dans le sens où elle s'affranchit de la distribution des clés privées à toutes les paires de nœuds. En revanche, TELSA occasionne une augmentation du délai d'authentification, ce qui vient au détriment de la réactivité du protocole.

Dans Ariadne, les mécanismes de sécurité mis en œuvre sont efficaces contre les attaquants indépendants. Un attaquant seul ne peut pas supprimer les identités de nœuds dans le chemin annoncé dans un message de demande sans être détecté, puisque la vérification des MAC par la source ou par la destination échouera. En revanche, il affiche des faiblesses face aux attaques par collusion de nœuds internes. Dans leurs travaux [BV04], Buttyán et Vajda montrent comment un attaquant peut s'ajouter en tant que successeur d'un nœud intermédiaire légitime dont il n'est pas voisin dans un chemin en construction entre une source et une destination. Cette attaque aboutit éventuellement à l'utilisation d'un chemin inopérant par la source, puisqu'aucun message ne peut franchir le lien entre le nœud légitime et l'attaquant. Dans d'autres travaux [ABV06], Ács *et al.* décrivent une autre attaque possible où deux attaquants en collusion parviennent à supprimer une séquence de nœuds intermédiaires dans le descriptif d'un chemin en construction. Bien que physiquement situés à différents endroits le long du chemin en construction, la particularité de ces deux attaquants est qu'ils partagent une même identité réseau. En outre, Ariadne souffre des mêmes limitations qu'ARAN, à savoir qu'il ne traite ni les attaquants qui ne retransmettent pas les messages de contrôle pour lesquels ils ont été sollicités, ni les attaquants qui injectent de fausses annonces de rupture de lien.

endairA Dans le but de corriger certains défauts de sécurité du protocole Ariadne, Buttyán et Vajda ont proposé une adaptation nommée facétieusement endairA [BV04]. Contrairement à Ariadne qui protège la construction du chemin dans la phase de protocole de découverte, ici ce ne sont que les réponses (en point à point) à une demande de chemin qui sont protégées par des signatures numériques. Au fur et à mesure que la réponse à la demande de chemin revient vers la source, chaque nœud intermédiaire vérifie son appartenance au chemin, vérifie que les nœuds suivant et précédent dans le chemin sont ses voisins, et appose une signature numérique. Cette signature est calculée à partir de tous les champs, incluant le chemin et les signatures des nœuds intermédiaires précédents. Finalement, il diffuse le message vers le prochain nœud sur le chemin de retour vers le nœud source. A la réception d'une réponse, le nœud source s'assure que le message a bien été délivré par un voisin, puis vérifie toutes les signatures pour le chemin formé.

De manière similaire à Ariadne, l'inconvénient est que la taille du message de réponse augmente linéairement avec le nombre de nœuds appartenant au chemin.

En outre, endairA est vulnérable à une attaque par collusion de nœuds internes. Nous montrons comment deux nœuds attaquants et non adjacents sur un chemin entre une

source et une destination, parviennent à raccourcir le descriptif d'un chemin en construction. Contrairement à l'attaque décrite par Ács *et al.*, cette attaque ne nécessite pas le partage d'une identité réseau. Ici, au lieu de recourir à une liaison dédiée, les deux attaquants forment un tunnel en s'appuyant uniquement sur l'existence de chemins fournis par le réseau lui-même et sur l'utilisation de techniques d'encapsulation. Le principe de cette attaque, qui peut être vue comme un cas particulier de l'attaque du tunnel, a été décrit par Khalil *et al.* [KBS05]. Examinons la figure 1.5 où il est illustré un scénario possible de l'attaque. Les attaquants sont dénotés par a_1 et a_2. On suppose que le nœud source

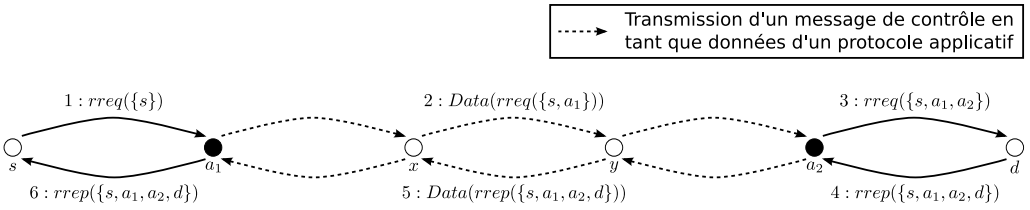


Fig. 1.5 – Exemple d'une attaque par encapsulation des messages de contrôle.

s envoie un message de demande de chemin vers la destination d. A la réception de ce message, le premier attaquant a_1 ajoute son identité au chemin, encapsule le message dans un paquet d'un protocole applicatif, puis le transmet au second attaquant a_2 à travers le chemin formé de la séquence de nœuds $\{a_1, x, y, a_2\}$. Comme les nœuds x et y reçoivent non plus un message de contrôle mais un paquet de données, ils le traitent comme tel en s'assurant uniquement de sa retransmission. A la réception de ce paquet, a_2 extrait la demande de chemin, y ajoute son identité, puis le retransmet vers d. Finalement, d reçoit une demande où le descriptif du chemin est plus court que ce qu'il est en réalité, car formé de la séquence de nœuds $\{s, a_1, a_2, d\}$ au lieu de $\{s, a_1, x, y, a_2, d\}$. d génère un message de réponse pour ce chemin. Ce message est de nouveau encapsulé par a_2 avant d'être transmis vers a_1. Etant donné que les attaquants sont des nœuds internes au réseau, toutes les signatures apposées au message de réponse peuvent être vérifiées. Dès lors que le chemin ainsi construit est choisi par la source, les nœuds attaquants ont le choix de délivrer correctement les paquets de données ou bien de les supprimer. A travers cette attaque, un sous ensemble de nœuds le long d'un chemin entre une source et une destination est court-circuité de la phase de construction. Par conséquent, elle atteint aussi l'extension de sécurité Ariadne.

Extensions de sécurité pour OLSR Dans la littérature, plusieurs extensions de sécurité pour le protocole OLSR ont été proposées [HTR+04, ACJ+03]. Leur point commun réside dans l'utilisation de signature numérique pour assurer l'authentification et intrinsèquement l'intégrité des messages de contrôle. Une telle authentification peut être réalisée soit de saut en saut, soit de bout en bout.

Dans Secure OLSR [HTR+04], Hafslund *et al.* ont proposé une approche d'authentification de saut en saut dans laquelle chaque nœud signe les paquets OLSR au fur et à mesure de leur retransmission. Ainsi, à la réception d'un paquet OLSR (un tel paquet pouvant contenir plusieurs messages OLSR de type HELLO et TC), un nœud intermédiaire

vérifie la signature du nœud précédent, la retire, puis appose sa propre signature. Cette approche permet d'inclure dans le calcul de la signature numérique les champs devant être modifiés en transit, tels que le TTL (*Time To Live*) et le nombre de sauts. Cependant, la signature assure seulement que le nœud qui a transmis le trafic est bien celui qui a signé le paquet, mais n'apporte aucune garantie sur l'intégrité du paquet original. Ici, les auteurs suggèrent l'utilisation de clés symétriques et d'une fonction de hachage telle que SHA-1 pour la génération des signatures numériques.

De manière similaire à Secure OLSR, Raffo *et al.* [ACJ+03] ont proposé une extension de sécurité pour le protocole OLSR basée sur l'utilisation de signatures numériques. Une première différence se situe au niveau du type des données protégées. Dans leur approche, une signature numérique est associée à chaque message de contrôle OLSR (*i.e.* HELLO ou TC) et non plus à chaque paquet OLSR. Ensuite, les auteurs proposent une approche d'authentification de bout en bout selon laquelle un nœud récepteur d'un message de contrôle authentifie le nœud d'origine plutôt qu'un nœud intermédiaire dans son cheminement. Ici, les champs TTL et nombre de sauts ne sont pas protégés par la signature, car ces derniers doivent être modifiés en transit par chaque nœud intermédiaire. En remplacement au TTL et pour déterminer si un paquet est trop ancien et s'il doit être rejeté, les auteurs proposent d'horodater chaque message OLSR. En outre, cette information temporelle est un indicateur qui sert à détecter les rejeux de messages de contrôle.

L'authentification des messages de contrôle représente une première ligne de défense pour contrecarrer « efficacement » les attaques externes contre le protocole OLSR. Or à elle seule, l'authentification n'empêche pas l'inoculation de fausses informations de routage par des attaquants internes au réseau. Pour pallier ce problème, d'autres méthodes plus originales ont été proposées et sont détaillées ci-après.

AdvSig En sus des mécanismes proposés dans [ACJ+03] et dans le but d'empêcher l'injection d'informations d'état de liens non valides par des nœuds attaquants internes, Raffo *et al.* ont proposé un schéma de signatures nommée *Advanced Signature* [RACM04, ACL+05]. La schéma proposée s'appuie sur le fait que la topologie du réseau évolue selon une séquence chronologique précise et en particulier, que l'état de lien entre deux nœuds au temps $(t+1)$ dépend directement de l'état de ce même lien au temps (t). L'idée est que pour tout message reçu d'un voisin, un nœud stocke les informations relatives à ses liens, puis les réutilise en tant que preuve de validité de son ensemble d'état de liens dans les messages qu'il émet ultérieurement. Pour éviter toute fabrication de faux messages, ces informations sont signées et encapsulées dans un message spécifique nommé AdvSig.

AdvSig+ Selon une étude présentée par Chen *et al.* [CXL06], il apparaît que les surcoûts induits par les opérations de calculs et de vérifications des signatures numériques sont à l'origine des pertes notables de paquets. Leur expérimentation montre qu'environ 70% des paquets de données ne sont pas délivrés lorsque AdvSig est utilisé, contre environ 20% pour la version d'OLSR sans mécanisme de sécurité. Une autre faiblesse de cette approche soulevée par les auteurs vient du fait qu'aucune preuve n'est requise dans la phase de déclaration d'un lien asymétrique entre deux nœuds. Il en résulte qu'un nœud attaquant est en mesure, de par une fausse déclaration d'un lien asymétrique, de conduire un autre

nœud à croire qu'il existe une relation symétrique entre eux. Pour corriger cette faille de sécurité, les auteurs suggèrent l'ajout d'une nouvelle preuve d'état de lien. Ensuite, des chaînes de hachage sont utilisées afin de réduire les coûts associés au traitement des messages de contrôle. Une autre particularité d'AdvSig+ est que, pour empêcher un nœud de disséminer de fausses informations sur la topologie dans les messages TC, tous les nœuds génèrent des messages TC dans lesquels ils déclarent leur ensemble de nœuds MPR, en plus de leur ensemble de nœuds qui les ont sélectionnés comme MPR. Par un recoupement des informations reçues, chaque nœud vérifie la cohérence des relations MPR déclarées, et met à jour sa connaissance locale sur la topologie en conséquence.

Comparativement à AdvSig, cette approche présente l'avantage de réduire sensiblement le nombre d'opérations de vérification de signature, ce qui la rend plus adaptée aux réseaux à faible bande passante et à puissance de traitement limitée. Néanmoins, certaines exigences telles que la connaissance de taille et de la durée de vie du réseau avant sa formation, et la présence d'un serveur central (accessible à tout instant et par tout nœud) dont le rôle est de générer puis de distribuer les chaînes de hachage, limitent l'utilisation de cette approche. En outre, contrairement aux spécifications d'OLSR, chaque nœud doit générer des messages TC. Par conséquent, cette solution affiche des coûts plus importants en terme de nombre de messages de contrôle. Enfin, la méthode proposée pour remédier aux problèmes des modifications présente l'inconvénient majeur de n'être valide que sous l'hypothèse d'un modèle d'attaquants sans collusion.

Discussion Dans le tableau 1.2, nous comparons ces différentes solutions de sécurité. Pour chaque solution, nous précisons le mécanisme cryptographique utilisé, leurs coûts en termes de nombre d'opérations cryptographiques et de taille des messages. Concernant les protocoles réactifs, le nombre total d'opérations de génération et de vérification de signatures et la taille d'un message pour l'établissement d'un chemin (incluant la recherche et la réponse), est donné en fonction de y, le nombre de nœuds sur le chemin entre une source et une destination. Dans notre analyse, nous considérons que le chemin formé suite à la phase de recherche est celui emprunté (en sens inverse) par le message de contrôle dans la phase de réponse. Dans le cas du protocole OLSR, les coûts par réception d'un message sont donnés en fonction de M, le nombre moyen de nœuds voisins, de M_r[9], le nombre moyen de nœuds MPR sélectionnés par un nœud et de N, le nombre total de nœuds dans le réseau. Pour estimer les coûts cryptographiques sur les messages TC, nous utilisons les résultats des travaux menés par Adjih *et al.* [AJV02], où il est prouvé que l'inondation MPR coûte en moyenne $\left(\frac{M_r N}{M}\right)$ retransmissions. Par ailleurs, gardons à l'esprit que le coût d'une opération utilisant la cryptographie à clé asymétrique est habituellement plus élevé que celui d'une opération symétrique.

Pour AdvSig+, il est à noter que tous les nœuds génèrent un message TC, et non plus uniquement les nœuds sélectionnés MPR comme cela est le cas dans la version de référence. Il en résulte une augmentation du nombre total dans le réseau d'opérations de génération et de vérification de signatures.

A partir des mécanismes de prévention fondés sur l'authentification des messages de

[9]Dans une étude menée par Jacquet *et al.* [JLMV02], il est démontré que $M_r \leq (9\Pi^2 M)^{\frac{1}{3}}$.

Protocole	Routage	Mécanismes cryptographiques	Coûts		
			Génération	Vérification	Taille
ARAN [SDL+02]	Réactif	Signature à clé publique	$O(2y)$	$O(2y)$	$O(1)$
SAODV [ZA02]	Réactif	Signature à clé publique, chaîne de hachage	$O(1)$	$O(2y)$	$O(1)$
SRP [PH02]	Réactif	MAC	$O(1)$	$O(1)$	$O(1)$
Ariadne [HPJ02]	Réactif	Signature à clé publique ou MAC ou MAC couplé à TESLA	$O(y)$	$O(3y)$	$O(y)$
endairA [BV04]	Réactif	Signature à clé publique	$O(y)$	$O(3y)$	$O(y)$
S-OLSR (paquet) [HTR+04]	Proactif	MAC	Hello : $O(1)$ TC : $O\left(\frac{M_r N}{M}\right)$	$O(1)$	$O(1)$
S-OLSR (message) [ACJ+03]	Proactif	Signature à clé publique ou MAC	$O(1)$	$O(1)$	$O(1)$
AdvSig [ACJ+03]	Proactif	Signature à clé publique	Hello : $O(M)$ TC : $O(1)$	Hello : $O(M)$ TC : $O(M_r)$	Hello : $O(M)$ TC : $O(M_r)$
AdvSig+ [CXL06]	Proactif	Signature à clé publique, chaîne de hachage	$O(1)$	$O(1)$	Hello : $O(M)$ TC : $O(M_r)$

Génération : nombre de signatures, symétriques ou asymétriques, calculées pour un chemin entre une source et une destination dans le cas des protocoles réactifs, et par message de contrôle pour le protocole OLSR.
Vérification : nombre de signatures, symétriques ou asymétriques, vérifiées pour un chemin entre une source et une destination dans le cas des protocoles réactifs, et par récepteur pour le protocole OLSR.

Tab. 1.2 – Comparaison des solutions de sécurité basées sur la cryptographie.

contrôle par chaque nœud, les attaques contre les protocoles de routage sont considérablement réduites. Comme nous venons de le voir, les attaques selon lesquelles un nœud interne modifie les informations de routage en transit dans l'intention de tromper d'autres nœuds ne sont que limitées, mais pas complètement éliminées. Par ailleurs, dans le cas où un attaquant agit seul, les attaques par usurpation d'identité peuvent être complètement empêchées. En revanche, cela est moins évident lorsque des attaquants forment une collusion.

Ces techniques affichent d'autres limitations. Entre autres, nous pouvons citer les coûts calculatoires importants liés à la génération et la vérification des signatures, les coûts induits sur la taille des messages, ainsi que les difficultés associées à la mise place d'une architecture de gestion de clés afin de gérer la confiance entre les nœuds dans le réseau (*i.e.* gestion des clés secrètes ou certificats). Notons par ailleurs que ces coûts sont aggravés lorsque la cryptographie asymétrique est utilisée. En outre, ces techniques ne permettent pas l'identification d'un attaquant.

Même si les procédés cryptographiques peuvent s'avérer efficaces pour contrer un

nombre important d'attaques, ils demeurent inopérants pour contrer l'attaque du tunnel (ou *wormhole*) présentée en section 1.3.1.2. La raison est que cette attaque ne requiert aucune génération ou modification des messages par l'attaquant. Ainsi, aussi bien les nœuds internes (c'est-à-dire les nœuds en possession des autorisations nécessaires pour participer aux opérations réseau) que les nœuds externes peuvent la mettre en œuvre. Cette attaque nuit tout particulièrement aux protocoles qui s'appuient sur une phase de découverte de voisinage par échange de messages de contrôle, car elle peut conduire à des conflits dans les relations de voisinage établies. La plupart des travaux de recherche dans ce domaine tentent de prévenir l'attaque du *wormhole* grâce à des techniques de calcul de distances, afin d'identifier les messages qui transitent au-delà de la portée de transmission. Ces techniques, qui exigent en général un matériel spécialisé, sont fondées sur des informations géographiques [HPJ03, HPJ06], des antennes directionnelles [HE04, LP04] ou la durée de transmission des messages [CBH03, Kor05]. D'autres travaux se fondent sur les informations de connectivité locale afin de détecter des structures interdites présentes lorsqu'un wormhole existe [BDV05b, MGD07] ou sur la reconnaissance des équipements radio grâce à des particularités des signaux (empreintes) [RC07]. Pour plus d'informations, le lecteur pourra se reporter à [Gor06] pour une analyse des différentes solutions existantes.

1.4.1.3 Détection d'intrusion

Puisque les systèmes de sécurité basés sur la cryptographie sont parfois coûteux et qu'ils ne permettent pas d'empêcher toutes les catégories d'attaques, d'autres travaux, souvent considérés comme complémentaires à la prévention, ont porté sur la conception de mécanismes de détection d'intrusions. Le principe général des systèmes de détection d'intrusions consiste en la surveillance et l'examen du comportement du protocole qui doit être protégé, de sorte à identifier et à réagir contre d'éventuelles attaques. Lors de l'examen, sont pris en considération le comportement de l'intrus et les comportements normaux et espérés du protocole. Selon la nature des comportements examinés, deux stratégies de détection se distinguent : la détection basée sur les anomalies et la détection basée sur les mauvais usages. Dans le premier cas, les actions réalisées sont comparées aux comportements normaux et attendus par le protocole. Si elles sont considérablement différentes, alors le protocole présente des anomalies et fait l'objet d'une intrusion. Dans le second cas, les actions réalisées sont comparées à une base de signatures représentant les actions illégales habituellement effectuées par un intrus. Une intrusion est détectée dès lors qu'une signature est identifiée parmi les actions réalisées. Compte tenu de la dynamique du réseau et du caractère parfois imprévisible des situations, la définition de la base de signature se révèle particulièrement difficile. A cela s'ajoute l'absence d'une entité centrale pour collecter et vérifier le trafic, ce qui a pour conséquence de compliquer la détection précise d'un intrus. En raison des limitations des systèmes conventionnels basés sur un répertoire de signatures, une grande partie de la littérature utilise la détection d'anomalies. Dans la suite de cette section, nous discutons quelques-unes de ces solutions.

Pour représenter le comportement normal attendu par un protocole, des techniques basées sur la spécification ont été proposées. En particulier, une détection basée sur une telle spécification nécessite dans une première étape la définition d'un ensemble de contraintes

qui décrit les opérations correctes du protocole étudié. Ensuite, la conformité de l'exécution du protocole est contrôlée au regard des contraintes définies.

Tseng *et al.* proposent une modélisation du protocole AODV sous la forme d'une machine à états finis [TBK+03]. Dans leur approche, ils distinguent deux types de nœuds, les nœuds ordinaires et les nœuds moniteurs. Ce sont ces derniers qui ont pour responsabilité de détecter les anomalies dans les traces observées sur le réseau, en utilisant la machine à états finis d'AODV comme référence. Certaines hypothèses fortes telles que l'unicité des adresses MAC (servant à identifier un nœud à l'origine d'une anomalie) font que cette approche n'est pas viable dans le contexte des réseaux ad hoc.

Dans le but de détecter les attaques sur la phase de découverte du voisinage, Orset *et al.* [OAC05] ont proposé une modélisation formelle du protocole OLSR sous la forme d'une machine à états finis étendue (EFSM[10]). L'EFSM définit le comportement correct du protocole sous la forme d'évènements d'entrée/sortie avec ou sans paramètres, de prédicats à satisfaire et d'actions à effectuer. Le processus de vérification et de détection des anomalies consiste à comparer les traces d'exécution du protocole, c'est-à-dire les messages reçus et envoyés, avec sa modélisation. La comparaison est effectuée à l'aide d'un algorithme de recherche en arrière [ACC+04] défini par les auteurs. Néanmoins, cette approche est limitée dans le sens où seules les attaques locales, qui violent directement le modèle OLSR, sont détectées.

Wang *et al.* ont proposé une technique de détection d'intrusion basée sur la vérification sémantique du protocole OLSR [WLMG05]. L'idée de base est de dériver les propriétés sémantiques implicites de la définition du protocole, afin de décrire le comportement valide dans les mises à jour des informations topologiques. Ces propriétés sont vérifiées par chaque nœud, à partir des messages de contrôle (HELLO et TC) qu'il émet et reçoit de son voisinage direct, et une intrusion est détectée lorsqu'une de ces propriétés n'est pas satisfaite. Toutefois, puisque les vérifications réalisées par un nœud concernent uniquement sa connaissance locale du réseau, toutes les attaques contre le protocole ne peuvent pas être détectées (par exemple, les attaques distantes et distribuées). Un autre désavantage est lié à l'absence d'un modèle formel pour spécifier et vérifier les propriétés sémantiques. Or à partir du moment où la sémantique ne permet pas de capturer finement tous les aspects d'un comportement, il est possible pour un attaquant de contourner le mécanisme de protection.

Dans d'autres travaux assez similaires, Orset et Cavalli [OC06] puis Cuppens *et al.* [CCBRT07, CCBNR07] ont précisément cherché à dériver formellement les propriétés sémantiques du protocole OLSR à partir de sa spécification. Pour ce faire, ils ont fait appel à la logique déontique et temporelle. Les travaux de Cuppens *et al.* se distinguent au niveau des réactions déclenchées : des contre-mesures de sécurité sont prises dès qu'une propriété n'est pas satisfaite. Ainsi, pour limiter les impacts de nœuds attaquants, les auteurs proposent de les exclure de la formation des chemins. A l'instar des travaux de Wang *et al.*, seules les traces de trafic local sont utilisées pour la vérification sémantique de la connaissance locale.

Adnane *et al.* [AdSJBM07] ont utilisé la notion de confiance afin d'analyser les proprié-

[10]EFSM : Extented Finite State Machine.

tés sémantiques du protocole OLSR. Les différentes relations de confiance établies implicitement entre les nœuds à travers les échanges de messages de contrôle sont exprimées à l'aide d'un langage formel. A partir de cette formulation qui décrit le comportement correct de chaque nœud OLSR, les auteurs ont proposé un mécanisme de détection d'intrusion basé sur la méfiance [AdSJBM08]. Contrairement à l'approche de Wang *et al.* [WLMG05], chaque nœud vérifie, en plus de sa propre connaissance locale du réseau, la cohérence du comportement des autres nœuds et valide les relations de confiance établies implicitement. Cette vérification est réalisée sur la base d'une corrélation entre les informations contenues dans les différents types de messages de contrôle reçus. Comme cela est souligné par les auteurs, lorsqu'un comportement anormal est détecté, il n'est pas toujours imputable à un nœud précis, mais à un groupe de nœuds qui inclut l'attaquant.

Discussion Ces techniques présentent l'avantage de détecter des intrusions sans recourir à une base de connaissances sur les signatures d'attaques, tout en produisant peu de fausses alarmes (*i.e.* détection de faux). En comparaison avec les solutions basées sur la cryptographie, elles affichent des coûts plus légers en terme de puissance de calcul. Néanmoins, un défi réside dans la définition des contraintes qui décrivent les opérations correctes du protocole : si le modèle n'est pas décrit assez finement (complexité), alors les attaques ne seront pas détectées efficacement. Entre autres, des attaques éventuellement plus complexes, et où les spécifications du protocole ne sont pas directement transgressées, peuvent passer inaperçues.

1.4.2 Coopération entre les nœuds

Les mécanismes décrits précédemment se révèlent efficaces pour fournir des services de sécurité classiques que sont l'authentification et l'intégrité, et ainsi assurer qu'aucune modification non conforme au protocole n'a été apportée sur les messages de contrôle. Alors qu'ils permettent de réduire le nombre d'attaques possibles (aussi bien externes qu'internes), ils s'avèrent inopérants lorsqu'il s'agit de traiter le problème de non-participation des nœuds dans l'opération de retransmission des paquets de contrôle et de données. Or dans le contexte des réseaux ad hoc, cette opération est fondamentale, car elle rend possible l'établissement et le maintien des communications entre les nœuds distants, sans recourir à une infrastructure prédéfinie.

C'est la raison pour laquelle, en sus des mécanismes de prévention, plusieurs travaux de recherche ont porté sur la définition de mécanismes de sécurité visant tout particulièrement à détecter les nœuds non coopératifs dans les opérations de retransmission des paquets ou à atténuer les effets néfastes de leurs comportements. Il existe essentiellement trois stratégies pour renforcer l'opération de retransmission. L'une d'elle consiste en l'identification des nœuds qui affichent des comportements de non-coopération, puis en la mise en œuvre de réactions en conséquence. Figurent dans cette catégorie les techniques basées sur la surveillance du comportement des nœuds, des acquittements explicites des paquets et le principe de conservation de flot. Une autre stratégie consiste non plus à détecter, mais à fournir des encouragements aux nœuds qui participent aux opérations du réseau. Dans cette catégorie figurent les systèmes basés sur des échanges d'argent virtuel. Une

dernière approche est la tolérance à la non-coopération. Elle consiste à concevoir l'opération de retransmission de sorte que son fonctionnement soit le moins compromis possible par d'éventuels nœuds non coopératifs. Dans la suite de cette section, nous présentons et discutons certaines de ces solutions.

1.4.2.1 Systèmes de réputation

La réputation est un indicateur qui est employé par les individus dans la vie de tous les jours pour d'une part valoriser une image, et d'autre part faciliter la prise de décision. Dans le cadre des réseaux ad hoc, la réputation peut être définie comme étant le niveau de participation d'un nœud dans l'opération de retransmission des paquets, tel que vu par d'autres nœuds. La réputation d'un nœud est ensuite utilisée pour différencier les bons nœuds (à savoir ceux qui coopèrent) des mauvais.

De façon générale, ces solutions utilisent trois mécanismes distincts : (1) un mécanisme local de surveillance afin d'observer le comportement des nœuds et de déterminer leur degré de réputation selon les actions observées, (2) un mécanisme de sanction ou d'isolation afin de protéger le réseau contre les comportements malveillants, et parfois (3) un mécanisme de dissémination des informations collectées.

Au vu de l'une des caractéristiques des réseaux ad hoc qu'est l'accès partagé au média de communication, tous les nœuds peuvent écouter et traiter les informations transmises par leurs voisins directs (à un saut), même si ces dernières ne leur sont pas destinées. Il s'agit du mode de fonctionnement *promiscuous* dans lequel peuvent être configurées les interfaces réseaux sans fil. Ainsi, selon les hypothèses que les nœuds transmettent avec la même puissance et que toutes les communications sont omnidirectionnelles, un nœud est en mesure de vérifier la conformité de comportement de ses voisins en fonction du contenu de chacune des trames qu'ils émettent (voir figure 1.6).

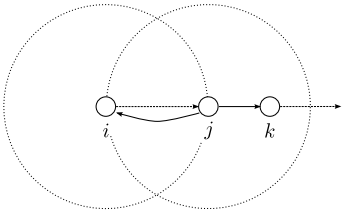


Fig. 1.6 – Transmission omnidirectionnelle et observation passive : le nœud i transmet un paquet à j qui le retransmet à k, et i est capable d'entendre cette retransmission.

Les premiers travaux exploitant cette caractéristique pour aborder le problème de non-coopération sont ceux de Marti *et al.* [MGLB00]. Dans leurs travaux, les auteurs traitent le cas du protocole de routage DSR en proposant un système fondé sur deux composants : le chien de garde (*Watchdog*) et l'évaluateur de chemins (*Pathrater*). Le *Watchdog*, utilisé localement par chaque nœud, a pour rôle de contrôler que le nœud suivant sur le chemin procède bien à l'opération de retransmission des paquets de données. Cette surveillance est possible car, s'agissant d'un protocole de routage par la source, chaque nœud intermédiaire

connaît le prochain nœud vers la destination grâce du descriptif du chemin inclus dans l'entête du paquet. Lorsqu'une action observée ne correspond pas à un résultat attendu, le nœud observateur comptabilise un échec de retransmission. A partir du moment où le compteur pour un nœud dépasse un seuil fixé, l'information est reportée au *Pathrater*. Le *Pathrater* est ensuite utilisé pour sélectionner les chemins les plus fiables entre une source et une destination, en évitant les nœuds qui ont été détectés comme non coopératifs. La faiblesse de cette approche est qu'elle ne permet ni de sanctionner ni d'isoler les nœuds qualifiés de non coopératifs. Ces derniers, bien qu'exclus de la construction des chemins, sont toujours en mesure d'utiliser les ressources des autres nœuds dans le réseau pour leurs propres communications.

D'autres travaux ont traité les problèmes de coopération entre les nœuds par le biais de systèmes de réputation plus complexes. Dans ce registre, Buchegger et Le Boudec ont proposé un système de surveillance distribué et collaboratif nommé CONFIDANT (*Cooperation of Nodes, Fairness In Dynamic Ad hoc NeTworks*) [BB02]. L'objectif de CONFIDANT est d'exclure les nœuds qui ne jouent leur rôle dans les opérations de routage, que ce soit au niveau du processus d'acheminement des données ou au niveau du processus de découverte des voisins. Il a été conçu comme étant une extension de sécurité des protocoles de routage réactifs à la source tels que DSR. Le système, maintenu par chaque nœud du réseau, définit quatre composants qui interagissent les uns les autres :

- Un moniteur.
- Un système de réputation.
- Un gestionnaire de confiance.
- Un gestionnaire de chemins.

Le rôle du moniteur consiste à vérifier, sur la base d'observations directes, le comportement des nœuds à l'égard des opérations de routage. Ces observations servent à classifier un nœud comme bienveillant ou malveillant. Dès lors que le moniteur détecte un événement suspicieux ou une incohérence (c'est-à-dire une déviation par rapport au motif de communication attendu), il en informe le système de réputation. Ce dernier a pour rôle de maintenir à jour les valeurs de réputation pour chaque nœud observé. Cependant, la perte d'un paquet peut être provoquée par plusieurs facteurs tels que la mobilité des nœuds, les congestions, les interférences ou collisions de trames. Etant donné ces caractéristiques, il est difficile, voire impossible, de discerner une perte accidentelle d'une destruction intentionnelle d'un paquet par un nœud non-coopératif. Dans le but de limiter les effets négatifs dus aux imprécisions du mécanisme de détection d'une part, et pour accélérer le mécanisme d'apprentissage des informations servant à évaluer les nœuds d'autre part, les valeurs de réputation sont éventuellement échangées entre les nœuds. Ainsi, en plus de ses observations directes, un nœud intègre les valeurs de réputation de voisins dans le calcul de réputation des autres nœuds. Dans CONFIDANT, seules les valeurs de réputation négatives sont diffusées à travers des messages d'alarme. C'est le rôle du gestionnaire de confiance de prendre la décision d'envoyer ce type de messages, puis de déterminer dans quelle mesure les informations reçues doivent être prises en considération pour le calcul de la valeur de réputation d'un nœud. Finalement, le gestionnaire de chemins évalue les chemins à partir de la topologie du réseau et des informations des autres composants. Il calcule les chemins les plus sûrs en utilisant comme métrique les valeurs de réputation des nœuds, et peut décider de rejeter les demandes de retransmission de paquets pour les

nœuds affichant une faible réputation.

Dans le même ordre d'idée, Michiardi et Molva ont proposé un système nommé CORE (*COllaborative REputation mechanism*) [MM02]. Le système a également été conçu comme étant une extension de sécurité des protocoles de routage réactifs. Le fonctionnement est très similaire à celui de CONFIDANT : des moniteurs observent le trafic afin de détecter des anomalies de comportement des nœuds et les résultats sont transmis à un système de réputation. Ici, la réputation d'un nœud, qui est une représentation de sa contribution aux opérations de routage, est calculée comme étant la combinaison de :

- La réputation subjective, calculée à partir des observations locales directes.
- La réputation indirecte, calculée à partir des valeurs de réputation positives fournies par d'autres nœuds).
- La réputation fonctionnelle, calculée en fonction du poids associé à une tâche spécifique.

Ainsi, les nœuds présentant un haut niveau de réputation peuvent utiliser le réseau pour leurs propres communications tandis que les nœuds à basse réputation sont graduellement exclus de la topologie du réseau. Afin d'atténuer les effets non désirés dus aux erreurs temporaires telles que la congestion réseau, les débordements de tampons ou les collisions, seuls les nœuds dont la réputation est inférieure à un seuil fixé sont exclus. Un mécanisme de rédemption offre la possibilité à un nœud de réintégrer progressivement le réseau en coopérant de nouveau aux opérations de routage.

Discussion D'un point de vue général, ces systèmes limitent les effets néfastes des attaques par non-coopération. Dans le cadre de simulations [MM02, BB02, BB03], il est montré que les dégâts sur les débits du réseau sont moindre, même lorsque le nombre de nœuds non coopératifs est élevé (allant jusqu'à 40 %). La définition d'une architecture distribuée, où chaque nœud intègre un système local de détection, fait qu'ils sont bien adaptés à la nature de l'environnement. De plus, l'utilisation d'un *Watchdog* présente l'avantage de ne pas engendrer de coûts additionnels pour identifier et évaluer le comportement des nœuds. Néanmoins, ces systèmes affichent plusieurs points de vulnérabilités qui peuvent être exploités par un attaquant :

1. Les messages de réputation échangés entre les nœuds représentent une première source d'attaques. Se pose d'une part le problème de l'altération des informations de réputation en chemin, et d'autre part le problème de leur véracité. Un attaquant, qui adopte un comportement conforme à l'égard des actions évaluées et qui affiche par conséquent une bonne réputation, peut faussement accuser d'autres nœuds de sorte à dégrader leur réputation. En cas de réussite de ce type d'attaque (sachant que les chances sont d'autant plus importantes si plusieurs nœuds attaquants sont en collusion), des nœuds pourtant conformes seront éventuellement isolés. Dans le but de protéger l'intégrité des informations de réputation disséminées, des mécanismes additionnels et basés sur la cryptographie doivent donc être envisagés. Ensuite, dans une tentative pour minimiser l'influence de cette attaque, il est suggéré dans CORE que seules les valeurs de réputations positives soient disséminées à travers le réseau. Cependant, rien n'empêche des attaquants en collusion de diffuser de faux rapports afin de maintenir un niveau de réputation acceptable, et ainsi éviter l'isolement.

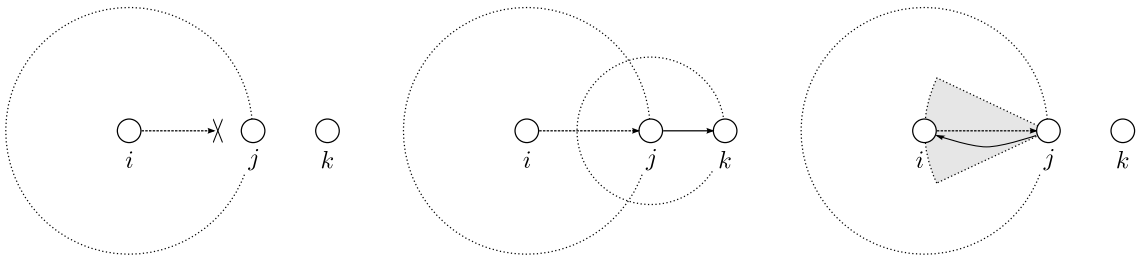
Dans des travaux ultérieurs, Buchegger et Le Boudec [BLB04] ont proposé une amélioration de leur système de confiance pour traiter ce problème. Dans leur cas, un nœud accepte une information indirecte de réputation associée à un autre nœud à la condition qu'elle ne diffère pas largement de son avis local ; dans le cas contraire, elle est considérée comme incompatible et est par conséquent ignorée. Bansal et Backer ont proposé OCEAN (*Observation-based Cooperation Enforcement in Ad hoc Networks*) [BB03] où, contrairement à CONFIDANT ou CORE, seules les observations directes sont utilisées dans le calcul de réputation d'un nœud. Cette approche présente l'avantage de réduire considérablement la complexité du système de réputation puisque les fraudes sur les échanges d'informations indirectes sont écartées. De plus, elle n'induit aucun coût en volume de trafic supplémentaire, contrairement à CONFIDANT ou CORE pour lesquels le coût sur une période de diffusion de messages de réputation est de $O(N^2)$, où N est le nombre de nœuds dans le réseau.

2. Une autre limitation vient du fait que des actions de suppression limitées en nombre (*i.e.* conduites de manière sporadique et/ou ciblée), même si elles sont détectées, passent inaperçues dans la note finale attribuée à un nœud. Par exemple, un attaquant peut supprimer des paquets qu'il est censé retransmettre uniquement pour un sous-ensemble de nœuds. Or à partir du moment où la proportion de nœuds avec qui il coopère est plus élevée que celle avec qui il ne coopère pas, alors il peut ne pas être classifié comme non coopératif par le système.

3. L'attaque Sybil, initialement présentée par Douceur dans [Dou02], consiste pour un attaquant à manipuler de multiples identités. La première forme de manipulation se caractérise par la création de nombreuses identités. Elle a pour conséquence d'enlever toute imputabilité des actions non conformes à l'attaquant, dans la mesure où les valeurs construites par le système de réputation deviennent obsolètes. La seconde se caractérise par l'usurpation d'une identité réelle dans le réseau. Elle offre l'opportunité à un attaquant de se bâtir une plus large influence sur le réseau en profitant de la bonne réputation associée à l'identité de sa victime, ou bien de dégrader la réputation de sa victime à son insu (de façon à ce qu'elle soit par exemple exclue du réseau). En d'autres termes, un attaquant est en mesure de perturber ou bien d'utiliser largement le réseau, sans y apporter sa contribution. La résistance d'un système de réputation à une attaque de type Sybil dépend de la facilité avec laquelle il est possible de forger des identités. En général, des solutions pour résoudre ce problème requièrent la mise en place d'une autorité centrale de certification afin d'offrir des services d'identification / authentification des nœuds.

4. Ils affichent également des défauts importants qui sont dus à l'utilisation d'un *Watchdog* comme technique de base pour l'observation et l'évaluation du comportement des nœuds. De l'aveu même des auteurs [MGLB00], le *Watchdog* peut soit échouer dans la détection d'un comportement non conforme, soit détecter des nœuds comme étant non conformes alors qu'ils ne le sont pas dans les cas suivants :
   - Un nœud peut ne pas être en mesure de capter une transmission. Cette situation peut survenir dans les cas où : les nœuds n'utilisent pas tous le même canal de communication, les antennes ne sont pas omnidirectionnelles, les nœuds utilisent des puissances d'émission différentes, une collision se produit au niveau du nœud

récepteur (voir figure 1.7(a)).
– Même si un nœud émetteur entend la transmission d'un de ses voisins, aucune garantie n'est apportée quant au fait qu'elle ait été correctement reçue par le prochain nœud appartenant au chemin vers la destination. Ce phénomène peut apparaître à la suite d'une collision entre le nœud relais et le prochain nœud vers la destination, lorsque les nœuds utilisent des puissances d'émission différentes et que la puissance d'émission du nœud relais est trop faible pour que la transmission soit reçue par le troisième nœud (voir figure 1.7(b)) ou que les nœuds utilisent des antennes directionnelles (voir figure 1.7(c))



(a) Collision à la réception : le nœud i transmet un paquet à j, mais ce dernier ne le reçoit pas à cause d'une collision sur le canal.

(b) Puissance variable d'émission : le nœud i n'entend pas la retransmission de j.

(c) Antenne directionnelle : le nœud j prétend à i que la retransmission a été réalisée.

Fig. 1.7 – Limitations de la technique *Watchdog*.

Par ailleurs, la technique du *Wacthdog* n'est pas adaptée à l'identification des nœuds non coopératifs dans la phase de signalisation. Le problème est qu'il n'est fait aucune distinction entre les situations de suppression conformes aux règles définies par le protocole et les situations de suppression malveillantes. En général, les protocoles de routage réactifs possèdent un mécanisme d'optimisation qui consiste en la suppression des messages de contrôle dupliqués. Lorsqu'un message de recherche de chemin est inondé, chaque nœud reçoit de multiples copies de ce message, même si les valeurs de certains champs tels que par exemple, le nombre de sauts et la séquence de nœuds du chemin vers la source, varient. Suivant ce mécanisme d'optimisation, chaque nœud ne transmet qu'une seule fois un même message (identifié par son numéro de séquence), les autres étant supprimés, réduisant ainsi considérablement les coûts de l'inondation. Or de telles suppressions sont identifiées à tort comme étant des comportements de non-coopération.

1.4.2.2 Acquittement explicite des messages

D'autres techniques de détection des nœuds non coopératifs, en remplacement du *Watchdog*, sont fondées sur des acquittements. L'idée n'est pas récente et des techniques d'acquittement des couches hautes du modèle OSI, telles que le TCP-ACK, permettent actuel-

lement de détecter des fautes [11] dans des communications de bout en bout. Néanmoins, la technique du TCP-ACK n'est pas utilisable telle quelle pour résoudre les problèmes de sécurité du routage dans les réseaux ad hoc. Ceci vient essentiellement du fait qu'elle ne donne aucune information précise de l'endroit où la faute s'est produite. Pour pallier ce problème, de nouvelles techniques ont été proposées.

Traceroute est un protocole utilisé pour déterminer le chemin emprunté par un paquet entre une source et une destination à travers un réseau. Un nœud source émet plusieurs paquets de contrôle vers la destination, avec une valeur de TTL de plus en plus grande (en commençant à 1) puis attend en retour les messages d'avertissement en provenance des nœuds intermédiaires qui ont reçu un paquet avec une valeur de TTL expirée. Padmanabhan et Simon ont proposé Secure Traceroute [PS03], une version sécurisée de Traceroute, pour localiser l'origine d'une faute sur un chemin de communication. Dans cette version, l'idée de base consiste à empêcher un attaquant de traiter de manière différente les paquets de contrôle des paquets de données. Pour ce faire, tous les paquets de contrôle sont authentifiés et masqués afin d'être confondus avec le trafic ordinaire. L'inconvénient ici est que la recherche d'un nœud fautif s'accompagne d'un coût en temps qui est linéaire avec la longueur du chemin.

Awerbuch *et al.* ont proposé [AHNRR02, ACH+08] un protocole de routage réactif résistant aux comportements byzantins, c'est-à-dire aux actions complètement arbitraires et imprévisibles d'un nœud interne (seul ou en collusion) qui résultent en une dégradation des services de routage. Les auteurs supposent l'existence d'un dispositif cryptographique pour assurer l'authenticité et l'intégrité des paquets en transit. Pour détecter les fautes byzantines, et plus particulièrement la suppression arbitraire de paquets de données sur des liens dans la phase d'acheminement, les auteurs proposent, un peu à la manière du TCP-ACK, une technique d'acquittement de bout en bout. Pour ce faire, les destinations doivent envoyer des acquittements explicites pour chaque paquet de données valide reçu. Les pertes, caractérisées par la non-réception d'un accusé de réception, sont détectées par le nœud source. Une faute byzantine est décelée à partir du moment où le taux de pertes dépasse un seuil prédéfini, ce qui conduit le nœud source à déclencher une procédure de recherche du nœud défaillant dans le chemin. Selon cette procédure, le nœud source inclut dans l'en-tête des paquets de données suivants, en plus de la destination, une liste de nœuds intermédiaires devant envoyer des acquittements. La source procède par dichotomie pour identifier le nœud défaillant, et pour un chemin de longueur n, $\lceil log(n) \rceil$ fautes successives de suppression sont nécessaires pour trouver sa position. Comme dans l'approche Secure Traceroute, les paquets de contrôle sont masqués. Cependant, la procédure de cherche peut être mise en défaut et ne pas identifier immédiatement un attaquant lorsque ce dernier supprime aléatoirement les paquets qu'il reçoit. Une fois le nœud défaillant trouvé, son poids associé est augmenté et un autre chemin vers la destination est calculé.

Afin d'accélérer la détection et l'identification des nœuds non coopératifs, Balakrishnan *et al.* ont proposé Two-Ack [BDV05a], une technique de détection qui repose sur des émissions d'acquittements à deux sauts dans le sens inverse au chemin de retransmission. Two-Ack est conçue comme étant une technique additionnelle pour n'importe quel proto-

[11] Ici, une faute désigne toute perturbation qui provoque des pertes ou des délais significatifs dans le réseau.

cole de routage à la source tel que DSR. Le principe de fonctionnement est le suivant. Un nœud qui émet un paquet de données attend la réception d'un acquittement en provenance d'un autre nœud à deux sauts suivant le chemin de routage vers la destination. S'il ne reçoit aucun acquittement pendant le temps imparti, alors un compteur local comptabilisant le nombre de paquets non retransmis est décrémenté. Lorsqu'un nœud non coopératif est détecté, un message d'erreur de route est envoyé en direction de la source. Tous les nœuds intermédiaires sur le chemin peuvent entendre ce message et enregistrer le nœud incriminé comme étant défaillant. Tout chemin impliquant ce nœud sera évité.

Une limitation à cette technique est qu'elle ne permet pas de détecter des comportements de non-coopération lorsque deux nœuds attaquants forment une collusion. Dès lors que deux nœuds attaquants se succèdent sur un chemin de retransmission, l'un peut couvrir les actions de suppression de l'autre soit en les ignorant soit en générant de faux acquittements. De plus, le système d'annonce de nœuds défaillants est lui-même source d'attaques, et ce pour deux raisons. La première vient du fait que les messages d'erreur qui incriminent des nœuds peuvent être générés par tout nœud inclus dans le chemin de retransmission. Un attaquant est alors en mesure d'injecter de fausses informations, conduisant éventuellement à des déclenchements intempestifs de reconstructions de chemins ou un partitionnement du réseau. Ensuite, des délais supplémentaires sont introduits dans la phase de réaction lorsqu'un attaquant supprime en chemin les messages d'annonce de nœuds défaillants qu'il reçoit au lieu de les retransmettre.

Discussion Le niveau de détection élevé affiché par les techniques de détection basées sur des acquittements explicites des paquets vient au coût d'une augmentation importante du nombre de paquets. Ce coût est d'autant plus important qu'il est indépendant de la présence de comportements de suppression. Plus particulièrement, ces techniques imposent un surcoût d'au moins un paquet de taille $O(1)$ par paquet de données émis par un nœud source, et de $O(n-1)$ communications pour un chemin de taille n ($O(2n-4)$ pour Two-Ack). Pour réduire ces coûts, une amélioration du schéma originel de Two-Ack a été proposée par Liu *et al.* [LDVB07]. Ils présentent une méthode où seulement une fraction des paquets de données sont acquittés, avec comme compromis une augmentation des délais pour la détection des comportements de suppression.

Par ailleurs, comme cela est le cas pour les techniques basées sur le Watchdog, elles répondent au problème de la non-coopération des nœuds dans la phase d'acheminement des paquets de données. En revanche, elles ne sont pas directement applicables à la détection des suppressions des messages de contrôle pour lesquels aucun récepteur intermédiaire spécifique n'est connu.

1.4.2.3 Principe de conservation de flot

Le principe de conservation de flot dans un réseau stipule que dans un intervalle de temps, tous les octets de données envoyés à un nœud et qui ne lui sont pas destinés doivent quitter ce nœud. Une mesure additionnelle pour identifier des anomalies dans la phase de retransmission des paquets de données consiste alors à traiter les paquets en transit comme des flots (qui sont comptabilisés) puis à vérifier ce principe dans le graphe topologique.

De façon évidente, il n'est pas envisageable de laisser les nœuds vérifier par eux-mêmes le respect du principe de conservation de flot exclusivement à partir de compteurs locaux des paquets qu'ils ont reçus et émis. De même, ces vérifications ne peuvent être réalisées uniquement sur la base d'une comparaison d'égalité entre les valeurs de leurs propres compteurs et celles de leurs voisins directs (*i.e.* un test d'égalité des compteurs des deux extrémités d'un lien). En effet, étant donné que ces compteurs sont manipulés par les nœuds eux-mêmes, un attaquant est en mesure de modifier les valeurs de ses compteurs afin de masquer ses actions de suppression.

Basée sur le principe de conservation de flot, Bradley *et al.* ont proposé une solution distribuée de détection nommée WATCHERS (*Watching for Anomalies in Transit Conservation : a Heuristic for Ensuring Router Security*) [BCP+98] dans laquelle les nœuds contrôlent les flots de leurs voisins directs grâce aux compteurs des voisins de leurs voisins. Dans une première phase, il s'agit pour chaque nœud de maintenir des compteurs distincts de paquets émis et reçus pour chacun de ses voisins. Ensuite, ces valeurs de compteur sont périodiquement échangées entre les nœuds du réseau. Ceci leur permet, par comparaison de valeurs de compteur concernant un même flot, de vérifier pour chacun de leurs voisins si le principe a bien été respecté. Dans cette approche, il est requis que chaque nœud soit en possession d'une copie de la table de routage de ses voisins. Par ailleurs, la technique WATCHERS a été proposée pour être utilisée dans les réseaux filaires. Par conséquent, certaines hypothèses qui ont été prises telles que l'immobilité des nœuds et une synchronisation de ces derniers pour la phase d'échange de compteurs font qu'elle n'est pas adaptée au réseau ad hoc. Dans le contexte des réseaux ad hoc, Raffo *et al.* ont proposé une technique similaire par-dessus un routage OLSR [ACL+05]. Cependant, les changements topologiques et la synchronisation des nœuds ne sont pas pris en considération.

Discussion La mobilité des nœuds rend difficile l'identification précise de l'ensemble des nœuds qui ont émis (reçu) des paquets vers (en provenance) d'un autre nœud du réseau. Afin de contourner ce problème, Gonzalez *et al.* ont proposé un mécanisme d'échange localisé des compteurs [GHP07]. Le but de ce mécanisme est de traquer tous les nœuds qui ont été en contact avec le nœud pour lequel un contrôle de flot est demandé, tout en réduisant le nombre de retransmissions de paquets de contrôle. Les principaux inconvénients de cette approche sont qu'elle ne prend pas en considération les attaques de nœuds en collusion ou les possibles suppressions de paquets contenant les compteurs des nœuds. Il en résulte des erreurs lors de la vérification du principe de conservation de flot. A l'instar de WATCHERS, elle requiert une synchronisation d'horloge.

Cette technique affiche également des lacunes pour apprécier avec justesse du comportement d'un nœud. Elle permet seulement de prouver que le bon nombre de paquets a été retransmis par un nœud, sans pour autant garantir que ce soit les bonnes données qui ont été retransmises. Ainsi, dans une étude menée par Hughes *et al.* [HAB00], quelques attaques qui exploitent cette faiblesse sont mises en exergue. Parmi celles-ci, nous pouvons citer la modification en transit de l'adresse de destination d'un paquet, l'injection de paquets avec usurpation de l'identité du nœud source. Les auteurs présentent également une attaque par laquelle deux attaquants consécutifs sur un chemin de transmission sont en collusion et modifient l'adresse source d'un paquet en transit ainsi que les valeurs de

leurs compteurs de paquets locaux. Ces attaques permettent à un attaquant de mener des actions conduisant soit à la perte de paquets sans pour autant être détecté, soit à l'accusation d'autres nœuds pourtant honnêtes dans le réseau.

1.4.2.4 Schémas de micropaiements

Dans d'autres travaux [HGLBV01, JHB03, BH03], le problème de la retransmission des paquets de données est modélisé comme un marché économique dans lequel les opérations d'émission et de retransmission ont un coût. Le principe de fonctionnement est similaire à la stratégie d'un prêté pour un rendu : un nœud perd des crédits lorsqu'il utilise les services offerts par le réseau, et il en reçoit lorsqu'il produit des services. Ainsi, au lieu de punir les nœuds non coopératifs, ces approches visent à stimuler la coopération en apportant des récompenses aux nœuds qui participent à ces opérations. Pour réguler les échanges entre les nœuds, la notion d'argent virtuel et de micropaiements a été introduite.

Buttyán et Hubaux ont proposé un mécanisme distribué de rétribution basé sur l'échange d'une monnaie virtuelle appelée *Nuglets* [BH03]. Le principe de fonctionnement est le suivant : chaque nœud maintient un compteur de crédit ; lorsqu'un nœud émet un paquet de données, il perd des crédits correspondant au coût de retransmission produit par les nœuds intermédiaires impliqués dans la transaction ; et inversement, lorsqu'un nœud retransmet un paquet, il accumule des crédits. Pour qu'un nœud puisse émettre ses propres paquets sur le réseau, son compteur de crédit doit être positif. Ainsi cette approche vise, au niveau applicatif, à encourager les nœuds à participer aux opérations d'acheminement des paquets de données. Pour assurer la sécurité du mécanisme de rétribution ou éviter toute manipulation frauduleuse du compteur de crédits, chaque nœud est muni d'un équipement matériel résistant à la manipulation. C'est à l'intérieur cette enceinte de sécurité que sont stockés les outils cryptographiques (paire de clés publique et privée du module de sécurité, clés de session) utilisés pour authentifier les paquets et le compteur local de crédits, et où sont exécutées les opérations de paiement.

Un autre système économique, nommé *Sprite*, a été proposé par Zhong *et al.* [ZCY03]. Contrairement à l'approche *Nuglets*, *Sprite* ne requiert pas d'équipement matériel résistant à la manipulation. Ici, la gestion des opérations de crédits et débits sur les comptes de chaque nœud participant au routage est accomplie par une autorité centrale de financement. Dans ce modèle, un nœud perçoit un reçu authentifié à chaque fois qu'il participe à une opération de retransmission d'un paquet de données. Les reçus accumulés par un nœud sont alors transmis à l'autorité de financement qui se charge de contrôler leur validité, puis de réaliser l'opération de débit et de crédit sur les comptes des nœuds impliqués. Cette autorité centrale est censée être accessible à tout instant par le biais d'une connexion à l'Internet, ce qui limite fortement l'utilisation d'une telle approche dans le contexte des réseaux ad hoc.

Dans les modèles économiques, une difficulté consiste à déterminer le prix exact qu'un nœud source doit payer pour émettre un paquet de données. Une solution au problème de la tarification de la source est proposée par Hauspie et Simplot-Ryl [HSR06]. Le principe consiste à utiliser un protocole de découverte de route pour calculer le prix associé à l'émission d'un paquet à travers un chemin. Ainsi, chaque nœud appartenant à un chemin

entre une source et une destination annonce le prix désiré pour la réalisation de la retransmission, ce prix étant fixé en fonction de plusieurs paramètres tels que le niveau d'énergie restant d'un nœud, la charge du réseau.

La qualité de service peut être définie comme étant un ensemble de besoins à garantir par le réseau pour le transport d'un flot entre une source et une destination. Ces besoins peuvent être mesurés en termes de délai de bout en bout, de variance de délai (ou gigue), de bande passante, de pertes de paquets, etc.... Contrairement aux modèles précédents où le prix par paquet est constant, un modèle économique dans lequel le prix par paquet est fixé en fonction des attributs de qualité de service présente l'avantage de favoriser une répartition du trafic en différents points du réseau. Dans cette optique, Crowford *et al.* ont proposé un modèle basé sur la tarification du trafic [CGKO04]. Ici, le coût d'acheminement des paquets est dynamiquement adapté en fonction de la consommation en énergie et de la charge du trafic infligée au niveau du nœud.

Discussion Le fait d'intégrer un équipement matériel résistant à la manipulation permet d'offrir une solution complètement distribuée et par conséquent, plutôt bien adaptée au contexte des réseaux ad hoc. Néanmoins, ces systèmes fournissent seulement des incitations pour les nœuds rationnels, c'est-à-dire ceux dont les objectifs sont d'utiliser les services réseau offerts par les autres nœuds tout en y contribuant le moins possible. En d'autres termes, la principale préoccupation de ce type de nœuds est l'économie de leurs ressources énergétiques. En revanche, ils demeurent inefficaces contre les nœuds dont les intentions sont purement et simplement de nuire au bon fonctionnement du réseau. La raison est que ces nœuds ne sont ni identifiés ni exclus des opérations de routage pour le compte d'autres nœuds, malgré leurs actions de malveillance.

Un autre problème associé au modèle économique est le manque d'équité entre les nœuds [CGKO04]. Il vient du au fait que les opportunités des nœuds pour cumuler des crédits dépendent de leur position géographique dans le réseau. Comme les nœuds périphériques sont rarement impliqués dans les opérations de retransmission, ils se retrouveront éventuellement dans l'incapacité de gagner suffisamment de crédits pour leurs propres émissions. Ce phénomène est d'autant plus pénalisant pour les réseaux ad hoc à topologie relativement statique.

1.4.2.5 Tolérance aux fautes

Pour atténuer les effets néfastes des suppressions dans la phase d'acheminement de paquets de données, certains travaux exploitent l'hypothèse selon laquelle dans un réseau ad hoc, de multiples chemins existent entre une paire de nœuds distants. Ici, la redondance des chemins sert d'une part à prévenir les attaques passives, et d'autre part à augmenter la fiabilité d'acheminement des paquets de données en présence de fautes (*i.e.* des pertes de paquets). La prévention des attaques passives, à savoir l'écoute des transmissions, peut être obtenue grâce à une dissémination du trafic à travers les multiples chemins reliant une source et une destination. De cette manière, les nœuds intermédiaires interceptent seulement une partie du message transmis, rendant l'interprétation d'une communication entre deux nœuds distants plus délicate. La seconde idée est que lorsque k chemins à nœuds

disjoints peuvent être établis entre une paire de nœuds source et destination en communication, alors une résistance aux attaques par suppression en présence d'au maximum $k-1$ nœuds en collusion est fournie puisqu'au moins un chemin sain pourra être utilisé.

Selon ce principe, Kotzanikolaou *et al.* ont proposé SecMR (*Secure Multipath Routing*) [KMD05, MKD07], un protocole réactif sécurisé contre un nombre fini d'attaquants en collusion. Ils mettent l'accent sur la définition d'un mécanisme sécurisé de découverte de chemins non cycliques et à nœuds disjoints entre une source et une destination. Une fois les multiples chemins établis, les paquets de données peuvent être émis à travers un seul chemin actif, auquel cas lorsqu'une erreur de transmission est détectée, un délai additionnel nécessaire à l'activation d'un chemin alternatif est introduit. Pour éviter ces délais, la seconde approche proposée repose sur une émission simultanée des paquets à travers les multiples chemins. Ici, une limitation vient du fait les auteurs ne mentionnent pas comment détecter les erreurs de transmission et comment réagir à cette détection (par exemple par un évitement des chemins défaillants).

Pour rendre encore plus fiable la phase d'acheminement contre les pertes de paquets, Papadimitratos et Haas ont proposé SMT (*Secure Message Transmission*) [PH03], un mécanisme qui introduit de la redondance et selon lequel plusieurs chemins (de préférence composés de nœuds disjoints) sont simultanément utilisés. Le mécanisme est basé sur le protocole réactif DSR. L'idée générale de fonctionnement du protocole est la suivante : un message devant être émis par une source est décomposé en plusieurs morceaux redondants au moyen d'un schéma de codage [Rab89]. Ce schéma assure en particulier que le paquet originel puisse être reconstruit à partir d'un nombre minimal de morceaux. Les morceaux sont ensuite acheminés vers la destination par un routage à la source, et empruntent les multiples chemins précédemment identifiés. Puisque de la redondance a été introduite, même si la destination ne dispose que d'une partie des morceaux, le message originel peut être reconstruit.

De plus, la destination émet un acquittement en direction de la source, et ce, pour chaque morceau correctement reçu. Ce mécanisme d'acquittement joue un rôle important dans le protocole puisqu'il permet à la source de déterminer quels sont les chemins opérationnels et ceux qui sont potentiellement défaillants. Ces informations d'acquittement reçues par la source servent ensuite à maintenir un degré de fiabilité pour chacun des chemins qu'elle connaît. Des décisions peuvent être prises pour éviter les chemins par lesquels des pertes ont été constatées dans le passé.

Discussion Ici, la fonction critique qu'est l'acheminement des paquets est conçue de sorte que son fonctionnement soit le moins compromis possible par un éventuel attaquant. Comme les activités de suppression ne sont ni empêchées, ni détectées de manière précise par une localisation de l'attaquant, des dégradations des performances du réseau sont toujours possibles. Ces approches présentent également l'inconvénient d'infliger des coûts supplémentaires importants en terme d'utilisation de la mémoire pour le stockage des chemins, et en nombre de messages de contrôle pour le maintien des multiples chemins existants entre une source et une destination.

Par ailleurs, il a été montré dans une étude menée par Ye *et al.* [YKT03] que le nombre de chemins formés de nœuds disjoints entre une paire de nœuds arbitraires dans un réseau

moyennement dense (avec un degré moyen de 6,7 nœuds par nœud) est faible (environ 2 lorsque le plus court chemin entre une source et une destination est de 7 sauts), et que cela est accentué lorsque la distance entre les nœuds augmente. Par conséquent, le degré de résistance aux attaquants en collusion fourni par cette approche est dépendant de la configuration physique du réseau, et peut s'avérer plus faible dans les réseaux de densité moyenne.

1.5 Conclusion

Dans un réseau ad hoc, tous les nœud doivent participer aux opérations de routage. Ils gèrent entre autre l'établissement des chemins, la dissémination de notifications de ruptures de chemins et la retransmission des données. Etant donné cette caractéristique, il devient relativement facile pour un nœud malveillant de mener divers types d'attaques, rendant ainsi le réseau inopérant. Qu'il s'agisse de nœuds malveillants internes ou bien de nœuds normaux compromis par un attaquant au cours de l'exploitation du réseau, ces nœuds déviants sont particulièrement difficiles à contenir. La raison en est qu'ils ont accès à toutes les clés cryptographiques nécessaires pour y participer, ce qui leur permet de mener des attaques sur par exemple les points les plus critiques des opérations du réseau ou sur les mécanismes de sécurité mis en œuvre.

Nous pouvons constater qu'il n'existe pas de solution complète pour protéger à la fois la signalisation des protocoles de routage et l'opération de retransmission des paquets. Les solutions de sécurité existantes basées sur la cryptographie ou sur la détection d'intrusions rendent plus robuste la phase de signalisation. Ces solutions utilisent soit des mécanismes légers comme l'authentification de bout en bout ou des chaînes de hachage pour l'authentification de saut en saut, soit des mécanismes coûteux comme la vérification et l'authentification de saut en saut de signatures à clé publique. Néanmoins, les solutions qui se veulent les plus complètes infligent souvent des traitements coûteux à chaque nœud, conduisent à une augmentation significative de la taille des messages de routage, et ne traitent que partiellement les attaquants internes ou en collusion.

Pour sécuriser l'opération de retransmission des paquets, certaines solutions s'appuient sur des échanges d'argent virtuel ou sur l'exploitation de multiples chemins, mais elles ne répondent pas aux problèmes de l'identification et de l'isolement des nœuds déviants. La détection et l'identification précise des nœuds à l'origine d'une perte d'un paquet font appel à des techniques d'observation des comportements ou d'acquittement explicite des paquets. Nous avons vu que les techniques basées sur l'observation présentent l'inconvénient d'être sujettes à des erreurs dans leur diagnostic. De plus, ces deux techniques ne répondent que partiellement au problème de la non-coopération dans l'opération de retransmission puisque seuls les paquets de données sont traités. Or la perte des paquets de contrôle peut s'avérer critique pour certaines applications. Il est essentiel que tous les nœuds participent à cette opération, faute de quoi des chemins sous-optimaux, un partitionnement du réseau ou des délais supplémentaires pour la détection des ruptures de lien peuvent apparaître. Un dernier écueil est que ces systèmes sont souvent complexes dans leur mise en œuvre puisqu'ils sont eux-mêmes sujets à des attaques. Ainsi, ils requièrent

au minimum le déploiement de services d'authentification pour se prémunir contre l'attaque Sybil d'une part, et pour empêcher les altérations dans leur phase de dissémination d'informations de réputation d'autre part.

Dans le chapitre suivant, nous proposons un modèle dans lequel chaque nœud intègre un équipement résistant à la manipulation. Grâce aux propriétés de sécurité offertes par cet équipement, nous définissons puis nous étudions une couche de communication sécurisée réduisant au minimum les attaques lancées par des nœuds malveillants sur les protocoles de routage. Les services offerts par cette nouvelle couche visent à rendre particulièrement difficile la compromission des fonctions essentielles de signalisation et de transmission des paquets. Par ailleurs, contrairement aux solutions existantes, l'idée est de permettre facilement l'intégration et la sécurisation de n'importe quel protocole de routage.

Chapitre 2

Sécurisation du routage à l'aide de matériels résistants à la manipulation

La couche réseau repose sur des protocoles distribués où une coopération entre les nœuds qui y participent est requise. A travers cette coopération, ils rendent possible l'obtention d'une représentation fidèle de la connectivité du réseau, puis le calcul et le maintien des chemins en conséquence. Ils servent d'autre part à assurer de manière transparente la retransmission des paquets de données de bout en bout. Dans la mesure où toutes les opérations définies par ces protocoles sont sous l'entière responsabilité des nœuds, l'apparition de comportements malveillants est très probable. Or avec de simples déviations par rapport à leurs spécifications, un attaquant peut entraîner des dégradations importantes dans le fonctionnement global du réseau.

Dans le but de maintenir un niveau de fonctionnement global acceptable, le besoin de faire confiance aux nœuds qui exécutent ces protocoles devient une priorité. Nous avons vu en section 1.4 que ce maintien de la confiance, lorsqu'il est principalement assuré par des moyens logiciels, pose de nombreux problèmes. Depuis plusieurs années, des architectures sécurisées basées sur des équipements matériels résistants à la manipulation sont utilisées. Elles le sont notamment dans de nombreuses applications pour lesquelles des garanties fortes dans l'exécution correcte d'une application et/ou la préservation d'informations confidentielles sont requises. Les propriétés de sécurité de ces équipements sont très attrayantes. Ils offrent un environnement d'exécution et de stockage des données sécurisé et autonome, ne pouvant être ni inspecté ni modifié par un attaquant. Ils permettent alors de garantir la confidentialité des applications et de leurs données, et leur intégrité, y compris en présence d'attaques logicielles et matérielles. Parmi les applications actuelles qui utilisent des équipements matériels résistants à la manipulation, nous pouvons citer le système de téléphonie mobile (GSM), le système de transactions bancaires, les grilles de calculs distribuées, les systèmes de gestion des droits numériques, etc. ... En raison de leurs propriétés d'inviolabilité, la disponibilité de ces équipements sur chacun des nœuds peut être utile pour établir et maintenir une confiance dans l'exécution correcte de protocoles de la couche réseau, et ainsi servir de fondement à chaque nœud pour leur permettre de

prendre localement des décisions adaptées à l'égard des autres nœuds.

Dans la suite de ce chapitre, nous proposons une couche de communication sécurisée qui tire avantage d'un équipement matériel résistant à la manipulation afin d'empêcher, ou dans le pire des cas de limiter, les attaques sur les opérations de routage présentées en section 1.3.1. Dans une première partie, nous présentons brièvement les équipements existants pour garantir la confidentialité et l'intégrité des applications et de leurs données, puis nous discutons leurs avantages et limitations pour une intégration dans le contexte des réseaux ad hoc. Dans une seconde partie, nous définissons les fonctionnalités gérées par cet équipement. Par une analyse des différentes interactions possibles entre les éléments intervenants dans le système, nous montrons qu'un problème restant est celui de la suppression des messages. Pour pallier ce problème, nous proposons un protocole de contrôle des échanges de messages adapté à la détection et l'identification des comportements arbitraires de suppression. Cette approche ne requiert ni de nœud central de confiance, ni de dissémination de messages de notification d'alarme à travers le réseau. Au lieu de cela, l'assurance dans le fonctionnement correct des opérations de routage et de sécurité repose sur la confiance accordée à ce matériel intégré sur chacun des nœuds.

2.1 Coprocesseur sécurisé

Pour garantir la confidentialité et/ou l'intégrité des applications et de leurs données contre les attaques matérielles et logicielles, des architectures basées sur un coprocesseur sécurisé ont été proposées. Leur principe est de rassembler tout ce qui est nécessaire à l'exécution des applications devant être protégées (à savoir un processeur, de la mémoire vive, un espace de stockage de « masse », etc...) à l'intérieur d'une enceinte matérielle résistante aux attaques. C'est notamment le cas du coprocesseur IBM 4758 [SW99, ibm06] et 4764 [ibm07], et des cartes à puce (et en particulier les cartes à microprocesseur[1]).

2.1.1 Caractéristiques

Dans la mesure où en terme de sécurité, des garanties fortes sur la confidentialité et l'intégrité des applications et de leurs données sont apportées, la solution proposée dans cette thèse peut se satisfaire de n'importe laquelle de ces architectures. Notons cependant que certains aspects tels que le coût, le format et les caractéristiques techniques influent sur le choix d'une de ces architectures.

En particulier, le coprocesseur IBM 4764 se présente sous la forme d'une carte PCI et peut par conséquent seulement être inséré dans un ordinateur de bureau classique. En raison de cette caractéristique, il se montre inapproprié pour les réseaux ad hoc composés d'équipements de taille réduite. En outre, il est actuellement affiché à un prix de plusieurs

[1] Selon leur architecture interne et leur mode de fonctionnement, on distingue deux principales catégories de cartes à puce : les cartes à mémoire et les cartes à microprocesseur intégré. Les premières cartes qui ont été conçues sont les cartes à mémoire. Ce sont des cartes relativement basiques, utilisées en particulier dans la téléphonie, où la mémoire contient un compteur qui représente les crédits disponibles et qui est décrémenté lors de leur utilisation. Suite à l'évolution de la technologie, un microprocesseur a été intégré, rendant ainsi les cartes capables d'exécuter des applications de manière autonome.

milliers de dollars, ce qui le rend difficile d'accès pour la majorité des applications grand public.

Quant à la carte à puce, elle se présente sous la forme d'une fine carte en plastique, de quelques centimètres de côté et de moins d'un millimètre d'épaisseur[2]. Au niveau matériel, toutes les ressources doivent être placées sur un micromodule monolithique afin de rendre extrêmement difficiles les attaques par espionnage du bus de données (voir figure 2.1). Il s'agit généralement d'un circuit intégré comportant au minimum : un micropro-

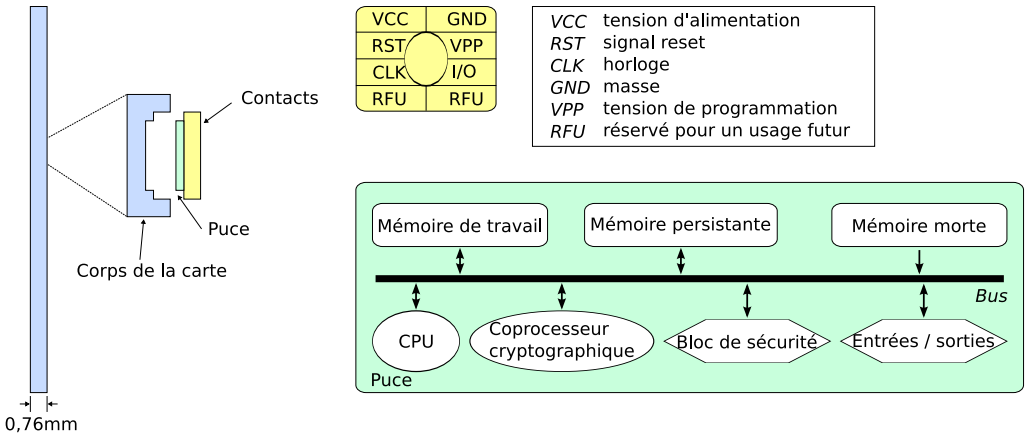


Fig. 2.1 – Architecture d'une carte à puce à microprocesseur.

cesseur, une mémoire morte, une mémoire de travail, une mémoire de stockage persistante et réinscriptible, un périphérique de communication (gestion des entrées/sorties). De plus, afin de contrôler l'environnement dans lequel la puce évolue et ainsi garantir son intégrité physique, un bloc de sécurité composé de détecteurs de conditions anormales de fonctionnement (variation de la luminosité, de la tension, de la fréquence, etc...) est également placé sur ce micromodule ; en réponse à la détection d'une anomalie, la carte arrête de fonctionner. C'est par-dessus ce micromodule que sont apposées des bornes de connexion lui permettant d'une part d'être alimentée, et d'autre part d'échanger des données avec l'extérieur. Or, afin de supporter les contraintes de flexion et de torsion que son porteur peut lui faire subir, la norme ISO[3] 7816 [ISO87] impose que la surface du circuit imprimé soit inférieure à 25mm^2, et son épaisseur de 0.3mm. La conséquence directe est que les ressources matérielles d'une carte sont très limitées (en comparaison avec les ressources actuellement disponibles sur les ordinateurs de bureau), et elles n'évoluent pas aussi rapidement qu'espéré (voir tableau 2.1) ; pour plus d'informations, le lecteur pourra se référer à [Gri00].

Actuellement, les cartes haut de gamme disponibles sur le marché sont dotées de 512 Ko

[2]Les dimensions habituelles sont 85,6 x 54mm et de 0,7 à 0,9mm d'épaisseur.
[3]ISO : *International Standardization Organisation.*

Année	Taille du bus	Fréquence	RAM	Mémoire persistante
1981	8 bits	4,77 MHz	36 octets	1 ko
1985	8 bits	4,77 MHz	128 octets	2 ko
1990	8 à 16 bits	4,77 MHz	256 octets	8 ko
1996	8 à 32 bits	4,77 à 28,16 MHz	512 octets	32 ko
2000	8 à 32 bits	4,77 à 28,16 MHz	1536 octets	64 ko
2003-2004	8 à 32 bits	4,77 à 100 MHz	16 ko	256 ko

Tab. 2.1 – Evolution des caractéristiques des cartes à microprocesseur (d'après [Gri00]).

de ROM[4], de 384 ko d'EEPROM[5], et de 24 ko de RAM[6] (pour l'Amtel AT91SC512384RCT) ou de 1280ko de FlashRAM[7] et de 30ko de RAM (pour le STMicroelectronics ST33F1M). En comparaison avec l'IBM 4764 [ibm07], les cartes à microprocesseur sont bien moins onéreuses. Même si elles affichent actuellement de fortes limitations matérielles et logicielles, elles n'en demeurent pas moins un équipement de sécurité particulièrement bien adapté pour les réseaux ad hoc. En effet, en raison de leur faible taille, elles peuvent facilement être intégrées sur chacun des nœuds du réseau et répondent par conséquent parfaitement aux besoins de décentralisation.

2.1.2 Attaques et contre-mesures

Etant donné que la sécurité globale du système repose sur la sécurité offerte par ces dispositifs, nous présentons dans la suite de cette section quelques-unes des attaques auxquelles ils sont sujets et les mécanismes de contre-mesure mis en œuvre pour y remédier. Ces attaques peuvent être classées selon deux grandes catégories [RE03] : les attaques physiques et les attaques logicielles.

Les attaques physiques reposent sur l'observation et/ou la modification de composants tels que le microprocesseur, les mémoires, et les bus. Parmi ces attaques figurent l'observation au microscope des cellules mémoires dans lesquelles est inscrite l'information recherchée, et la pose d'une sonde sur le bus reliant la mémoire au microprocesseur afin d'espionner les transferts de données. Des attaques plus subtiles sont celles par canaux cachés. Elles sont pour certaines basées sur l'observation des propriétés physiques liées au fonctionnement des composants telles que le temps d'exécution d'une opération, la consommation électrique, ou encore le rayonnement électromagnétique du microproces-

[4]ROM : *Read-Only Memory*. La ROM est une mémoire persistante mais non modifiable. Elle est utilisée pour stocker des données qui n'auront pas à être modifiées durant tout le cycle de vie de la carte. Par exemple, sont stockées en ROM les parties critiques du système d'exploitation.

[5]EEPROM : *Electronic Erasable Programmable Read-Only Memory*. L'EEPROM est un type de mémoire persistante et modifiable utilisée pour le stockage des applications.

[6]RAM : *Random Access Memory*. La RAM est une mémoire volatile et modifiable. En raison de ses temps d'accès rapides en lecture et écriture, elle est utilisée comme espace de travail par le processeur afin de stocker temporairement des données lors de l'exécution d'un processus.

[7]FlashRAM : la mémoire FlashRAM (ou Flash) est un type de mémoire EEPROM particulier qui permet une granularité d'écriture moins fine, mais réduisant les délais d'écriture.

seur. A partir de l'analyse de ces comportements, il devient possible pour un attaquant de déduire des informations sur les données manipulées par le processeur. Pour prévenir ces attaques, des mécanismes de protection sont implémentés aussi bien au niveau matériel que logiciel. Par exemple, des capteurs sont introduits de sorte à détecter toute utilisation en dehors de la norme. Dès lors qu'une telle situation survient, l'exécution des applications est définitivement interrompue et leurs données sont éventuellement détruites. La mise en œuvre d'une vérification périodique de l'intégrité des données présentes en mémoire assure qu'aucune altération par une entité extérieure n'est possible.

Outre les attaques matérielles, les logiciels maintenus à l'intérieur de l'enceinte matérielle de sécurité (*i.e.* le système d'exploitation et les applications qui y sont installés) représentent des cibles pour un attaquant. Une première étape de vérification de l'intégrité de l'application, avant son chargement, vise à éviter l'introduction d'applications malicieuses. En ce qui concerne la sécurité de l'environnement d'exécution, une séparation complète entre les applications installées est assurée grâce à des mécanismes d'isolation fournis par le système d'exploitation. Puisqu'il est prévu que chaque application s'exécute dans son propre espace mémoire, même si une application malicieuse franchit l'étape de vérification de son code, elle ne peut ni mettre en péril le système, ni affecter les données ou l'exécution des autres applications.

Les attaques pour mettre en défaut la sécurité de ces équipements demandent un niveau de connaissance et de compétence élevé, et parfois des équipements spécifiques très coûteux. Dans la suite de nos travaux, nous partons de l'hypothèse (que nous estimons à notre avis raisonnable) qu'un attaquant ne peut pas compromettre les applications et les données stockées à l'intérieur de ces équipements.

2.2 Architecture de sécurité pour les réseaux ad hoc

Notre objectif est réduire au minimum les attaques de nœuds aussi bien internes qu'externes, et à partir desquelles le fonctionnement du routage est compromis. A cet effet, nous définissons une couche de communication sécurisée qui est maintenue à l'intérieur d'une enceinte matérielle de sécurité. En particulier, cette couche comprend toutes les fonctionnalités des protocoles de routage. Parmi ces fonctionnalités figurent la gestion des messages de signalisation, le maintien des structures de données (*i.e.* la table de routage, la table de messages dupliqués, la table de voisinage, etc...), et le mécanisme de retransmission des paquets de données de la couche réseau. Notons que dans des travaux menés de manière indépendante [CLM08], Chaouchi et Laurent-Maknavicius ont proposé une architecture similaire à la notre, à savoir basée sur l'intégration d'une enceinte matérielle de sécurité sur chaque nœud sous le contrôle d'un opérateur réseau.

L'enceinte de sécurité fournit l'assurance d'une exécution correcte des opérations des protocoles de la couche de communication. Or dans la pratique, ces protocoles font intervenir de multiples nœuds à travers des échanges de messages. Une première étape de notre travail est alors d'identifier et de modéliser les différents éléments qui composent cette architecture de sécurité. Le but de cette étude est de mettre en exergue les rôles, les responsabilités et les interactions entre ces différents éléments, afin de mieux cerner les limites de la sécurité offerte par une telle architecture.

2.2.1 Structure d'un nœud

Dans notre modèle, chaque nœud du réseau est équipé d'un équipement matériel résistant à la manipulation à l'intérieur duquel sont exécutées les fonctions de routage. Plus précisément, nous représentons un nœud comme étant composé de deux parties interdépendantes (voir figure 2.2) : l'hôte (dénoté H) et l'enceinte de sécurité (dénotée TRD pour *Tamper Resistant Device*). L'hôte est la partie qui sert de support au TRD, pour par exemple son alimentation. Il doit être vu comme un ordinateur classique, composé de matériels (*i.e.* un microprocesseur, des mémoires et des périphériques d'entrée/sortie) et de logiciels dont le système d'exploitation. Etant donné que tous ces éléments sont entièrement sous le contrôle de l'hôte, aucune garantie n'est offerte à une application concernant la bonne réalisation d'une opération. Il peut ainsi, par exemple, modifier le comportement de son système d'exploitation, modifier le comportement de la pile réseau, etc.... Ces caractéristiques nous conduisent à considérer la partie hôte comme étant hostile, et potentiellement source d'attaques.

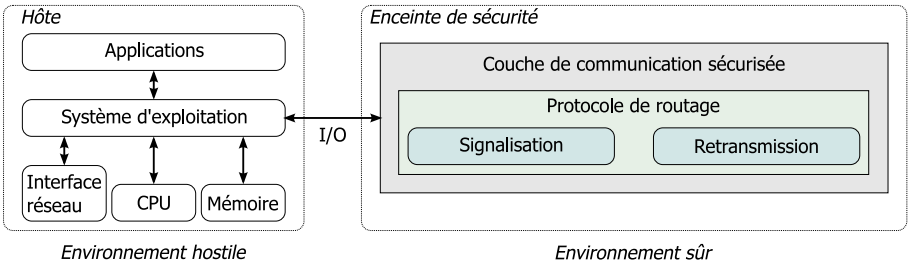


Fig. 2.2 – Structure d'un nœud.

A contrario, étant donné que l'hôte n'a ni directement ni indirectement accès à ce qui est situé à l'intérieur de l'enceinte de sécurité, cette dernière représente la partie sûre d'un nœud. Elle fournit l'assurance d'une exécution correcte des protocoles de la couche de communication qu'elle embarque au regard de leurs spécifications (telles que décrites dans les RFC).

2.2.2 Interactions entre l'hôte et l'enceinte de sécurité

En général, les protocoles de la couche de réseau sont exécutés par chaque nœud du réseau. Ils fonctionnent sur la base d'un échange de messages, suivant des règles précises décrites dans leurs spécifications. Lorsqu'un nœud du réseau participe à un de ces protocoles, plusieurs responsabilités lui sont confiées. Ces responsabilités se résument comme suit :

- Il doit générer et émettre des messages de signalisation soit de manière périodique, soit à la suite d'un événement particulier survenu dans le réseau.
- Il doit recevoir et traiter les messages de signalisation en provenance des autres nœuds présents dans son champ direct de transmission (ses voisins), afin de mettre à jour sa base de connaissances locale.
- Et certains des messages de signalisation ou de données reçus doivent être réémis.

Une solution idéale aurait été que l'enceinte de sécurité soit d'elle-même capable de réaliser l'ensemble de ces opérations. Or en pratique, une carte à puce ou un coprocesseur sécurisé ne sont pas considérés comme des points de communication autonomes[8]. Les périphériques d'entrée/sortie dont ils bénéficient servent seulement de support pour l'établissement de communications directes avec son hôte.

Etant donnée cette caractéristique, un TRD requiert explicitement les ressources matérielles de communication de son hôte pour interagir avec des TRD distants. Ceci signifie en particulier que dans notre modèle, un hôte joue un rôle d'intermédiaire incontournable dans le flux de messages entre les TRD. Il a pour responsabilité de relayer les messages, c'est-à-dire, d'émettre tous les messages générés par son TRD sur le réseau, puis de transmettre tous les messages reçus sur son interface de communication à son TRD.

Bien sûr, puisque les canaux physiques de communication vers le monde extérieur sont fournis par l'hôte, ils sont sujets à tous les problèmes traditionnels inhérents aux communications à travers un canal non sécurisé. La raison est qu'un hôte agit suivant son libre arbitre et peut, par exemple, espionner, modifier, injecter, retarder, réorganiser, usurper ou tout simplement refuser de délivrer des messages à son propre TRD ou à des TRD distants.

2.2.3 Sécurité des communications

2.2.3.1 Besoins

Pour limiter les actions malveillantes d'un hôte sur les messages lors de leur transfert, l'architecture de sécurité doit satisfaire les besoins élémentaires suivants :

Vérification de l'exactitude du contenu des messages. Le récepteur d'un message doit avoir la garantie que les informations qu'il contient ne sont pas falsifiées. A partir du moment où il est possible de prouver qu'un message a bien été envoyé par un protocole authentique, c'est-à-dire exécuté à l'intérieur de l'enceinte de sécurité, alors nous pouvons être sûrs que les données qu'il contient n'ont pas été volontairement modifiées. Ainsi, l'authentification de la couche de communication plutôt que du nœud à l'origine d'un message permet d'éviter l'inoculation de fausses informations.

Intégrité des messages. Le récepteur d'un message doit avoir la garantie qu'aucune modification non autorisée, à savoir l'insertion, la suppression ou la substitution de données, n'a été apportée sur un message en transit.

Identification des responsabilités. Les nœuds participants aux opérations du réseau sont responsables de leurs actions, délibérées ou accidentelles, qui perturbent le fonctionnement des autres nœuds ou les services de transport. Le système devrait fournir des informations permettant d'identifier ou d'aider à l'attribution d'une responsabilité.

[8] A l'exception des cartes à puce sans contact. Elles communiquent par radio, au moyen d'une antenne interne. Cependant, cette technologie impose que les cartes soient placées à proximité d'un lecteur pour fonctionner, à savoir entre 3 à 10 centimètres.

Comme nous venons de le souligner, s'assurer de la légitimité des messages émanant de la couche de communication est nécessaire pour empêcher l'inoculation de fausses informations de routage par des attaquants externes, mais aussi par des hôtes en possession d'un TRD valide. En revanche, la confidentialité des messages vise à garantir que des données puissent être échangées entre des protagonistes légitimes sans crainte qu'elles soient comprises, même dans le cas où elles seraient interceptées par un attaquant. Cet aspect de la sécurité peut facilement être obtenu grâce à l'utilisation de mécanismes additionnels de chiffrement des messages en transit sur le canal de communication. Néanmoins, nous l'avons exclu de nos travaux, car il n'est pas essentiel pour assurer le bon déroulement des opérations des protocoles de routage.

2.2.3.2 Configuration initiale

Les propriétés d'un canal ne sont généralement pas intrinsèques au canal physique, mais plutôt liées à des mécanismes supplémentaires basés par exemple sur des primitives cryptographiques. Dans la mesure où les TRD ont la capacité de garder secrètes des données (telles que des clés, des certificats) et d'exécuter des primitives cryptographiques (de manière efficace grâce à ses accélérateurs matériels), des canaux sécurisés entre les TRD peuvent être établis par-dessus les canaux physiques fournis par les hôtes, et ce sans recourir à l'intervention de la partie hôte. Des garanties en termes d'authentification et d'intégrité sur les données véhiculées peuvent ainsi être obtenues.

Dans les architectures de sécurité traditionnelles, l'établissement d'un canal sécurisé s'appuie soit sur des procédés cryptographiques basés sur des algorithmes à clé asymétrique (utilisation d'une paire de clés privée/publique), soit sur des algorithmes à clé symétrique. Ces derniers sont souvent considérés comme moins coûteux. En effet, ils présentent l'avantage d'être « simples » dans le sens où ils ne nécessitent pas une capacité de traitement importante, et offrent par conséquent de meilleures performances lors de leur exécution. Ce point représente un critère important pour les systèmes à ressources limitées, et c'est la raison pour laquelle nous avons privilégié leur utilisation.

Les algorithmes à clé symétrique requièrent que les différents protagonistes se mettent au préalable d'accord sur la clé secrète à utiliser dans les phases d'échange de messages. Pour ce faire, en plus de gérer les protocoles de la couche réseau, la couche de communication sécurisée installée à l'intérieur de l'enceinte de sécurité stocke et manipule une clé secrète commune dénotée k_{TRD}. Concernant la configuration de cette clé, deux approches peuvent être envisagées :

- La première repose sur une procédure d'initialisation de la couche de communication sécurisée avant même la distribution des TRD aux utilisateurs finaux. C'est durant cette phase d'initialisation, par exemple assurée par l'administrateur du réseau, que la clé secrète est chargée.
- La seconde, plus flexible, consiste à mettre en place une procédure de façon à ce que cette clé puisse être obtenue dynamiquement, après distribution des TRD. En initiant une requête auprès du TRD d'un nœud voisin, tout TRD d'un nouveau nœud intégrant le réseau doit pouvoir obtenir cette clé. L'obtention de cette clé est bien évidemment soumise à une condition : le nœud doit au préalable prouver qu'il

possède un TRD valide, par exemple grâce à un certificat à clé publique (racine de la confiance) chargée dans l'enceinte de sécurité au cours de la procédure d'initialisation de la couche de communication sécurisée.

Par ailleurs, nous considérons que tous les nœuds sont identifiés de manière unique dans le réseau. Une identité associée à un nœud est dénotée id_x, et est stockée à l'intérieur du TRD. Une fois initialisée, elle ne peut donc pas être modifiée de l'extérieur, que ce soit par des moyens matériels ou des logiciels.

2.2.3.3 En-tête de sécurité

Pour satisfaire les besoins de sécurité précédemment cités, tous les messages sont authentifiés par le TRD d'un nœud émetteur et sont vérifiées par tout TRD d'un nœud récepteur à l'aide de la clé cryptographique k_{TRD}. En particulier, un en-tête de sécurité est ajouté à tout message produit ou traité par un protocole de la couche de communication avant de quitter le TRD. Cet en-tête, illustré figure 2.3, comprend : l'identité réseau du nœud émetteur (dénoté id_s), un numéro de séquence identifiant le message (dénoté sn_m), et des données d'authentification basées sur l'utilisation d'un code d'authentification de message (MAC). L'identité contribue à imputer de manière sûre une action d'émission d'un message à un nœud. Le couple (id_s, sn_m) de l'en-tête de sécurité est considéré comme unique et sert à rendre possible la détection des attaques par rejeu de messages obsolètes[9]. Le MAC est calculé par application d'une fonction de hachage cryptographique (telle que SHA-1 ou DES) en combinaison avec la clé secrète commune, sur les champs id_s et sn_m de l'en-tête de sécurité et les données du message à protéger.

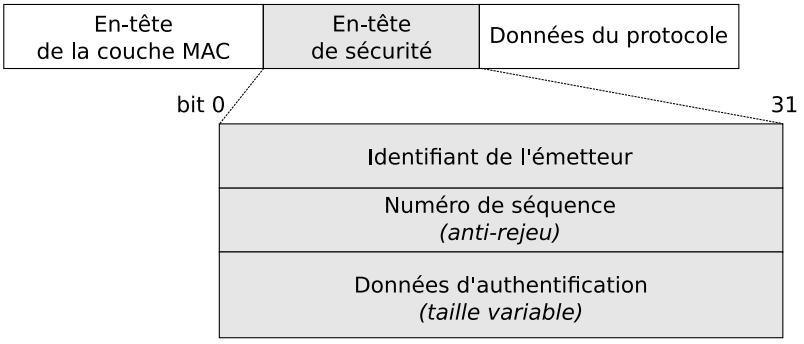


Fig. 2.3 – Format de l'en-tête de sécurité.

Ainsi, à la réception d'un message, un TRD vérifie le MAC associé afin d'en authentifier l'origine. Pour ce faire, il calcule le MAC à partir des données du message et le compare avec celui reçu. Tout message entrant présentant une signature valide (c'est-à-dire avec des valeurs de MAC qui coïncident) est accepté, sinon il est supprimé. Nous considérons que

[9]Un numéro de séquence représenté sur 16 bits laisse la possibilité de générer de 0 à 65535 messages sur un tour, avant la remise à zéro du compteur. Ainsi, un message est unique seulement sur un tour, et rend alors possible le rejeu de messages déjà traités au tour suivant. Pour traiter ce problème de remise à zéro du numéro de séquence, la solution présentée dans la section 19 de la RFC d'OLSR [CJ03] peut être envisagée.

la signature générée présente la caractéristique d'être unique et généralement impossible à forger, à moins que la clé secrète ne soit découverte.

2.2.4 Définition du problème

L'utilisation d'un équipement matériel résistant à la manipulation est loin d'offrir une solution de sécurité complète pour assurer le fonctionnement correct du routage. Une architecture de sécurité basée sur l'intégration d'un tel dispositif dans un système complexe, qui implique en particulier la participation de plusieurs entités autonomes, doit être conçue avec précaution. Dans notre modèle, les TRD peuvent seulement communiquer entre eux en échangeant des messages à travers les canaux non sécurisés offerts par leur partie hôte (éventuellement hostile). L'utilisation de primitives cryptographiques, couplée avec les informations de l'en-tête de sécurité ajouté à tous les messages, permet aux TRD d'établir des canaux authentifiés entre eux. Ces procédés limitent les attaques possibles de nœuds externes et internes. Premièrement, un attaquant sans TRD valide ne peut ni participer aux opérations de routage, ni inciter des nœuds légitimes à transmettre ses propres messages. Ensuite, les actions exhibées par un hôte hostile telles que la modification, la création ou la rediffusion de messages anciens sont évitées. Toutefois, ces mesures demeurent inefficaces contre le rejeu de messages qui ne sont pas supposés être retransmis, et le refus ou l'introduction de retards dans les transmissions de messages.

En particulier, selon la façon dont un hôte joue son rôle d'intermédiaire, nous identifions deux types d'attaques. Le premier est l'omission en réception. Il se caractérise par le fait qu'un hôte omet de retransmettre à son TRD un message reçu sur son interface de communication. Le second est l'omission en émission. Il se traduit par le fait qu'un hôte refuse d'émettre sur le réseau un message généré par son TRD. La figure 2.4 illustre les points de vulnérabilité dans le flux des messages entre les TRD et les hôtes sur le réseau. Lorsque l'on considère le cas simple d'une communication entre deux nœuds, les attaques surviennent donc soit au niveau des interactions entre l'enceinte de sécurité et son hôte, soit au niveau des interactions entre l'hôte distant et son enceinte de sécurité.

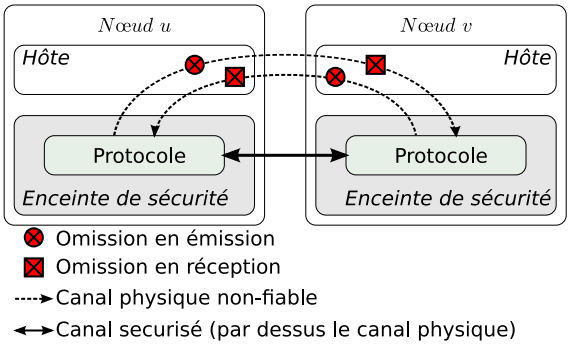


Fig. 2.4 – Vulnérabilités des canaux de communication.

Le résultat est que, même si tous les TRD exécutent correctement leur part des opérations, le comportement global du protocole est contrarié par ces suppressions. Notons que ces suppressions peuvent être lancées aléatoirement par un hôte ou à des points spé-

cifiques des échanges, ce qui les rend d'autant plus pernicieuses. A partir du moment où les calculs effectués par les protocoles de routage reposent sur un ensemble d'informations incomplètes, alors des erreurs dans les résultats produits sont possibles.

Nous ne pouvons pas garantir un environnement fiable. Notre objectif est donc de définir une couche de contrôle supplémentaire pour faire face à ces suppressions. En partant du principe qu'il est impossible d'empêcher les attaques par suppression menées par un hôte, nous proposons de comptabiliser les pertes inopinées de messages comme étant des fautes de fonctionnement. Nous visons à fournir aux nœuds les moyens de savoir à qui ils ont affaire. On peut s'attendre à ce que les nœuds exploitent les informations collectées localement pour tenter de parvenir à leurs objectifs. Ces objectifs varient en fonction du contexte d'application du réseau. Il peut s'agir de simplement éviter ces nœuds fautifs lors de la formation des chemins ou de les isoler de toute participation aux opérations du réseau. Notons cependant que grâce à une couche de détection, des améliorations dans la sécurité et les performances du réseau peuvent être apportées seulement pour les protocoles de routage (et à un niveau supérieur, aux applications) qui sont d'une part susceptibles d'admettre des pertes légères de messages, et qui peuvent d'autre part prendre des décisions appropriées en réaction aux informations recueillies par notre couche supplémentaire.

Dans la section suivante, nous étudions le plus simple des protocoles dans un réseau, à savoir la transmission d'un message par diffusion à partir d'un nœud vers l'ensemble de ses voisins. Ce type de fonctionnement représente la base de la plupart des protocoles de routage décrits en section 1.2. Ainsi, l'idée principale est que la détection d'un mauvais comportement sur ce protocole, afin de limiter les effets néfastes des attaques par suppression, a une incidence directe sur d'autres protocoles plus complexes.

2.3 Détection de comportement

Afin d'évaluer la capacité d'un hôte à jouer son rôle d'intermédiaire au niveau des couches basses de communication, nous proposons l'ajout d'une couche additionnelle simple de contrôle. Selon le modèle OSI, une pile de protocoles réseau se compose d'une couche application, d'une couche de transport, d'une couche réseau, d'une couche liaison de données et d'une couche physique. Au regard de cette pile réseau, notre couche de contrôle se situe entre la couche réseau et la couche liaison. Tous les autres aspects de la pile de protocoles demeurent quant à eux inchangés. Cette couche de contrôle est exécutée à l'intérieur du TRD, et est fondée sur un schéma d'acquittement explicite de tous les messages. Ces travaux sont assez proches des techniques existantes basées sur des acquittements. La principale différence avec ces techniques est que nous tirons parti de l'autonomie de traitement offerte par le TRD pour définir un schéma purement local, et indépendant du protocole. L'intérêt d'une approche locale est de permettre d'accélérer la détection et l'identification du nœud à l'origine d'une suppression. Ensuite, en raison de l'autonomie de traitement, les interactions parmi les nœuds sont limitées aux voisins directs. Les informations collectées localement par les TRD sont directement exploitables pour aider à la prise de décision telle que l'évitement ou l'isolement d'un nœud. Ainsi, contrairement aux techniques existantes, des échanges de messages de notification sur la réputation des nœuds ne sont plus requis, ce qui réduit les surcoûts en communication.

2.3.1 Modèle du réseau et notations

Nous présentons dans cette section nos hypothèses en ce qui concerne la couche physique du réseau. Nous supposons que les nœuds communiquent au moyen de liaisons sans fil. Nous considérons un modèle de communication dans lequel les liaisons présentent les propriétés suivantes :

- Symétriques : si un nœud u est capable de recevoir les messages émis par un nœud v, alors v est également capable de recevoir les messages émis par u.
- Longue durée : les nœuds sont stationnaires, et les liaisons qui les relient dans le graphe de connectivité sont par conséquent statiques.

De plus, nous introduisons les notations suivantes :

- TRD_u et H_u représentent respectivement le TRD et la partie hôte du nœud identifié u.
- $\vartheta(u)$ est l'ensemble des nœuds voisins de u, c'est-à-dire tous les nœuds dans son champ de transmission. Au regard de la propriété de symétrie des liaisons, nous avons donc la relation suivante : $\forall v \in \vartheta(u)$, $u \in \vartheta(v)$.
- $u \rightarrow v : m$ est l'émission d'un message m d'un nœud u vers un nœud v, et inversement, la réception d'un message est dénotée $u \leftarrow v : m$.

2.3.2 Idée de base

Foncièrement, nous étudions le succès ou l'échec dans la transmission d'un message par diffusion, à partir du TRD d'un nœud émetteur vers les TRD de l'ensemble des nœuds présents dans son voisinage. Pour cela, nous proposons un schéma d'acquittement local pour chaque message diffusé. Néanmoins, un tel schéma ne peut être appliqué de manière traditionnelle pour localiser de manière précise l'hôte à l'origine d'une suppression. Ceci vient du fait qu'un message d'acquittement est lui-même sujet à des attaques par suppression menées par un hôte. Il s'ensuit que la non-réception d'un acquittement par un TRD (et pour un message donné) ne constitue en rien une information suffisante pour affirmer que l'hôte distant est à l'origine de la suppression.

C'est pourquoi dans le schéma proposé, l'idée est que le TRD d'un nœud émetteur, avec le support des TRD de ses voisins, contrôle d'une part le comportement de son propre hôte au regard de l'accomplissement de l'opération d'émission pour un message donné, puis d'autre part, contrôle pour chacun de ses hôtes voisins la réalisation de l'opération de réception de ce même message.

Dans notre approche, nous tirons profit de l'omnidirectionnalité des transmissions sans fil afin de permettre cette identification précise de l'hôte à l'origine de la perte d'un message dans un échange. En particulier, une transmission est dite omnidirectionnelle si pour tout message m émis par un un nœud u sur le réseau, alors tous les nœuds dans son champ de transmission le reçoivent. Cela permettra à un TRD de recouper les informations reçues et de détecter l'hôte à l'origine de la perte.

2.3.3 Hypothèses

Dans une configuration réaliste, les transmissions entre les nœuds sont sujettes à des pertes bénignes de messages. Par exemple, la perte d'un message peut survenir à la suite d'une suppression involontaire en raison d'un dépassement de tampon, d'une corruption d'un message en raison d'interférences, mais également d'une suppression intentionnelle par un hôte. Dans notre définition de pertes de messages, nous ne faisons pas de distinction entre celles causées par les hôtes paresseux, égoïstes ou malveillants, et celles causées par une défaillance temporaire de l'hôte. La raison en est qu'un hôte hostile pourra toujours faire en sorte qu'une suppression intentionnelle est l'air d'une défaillance.

En outre, nous estimons qu'au moins un nœud dans le voisinage de l'émetteur agit en conformité avec les règles définies par le protocole. Ce nœud réalise sans distinction toutes les opérations d'émission (*i.e.* de transmettre les messages générés par son TRD sur le réseau), et toutes les opérations de réception (*i.e.* de transmettre les messages en provenance du réseau à son TRD). Ce nœud servira de témoin en cas d'ambiguïté dans l'identification du comportement d'un autre nœud. En effet, l'esprit de nos travaux est de proposer une méthode simple et locale au niveau des couches basses de communication, et qui offre de bonnes propriétés émergentes dans le système global. Sur la base d'une évaluation locale, si un nœud est exclusivement entouré de nœuds malveillants, il est peu probable que son TRD puisse identifier correctement un hôte émetteur ou un ou plusieurs récepteurs à l'origine d'une suppression : il n'aura aucun témoin pouvant lui donner des indices sur un éventuel comportement déviant de son hôte ou des hôtes de ses voisins.

Les attaques par déni de service sont inévitables et ne sont donc pas traitées dans nos travaux. Par exemple, il suffit pour un hôte hostile de ne pas alimenter en électricité son enceinte de sécurité. Cette situation n'induit pas de pénalité importante dans le sens où le nœud ne sera pas considéré comme étant un intervenant dans les échanges du protocole. Cependant, des dommages bien plus importants surviennent si un hôte rejoue volontairement de nombreux messages valides de son TRD à son voisinage. Ce type d'attaque conduit d'une part à une surcharge en traitement des TRD voisins, et d'autre part un épuisement prématuré des ressources énergétiques des nœuds.

2.3.4 Description du protocole

Dans cette section, nous décrivons plus en détails le fonctionnement de notre couche de contrôle. La figure 2.5 illustre les échanges de messages. Le contrôle commence dès lors qu'un TRD est sollicité par un protocole de la couche de communication pour la diffusion d'un message et se termine une fois le délai de transmission d'un acquittement écoulé. Le nœud émetteur u et tout nœud récepteur v (tel que $v \in \vartheta(u)$) d'un message diffusé localement agissent de la façon suivante :

1. Pour un message m devant être diffusé sur le réseau, le TRD_u ajoute un en-tête de sécurité de la forme $< id_u, sn_m, hmac >$ avec $hmac = H(m|id_u|sn_m|k_{TRD})$. H est une fonction de hachage, $|$ l'opérateur de concaténation, et k_{TRD} la clé secrète connue uniquement des TRD. Avant de transmettre ce message à son hôte H_u, le TRD_u détermine quel est l'ensemble de nœuds voisins censé y répondre par un acquittement,

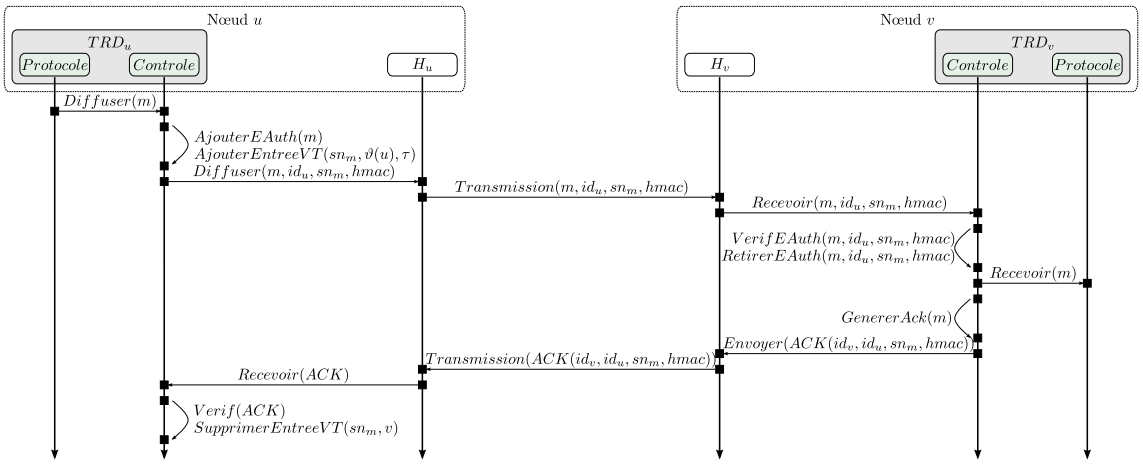


Fig. 2.5 – Echanges pour la diffusion d'un message.

et ajoute une nouvelle entrée dans sa table de vérification. Cette table sert à garder une trace des réponses escomptées des nœuds voisins. Une entrée de cette table représente un message pour lequel le TRD est en attente d'un acquittement. Elle est formée du numéro d'identifiant du message, de $\vartheta(u)$, et d'un *timer*. Ce *timer* désigne le temps d'expiration de l'entrée ; à chaque fois qu'une entrée est ajoutée, le *timer* est initialisé pour une durée de τ secondes, ce qui correspond au délai de réception d'un acquittement.

2. H_u diffuse le message m sur le réseau.
3. Chaque hôte H_v d'un nœud v présent dans le voisinage de u transmet le message m reçu à son TRD.
4. A la réception du message m, chaque TRD_v génère un acquittement de la forme $ACK :< id_v, id_u, sn_m, hmac >$ avec $hmac = H(id_v|id_u|sn_m|k_{TRD})$, puis le transmet à son hôte. Tandis que le premier champ id_v sert à identifier le nœud récepteur à l'origine de l'acquittement, le couple (id_u, sn_m) sert à déterminer à qui l'acquittement est destiné et pour quel message.
5. Chaque hôte H_v diffuse l'acquittement sur le réseau.
6. L'hôte H_u transmet les messages ACK reçus à son TRD.
7. Pour chaque ACK d'un nœud voisin v, le TRD_u supprime l'entrée qui lui est associée de sa table de vérification.

Remarquons que seul un TRD est capable de générer une signature valide pour un message, et que toute génération d'acquittement est couplée à une réception au préalable d'un message. De plus, chaque TRD maintient une identité de nœud unique et cette identité est incluse dans tout message d'acquittement qu'il génère. Etant donné ces caractéristiques, tout message d'acquittement de la forme $ACK :< id_v|id_u|sn_m|hmac >$ contient alors la preuve irréfutable que le nœud v a accompli une opération de réception du message identifié sn_m émis par u.

2.3.5 Vérification de la conformité d'un hôte

Nous exploitons l'ensemble des informations collectées durant ces échanges pour détecter les déviations sur ce protocole, et identifier avec précision les hôtes à l'origine de la perte d'un message. La procédure d'identification repose sur deux étapes successives : une première dont le rôle est de déterminer si l'hôte émetteur a transmis ou non sur le réseau le message pour lequel il a été sollicité, puis une seconde, qui selon le résultat de la première, évalue le comportement en réception des hôtes voisins.

2.3.5.1 Détection d'un hôte non conforme en émission

Dans une première étape, seul le comportement de l'hôte émetteur est vérifié. A la réception d'un message m en provenance du TRD, l'hôte H_u a deux choix possibles : soit il diffuse le message m sur le réseau, soit il le supprime. Selon l'hypothèse qu'au moins un nœud est conforme dans le voisinage $\vartheta(u)$, si aucun ACK correspondant à l'émission en attente dans la table de validation n'est reçu par le TRD_u avant l'expiration du délai τ, alors le comportement de l'hôte H_u est identifié comme étant non conforme dans l'opération d'émission. Un compteur de fautes dans l'opération d'émission (dénoté E_{miss}), maintenu localement par le TRD_u, est incrémenté.

L'opération d'émission réalisée par l'hôte H_u est validée à partir du moment où le TRD_u reçoit au moins un ACK en provenance d'un nœud quelconque dans son voisinage. La raison de ce choix est que des hôtes hostiles pourraient omettre de transmettre des ACK afin de faire faussement accuser des hôtes comme étant de mauvais émetteurs par leur TRD.

2.3.5.2 Détection d'un hôte non conforme en réception

Une fois l'opération d'émission validée, le succès ou l'échec des hôtes voisins de u dans l'opération de réception peut être évalué par le TRD_u. Ce dernier passe dans un état d'attente d'un ACK en provenance de chacun des nœuds présents dans $\vartheta(u)$. De plus, le TRD_u maintient un compteur afin de comptabiliser le nombre de fautes dans l'opération d'émission pour chacun de ses hôtes voisins. Ainsi, tout hôte pour lequel le TRD_u ne reçoit pas d'ACK dans le délai imparti est détecté comme étant non conforme dans l'opération de réception : le compteur de fautes en réception (dénoté R_{miss}) associé à l'hôte incriminé est alors augmenté.

2.3.5.3 Délai de réception d'un acquittement

Le paramètre de temps τ permet de fixer le délai de réception d'un acquittement. Si le délai pour une entrée de la table de validation expire avant la réception d'un message d'acquittement, le compteur de fautes E_{miss} ou R_{miss} associé au nœud sera incrémenté. La configuration de τ est donc importante pour assurer une identification correcte des comportements des nœuds à travers ce schéma d'acquittement.

De nombreuses fausses alarmes peuvent être déclenchées si la valeur fixée pour τ est trop basse. D'un autre côté, une valeur pour τ trop élevée a deux effets. Premièrement,

chaque TRD devra maintenir plus d'entrées dans sa table de validation, ce qui impose une plus grande capacité de stockage. Ensuite, une valeur élevée pour τ offre de plus grandes opportunités pour un hôte hostile de retarder la réception d'un message sans pour autant être détecté, ce qui vient au détriment des performances du réseau.

Par conséquent, la valeur pour τ doit être fixée de sorte à au moins permettre l'apparition d'une défaillance temporaire sur un lien entre deux nœuds (par exemple, un échec de transmission due à une congestion locale du trafic). Il est essentiel pour τ de satisfaire la relation suivante :

$$\tau > 2 * (\text{délai de transmission à un saut}),$$

où le délai de transmission à un saut inclut les délais d'accès au canal, de transmission, et de traitement d'un message.

2.3.6 Analyse de sécurité du protocole face aux attaques

Un hôte hostile peut tenter, par des actions de suppression à différents points du protocole de contrôle, de biaiser les résultats d'évaluation produits par les TRD. La question est alors de savoir si la détection des hôtes fautifs reste fiable et précise malgré ces situations adverses. Autrement dit, un hôte hostile peut-il éviter d'être qualifié de non conforme dans l'opération d'émission ou de réception alors qu'il ne suit pas les règles du protocole ? Peut-il faussement faire désigner un hôte comme étant non conforme alors qu'il ne l'est pas ? Dans le but d'analyser la fiabilité de notre schéma d'acquittement face aux attaques byzantines des hôtes, nous modélisons les interactions possibles entre le TRD émetteur d'un message et son hôte, puis entre ses hôtes voisins, récepteur de ce message, et leur TRD.

Dans un réseau à topologie statique, les voisins d'un nœud sont en nombre fini et ne changent pas au cours du temps. A partir d'un modèle dans lequel sont représentés le nœud émetteur et l'intégralité de son entourage, il est bien trop complexe d'analyser toutes les interactions possibles. En pratique, nous pouvons distinguer deux catégories de nœuds voisins selon la façon dont leur hôte agit au regard des règles définies par notre protocole. Un nœud est conforme lorsque son hôte réalise sans distinction toutes les opérations qui lui sont confiées. A l'opposé, il est non conforme lorsque son hôte agit arbitrairement. Nous proposons donc un analyse sur un modèle de communication simplifié (mais qui demeure complet) dans lequel seulement trois nœuds interviennent : un nœud émetteur d'un message (dénoté u), et son entourage qui comprend un nœud conforme (dénoté c) et un nœud non conforme (dénoté nc).

La phase de contrôle est initialisée par le TRD du nœud u, qui génère un message m devant être diffusé par son hôte. Les actions de suppression éventuellement menées par la partie hôte des nœuds impliqués dans l'échange sont les suivantes :

- Du point de vue du nœud u, l'hôte H_u peut :
  - Supprimer le message m généré par son TRD_u.
  - Transmettre le message m sur le réseau, puis supprimer complètement ou de manière sélective les ACK en provenance des hôtes H_c et H_{cn}.
- Du point de vue du nœud nc, l'hôte H_{nc} peut :

– Supprimer le message m reçu.
– Accepter la réception du message m, puis supprimer le message ACK généré par son TRD_{nc}.

La figure 2.6 illustre les différents résultats dans l'évaluation des hôtes H_u, H_{cn} et H_c par le TRD_u, selon les actions de suppression menées à différents points du protocole par ces derniers. Chaque nœud de l'arbre décrit l'identité de l'hôte et l'action qu'il est censé réaliser. Les arêtes représentent quant à elles les interactions possibles entre l'hôte et le réseau dans le cas d'une action d'émission, ou entre l'hôte et son TRD dans le cas d'une action de réception. Elles sont chacune numérotées par un entier n : $n = 1$ pour caractériser le fait que l'action attendue a correctement été réalisée par l'hôte, et $n = 0$ dans le cas contraire. Il existe une situation dans l'arbre pour laquelle $n = 2$. Il s'agit du cas où les hôtes H_{nc} et H_c ont tous deux correctement suivi les règles de fonctionnement du protocole. H_u reçoit par conséquent deux messages d'acquittement et n est alors égal à deux lorsque ces deux messages sont transmis au TRD_u. Finalement, ce sont les feuilles de l'arbre qui représentent le résultat de l'évaluation du comportement des hôtes H_u, H_{cn} et H_c par le TRD_u . Ces résultats, obtenus en fonction des acquittements reçus par le TRD_u, mettent uniquement en évidence la détection de fautes en émission ou en réception. Elles sont soulignées par une incrémentation des compteurs de fautes locaux. En d'autres termes, lorsqu'aucune faute n'est décelée pour un hôte, le compteur local qui lui est associé reste inchangé.

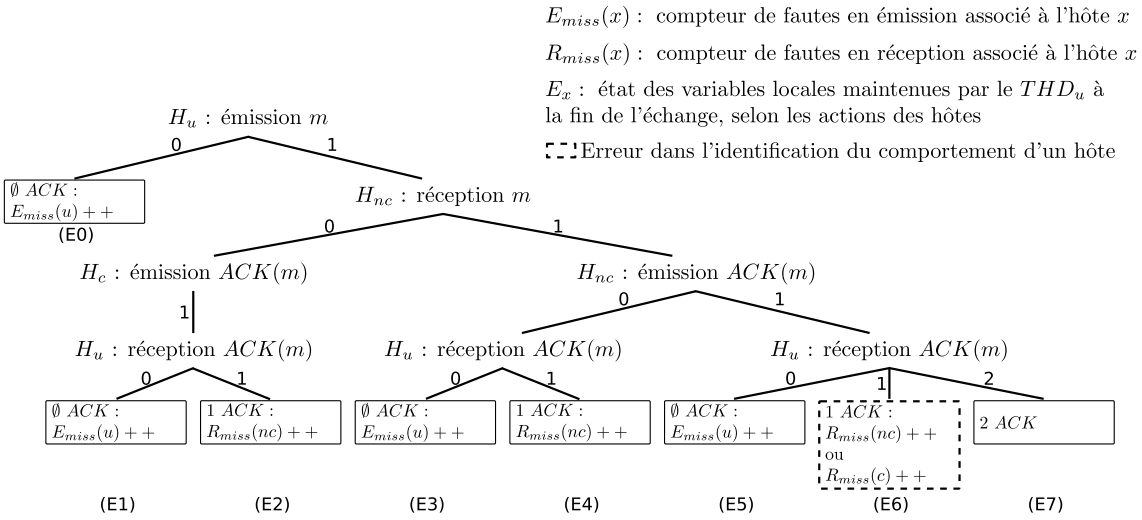


Fig. 2.6 – Résultats d'évaluation du comportement des hôtes selon les actions de suppression menées à différents points du protocole : les hôtes H_u, H_{nc} et H_c sont évaluées par le TRD du nœud u pour la transmission d'un message m.

Nous pouvons constater que dans la plupart des cas, le comportement des hôtes est correctement identifié par le TRD_u. Cependant, nous observons une situation dans laquelle un hôte H_u parvient à fausser la procédure d'identification de son TRD_u : il s'agit de l'état $E6$ de la figure 2.6. Elle survient lorsque l'hôte H_u supprime de manière sélective un des multiples acquittements qu'il reçoit (celui de l'hôte H_c ou H_{nc}). En effet, sur les deux acquittements que l'hôte H_u reçoit de ses voisins, s'il transmet seulement celui de l'hôte

H_{nc} (et inversement celui de H_c) à son TRD_u, alors l'hôte H_u sera identifié comme étant un bon récepteur tandis qu'une faute en réception sera comptabilisée pour l'hôte H_c (et inversement H_{nc}).

Comme nous le verrons dans la section suivante, cette erreur d'évaluation peut se reproduire par d'autres moyens par les hôtes hostiles, et met en avant certaines limitations dans l'évaluation précise des comportements.

2.3.7 Limitations

Une technique basée sur des acquittements sert à un TRD pour conclure positivement ou négativement sur la réalisation d'une opération d'émission par son hôte, puis d'autre part pour conclure positivement ou négativement sur la réalisation d'une opération de réception pour chacun de ses hôtes voisins. Cependant, cette technique a certaines limitations. Le principal problème vient du fait que nous sommes dans l'incapacité de définir sous quelle forme une opération d'émission est conduite par un hôte. Nous ne pouvons actuellement pas discerner une émission en diffusion locale (soit $u \rightarrow \vartheta(u) : m$) et une émission ciblée (soit $u \rightarrow v : m$). Il s'ensuit que des attaques visant à tromper le mécanisme d'évaluation du comportement d'un hôte sont possibles. Elles surviennent lorsque des hôtes hostiles forment une collusion, lorsqu'un hôte embarque plusieurs TRD, ou lorsqu'un hôte utilise une antenne directionnelle pour transmettre ses messages.

La figure 2.7 illustre ce problème dans le cas d'une collusion entre les hôtes H_u et H_v. Ces deux hôtes établissent un tunnel. Ce tunnel peut par exemple être obtenu par chiffrement des messages au moyen d'une clé secrète k_c connue uniquement par H_u et H_v. Ainsi, à la réception d'un message m de son TRD_u, l'hôte hostile H_u chiffre m avec k_c avant de le diffuser sur le réseau (étape 2 de la figure 2.7). Il en découle que seul l'hôte H_v (ou plus généralement tout hôte appartenant à la coalition de H_u) est en mesure d'interpréter, de faire traiter par son TRD, puis d'accuser la réception de ce message (étapes 3, 4, 5 de la figure 2.7). A travers cette attaque, l'hôte H_u parvient à tromper la fonction d'évaluation de son TRD_u, et ce en deux points :

- Tous les hôtes présents dans le voisinage de u et en dehors de sa collusion seront identifiés comme fautifs au regard de l'opération de réception.
- H_u échappe à la procédure de détection : il ne sera pas identifié comme étant non conforme dans l'opération d'émission.

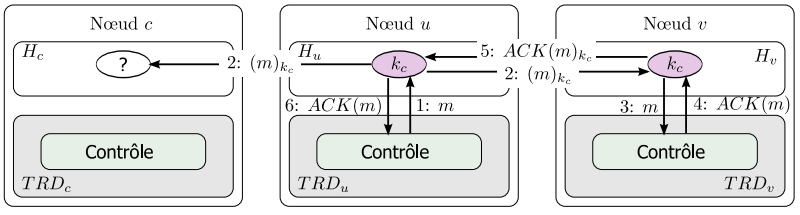


Fig. 2.7 – Attaque par collusion entre les hôtes H_u et H_v.

En considérant qu'un nœud embarque deux TRD, le même type d'attaque peut être mené.

Contrairement à une antenne omnidirectionnelle pour laquelle les émissions et les réceptions s'effectuent dans toutes les directions (360°), une antenne directionnelle est une antenne grâce à laquelle les angles d'émission et de réception peuvent être paramétrés. Grâce à ce type de matériel de communication, un hôte hostile à la capacité de cibler les hôtes récepteurs de ses messages et les mêmes effets peuvent être atteints.

2.3.8 Vérification de la conformité d'un lien entre deux hôtes

Cette première analyse met en évidence le fait qu'il est particulièrement difficile d'identifier de manière précise le comportement des hôtes dans leurs opérations d'émission et de réception. Il subsiste actuellement de trop nombreuses situations de détournement pour que la couche de contrôle puisse être utilisée telle quelle. Une solution pour circonvenir ces problèmes de détournements consiste à restreindre l'évaluation au niveau des liens entre les hôtes. L'alternative proposée vise à comptabiliser toute suppression d'un message comme une faute sur un lien plutôt que de chercher à déterminer avec précision quelle est l'extrémité du lien qui en est à l'origine.

Pour cette solution, aucune modification n'est requise dans les phases d'échanges du protocole. Seule la manière dont les informations collectées sont exploitées est revue. Afin de comptabiliser les fautes sur les liens, un TRD maintient une liste avec les identifiants de tous ses voisins et leur compteur de fautes associé. Suite à la diffusion d'un message m par un TRD et quel que soit le comportement des hôtes impliqués dans l'échange, les compteurs sont mis à jour de la façon suivante : tout nœud voisin pour lequel un TRD ne reçoit aucun accusé de réception dans le délai imparti comptabilise cette perte comme étant une faute sur leur lien. Il incrémente en conséquence le compteur qui lui est associé.

Nous avons traité jusqu'à présent uniquement le cas des messages émis par diffusion. Il est important de noter que cette alternative présente l'avantage d'être applicable, sans imposer de surcoûts en nombre de messages ou de modifications importantes, à la vérification des émissions point à point, c'est-à-dire à destination d'un seul nœud voisin. Au regard de l'évaluation, la principale différence avec l'émission par diffusion est qu'au lieu d'évaluer l'ensemble des liens avec son voisinage, le TRD évaluera seulement le lien avec le nœud destinataire du message.

Bien que cette solution entraîne une perte inévitable de précision, nous pensons que les informations collectées localement peuvent être utiles. Elles peuvent, par exemple, servir à des protocoles de routage qui se fondent sur une évaluation de la qualité des liens pour construire des routes fiables entre des nœuds tels que Associativity-Based long-lived Routing protocol (ABR) [Toh96].

De toute évidence, l'utilisation directe des résultats d'évaluation des liens (compteurs de fautes) par les TRD pour mettre en place des mécanismes de contre-mesures, telles que l'exclusion ou l'évitement d'un lien n'est pas souhaitable. La principale raison est que plusieurs facteurs tels que les capacités de traitement limitées des nœuds ainsi que les conditions variables de transmission par onde radio conduisent à l'apparition de pertes fortuites de messages. Pour répondre à ce problème, des degrés de confiance doivent être localement calculés pour chaque lien, et ce à partir des évènements observés par un TRD. Le degré de confiance local pour un lien traduit plus particulièrement l'idée que se fait un

TRD sur la fiabilité d'un lien avec un autre TRD voisin pour lequel des échanges dans le passé ont été soumis à vérification. Une solution minimaliste consiste alors à calculer le taux de pertes pour chacun des liens, puis à utiliser le résultat comme degré de confiance ; ainsi, plus le taux de perte sera faible, plus grand sera le degré de confiance accordé dans le lien. Selon les besoins, un seuil critique peut être fixé par le système et lorsque le taux de pertes sur un lien dépasse ce seuil, alors des actions de contre-mesures plus incisives pourront être entreprises. Il ne s'agit ici que d'un exemple simple d'utilisation des compteurs locaux de fautes sur les liens. Mais de telles informations peuvent bien évidemment servir à la mise en place de systèmes de réputation plus complexes, qui prendraient par exemple en considération la notion de temps (avec ou sans échec) pour faire évoluer les degrés de confiance des liens.

2.4 Application aux protocoles de routage réactifs

Les échanges d'informations constituent une communication à laquelle les TRD réagissent en changeant leurs états. Notre objectif est de montrer comment ces décisions locales peuvent servir à faire émerger des propriétés intéressantes dans l'évaluation plus générale de l'opération de retransmission. Pour cela, nous avons cherché à appliquer la technique d'identification des fautes sur les liens par les TRD à l'opération de découverte de chemins des protocoles de routage réactifs.

Cette opération est initiée lorsqu'un nœud source cherche à communiquer avec un nœud distant (à plus d'un saut). Elle est fondée sur un protocole de diffusion de proche en proche : le nœud source diffuse un message de demande de chemin vers ses voisins qui les relaient à leur tour jusqu'à ce que tous les nœuds du réseau soient joints. Une fois atteint, le nœud ciblé répond à la demande par un message point à point. Ce message sert à informer le nœud source de sa présence ainsi que du chemin à emprunter pour le joindre. La figure 2.8 illustre cette opération pour la construction d'un chemin entre le nœud s et d.

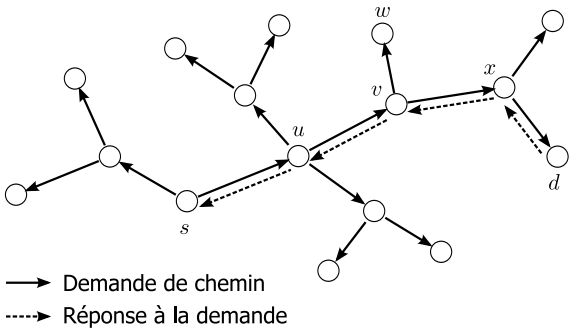


Fig. 2.8 – Diffusion dans les protocoles réactifs pour la recherche d'un chemin.

Dans la mesure où un préalable à l'identification des fautes sur les liens est la connaissance du voisinage des nœuds par les TRD, nous présentons et discutons dans la section suivante une approche simple qui nous permet d'obtenir cette connaissance.

2.4.1 Connaissance approximative du voisinage

Dans le cas de l'émission d'un message par diffusion, le TRD a l'origine de cette émission doit, avant toute évaluation, déterminer quels sont les hôtes voisins pour lesquels un acquittement est attendu. C'est grâce à cette connaissance que seuls les hôtes atteints par l'émission sont contrôlés par le TRD jusqu'à la réception d'un acquittement ou jusqu'à l'expiration du temps d'attente. Selon un fonctionnement classique, la découverte de la topologie du réseau (incluant le voisinage) est assurée au moyen d'échanges de messages de contrôle. Or c'est justement l'émission et la réception (et par transitivité, la retransmission) de ces messages que nous cherchons à vérifier. Pour répondre à ce problème, nous proposons de construire localement une vue approximative du voisinage d'un nœud sur la base des messages de contrôle et de données qu'il reçoit.

2.4.1.1 Principe

Chaque TRD maintient une table de voisinage. Une entrée de cette table contient l'identité du nœud émetteur d'un message et l'heure à laquelle il a été réceptionné. A la réception d'un message (de contrôle ou de données), le TRD extrait l'identité du nœud émetteur. Si une entrée associée à l'identité extraite existe, alors l'heure de réception est mise à jour ; sinon, une nouvelle entrée est créée. En raison de la mobilité des nœuds, les informations de voisinage enregistrées dans cette table ne sont valides que pour une durée limitée. En particulier, il est possible de considérer les entrées des nœuds pour lesquels aucune activité n'est observée pendant un certain temps comme ne faisant plus partie du voisinage, et doivent par conséquent être supprimées. Une difficulté est alors d'estimer le temps à partir duquel une entrée associée à un nœud est considérée comme caduque : si le temps fixé est trop élevé, alors des nœuds en dehors du voisinage physique d'un autre nœud continueront à être évalués par le TRD de ce dernier ; à l'inverse, s'il est trop court, des nœuds appartenant toujours à son voisinage ne seront pas évalués. Il est à noter qu'essentiellement deux paramètres interviennent dans l'estimation d'un temps approprié : le volume du trafic de données et la mobilité des nœuds. Avec un trafic faible, moins de messages seront reçus. Par conséquent, le temps fixé devra être plus important de sorte à éviter que des nœuds pourtant voisins soient exclus de la table de voisinage. Par contre, en cas de forte mobilité, le temps fixé devra être réduit pour éviter que des nœuds en dehors du voisinage physique soient toujours présents dans la table de voisinage.

2.4.1.2 Discussion

La table de voisinage maintenue par un TRD est mise à jour à partir des messages de contrôle et de données. Or ces messages passent par l'intermédiaire de l'hôte. En raison des pertes éventuelles de messages, la connaissance sur le voisinage ainsi construite est approximative, car elle échouera à découvrir tous les voisins. Est-il alors possible pour un hôte hostile de tirer profit de ces suppressions afin de tromper le mécanisme d'évaluation de son TRD ?

Le TRD d'un nœud comptabilise des fautes seulement sur les liens avec les nœuds présents dans la table de voisinage. A travers des actions de suppression (éventuellement

de manière sélective) des messages qu'il reçoit de ses voisins, un hôte hostile parvient à altérer la table de voisinage construite par son propre TRD. Il s'ensuit que les liens avec les TRD de nœuds pourtant voisins et pour lesquels des messages ont été supprimés ne seront pas évalués par le TRD de l'hôte hostile. Même si ce type de comportement conduit à une réduction du nombre de liens évalués, il ne procure que peu d'avantages à l'hôte hostile. La raison en est que la suppression en réception d'un message par hôte hostile n'influence pas le résultat d'évaluation réalisé par le TRD d'un nœud voisin.

Par exemple, considérons les hypothèses suivantes :
- Les nœuds u et v sont voisins, soit $u \in \vartheta(v)$ et $v \in \vartheta(u)$.
- L'hôte H_u est conforme.
- L'hôte H_v est hostile : il omet de transmettre à son TRD_v les messages de u, mais continue de transmettre les messages de son TRD_v sur le réseau. (Notons que cette hypothèse est réaliste, car un nœud dont l'hôte ne transmet aucun des messages de son TRD est inexistant dans le réseau.)

Dès lors que le TRD_u reçoit un message de v, il reconnait v comme étant un voisin et l'enregistre dans sa table de voisinage. Ainsi, en l'absence d'acquittements du nœud v pour les messages m, le TRD_u comptabilisera ces pertes sur le lien avec v.

2.4.2 Protection de la découverte de chemins

2.4.2.1 Attaque sur la retransmission

Dans la suite de cette section, nous étudions la première phase de l'opération de découverte de chemins, à savoir la diffusion d'une demande de chemin vers la destination. A partir de la configuration réseau de la figure 2.8, nous étudions plus particulièrement trois scénarios possibles d'attaques dans les interactions entre les hôtes et les TRD. Un premier dans lequel l'hôte H_v omet de transmettre à son TRD_v le message de demande de chemin m reçu de l'hôte H_u, un second dans lequel l'hôte H_v omet de retransmettre sur le réseau le message m à ses voisins, et un dernier cas dans lequel les hôtes H_v et H_w sont en collusion. Remarquons que si le protocole le nécessite, le message m peut subir des modifications lors de son traitement par un TRD. En outre, les paragraphes suivants n'ont pas pour vocation de traiter dans leur globalité tous les cas possibles d'attaques, mais seulement d'en détailler quelques-uns afin de mieux comprendre comment notre méthode d'identification des fautes sur les liens peut être appliquée à la fonction de retransmission des messages de diffusion et à la détection de l'attaque par encapsulation.

Scénario 1 Dans ce premier exemple, l'hôte H_u émet en diffusion locale le message m généré par son TRD_u. Nous considérons que l'hôte H_v omet de retransmettre à son TRD_v le message m. D'un point de vue local, le résultat de cette action est que TRD_u ne recevra en retour aucun acquittement authentifié pour m par le TRD_v. Il s'agit du chemin $E2$ de l'arbre décrit figure 2.6. Le TRD_u, émetteur du message devant être retransmis comptabilisera une faute sur le lien entre u et v.

Scénario 2 Dans ce scénario, nous considérons que l'hôte H_v reçoit le message m en provenance de H_u et le transmet à son TRD_v. Soulignons que le message m est par nature

un message devant être retransmis. Ainsi, au regard du TRD_v, le traitement de ce message a deux conséquences :

- La génération d'un message ACK destiné au TRD_u.
- Et s'il s'agit de la première fois que le message m est traité, alors le TRD_v appose un nouvel en-tête de sécurité (lequel contient son identité de nœud) au message m, et le transmet à son hôte H_u. Dans le même temps, le statut du TRD_v évolue. L'identifiant de ce message est enregistré dans la table de vérification des messages, et il devient donc un émetteur de m en attente d'un acquittement par l'ensemble de ses nœuds récepteurs voisins.

En considérant que l'hôte H_v ne diffuse ni l'ACK, ni le message m, alors du point de vue du TRD_u, une faute sur le lien entre u et v sera comptabilisée (chemin EA de l'arbre figure 2.6). Du point de vue du TRD_v, une faute sur le lien entre v et chacun de ses voisins (à l'exception de u, initialement émetteur du message) sera comptabilisée (chemin $E0$ de l'arbre Figure 2.6).

Scénario 3 Nous prenons un dernier exemple dans lequel les hôtes H_v et H_w sont en collusion. Dans ce scénario, l'hôte H_v reçoit le message m en provenance de H_u sur son interface de communication et le transmet vers son TRD_v. Contrairement à l'exemple précédent, le message m est retransmis par l'hôte H_v. Cependant, cette retransmission est réalisée dans des conditions particulières, à savoir uniquement vers le nœud w par le biais d'un tunnel. Le résultat direct de cette action est que du point de vue du TRD_v, puisque aucun nœud présent dans le voisinage de v (à l'exception de w) n'est en mesure d'émettre un acquittement pour m (car non interprétable par ces derniers), une faute sur leur lien sera comptabilisée.

Discussion Une propriété intéressante que nous pouvons observer est qu'à partir du moment où un hôte mène des actions incorrectes au regard des règles de réception et d'émission (et par extension de retransmission) définies par le protocole, alors un, voire plusieurs de ses liens avec ses voisins seront toujours incriminés par les TRD.

Ensuite, des degrés de confiance dans les liens peuvent être déterminés par les TRD des nœuds, et ce avec chacun des autres nœuds présents dans leur voisinage. Ces valeurs évoluent à partir des messages d'acquittement reçus pour chaque émission réalisée. Dans les futurs échanges, ces degrés de confiance pourront être utilisés afin d'aider la prise de décision locale, comme le calcul des chemins à emprunter ou l'isolement de nœuds, apportant ainsi plus de fiabilité dans l'opération d'acheminement des messages. Par exemple, tout lien présentant un faible degré de confiance pourra être évité dans l'établissement d'un chemin.

Notons cependant que la perte de messages de contrôle en général, et de demande de chemin en particulier, ne devrait pas être tolérée, et ce pour deux raisons principales. La première est qu'elle permet à des nœuds non coopératifs de s'exclure complètement de la formation des chemins. La seconde est qu'elle rend possible l'accomplissement de l'attaque du tunnel par encapsulation décrite en section 1.4.1.2. Par conséquent, la comptabilisation d'une perte d'un tel message sur un lien entre deux nœuds devrait être traitée avec une

attention particulière, et éventuellement conduire à des réactions plus sévères. Nous montrons dans la section suivante comment les informations collectées par les TRD peuvent être exploitées pour éviter l'attaque par encapsulation.

2.4.2.2 Attaque par encapsulation

Dans une attaque par encapsulation des messages de contrôle, deux nœuds attaquants et non adjacents sur un chemin entre une source et une destination parviennent à raccourcir le descriptif d'un chemin en construction. Dans cette section, nous montrons comment utiliser simplement une information relative à la perte d'un message de demande pour détecter et empêcher la construction d'un tel chemin.

Comme mentionné en section 1.4.1.2, deux attaquants en collusion échangent des messages de contrôle entre eux en utilisant des paquets de données. Ils utilisent en particulier les chemins offerts par le réseau pour procéder à ces échanges. Etant donné que le message de contrôle, inclus dans un paquet de données, est relayé par plusieurs nœuds intermédiaires avant d'atteindre le nœud attaquant, les délais de transmissions sont sensiblement plus élevés que celui fixé pour τ. Par conséquent, le TRD du premier nœud attaquant détectera la perte d'un message de demande de chemin sur le lien avec le second nœud attaquant. En revanche, même si une perte est localement détectée par le TRD du premier attaquant, le message de demande de chemin poursuit son cheminement jusqu'à atteindre la destination. Cette dernière y répond par un nouveau message, contenant alors un descriptif de chemin raccourci par les nœuds attaquants. Dans le but d'empêcher la validation par le nœud source d'un chemin éventuellement corrompu, les informations sur la connectivité des liens identifiées lors de la retransmission de la demande sont ajoutées au message de réponse par les TRD des nœuds intermédiaires appartenant au chemin formé.

Les opérations de contrôle associées à la diffusion d'une demande de chemin sont illustrées figure 2.9. Nous supposons que le nœud source s envoie un message de demande de chemin vers la destination d. Ce message est de la forme $< rreq, id_s, id_d, id_{req}, chemin >$, avec s l'identité du nœud source, d celle du nœud destinataire, id_{req} l'identifiant du message, et *chemin* le descriptif du chemin. Les opérations de contrôle réalisées par les TRD sont les suivantes (dans le but d'en faciliter la lecture, toutes les manipulations réalisées par les TRD sur les en-têtes de sécurité ont volontairement été omises de cette description) :

- A la réception du message $rreq$ par le TRD d'un nœud intermédiaire, un acquittement est généré et transmis à l'hôte. Ensuite, s'il s'agit de la première fois que le message est traité par le processus de routage, alors l'identité du nœud est ajoutée au descriptif du chemin et le message $rreq$ doit être retransmis à l'hôte. Avant de le retransmettre, le TRD l'enregistre dans une table de vérification des demandes de chemin (dénotée VT_{rreq}). Une entrée de cette table contient le triplet (id_s, id_d, id_{req}) identifiant de manière unique le message $rreq$, la liste des nœuds voisins devant y répondre par un acquittement, le délai de réception de l'acquittement, et une information R_{ack} caractérisant la réception (si R_{ack} vaut 1) ou non (si R_{ack} vaut 0) d'un acquittement. Cette table sert à garder une trace des réponses escomptées des nœuds voisins. Cependant, contrairement à la table de vérification décrite en section 2.3.4, une entrée de cette table est maintenue jusqu'à la réception d'une réponse associée

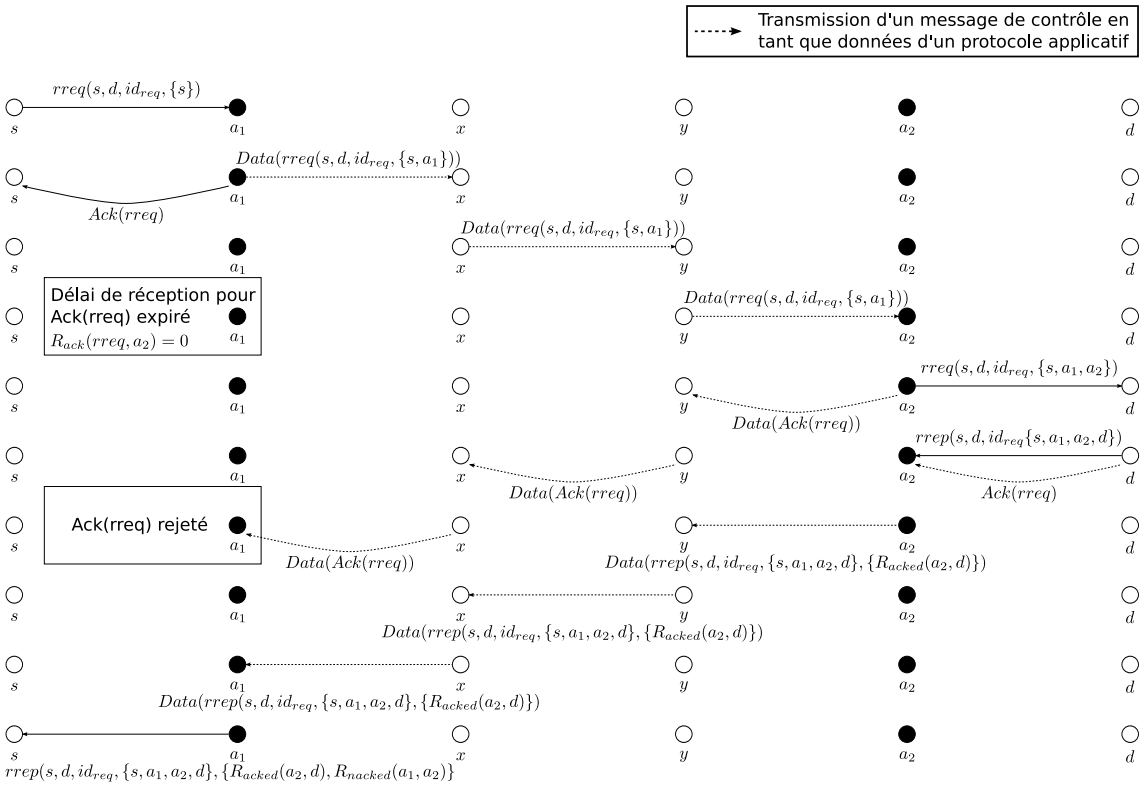


Fig. 2.9 – Exemple des opérations et des échanges de messages entre les nœuds pour la détection de l'attaque par encapsulation. L'initiateur de la demande d'un chemin est s et la destination est d. id_{req} est l'identifiant de la demande. Les nœuds intermédiaires a_1 et a_2 encapsulent les messages de contrôle fournis par leur TRD dans des paquets de données avant de les retransmettre.

à la demande de chemin enregistrée, ou jusqu'à expiration d'un délai plus important (en cas d'absence de réponse).

- A la réception d'un message d'acquittement par un TRD, l'identité du nœud émetteur est extraite. Si une entrée correspondante au message $rreq$ acquitté est trouvée dans le table VT_{rreq}, alors la valeur R_{ack} associée au nœud émetteur est mise à jour (elle passe à 1).
- A la réception du message $rreq$, le TRD_d génère un message de réponse de la forme $< rrep, s, d, id, chemin >$.
- Au fur et à mesure que la réponse à la demande de chemin revient vers la source, le TRD d'un nœud intermédiaire vérifie son appartenance au chemin fourni. Si c'est le cas, il extrait l'identité du nœud successeur et recherche dans sa table VT_{rreq} une entrée correspondante au message $rrep$ reçu. Une fois trouvée, la valeur R_{ack} associée au nœud successeur dans le chemin est apposée au message $rreq$. En cas d'échec lors d'une vérification, le message est supprimé, sinon il est retransmis par le TRD.
- A la réception du message $rreq$, le TRD_s a connaissance des éventuels échecs survenus lors de la transmission de son message de demande de chemin et peut décider d'accepter ou rejeter le chemin construit.

Discussion Notre approche pour détecter les attaques par encapsulation repose sur le contrôle des délais de transmission à un saut des messages. Elle est adaptée aux protocoles de routage réactifs qui construisent les chemins les plus courts en termes de nombre de sauts. Elle exploite directement les informations de fautes collectées sur les liens et exige peu de modifications des messages de contrôle du protocole de routage. Néanmoins, ce type de détection échouera très certainement dès lors que les deux attaquants en collusion seront relativement proches l'un de l'autre. Cependant, le principal objectif de l'attaquant à travers cette attaque est d'appartenir au chemin sélectionné par le nœud source. Pour y parvenir, le chemin construit doit être plus court que tous les chemins existants. Etant donné que plus la distance en nombre de sauts séparant les deux nœuds attaquants est importante, plus le raccourcissement sur le chemin final construit est significatif, notre approche permet de réduire les chances de réussite de cette attaque.

2.5 Analyse théorique de la complexité

Plusieurs facteurs tels que la fonction de hachage cryptographique, la taille du code d'authentification et les capacités de calcul de l'environnement de confiance peuvent avoir une influence sur les performances globales du système. C'est la raison pour laquelle nous proposons dans cette section une évaluation théorique des coûts induits par notre schéma d'acquittement.

2.5.1 Coût en volume de données supplémentaire

Nous avons envisagé l'utilisation de la cryptographie symétrique pour ses plus faibles coûts calculatoires, et plus particulièrement l'ajout d'un code d'authentification de message pour assurer l'authenticité et l'intégrité des messages échangés entre les TRD. La taille de l'en-tête de sécurité ajouté à tout message peut être considérée comme constante et dépendante de trois paramètres : la taille du condensé, la taille du numéro de séquence, et la taille de l'identité du nœud émetteur. En prenant l'hypothèse d'un numéro de séquence et d'une identité de nœud chacun représentés sur 32 bits, puis d'un condensé de 160 bits (pour la fonction de hachage SHA-1), la taille de l'en-tête de sécurité est alors de 224 bits par message, et la taille d'un message ACK est de 256 bits. De plus, le volume supplémentaire d'octets associé aux messages ACK est linéaire avec le nombre de messages et de voisins.

Notons que tout système assurant l'authenticité et l'intégrité des messages induit un coût dû au temps de calcul pour la génération du condensé. Sur coprocesseur sécurisé ou une carte à puce, grâce aux accélérateurs cryptographiques matériels, ce coût peut être considéré comme relativement faible.

2.5.2 Coût en nombre de messages supplémentaire

Tout message émis par un nœud source est immédiatement acquitté par chacun des nœuds présents dans son voisinage direct de communication, et ce message ACK valide la réception d'un seul message. Cette approche a pour conséquence une augmentation importante du trafic en nombre de messages et peut conduire à une congestion temporaire du

canal de communication. Dans le cas où tous les nœuds voisins émettent leur acquittement à intervalles rapprochés, alors des collisions sont à prévoir, entraînant éventuellement des pertes accidentelles de messages. Pour pallier ce problème, d'autres schémas moins coûteux en nombre de messages additionnels peuvent être envisagés tels que :

Cumuler les acquittements pour plusieurs messages reçus. Comme cela a été présenté par Balakrishnan *et al.* dans [BDV05a], au lieu d'exiger un acquittement explicite pour chaque message, une approche consiste pour un nœud à cumuler les acquittements pour plusieurs messages reçus (éventuellement en provenance de plusieurs émetteurs), puis à les envoyer dans un seul message ACK. Cette approche permet de réduire considérablement le nombre de messages. Dans le pire des cas, le nombre de messages sur une unité de temps est linéaire avec le nombre de voisins. En contrepartie, comme le délai accordé par le nœud émetteur pour la réception de ce type d'acquittement augmente, la détection des fautes est moins efficace. Des comportements hostiles tels que l'introduction de retards ou l'encapsulation des messages reçus peuvent passer inaperçus.

Acquitter seulement un sous-ensemble de messages. La méthode proposée par Liu *et al.* [LDVB07] pour laquelle seulement une fraction des messages sont acquittés induit également une augmentation des délais de détection des comportements de suppression.

Inclure les acquittements dans d'autres messages du protocole. Une autre solution pour maintenir un délai de réception des acquittements relativement bas tout en limitant au mieux le nombre de messages consiste à inclure les acquittements dans les messages « normaux » des protocoles traités par le TRD. Comme la quantité de messages « normaux » varie dans le temps, leur seule utilisation pour l'envoi des acquittements n'est pas envisageable. En effet, si dans l'intervalle de temps entre le traitement d'un message par un nœud récepteur et l'expiration du délai de ce même message par le nœud émetteur, alors une faute sera imputée à tort au nœud récepteur. Pour traiter cette situation, nous suggérons une approche hybride où la méthode d'envoi d'un acquittement est choisie en fonction du volume du trafic. En cas de trafic faible, un nœud récepteur confirme la réception d'un message par l'émission d'un acquittement dédié, tandis qu'en cas de trafic important, le ou les acquittements pour plusieurs messages sont inclus dans des messages « normaux ».

2.5.3 Coût en espace mémoire

Puisque les ressources fournies par les cartes à puce sont fortement limitées, une préoccupation majeure est le stockage. Dans cette section, nous étudions les coûts en stockage induit par notre schéma d'acquittement.

Chaque nœud maintient une table de vérification contenant le numéro de séquence, l'identité d'un nœud voisin, et le temps d'expiration associés aux messages pour lesquels il est en attente d'un acquittement. Ici, le temps d'expiration τ représente le temps pendant lequel un message peut être acquitté par un nœud voisin. Une fois ce temps dépassé, l'entrée est supprimée de la table. En prenant l'hypothèse qu'une entrée de cette table est

représentée sur 96 bits (3*32), si un nœud émet M messages à N nœuds dans son voisinage par unité de temps (ut), alors dans le pire des cas, il en résulte une occupation mémoire de $\tau * 96 * N * M$ bits (si $\tau \geq 1ut$) à un instant donné, ou de $1ut * 96 * N * M$ bits sinon. Soit pour un scénario où un nœud est entouré de $N = 50$ voisins et émet $M = 10$ messages par seconde. Pour $\tau = 0.25$ secondes (d'après [ACH+08]), l'occupation mémoire est de 6 ko.

Une autre table contenant l'ensemble des identités de nœuds voisins et leur compteur de fautes (calculé à partir des messages acquittés) est également maintenue. Une entrée de cette table est représentée sur 64 bits ($2 * 32$). L'occupation mémoire de cette table est linéaire avec le nombre de nœuds voisins. Si chaque nœud a en moyenne $N = 50$ voisins, il en résulte une occupation mémoire de 400 octets.

Actuellement, une carte à puce typique est dotée de 64 à 256 ko de mémoire de stockage persistante, ce qui couvre les besoins en espace mémoire requis par notre approche. Notons par ailleurs que les cartes à puce de nouvelle génération offrent de plus grandes capacités de stockage et peuvent par conséquent être déployées pour répondre à des scénarios plus complexes (faisant interagir plus de nœuds, et impliquant plus de messages).

2.6 Conclusion

Dans ce chapitre, nous avons proposé l'utilisation de cartes à puce (ou de coprocesseurs sécurisés) pour sécuriser les protocoles de routage.

Nous avons tout d'abord montré (section 2.2.4) que du fait de la non-fiabilité des canaux de communication entre un TRD et son hôte, il était impossible de garantir le fonctionnement correct de ces protocoles uniquement en sécurisant leur exécution sur chacun des nœuds participants.

Dans ce cadre, nous avons décrit un schéma d'acquittement local (section 2.3.5), exécuté à l'intérieur d'un matériel résistant à la manipulation, permettant de détecter de manière fiable le comportement des hôtes au regard des opérations bas niveau que sont l'émission et la réception des messages.

Ensuite, nous avons effectué l'analyse détaillée de ce schéma (section 2.3.6). Elle a fait ressortir des mesures potentiellement exploitables par un TRD pour rendre plus fiables l'opération de retransmission. Cependant, nous avons montré qu'à cause des attaques par émission ciblée, nous sommes limités, dans notre cas, à compter les pertes sur liens entre les nœuds. Néanmoins, même si ces informations sont beaucoup moins précises que ce que nous aurions pu espérer obtenir, nous considérons qu'elles pourront servir à la prise de décision pour les protocoles de routage ou des systèmes de plus haut niveau tels que des systèmes de réputation.

Nous avons ensuite appliqué notre méthode d'identification des fautes sur les liens à l'opération de découverte de chemins des protocoles de routage réactifs. Nous avons montré (sections 2.4.2.1 et 2.4.2.2) que les informations collectées sur les liens avec notre méthode se révèlent particulièrement utiles pour détecter les fautes de retransmission et réduire les possibilités de réussite de l'attaque par encapsulation dans la phase de recherche de chemins.

Enfin nous avons évalué le coût de l'utilisation de notre méthode au niveau du réseau (coût en nombre et taille des messages) et des nœuds (coût mémoire).

Chapitre 3

Cadre d'analyse de la sécurité des protocoles de routage

Pour sécuriser la phase de découverte de la topologie des protocoles de routage, nous avons vu au chapitre 1 (sections 1.4.1.1 et 1.4.1.2) que plusieurs travaux basés sur la cryptographie ont été entrepris. Cependant, avant de déployer une quelconque solution de routage en situation réelle, il est d'une première nécessité pour les concepteurs de connaître dans quelle mesure leur fiabilité est conservée, tout en tenant compte de la présence d'éventuels attaquants. Il s'agit de comprendre en profondeur quelles sont les propriétés de sécurité garanties pour une solution donnée, et ce, dans un environnement de déploiement donné.

Dans ce chapitre, nous discutons dans un premier temps les différentes approches existantes pour évaluer la sécurité des protocoles de routage dans les réseaux ad hoc. Ensuite, nous proposons un nouveau cadre d'évaluation systématique de la sécurité du protocole de routage OLSR. A partir des résultats d'analyse obtenus sur ce protocole de référence, nous étudions puis nous comparons deux versions renforcées d'OLSR : la première est Advanced Signature [RACM04] et la seconde est une solution utilisant un équipement matériel résistant à la manipulation. Notre objectif est de mettre en évidence les apports et les limitations de ces deux versions sécurisées d'OLSR selon différents environnements adverses.

3.1 Evaluation de la sécurité des protocoles de routage

Pour analyser la sécurité des protocoles de routage, il est nécessaire de définir le ou les objectifs du protocole, les capacités de l'attaquant à les mettre en défaut, puis de préciser quelles sont les autres hypothèses (par exemple, l'environnement de déploiement). C'est en fonction de ces réductions du problème qu'il est ensuite déterminé si un protocole satisfait ou pas une propriété de sécurité donnée.

Actuellement, les méthodes d'analyse des propriétés de sécurité des protocoles de routage dans les réseaux ad hoc sont assez disparates. Nous pouvons distinguer trois catégories de méthodes :

– L'inspection humaine.
– La simulation.
– La spécification et vérification formelle.

Les deux premières méthodes ne sont en général pas guidées par un cadre complet et systématique d'analyse. Elles produisent par conséquent des résultats qui ne sont valables que dans certains cas de figure. A l'opposé, dans la mesure où les méthodes formelles de spécification et de vérification de protocoles fournissent un cadre complet et structuré, elles offrent une certaine forme de complétude dans les résultats de validation produits. Pour une propriété de sécurité et un scénario donné, elles parviennent soit à prouver qu'une propriété de sécurité est satisfaite soit à évaluer toutes les possibilités de sorties du protocole. Ainsi, s'il existe une attaque sur la spécification formelle d'un protocole, alors elle sera découverte. Dans le cas contraire, l'absence d'attaque dans les limites fixées par le scénario sera la preuve que la spécification formelle ne contient, a priori, pas de failles. Néanmoins, ce type de méthode s'avère particulièrement difficile à appliquer, car des restrictions ou des concessions doivent être faites sur la représentation du protocole et/ou sur l'environnement dans lequel il opère (sur les canaux de communication par exemple).

Dans les sections suivantes, nous discutons les différentes méthodes existantes pour analyser la sécurité des protocoles de routages dans les réseaux ad hoc.

3.1.1 Inspection humaine

L'inspection humaine constitue une première étape dans l'évaluation des propriétés de sécurité atteintes par les protocoles. Elle est conduite à partir de l'examen des différents éléments qui définissent le fonctionnement d'un protocole, à savoir ses intervenants, leurs rôles, leurs états locaux, et leurs interactions à travers les échanges de messages. Puisque les interactions entre les différents intervenants sont parfois complexes, elle n'est en général pas considérée comme une méthode exhaustive d'identification de tous les scénarios d'attaques possibles. Elle n'en demeure pas moins une méthode utile, et est actuellement la plus couramment utilisée dans le processus d'évaluation de la sécurité des protocoles de routage. La raison principale est qu'elle est relativement simple. Elle permet de révéler des scénarios d'attaques auparavant inconnus, sans pour autant demander d'investissements particuliers, à l'exception d'une compréhension fine du protocole étudié.

En réponse aux attaques décelées par inspection humaine, des extensions de sécurité telles que AdvSig [RACM04], ARAN [SDL+02], Ariadne [HPJ02, HPJ05], SEAD [HJP02], Secure-OLSR [ACJ+03, HTR+04] ont été proposés pour y faire face. Ces extensions utilisent des mécanismes cryptographiques pour protéger la phase de découverte de chemins. Bien qu'intéressants, les scénarios d'attaques identifiés dans ces travaux ne représentent que quelques-unes des innombrables possibilités d'attaques, ce qui peut conduire à une remise en question de l'étendue de ces mécanismes de protection. Nous pouvons en particulier constater l'absence d'un cadre structuré d'analyse pour évaluer l'apport de ces extensions dans différents environnements de déploiement. En prenant en exemple le cas de l'utilisation d'un de ces protocoles sur un champ de bataille, la capture physique d'un nœud et/ou la compromission des clés cryptographiques sont des événements qui peuvent se produire.

Par conséquent, des nœuds internes dans le réseau sont potentiellement à l'origine d'attaques. Or en règle générale, l'attaquant interne n'est pas étudié de manière approfondie et systématique dans la phase d'analyse de sécurité de ces extensions. Il devient alors difficile de conclure sur leur fiabilité dans diverses situations adverses.

Pour illustrer notre propos, nous étudions les phases de développement et d'analyse des mécanismes de sécurité pour le protocole de routage proactif DSR [JMH07]. Les premiers travaux à traiter de la sécurité de DSR sont ceux de Papadimitratos et Haas [PH02]. Ils proposent l'extension SRP (voir section 1.4.1.2) dont le but est de fournir une sécurité de bout en bout entre une source et une destination, durant la phase de découverte d'un chemin. Par une inspection humaine, les auteurs discutent des scénarios d'attaques possibles et concluent que SRP est sécurisé tant que les attaquants n'entrent pas en collusion.

Dans leurs travaux, Buttyán et Vajda [BV04] montrent que l'analyse de sécurité de SRP n'est pas valide. Ils illustrent leur propos en présentant une attaque selon laquelle un attaquant indépendant parvient à injecter activement des informations erronées lors de la construction d'un chemin avec SRP. Les auteurs ont trouvé l'attaque par inspection humaine, après avoir décelé que SRP n'était pas sécurisé de manière prouvable dans leur modèle d'analyse basé sur la vérification d'équivalence (de deux abstractions du protocole).

L'extension de sécurité Ariadne (voir section 1.4.1.2) a été proposée par Hu *et al.* [HPJ02, HPJ05] dans le but de raffiner les services de sécurité offerts par SRP. Elle vise à fournir une sécurité de saut en saut en requérant une authentification par chaque nœud intermédiaire sur un chemin entre une source et une destination. Une particularité de ces travaux est la proposition d'un modèle représentant les capacités de l'attaquant. Les auteurs analysent par inspection humaine les propriétés de sécurité satisfaites par les mécanismes proposés, et ce à travers différentes classes d'attaquants. Cette analyse, bien que plus complète grâce à la prise en compte de plusieurs classes d'attaquants, a été invalidée par Buttyán, Vajda et Ács [BV04, ABV06]. Les auteurs ont mis en évidence plusieurs attaques contre Ariadne. A l'instar de SRP, le mode opératoire de ces attaques a été identifié par inspection humaine, après que leur modèle ait montré que ce protocole n'était pas sécurisé de manière prouvable.

Dans le but de corriger les défauts de sécurité du protocole Ariadne, Buttyán et Vajda ont proposé une adaptation nommée endairA [BV04] (voir section 1.4.1.2). Nous avons montré que cette extension est vulnérable à l'attaque du tunnel où deux attaquants encapsulent les messages de contrôle dans des paquets de données.

Dès lors que la conception et l'analyse d'une extension de sécurité s'appuient sur des informations partielles, des erreurs ou des attaques sont souvent révélées à postériori. Ceci soulève la nécessité d'une méthodologie complète et systématique d'analyse pour évaluer la sécurité des protocoles de routage dans les réseaux ad hoc. Dans cette optique, un premier travail est celui de Ning et Sun [NS03, NS05]. Les auteurs présentent une approche systématique pour étudier l'attaquant interne contre le protocole de routage AODV. Il y est fait état des divers buts qui peuvent être atteints par un attaquant indépendant interne, avec des attaques simples telles que le rejeu et la suppression ou modification d'une ou de plusieurs informations contenues dans un message de contrôle. Alors que cette approche met en évidence les points de faiblesses d'AODV, elle ne permet pas directement d'en évaluer des versions sécurisées, car des paramètres liés à l'environnement de déploiement

tels que les capacités de l'attaquant ou les interactions entre un nœud et son voisinage n'ont pas été introduits.

Huang et Lee [aHL04] ont également proposé une analyse systématique d'AODV. Ils ont examiné les états internes du protocole afin d'analyser les possibilités d'attaques. Les transitions et les états erronés possibles ont été identifiés, et les exploitations du protocole de routage ont été analysées. Cependant, similairement à l'approche de Ning et Sun, ils ont orienté leurs travaux autour de l'analyse du protocole AODV de référence, tel que défini dans la RCF 3561 [PBRD03], plutôt que de fournir une approche complète et systématique adaptée à l'évaluation de versions sécurisées des protocoles de routage.

3.1.2 Simulation réseau

Dans la communauté des réseaux ad hoc, les outils de simulation tels que Glomosim, Network-Simulator (plus connu sous le nom de NS-2) [ns], Omnet++ [omn], Opnet [Tec] font partie intégrante du processus de développement et de validation des protocoles de routage. Le rôle premier de la simulation est d'estimer de manière précise les performances (en termes de débit, de passage à l'échelle, de surcoût en nombre et taille des messages, etc...) d'un protocole. Plusieurs paramètres tels que la distribution géographique des nœuds, leur mobilité, leur volume (et motif) de trafic, la modélisation de l'attaquant, etc... affectent les résultats produits par une simulation du protocole. Vers une recherche d'une certaine forme d'exhaustivité, le principe de la simulation consiste alors à simuler un protocole dans différentes configurations réseau possibles. Ensuite, sur la base d'une analyse statistique des données obtenues à travers les multiples simulations indépendantes, une estimation de ses performances est fournie.

Concernant l'analyse de sécurité, les informations extraites à partir de plusieurs simulations permettent de répondre à la question « comment un protocole réagit en moyenne contre des attaques spécifiques, dans des configurations réseau données » ? De notre point de vue, la portée de ces résultats est limitée. Comme les résultats sont lissés, il devient par exemple difficile de déterminer si certaines configurations offrent de plus grandes opportunités à la réussite d'une attaque que d'autres. Un autre point délicat est la modélisation de l'attaquant. En général, son activité est définie de manière statistique : ses actions se produisent soit de manière aléatoire soit à un taux uniforme, sans prise en considération de l'évolution de l'environnement. Cette distribution statistique des actions est discutable, car elle peut limiter les opportunités de l'attaquant dans la réussite d'une attaque. Par ailleurs, même si un protocole fonctionne correctement selon un ensemble de conditions prédéfinies, il peut opérer complètement différemment dans une configuration réseau non testée par la simulation, ou lorsqu'il est sujet à des variantes d'une attaque. Il est alors difficile de déterminer de façon générale si un protocole satisfait les objectifs de sécurité fixés.

Néanmoins, la simulation est actuellement un outil utile pour estimer le surcoût introduit par les extensions de sécurité déployées. Par exemple, Hu *et al.* [HPJ05] ont estimé le surcoût introduit par leur extension de sécurité Ariadne en le comparant avec le protocole de référence DSR. En revanche, pour les raisons précédemment citées, lorsqu'il s'agit d'estimer les apports de sécurité d'une extension face aux différentes intentions de l'attaquant,

les résultats produits par la simulation sont discutables.

3.1.3 Méthodes de vérification formelle

Étant donné un modèle de la spécification d'un protocole, un modèle de l'attaquant et une propriété de sécurité à vérifier, nous voulons déterminer si le modèle satisfait bien la propriété considérée en présence de cet attaquant. Les méthodes de vérification formelle de protocoles apportent une réponse à ce problème. Ces méthodes reposent essentiellement sur une description mathématiquement formalisée des échanges de messages entre les différents intervenants du protocole, du graphe de connectivité des intervenants, des actions de l'attaquant, et des propriétés de sécurité attendues sur le protocole considéré. Une fois le système complet décrit formellement, il peut être automatiquement analysé afin de mettre en exergue des attaques ou à l'inverse, d'en garantir la sécurité. A cet effet, une procédure de recherche exhaustive de tous les états atteignables lors de l'exécution du protocole peut être utilisée. Si des états non valides surviennent, alors une trace d'attaque invalidant en conséquence la propriété de sécurité déclarée est produite.

Ces méthodes ont largement été utilisées pour évaluer la sécurité des protocoles cryptographiques [Low98, YW99, SBP01, ABB$^+$05]. Un exemple est le cas du protocole de Needham-Schrœder [NS78] - utilisé pour permettre l'échange d'une clé symétrique entre deux participants - et de ses variantes [Low95, Low96]. La propriété de sécurité considérée est alors le secret. Elle vise à déterminer si un attaquant, indépendamment des nœuds intermédiaires dans le chemin de communication, peut obtenir une information échangée entre les participants du protocole. Dans ce protocole, seulement deux participants interviennent et les échanges entre ces derniers sont simples et en nombre limité : au nombre de trois au total. Sa traduction dans un langage de description en est alors relativement aisée.

En revanche, les protocoles de routage dans les réseaux ad hoc sont plus complexes dans la mesure où ils font intervenir plusieurs nœuds intermédiaires. Etant donné que les actions de chacun de ces intermédiaires doivent être prises en considération lors de l'évaluation, la traduction du protocole dans le langage cible s'en trouve complexifiée. Cette traduction est d'autant plus délicate que certaines restrictions doivent être faites par l'analyste. Ces restrictions sont en partie dues au fait que si tous les éléments décrivant le protocole sont traduits, alors l'analyse exhaustive de toutes les possibilités de sorties du protocole peut échouer, car le système d'états qui lui est associé peut rapidement augmenter au-delà de la mémoire disponible. Pour réduire l'espace de recherche, l'analyste doit alors s'assurer que le modèle du protocole capture seulement les éléments nécessaires à l'évaluation de la propriété de sécurité étudiée. Concernant l'environnement physique, comme les nœuds ont par nature des capacités de communication limitées, le graphe de connectivité servant à décrire les liaisons directes possibles entre les participants doit également être modélisé. Or cette modélisation induit une réduction du problème, car la définition d'un graphe de connectivité particulier limite inévitablement les opportunités d'attaques sur le protocole. En particulier, la position du ou des attaquants dans le graphe de connectivité influence la quantité d'informations accumulée à partir des messages reçus, et joue par conséquent un rôle déterminant dans la réussite ou non d'une attaque. En outre, pour prendre en

considération l'attaquant dans la vérification, il est nécessaire de définir ses capacités et son comportement. En conséquence, l'étendue de l'analyse est actuellement limitée par les restrictions appliquées sur la description du protocole, les capacités de l'attaquant, voire sur le graphe de connectivité considéré.

Quelques travaux ont considéré l'utilisation de méthodes de vérification formelle pour évaluer la sécurité de la procédure de découverte de chemin des protocoles réactifs. Nous pouvons citer ceux de Nanz et Hanken [NH06] dans lesquels l'extension de sécurité Secure-AODV [ZA02] (voir section 1.4.1.1) est évaluée. Les auteurs définissent la sécurité de la découverte de chemin en terme de cohérence des tables de routage avec le graphe de connectivité. Ils illustrent leur approche en montrant qu'un attaquant interne au réseau parvient à injecter des informations topologiques incohérentes. Dans [Mar03], Marhall a proposé une évaluation de la procédure de recherche de chemin du protocole SRP afin de garantir qu'aucun chemin retourné et invalide dans le graphe de connectivité ne soit accepté par l'initiateur. Yang et Baras [YB03] ont proposé une méthode basée sur les *strand spaces* et mettent en évidence la faisabilité d'une vérification automatique.

De manière générale, ces travaux révèlent quelques attaques sur la procédure de découverte de chemins de protocoles pourtant sécurisés. Néanmoins, peu de graphes de connectivité ont été pris en considération lors de la simulation des protocoles étudiés (composés d'au maximum cinq nœuds). Nous pensons que ces restrictions trop fortes appauvrissent les résultats de validation obtenus.

3.2 Cadre d'analyse de la sécurité d'OLSR

Dans cette section, nous proposons un cadre d'analyse systématique de la sécurité des protocoles de routage. Notre approche se fonde sur une analyse par inspection humaine, et vise à mettre en exergue les opportunités données à l'attaquant pour compromettre l'intégrité des informations de routage. Elle se décompose en trois étapes successives :

1. Dans une première étape, les différents éléments qui définissent le fonctionnement du protocole de référence sont étudiés. Ces éléments comprennent : les intervenants du protocole, leurs rôles, leurs états locaux, leurs interactions à travers des échanges de messages. Cette analyse est essentielle, car elle permet de cerner la portée de chaque type de messages de contrôle et donc de mieux comprendre, en situation d'attaque, quelle sera la propagation des informations erronées ainsi que les informations utiles que pourra extraire et réutiliser un attaquant.
2. Dans une seconde étape, nous proposons d'identifier les attaques possibles contre le protocole. Nous partons de l'hypothèse qu'il est complexe, voire impossible, d'énumérer de manière exhaustive toutes les déclinaisons d'attaques possibles et réalisables par un attaquant pour nuire à l'intégrité des données de routage. En effet, les procédures d'attaques sont pour la plupart composées d'une séquence de comportements élémentaires d'attaques et de comportements élémentaires en conformité au regard du protocole. Afin d'affecter de façon précise certaines données du protocole, cette séquence doit être exécutée d'une manière particulière dans le temps et l'espace.

Il s'avère alors particulièrement difficile de combiner tous les comportements élémentaires possibles pour dériver un ensemble exhaustif de toutes les déclinaisons d'attaques. C'est la raison pour laquelle nous proposons une représentation des attaques en termes d'**actions non conformes**. Dans notre modèle, une **action non conforme** est définie comme étant une manipulation unitaire et non conforme au regard de la logique du protocole et portant sur les messages de contrôle.

3. Dans une troisième étape, les **relations de causalité** entre les différentes actions non conformes et l'étendue des perturbations sur les données de routage sont déduites, ceci en prenant en considération les interactions possibles entre un nœud et son voisinage.

Pour illustrer notre approche d'analyse systématique de la sécurité, nous prenons en exemple le protocole de routage OLSR. Notre choix s'est porté sur ce protocole pour deux raisons principales. Tout d'abord, OLSR est un protocole bien établi dans la communauté des réseaux ad hoc. Ensuite, alors que les protocoles de routage réactifs tels que DSR ou AODV ont largement été étudiés dans la littérature, les protocoles proactifs en général (dont OLSR) n'ont pas fait l'objet d'une analyse de sécurité approfondie. Or en raison de leur caractère proactif, les attaques contre ces protocoles et les mécanismes de sécurité pour y faire face sont de portée différente. Contrairement aux protocoles réactifs où une attaque sur une demande de chemin affecte seulement le chemin entre les nœuds source et destination impliqués, une attaque sur une information de routage dans OLSR affecte potentiellement de multiples chemins. En outre, selon le mécanisme de découverte de voisinage d'OLSR, des messages HELLO sont diffusés localement par les nœuds en fonction de leur connaissance du réseau. Comme ces messages n'ont pas vocation à être rediffusés, les options de sécurité sont réduites et l'acceptation des informations présentes dans ces messages repose sur une confiance entre les nœuds.

3.2.1 Analyse du protocole

Les protocoles de routage définissent les interactions entre les nœuds participants à travers des échanges de messages de contrôle. Ces messages véhiculent les informations requises par le protocole pour permettre la construction des chemins. Ils sont formés, selon les règles définies par le protocole, à partir de la connaissance qu'a un nœud sur l'état courant du réseau. Le traitement de ces messages par des nœuds récepteurs à des effets directs sur leur connaissance du réseau qu'ils maintiennent localement, puisque des mises à jour seront éventuellement réalisées sur la base des informations reçues.

Les messages de contrôle représentent la principale ressource utilisée par l'attaquant contre le protocole de routage. L'objectif de cette étape est donc de déterminer, selon le fonctionnement normal du protocole, dans quelle mesure les informations reçues dans un message de routage affectent les états locaux d'un nœud et comment elles sont propagées dans le réseau. Les effets des événements de routage sur la connaissance du réseau propre à un nœud, telle que les liaisons avec le voisinage et les tables de routage, sont étudiés. Dans une deuxième étape, l'étendue de la propagation des informations de routage pour les différents types de message introduits dans OLSR, à savoir les HELLO et TC, est examinée. Ces relations ont été obtenues à partir de la RFC 3626 [CJ03] du protocole.

3.2.1.1 Connaissance du réseau du point de vue d'un nœud OLSR

Dans OLSR, la connaissance du réseau pour un nœud est représentée à partir de plusieurs ensembles d'informations :

- LS_u (Link Set) est l'ensemble des liens avec le voisinage connus du nœud u. Selon la spécification d'OLSR, chaque nœud voisin inclus dans cet ensemble a un statut de lien qui lui est associé. Ce statut est soit symétrique soit asymétrique. Symétrique indique que le statut du lien avec un nœud voisin a été vérifié comme étant bidirectionnel, c'est-à-dire que les données peuvent être envoyées dans les deux directions. A l'opposé, asymétrique indique que le message HELLO d'un nœud voisin a été entendu, mais qu'il n'a pas été confirmé que ce dernier est capable de recevoir les messages. Il existe une relation évidente entre l'ensemble des liens et l'ensemble des voisins puisqu'un nœud fait partie du second ensemble si et seulement si il existe au moins un lien le concernant dont le statut est symétrique dans le premier. Pour faciliter la compréhension de notre analyse, nous considérons dans la suite de nos travaux que seuls les liens de voisinage asymétriques seront maintenus dans cet ensemble, les voisins symétriques étant connus grâce au NS_u.
- NS_u (Neighbor Set) est l'ensemble des voisins symétriques connus du nœud u.
- $2HNS_u$ (2-Hop Neighbor Set) est l'ensemble des voisins à deux sauts du nœud u.
- $MPRS_u$ est l'ensemble des nœuds sélectionnés comme MPR par le nœud u.
- $MPRSS_u$ (MPR Selector Set) est l'ensemble des voisins symétriques qui ont sélectionné le nœud u comme MPR.
- TS_u (Topology Set) est l'ensemble des informations sur la topologie du réseau du point de vue du nœud u.
- DS_u (Duplicate Set) est l'ensemble des informations sur les messages TC récemment reçus par le nœud u. Cet ensemble sert à éviter qu'un même message soit traité (et éventuellement retransmis) plusieurs fois.
- RT_u (Routing Table) est la table de routage du nœud u.

La figure 3.1 illustre les effets de l'arrivée d'informations de routage sur les différents ensembles locaux qui caractérisent la connaissance du réseau d'un nœud. Nous pouvons constater que certains ensembles sont influencés par un seul type de message de routage, alors que d'autres sont touchés par de multiples types de messages.

Par exemple, le calcul de l'ensemble $MPRS$ repose sur une heuristique gloutonne. Elle consiste à sélectionner dans une première étape (et donc en priorité) tous les nœuds de l'ensemble NS présentant une connectivité vers des nœuds isolés[1]. Dans une seconde étape, sont sélectionnés tous les nœuds de l'ensemble NS permettant de rattacher le plus de nœuds de l'ensemble $2HNS$ des voisins à deux sauts non encore couverts. Le résultat est donc régi par le message HELLO puisque les informations qu'il véhicule affectent le calcul des ensembles de voisins à un et deux sauts. A l'opposé, les informations maintenues dans l'ensemble topologique TS sont uniquement contrôlées par le message TC. Quant à la table de routage, elle représente en quelque le sorte le résultat final d'OLSR. Elle est un ensemble de données affecté par les deux types de messages de routage, car le calcul de ses entrées s'appuie sur les informations maintenues dans les trois ensembles NS, $2HNS$ et TS.

[1] Un nœud est dit isolé lorsqu'il n'existe qu'un seul chemin pour l'atteindre.

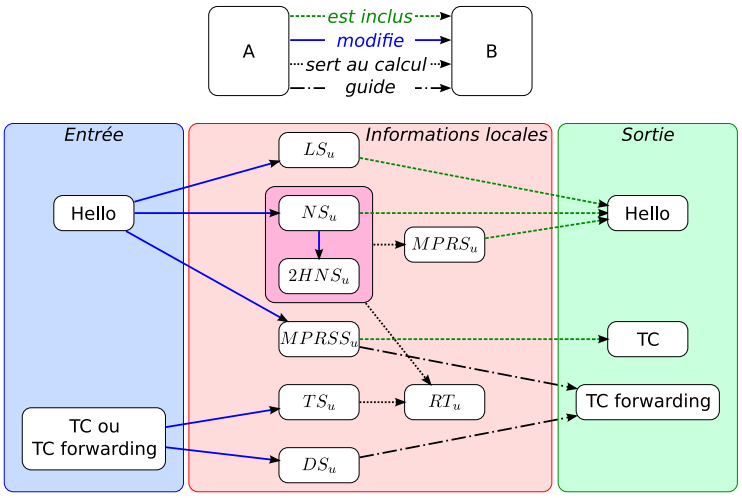


Fig. 3.1 – Analyse des relations entre les états locaux et les messages en entrée / sortie d'OLSR.

Toutefois, il convient de noter que l'arrivée d'un message de contrôle n'est pas la seule source d'informations influençant directement l'état de la connaissance du réseau d'un nœud. En effet, dans OLSR, chaque nœud est autonome, ce qui signifie qu'il est le seul responsable du maintien de sa connaissance du réseau. Ensuite, des messages de contrôle sont générés en fonction de cette connaissance, puis diffusés. Du point de vue de la sécurité, puisque des attaques comme la modification à un niveau local de certaines informations sont possibles, des informations erronées peuvent être diffusées. Or en l'absence de mécanisme de sécurité ou de procédure de validation des informations échangées, ces attaques conduisent des nœuds voisins à réaliser des mises à jour de leurs ensembles locaux sur une base d'informations incorrectes. De manière générale, il en résulte des perturbations dans le mécanisme de sélection des nœuds MPR et dans la construction des chemins.

Ceci signifie que ces ensembles de connaissance doivent être vus comme les points critiques d'OLSR, car ce sont eux qui, *in fine*, conduisent les résultats des procédures de calculs locaux (*i.e.* l'ensemble $MPRS$) et globaux (*i.e.* la table de routage). Il est donc nécessaire de comprendre comment une information, incluse dans un message de contrôle et construite à partir de la connaissance du réseau d'un nœud, se propage vers d'autres nœuds présents dans son voisinage. C'est l'objet de la section suivante.

3.2.1.2 Propagation des informations de routage

A partir de la figure 3.1, nous pouvons remarquer qu'à la fois les informations incluses dans messages de contrôle HELLO et TC sont propagées. Cependant, par nature, les messages HELLO ne sont pas censés être rediffusés. Un nœud maintient ses ensembles de voisins à un et deux sauts à partir informations annoncées dans les messages HELLO de ses voisins directs qui eux-mêmes incluent des informations obtenues à partir des messages HELLO de leurs propres voisins. La propagation directe des informations contenues dans un message HELLO est donc limitée à une distance d'au maximum deux sauts.

Seuls les messages TC sont propagés à travers tous les nœuds du réseau, et leur traitement a une incidence directe sur leur ensemble topologique TS. Toutefois, il est important de souligner que ce sont les informations annoncées dans le message HELLO d'un nœud qui influencent directement ou indirectement le contenu des messages TC générés par ses nœuds voisins à un et deux sauts. La raison est la suivante. Le message TC d'un nœud source contient son ensemble courant de sélecteurs MPR ($MPRSS$), c'est-à-dire tous les nœuds qui l'ont désigné comme MPR. Cet ensemble est mis à jour à partir des annonces, dans les messages HELLO de ses voisins, de leurs choix de MPR. Or comme indiqué figure 3.1, pour désigner ses nœuds voisins MPR, un nœud utilise ses ensembles de voisins à un et deux sauts. Etant donné que les messages HELLO sont à l'origine de ces ensembles, ils influencent par conséquent le mécanisme de sélection des MPR, qui au final détermine le contenu des messages TC.

Après la sélection MPR, un nœud doit annoncer son choix dans ses futurs messages HELLO en modifiant le statut des liens. Un deuxième moyen plus direct de modification du contenu d'un TC à travers le message HELLO est alors la rétention ou l'ajout de nœuds MPR.

Pour des nœuds au sein d'un même voisinage, nous venons d'examiner la relation existante entre les messages HELLO et TC. Cette analyse révèle l'importance des messages HELLO dans la propagation d'informations de routage erronées. En particulier, une information erronée dans message HELLO, si elle n'est pas évitée, a deux conséquences :

- Elle peut conduire à des erreurs dans le calcul des entrées des tables de routage, et ce pour tous les nœuds à une distance d'au maximum deux sauts de l'origine du message.
- Elle peut entrainer la génération d'un message TC au contenu incohérent par rapport à l'état réel du réseau. Dans ce deuxième cas, les informations erronées sont alors propagées à travers le réseau entier.

Cette connaissance nous donne un aperçu de l'emplacement optimal d'un mécanisme de sécurité pour empêcher ou détecter les attaques. En particulier, son principal objectif devrait être de veiller à l'intégrité et la véracité du contenu des messages HELLO afin d'éviter la propagation d'informations erronées.

3.2.2 Etudes des actions non conformes

Dans cette section, nous proposons une approche d'analyse des attaques possibles sur les messages de contrôle HELLO, car ils sont à la base du maintien de la cohésion des multiples ensembles utilisés dans les procédures de routage d'OLSR. Premièrement, nous identifions les attaques de base nommées actions non conformes qu'un attaquant peut réaliser sur le contenu de ces messages. Ensuite, nous examinons quels sont leurs effets néfastes directs sur le réseau.

3.2.2.1 Identification des actions

Les messages HELLO sont échangés périodiquement entre les nœuds. Ils impliquent la transmission de trois ensembles maintenus localement, à savoir le LS, NS et $MPRS$. Nous

considérons qu'une action non conforme se produit dès lors qu'une information incluse dans ce message ne coïncide plus avec l'état réel du réseau représenté par ces trois ensembles.

Nous identifions trois classes d'actions non conformes sur le contenu d'un message HELLO généré par un attaquant :

- **L'ajout unitaire d'une information non valide relative à l'identité du nœud d'origine.** Cette action est dénotée $Usurpate(u)$.
- **L'ajout unitaire d'une information non valide relative à un état de lien.** Cette action est dénotée $Add(u, link)$ où u représente l'identité du nœud visée par l'action et pour lequel un état de lien $link = (asym \vee sym \vee mpr)$ déclaré.
- **Le retrait unitaire d'une information valide relative à un état de lien.** Cette action est dénotée $Remove(u, link)$ où u représente l'identité du nœud visée par l'action et pour lequel un état de lien $link = (asym \vee sym \vee mpr)$ n'est pas déclaré.

3.2.2.2 Illustration de quelques procédures d'attaques

Notons qu'il est particulièrement complexe, voire impossible, d'énumérer de manière exhaustive toutes les déclinaisons d'attaques possibles et réalisables par un attaquant pour causer un effet précis sur l'intégrité du routage, car celles-ci sont fondées sur des séquences temporelles et spatiales d'actions à la fois conformes et non conformes. Dans nos travaux, nous considérons que les scénarios d'attaques peuvent être formés sur la base d'une ou de plusieurs des actions non conformes identifiées. Une grande proportion des attaques existantes dans la littérature se retrouve dans ce cas, même s'il est évident que ce modèle ne peut pas décrire l'ensemble des attaques possibles. Ainsi, bien que nous ne soyons pas en mesure de prouver que l'ensemble des actions non conformes identifié est exhaustif, nous considérons qu'il forme une base de référence substantielle pour conduire notre analyse de sécurité. Dans la suite de cette section, nous montrons par quelques exemples comment des attaques connues peuvent être exprimées sous la forme d'une composition d'actions non conformes successives.

Usurpation de liens. Une des attaques élémentaires sur un message HELLO est l'usurpation de liens. Elle consiste en la fabrication d'un message HELLO dans lequel sont annoncées des relations de voisinage qui ne coïncident pas avec la connectivité réelle du réseau. Dans notre modèle, nous pouvons exprimer cette attaque par une simple composition de plusieurs actions non conformes de type ajout et/ou retrait unitaire d'une information de la manière suivante :

$$\forall u, v \ : \ \begin{array}{l} \underset{u}{\cup} Add(u, link) \\ et/ou \\ \underset{v}{\cup} Remove(v, link) \end{array} ,$$

quelles que soient les valeurs pour $link$ (tel que $link = (asym \vee sym \vee mpr)$).

Usurpation d'identité. L'attaque par usurpation d'identité du nœud à l'origine d'un message est assez similaire. Elle est exprimée comme étant la composition d'une action de type usurpation d'identité et d'un ensemble d'actions de type ajout unitaire d'une information de la manière suivante :

$$\forall u, v \ : \ \begin{array}{l} Usurpate(u) \\ et \\ \bigcup\limits_{v} Add(v, link) \end{array} .$$

Rejeu. Un message HELLO est diffusé aux voisins à un saut, et n'est pas censé être relayé. Une autre attaque est le rejeu à l'identique d'un message HELLO précédemment entendu par l'attaquant. Cette attaque peut être exprimée comme la somme d'une action de type $Usurpate(u)$ et de plusieurs actions de type $Add(v, link)$. A première vue, nous pourrions être amenés à penser que cette attaque est de nature identique à l'attaque par usurpation d'identité, car elle est composée des mêmes actions. Cependant, la principale différence peut être définie en terme de granularité de contrôle. Contrairement à l'attaque par usurpation d'identité, la somme des actions est dans ce cas indivisible : toutes les informations contenues dans le message original sont copiées à l'identique, et aucune autre action supplémentaire n'est possible.

Election seul MPR. Une attaque plus complexe est celle où l'attaquant vise à augmenter ses chances d'élection MPR, ou de manière encore plus subtile, à se faire élire seul MPR par un nœud dans son voisinage. L'élection en tant que seul MPR est souvent considéré comme le but ultime de l'attaquant, car étant le seul à diffuser des messages TC annonçant une connectivité avec la victime, cette position lui permettra de contrôler tous les flux de messages à destination de sa victime. Dans une étude menée par Kannhavong *et al.* [KNK+06], une déclinaison de cette attaque est décrite. Elle consiste pour un attaquant à fabriquer un message HELLO dans lequel il annonce être le voisin symétrique de tous les nœuds voisins à deux sauts sa victime, plus un nœud fictif. Selon la procédure de sélection des nœuds MPR, à la réception de ce message HELLO, la victime choisira l'attaquant en tant que seul MPR. Toutefois, l'attaquant doit procéder en plusieurs étapes pour mener à terme cette attaque :

1. Dans un premier temps, il doit recevoir un message HELLO de sa victime afin d'identifier tous ses voisins symétriques.
2. Il doit ensuite analyser les messages TC diffusés par les voisins à un saut de sa victime afin de déterminer quels sont tous ses voisins à deux sauts.
3. Il utilise cette connaissance pour forger son message HELLO, puis il le diffuse sur le réseau.

En termes d'actions non conformes sur le message HELLO, cette attaque n'est finalement qu'un cas particulier de l'attaque par usurpation de liens, et peut être exprimée comme suit :

$$\forall fictif \ : \ \begin{array}{l} Add(fictif, sym) \\ et \\ \bigcup\limits_{v \in 2HNS_{victime}} Add(v, sym) \end{array} .$$

Variante à l'élection seul MPR. Une variante à cette attaque, qui n'est à notre connaissance pas décrite dans la littérature, consiste pour un attaquant invalider tous les liens symétriques que sa victime entretient avec ses voisins à un saut. En effet, si l'attaquant est perçu par sa victime comme étant le seul voisin, alors il sera d'une part sélectionné comme seul MPR, et d'autre part, il sera le seul intermédiaire pour les flux de messages de et vers sa victime. L'attaque se déroule suivant la séquence suivante :

1. A la réception d'un message HELLO de sa victime, l'attaquant identifie tous ses voisins symétriques.
2. Pour chaque voisin symétrique, l'attaquant usurpe son identité afin de fabriquer un message HELLO dans lequel il annonce le lien associé à sa victime comme inexistant soit :

$$\forall v \in NS_{victime} \ : \ \begin{array}{l} Usurpate(v) \\ et \\ Remove(victime, sym) \end{array} .$$

3. Ensuite, il diffuse ce message sur le réseau.

Autres exemples. D'autres attaques permettent à un attaquant d'être sélectionné comme MPR par une victime. Elles consistent pour l'attaquant à fabriquer un message HELLO dans lequel il annonce une ou plusieurs des informations suivantes :

- Un lien symétrique avec un nœud fictif.
- Un lien symétrique avec un nœud légitime dans le réseau, mais n'appartenant pas à l'ensemble des voisins à deux sauts de la victime.
- Des liens symétriques avec les voisins à deux sauts de la victime afin d'obtenir un degré de connectivité supérieure aux autres nœuds voisins à saut de la victime.

Comme précédemment, nous pouvons remarquer que toutes ces attaques ne sont que des cas particuliers de l'attaque par usurpation de liens, et peuvent donc être représentées dans notre modèle.

3.2.3 Etude des effets unitaires directs

3.2.3.1 Identification des cibles

Nous définissons une cible comme étant l'identité d'un nœud pouvant être manipulée par un attaquant pour servir ses actions non conformes. Afin de mesurer les opportunités d'un attaquant, nous devons connaître quelles sont les cibles admissibles en entrée pour chacune des actions non conformes. Ces cibles dépendent principalement de la connaissance du réseau réutilisable par un attaquant.

Etant donné qu'OLSR ne définit aucun mécanisme d'authentification, la cible des actions non conformes peut être l'identité de n'importe quel nœud, qu'elle soit réelle ou fictive. Une information relative à l'identité d'un nœud réel dans le réseau peut être instantanément extraite à partir des messages de contrôle capturés, puis être réutilisée telle qu'elle par un attaquant. Quant aux identités fictives, elles ne font référence à aucun nœud réel dans le réseau : elles sont explicitement construites par un attaquant pour servir ses actions.

3.2.3.2 Victimes et environnement de l'attaquant

Dans la mesure où aucune procédure de vérification n'est précisée dans les spécifications d'OLSR pour assurer la véracité d'une information sur l'état d'un lien annoncée dans un message HELLO, un nœud accepte telle quelle toute information qu'il reçoit et modifie directement sa connaissance du réseau en conséquence. Une action non conforme sur le contenu d'un message de contrôle entraîne alors une modification unitaire et immédiate d'ajout ou de suppression d'un état de lien sur la connaissance d'un nœud à portée de transmission de l'attaquant.

Toutefois, la nature de l'information erronée et la connectivité réelle entre l'attaquant et ses voisins joue un rôle majeur dans la réussite de l'action non conforme d'une part, et dans la détermination de sa portée locale d'autre part. Par exemple, une action de type $Add(u, sym)$ affecte tous les voisins symétriques de l'attaquant, car ceux-ci enregistreront u comme étant un voisin à deux sauts directement accessible par l'attaquant. En revanche, une action de type $Add(u, asym)$ affecte exclusivement le nœud u, si et seulement si u est capable de recevoir le message HELLO de l'attaquant. C'est pourquoi, pour mesurer l'étendue des actions non conformes de l'attaquant sur son voisinage, nous ajoutons deux nouvelles dimensions qui sont (1) la victime et (2) l'environnement de communication de l'attaquant. Nous définissons la victime comme étant un nœud ou un ensemble de nœuds acceptant une information erronée suite à une action non conforme. Soit A l'identité de l'attaquant, nous définissons p_A comme étant l'ensemble des nœuds à portée de transmission de l'attaquant. Tout nœud u appartenant à p_A est capable de recevoir directement les messages émis par l'attaquant, et l'attaquant est capable de recevoir ses messages.

3.2.3.3 Relations de causalité

A partir de la connectivité de l'attaquant couplée à sa connaissance réelle du réseau et aux cibles potentielles identifiées dans la section précédente, nous explorons pour chacune des actions non conformes les différentes façons dont les nœuds dans le voisinage de l'attaquant réagiront à une information erronée. Les notations utilisées pour caractériser la modification d'une information sur la connaissance du réseau d'un nœud sont les suivantes :

- C_u : $Create(v, link)$: signifie une création d'un état de lien asymétrique, symétrique ou mpr dénoté $link$ entre u et v sur la connaissance du nœud u.
- C_u : $Delete(v, link)$: signifie une suppression d'un état de lien asymétrique, symétrique ou mpr dénoté $link$ entre u et v sur la connaissance du nœud u.

– C_u : $Simulate(v)$: signifie une simulation d'un état de lien entre u et v sur la connaissance du nœud u.

Le tableau 3.1 décrit les effets unitaires directs néfastes pour toutes les instances d'actions non conformes possibles. Dans cette analyse, nous partons de l'hypothèse d'une configuration statique du réseau.

Effet unitaire direct	Action non conforme	Cible de l'action	Victime
$C_u : Create(A, sym)$	$Add(u, asym)$	$\forall u \vert (u \notin LS_A) \wedge (u \in p_A)$	u
	$Add(u, sym)$	$\forall u \vert (u \notin NS_A) \wedge (u \in p_A)$	u
$C_u : Create(A, mpr)$	$Add(u, mpr)$	$\forall u \vert (u \notin MPRS_A) \wedge (u \in p_A)$	u
$C_u : Create(v, sym/A)$	$Add(v, sym)$	$\forall v \vert v \notin NS_A$	$\forall u \vert A \in NS_u$
$C_u : Delete(A, sym)$	$Remove(u, asym)$	$\forall u \in LS_A$	u
	$Remove(u, sym)$	$\forall u \in NS_A$	u
$C_u : Delete(A, mpr)$	$Remove(u, mpr)$	$\forall u \in MPRS_A$	u
$C_u : Delete(v, sym/A)$	$Remove(v, sym)$	$\forall v$	$\forall u \vert A \in NS_u$
$C_u : Create(v, asym)$	$Usurpate(v)$	$\forall v$	$\forall u \vert u \in p_A$
$C_u : Simulate(v)$	$Usurpate(v)$ *et* $Add(u, link)$ *ou* $Remove(u, link)$	$\forall v$	$u \vert u \in p_A$

Tab. 3.1 – Actions non conformes et effets unitaires directs associés.

Actions Add(u,asym) et Remove(u,asym) Dans OLSR, un nœud récepteur d'un message HELLO réagit à une action de type $Add(u, sym)$, et modifie sa connaissance du réseau en créant un lien symétrique avec l'origine du message si et seulement si il est directement impliqué dans la relation annoncée (voir la première ligne du tableau 3.1). Il se dégage deux éléments d'informations de cette caractéristique. Tout d'abord, la portée directe de cette information erronée est limitée à la connaissance d'une seule victime, à savoir le nœud u. Ainsi, pour qu'une action de type $Add(u, asym)$ aie un effet, la victime doit être capable de recevoir le message émis par l'attaquant. Ensuite, l'ensemble des cibles possibles pour l'action $Add(u, asym)$ est formé de tous les nœuds à portée de communication de l'attaquant et non connus par ce dernier comme présentant un lien asymétrique.

Une action de type $Remove(u, asym)$ n'a de sens que si l'attaquant a reçu au préalable un message HELLO de la cible u, et la connait comme présentant un lien asymétrique. Ainsi, toutes les cibles possibles pour l'action $Remove(u, asym)$ sont les nœuds à portée de communication de l'attaquant, et connus par ce dernier comme présentant un lien asymétrique. L'effet direct de l'action est la suppression d'un lien symétrique entre u et A de la connaissance de u (voir la cinquième ligne du tableau 3.1).

Actions Add(v, sym) et Remove(v, sym) Le traitement d'une information de lien symétrique entre v et A dans un message HELLO dépend du type de la relation existante entre l'origine A et le nœud récepteur u.

Premièrement, pour n'importe quelle identité v de nœud cible annoncée, elle affecte tous les récepteurs u qui connaissent l'attaquant comme étant un voisin symétrique et qui ne sont pas directement impliqués dans le lien symétrique annoncé (soit $v \neq u$). Pour une action d'ajout (resp. de suppression), tout nœud u connaissant A comme voisin symétrique (soit $A \in NS_u$) traite cette information et en déduit que A est un relais direct vers v (resp. n'est pas un relais vers v). Un état de lien symétrique entre u et v en passant par A est créé (resp. supprimé) de la connaissance de la victime u, soit $C_u : Create(v, sym/A)$ (resp. $C_u : Delete(v, sym/A)$) (voir la quatrième et la sixième ligne du tableau 3.1).

Dès lors que la cible v est telle que A ne connaît pas (resp. connaît) v comme étant un voisin symétrique et que le nœud u (directement impliqué dans la relation de lien symétrique annoncée, soit $v = u$) reçoit le message de A, alors une autre victime est la cible v elle-même. Cette dernière est amenée à créer (resp. supprimer) un état de lien symétrique avec A dans sa connaissance (voir la deuxième ligne du tableau 3.1).

Actions Add(u,mpr) et Remove(u, mpr) Comme pour l'annonce d'un lien asymétrique, un nœud récepteur d'un message HELLO traite une information de lien mpr, et modifie sa connaissance du réseau en créant ou supprimant un lien mpr avec l'origine du message si et seulement si il est directement impliqué dans la relation annoncée (voir la troisième et septième ligne du tableau 3.1). Ainsi, la portée directe de cette information erronée est limitée à la connaissance d'une seule victime, à savoir le nœud récepteur u à portée de transmission de l'attaquant.

L'ensemble des cibles possibles pour l'action $Add(u, mpr)$ (resp. $Remove(u, mpr)$) est formé de tous les nœuds non connus (resp. connus) par ce dernier comme présentant (resp. ne présentant pas) une relation mpr.

Remarques Suivant les résultats présentés dans le tableau 3.1, nous observons deux types d'effets. D'une part, pour toute instance d'action non conforme $Add(u, link)$ ou $Remove(u, link)$, l'attaquant reste l'origine du message HELLO. Par conséquent, il parvient, selon un fin niveau de granularité, à altérer la connaissance d'un ou plusieurs de ses voisins à portée de communication au regard de l'état d'un lien qu'ils entretiennent mutuellement. Précisons par ailleurs que les effets ne sont pas réciproques dans le sens où la connaissance de l'attaquant reste intacte au regard de la victime. Il peut par exemple continuer à exploiter la victime pour laquelle il a retiré un état de lien symétrique comme relais direct vers d'autres nœuds du réseau.

D'autre part, lorsqu'un attaquant combine toute instance d'action non conforme d'ajout d'une identité à toute instance d'action non conforme d'ajout ou de retrait d'un état de lien ($Usurpate(v)$ *et* $Add(u, link)$ ou $Usurpate(v)$ *et* $Remove(u, link)$), les effets sont similaires à ceux des actions non conformes $Add(u, link)$ et $Remove(u, link)$. La différence est qu'au lieu d'être associés à l'identité de l'attaquant, ils portent sur un état de lien impliquant l'identité du nœud usurpée sur la connaissance de la victime. Par exemple, la combinaison $Usurpate(v)$ *et* $Add(u, link)$ conduit à la création d'un état de lien (dépendant du type d'état de lien ajouté dans l'action) entre u et v sur la connaissance de u. Nous notons l'effet de cette combinaison d'actions non conformes comme étant une simulation de l'état

d'un lien entre un nœud u et un nœud v auprès de u (voir la dernière ligne du tableau 3.1).

3.2.4 Discussion

Cette première étape d'analyse du protocole nous a permis de mettre en évidence les points critiques du protocole que sont d'une part les ensembles de connaissance du réseau maintenus localement par les nœuds, et d'autre part les messages HELLO. Ensuite, nous avons identifié trois classes d'actions non conformes sur les messages HELLO. Pour toutes les instances possibles de ces actions, nous avons identifié les victimes potentielles et les effets unitaires directs sur leur connaissance de la topologie du réseau. Cette analyse ne cherche pas à évaluer quels sont les effets complexes qui peuvent être atteints par un attaquant suite à une combinaison de plusieurs actions non conformes. Néanmoins, les résultats obtenus font apparaître des situations de base à partir de laquelle la sécurité du protocole peut être compromise. A défaut de fournir une vue exhaustive, cette compréhension approfondie des effets peut se révéler utile pour appréhender les possibilités et les limites d'attaques sur le protocole. Elle permet notamment de savoir dans quelle mesure (en nombre de cibles et victimes) la connaissance de la topologie du réseau construite localement par les nœuds, et ce, à partir des informations contenues dans les messages de contrôle reçus, peut être altérée par un attaquant.

3.3 Analyse de versions sécurisées d'OLSR

Le déploiement de mécanismes de sécurité basés par exemple sur l'authentification des messages de contrôle conduit à une réduction naturelle de l'ensemble des actions non conformes exploitables par un attaquant. Intuitivement, moins un protocole est exposé à des actions non conformes, moins grandes sont les opportunités d'attaques permettant à un attaquant de causer des effets nuisibles dans son fonctionnement. En effet, il semble raisonnable de considérer qu'une forte réduction dans les instances d'actions non conformes réalisables par un attaquant aboutira en une complexification considérable de la mise en place d'attaques.

Nous proposons donc de raisonner sur l'ensemble des actions non conformes, de leurs cibles et de leurs effets afin de mettre en relief les apports et les limitations de deux versions sécurisées d'OLSR : une première basée sur la cryptographie traditionnelle, et une seconde dans laquelle l'exécution du protocole se fait à l'intérieur d'un matériel résistant à la manipulation.

Plus spécifiquement, pour une version sécurisée du protocole et selon un modèle de l'attaquant (décrit plus en détail dans la section suivante), nous étudions :

- Les choix de cibles restants pour les actions non conformes identifiées sur le protocole OLSR de référence.
- Les nouvelles classes d'actions non conformes qui permettent d'atteindre les effets unitaires directs identifiés en section 3.2.3.3 .

– Pour ces nouvelles classes d'actions, quelles sont leurs cibles, quelles sont leurs victimes, et dans quelle mesure en terme de degré de granularité de résultat les effets unitaires sont-ils atteints.

3.3.1 Modèle de l'attaquant

Dans les approches d'analyse de sécurité existantes et basées sur l'inspection humaine, les auteurs font souvent des hypothèses sur le contexte d'utilisation du protocole ou sur les capacités de l'attaquant. En plaçant de telles restrictions, les résultats obtenus ne sont valides que dans les seuls cas prévus par les auteurs. Or les protocoles ont souvent des usages inattendus et sont exposés à des classes d'attaquants plus larges que prévu. Il n'est donc pas rare de trouver des situations dans lesquelles ces extensions peuvent être mises en défaut. Un second inconvénient est qu'il devient relativement difficile de comparer, puis de choisir une solution adaptée à un environnement de déploiement, puisque ces dernières sont évaluées dans des contextes particuliers.

Notre objectif est de déterminer si, pour une solution de sécurité donnée, la connaissance du réseau maintenue par les nœuds à partir des informations contenues dans les messages HELLO reçus est cohérente avec l'état réel du réseau. Partir sur la base d'un modèle de l'attaquant dans lequel seul le plus puissant est représenté présente l'inconvénient de ne pas permettre l'identification des capacités minimales requises pour corrompre le fonctionnement normal du protocole. Or à notre sens, cette connaissance est utile, car elle permet d'apprécier les environnements dans lesquels une version renforcée d'un protocole est susceptible de résister aux attaques. Afin de fournir une analyse de sécurité plus complète, nous étudions les versions renforcées d'OLSR en prenant en considération plusieurs types d'attaquants, classifiés en fonction des capacités dont ils disposent pour les mettre en défaut.

3.3.1.1 Capacités

Afin d'évaluer les efforts requis en termes de moyens physiques et logiques par un attaquant pour atteindre un effet précis, nous utilisons le modèle de l'attaquant décrit par Hu *et al.* [HPJ02]. Dans ce modèle, les auteurs formalisent les capacités de l'attaquant sous la forme $Active - n - m$, où n est le nombre de nœuds internes compromis par l'attaquant, et m est le nombre total de nœuds physiques possédés par l'attaquant. Ici, les nœuds internes compromis sont des nœuds autorisés dans le réseau, mais pour lesquels les clés cryptographiques utilisées dans les échanges de messages de contrôle ont été révélées à l'attaquant. En outre, un attaquant peut distribuer les clés compromises aux $m-1$ nœuds physiques qu'il possède. En terme de communication, l'attaquant affiche des capacités identiques à celles des autres nœuds du réseau, à savoir qu'il ne peut recevoir que les messages transmis par son voisinage direct et que seul son voisinage direct peut recevoir ses messages.

A partir de ce modèle, nous étudierons les possibilités d'actions non conformes pour les attaquants décrits dans le tableau 3.2.

Type de l'attaquant	Possessions
$A_1 : Active - 0 - 1$	– Un nœud physique.
$A_2 : Active - 1 - 1$	– Un nœud physique. – Une clé cryptographique.
$A_3 : Active - n - 1$	– Un nœud physique. – Plusieurs clés cryptographiques.
$A_4 : Active - 0 - m$	– Plusieurs nœuds physiques.
$A_5 : Active - 1 - m$	– Plusieurs nœuds physiques. – Une clé cryptographique.
$A_6 : Active - n - m$	– Plusieurs nœuds physiques. – Plusieurs clés cryptographiques.

(n et m sont des entiers supérieurs à un.)

Tab. 3.2 – Capacités des attaquants étudiés.

3.3.1.2 Connaissance utile du réseau

Dans l'utilisation de ce modèle de l'attaquant par Hu *et al.*, la connaissance du réseau utile à l'attaquant et prise en considération pour l'analyse des attaques est réduite à la possession ou non d'informations sur la configuration initiale du réseau. En particulier, ces informations de configuration comprennent les identités et les clés cryptographiques associées qui sont utilisées dans les phases d'échanges de messages de contrôle. Elles présentent la caractéristique d'être sous une forme élémentaire et peuvent directement être utilisées par l'attaquant pour servir ses actions.

Cette représentation de la connaissance du réseau de l'attaquant est limitée. Elle est insuffisante pour déterminer quelles sont toutes les opportunités d'actions non conformes. La raison principale est que la connaissance du réseau utile à l'attaquant n'est pas statique, mais augmente au fur et à mesure des échanges de messages entre les nœuds. Dans la mesure où les messages de contrôle sont échangés en clair lors des opérations normales du protocole, les informations qu'ils contiennent peuvent éventuellement être accumulées, puis exploitées à postériori par un attaquant, par exemple en combinaison avec les clés qu'il détient. Nous proposons d'ajouter une nouvelle dimension au modèle de l'attaquant proposé par Hu *et al.* : la **connaissance du réseau utile à l'attaquant** pour mener ses actions non conformes, et formée sur la base des messages qu'il capture.

Concernant cette connaissance, le premier point à souligner concerne la granularité des

informations utiles extraites à partir des messages de contrôle capturés. Elle est variable et dépend des mécanismes de sécurité déployés ainsi que de la connaissance initiale de l'attaquant. Au plus fin niveau de granularité, et comme nous avons pu l'observer lors de l'analyse du protocole OLSR de référence, il s'agit de l'identité d'un nœud. Au plus haut niveau de granularité, il s'agit des messages complets interceptés par l'attaquant. Afin d'éviter de représenter plusieurs fois la même information, nous identifions pour chaque extension de sécurité quelles sont les informations utiles apprises par un attaquant, puis nous les représentons sous leur forme élémentaire.

Le deuxième point à souligner est qu'à la fois la portée de communication de l'attaquant et sa capacité à partager des informations avec d'autres attaquants définissent sa capacité à apprendre de l'information utile. Par exemple, à partir d'échanges de messages HELLO, un nœud u peut accumuler des informations sur les états des liens avec ses nœuds voisins dans ses ensembles LS_u, NS_u, $2HNS_u$. De plus, lorsqu'un attaquant gouverne plusieurs nœuds physiques répartis à différents endroits sur le réseau, ses informations peuvent être échangées. Il est à noter que pour réaliser ce partage, l'existence d'une liaison de communication directe (éventuellement filaire) entre les nœuds physiques de l'attaquant n'est pas nécessaire. Il est permis grâce à l'utilisation des chemins offerts par le réseau lui-même. Cette distinction est importante, car elle montre que des attaques exploitant exclusivement les ressources fournies par le réseau sont possibles.

3.3.2 Analyse d'OLSR couplé avec Advanced Signature

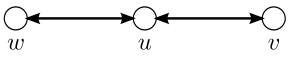
Comme nous l'avons vu au chapitre 1 (section 1.4.1.2), plusieurs extensions de sécurité d'OLSR basées sur la cryptographie, et plus particulièrement sur des mécanismes de signature et d'authentification des messages de contrôle, ont été proposées [ACJ+03, HTR+04, RACM04, CXL06]. Ces solutions garantissent toutes que des nœuds ne possédant pas les autorisations nécessaires ne puissent pas participer aux opérations de routage. En revanche, seules les extensions de sécurité de Raffo *et al.* [RACM04] et de Chen *et al.* [CXL06] tentent d'empêcher l'inoculation d'informations d'état de liens erronées par des nœuds attaquants internes. Pour ce faire, ces deux extensions reposent sur l'ajout de preuves cryptographiques sur les états des liens annoncés dans les messages de contrôle. Toutes deux sont relativement similaires. Nous étudierons dans la suite de cette section l'extension de Raffo *et al.* nommée AdvSig.

3.3.2.1 Description

Le principe d'AdvSig est que tout nœud enregistre les informations concernant ses liens envoyées par ses voisins, et les réutilise comme preuves dans ses messages de contrôle successifs. Ainsi, lorsqu'un nœud u veut déclarer un lien de type $link$ (avec $link = asym \vee sym \vee mpr$) avec un nœud v dans un message HELLO, il doit fournir la preuve de validité du lien. Cette preuve, obtenue à partir des informations contenues dans un message HELLO précédemment émis par v, est formée de l'identité de u, de son état de lien avec v de type $link_p$, et d'un horodatage. Ensuite, la preuve de validité requise dépend de l'état de lien annoncé. Elle est définie comme suit :

- Si u annonce un lien asymétrique avec v, soit $(v, link = asym)$ alors aucune preuve n'est requise puisque v n'entend pas u.
- Si u annonce un lien symétrique avec v, soit $(v, link = sym)$, alors une preuve $Preuve(u, link_p)_v$ avec $link_p = (asym \vee sym)$ est requise.
- Si u annonce un lien mpr avec v, soit $(v, link = mpr)$, alors une preuve $Preuve(u, link_p)_v$ avec $link_p = sym$ est requise.

La figure 3.2 illustre la procédure pour l'établissement d'un lien symétrique entre les nœuds u et v selon AdvSig. La notation $u \rightarrow v : \{M, P, T_u(t_x)\}_u$ signifie que u diffuse le message M avec la preuve P, l'horodatage de u au temps t_x, le tout signé par la clé privée de u. Au début de l'échange, u ne possède aucune information sur v et vice versa.



1. $u \rightarrow *(v) : \{\emptyset, \emptyset, T_u(t_0)\}_u$
2. $v \rightarrow *(u) : \{\{(u, asym), T_v(t_1)\}_v, \emptyset, T_v(t_1)\}_v$
3. $u \rightarrow *(v, w) : \{\{(v, sym), T_u(t_2)\}_u, \{(u, asym), T_v(t_1)\}_v, T_u(t_2)\}_u$

Fig. 3.2 – Etablissement d'un lien symétrique entre les nœuds u et v selon AdvSig.

A t_0 (étape 1 de la figure 3.2), le nœud u diffuse un message HELLO. v entend ce message et annonce dans son message HELLO qu'il a entendu u avec un état de lien asymétrique (étape 2 de la figure 3.2). A la réception de ce message, u est en possession de la signature de v. Cette signature prouve l'existence du lien entre u et v au temps $T_v(t_1)$. Dans le troisième message HELLO (étape 3 de la 3.2), u déclare un lien symétrique avec v grâce à la preuve $\{(u, asym), T_v(t_1)\}_v$ signée par v. A la réception d'un message contenant une telle preuve, les récepteurs (w) ont la garantie que u annonce des états de liens en cohérence avec la connectivité réelle du réseau.

Notons que dans ce schéma, l'estampille temporelle sert à garantir la fraîcheur d'un message et empêche ainsi les attaques par rejeu d'informations anciennes.

3.3.2.2 Connaissance utile du réseau

Le tableau 3.3 résume la connaissance du réseau initiale et apprise par l'attaquant selon ses capacités physiques et logiques.

Premièrement, indépendamment du nombre de nœuds physiques possédés par l'attaquant, s'il ne connaît aucune clé cryptographique valide (attaquant de type A_1 ou A_4), alors sa connaissance du réseau est formée des seuls messages complets qu'il capture.

Ensuite, lorsque l'attaquant gouverne un nœud physique et qu'il est en possession d'au moins une clé cryptographique (attaquant de type A_2 ou A_3), il est pleinement autorisé à participer aux échanges de messages du protocole. En particulier, les messages qu'il génère sous le couvert de son identité influencent les états de ses voisins. Les informations sous leur forme minimale qu'il peut utiliser dans ses actions non conformes sont les suivantes :

- L'ensemble des identités et leurs clés cryptographiques associées.

Attaquant	Connaissance initiale des identités (dénotée Y)	Connaissances sous leur forme élémentaire (dénotée C_{A_x})	Connaissance des messages
A_1	$Y = \emptyset$	C_{A_1} : Aucune.	Tous les messages HELLO reçus de ses voisins directs.
A_2	$Y = \{y\}$	C_{A_2} : – Nœuds à portée de transmission. – Nœuds asymétriques. – Ensemble des nœuds voisins connus comme étant symétriques. – Ensemble des nœuds voisins connus comme étant MPR.	Tous les messages HELLO reçus de ses voisins directs.
A_3	$Y = \{y_1, ..., y_n\}$	C_{A_3} : idem C_{A_2}.	Tous les messages HELLO reçus de ses voisins directs.
A_4	$Y = \emptyset$	C_{A_4} : Aucune.	Tous les messages HELLO capturés par ses nœuds physiques.
A_5	$Y = \{y\}$	C_{A_5} : – C_{A_2}. – Preuves de liens symétriques entre y et chacun des nœuds voisins symétriques de ses nœuds physiques.	Tous les messages HELLO capturés par ses nœuds physiques.
A_6	$Y = \{y_1, ..., y_n\}$	C_{A_6} : – C_{A_2}. – Pour tout $y \in Y$, preuves de liens symétriques entre y et chacun des nœuds voisins symétriques de ses nœuds physiques.	Tous les messages HELLO capturés par ses nœuds physiques.

Tab. 3.3 – AdvSig : connaissance utile du réseau selon les différents types d'attaquant.

– L'ensemble des nœuds dans son champ de transmission, mais pour lesquels il ne reçoit pas les messages ; cette information est extraite à partir des relations de voisinage annoncées dans les messages HELLO de ses voisins directs.
– L'ensemble des nœuds qu'il connait comme étant asymétriques, symétriques ou mpr et qui sont associés à ses identités ; ces ensembles représentent la connaissance réelle de l'état du réseau.

Finalement, lorsque l'attaquant gouverne plusieurs nœuds physiques et qu'il est en possession d'au moins une clé cryptographique (attaquant de type A_5 ou A_6), nous considérons alors que sa connaissance du réseau n'est plus strictement locale, mais qu'elle est étendue à la somme de la connaissance de chacun de ses nœuds physiques. La figure 3.3

illustre une configuration du réseau pour un attaquant de type $Active - 2 - 3$. L'attaquant est dénoté Att. Les nœuds physiques sous son contrôle sont dénotés Att_1, Att_2 et Att_3. Il a connaissance des clés cryptographiques associées à deux identités de nœud du réseau : att_1 et att_2. Ces clés sont partagées entre tous ses nœuds physiques et sont donc potentiellement utilisées par ces derniers dans la phase d'échange de messages HELLO. A l'opposé, les nœuds honnêtes utilisent une seule identité qui n'est pas compromise. Ainsi, la notation $X : \{y_1, ..., y_n\}$ signifie que le nœud physique X peut utiliser l'ensemble des identités $Y = \{y_1, ..., y_n\}$ lors des échanges de messages HELLO. A partir cette configuration, l'attaquant Att peut établir des états de liens et accumuler leurs preuves associées entre :

- l'identité att_1 et l'ensemble des voisins directs de chacun de ses nœuds physiques,
- puis entre l'identité att_2 et l'ensemble des voisins directs de chacun de ses nœuds physiques.

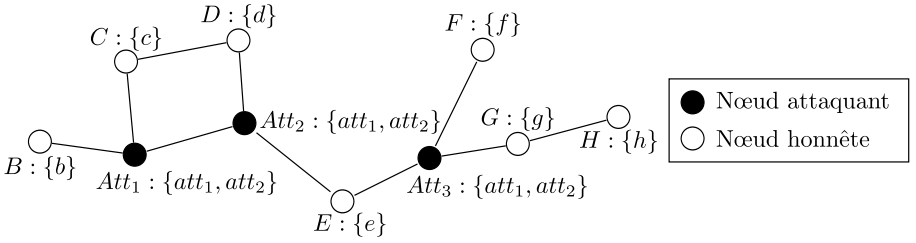


Fig. 3.3 – Illustration d'une configuration réseau de l'attaquant.

De façon générale, en nous plaçant du point de vue d'un seul nœud physique x, l'ensemble des preuves de liens accumulées entre l'identité y et chacun des voisins directs v peut être formulé comme suit :

$$Preuve_y(x) = \bigcup_{v \in p_x} Preuve(y, link)_v.$$

La connaissance globale des preuves pour tout nœud physique d'un attaquant de type $Active - n - m$ est alors la suivante :

$$\forall y \in Y, Preuve_y = \bigcup_{x \in X} Preuve_y(x),$$

avec Y l'ensemble des identités possédées par l'attaquant (et $card(Y) = n$) et X l'ensemble de ses nœuds physiques (et $card(X) = m$).

La preuve d'un état de lien représente une information élémentaire qui peut être réutilisée localement par un attaquant. Les preuves accumulées sont des preuves d'états de liens asymétriques, symétriques, et MPR. Rappelons que dans notre analyse du protocole OLSR de référence, nous avons observé qu'une information de lien asymétrique ou mpr est traitée par un nœud récepteur si et seulement si il est directement impliqué dans la relation annoncée. Par conséquent, au regard des effets potentiels sur le voisinage direct d'un seul nœud physique, les seules informations utiles sont les preuves de liens symétriques. En considérant l'ensemble des preuves accumulées par l'attaquant Att pour l'identité att_1 suivant :

$$Preuve_{att_1} = \{Preuve(att_1, sym)_b, Preuve(att_1, sym)_c, Preuve(att_1, asym)_d, Preuve(att_1, mpr)_e, Preuve(att_1, asym)_f, Preuve(att_1, sym)_g\}.$$

Du point de vue du nœud physique Att_1, les preuves de liens $Preuve(att_1, asym)_d$, $Preuve(att_1, mpr)_e$, et $Preuve(att_1, asym)_f$ lui seront inutiles dans ses actions puisque les nœuds d, e, et f ne recevront pas directement ses messages HELLO.

Nous venons d'identifier les informations accumulées par l'attaquant et lui permettant de mener ses actions non conformes. Nous utilisons ces informations pour déterminer dans quelle mesure (en nombre de cibles et en granularité des actions non conformes) les effets unitaires que nous avons précédemment identifiés sur la version non sécurisée d'OLSR peuvent être atteints.

3.3.2.3 Ajout ou retrait d'un état de lien

Add(u,asym) Dans le schéma AdvSig, la déclaration d'un état de lien asymétrique ne requiert pas de preuve pour être acceptée par un nœud. Par conséquent, un attaquant interne au réseau de type A_2, A_3, A_5 ou A_6 peut utiliser n'importe quelle identité de sa connaissance de nœuds à portée de transmission, mais non connue comme étant asymétrique en tant que cible d'une action d'ajout d'un lien asymétrique.

Add(v, sym) Pour annoncer un état de lien symétrique avec une cible v dans un message HELLO, l'attaquant est dans l'obligation de fournir une preuve signée par v de l'existence de ce lien. Dans AdvSig, cette preuve est indispensable pour que l'information soit acceptée par ses voisins. Ainsi, les actions d'ajout unitaire d'un état de lien symétrique sont éliminées pour un attaquant indépendant et sont fortement réduites pour un attaquant possédant plusieurs clés cryptographiques (voir la troisième ligne du tableau 3.4). Pour fournir cette preuve, nous pouvons distinguer deux cas.

Tout d'abord, un attaquant de type A_3 ou A_6 connait plusieurs identités de nœuds et leurs clés cryptographiques associées. Cette connaissance initiale lui permet de fabriquer localement des preuves valides (au sens d'AdvSig) d'état de liens symétrique entre chacune des identités de nœud qu'ils possèdent. La cible d'une action d'ajout d'un état de lien symétrique est alors n'importe laquelle de ces identités.

Le second cas est celui où l'attaquant est de type A_5 ou A_6. Nous avons étudié dans la section précédente comment ces attaquants, repartis à plusieurs endroits du réseau, pouvaient construire et accumuler des preuves de liens symétriques pour une identité donnée. La cible pour cette action peut alors être n'importe quelle identité de nœud pour laquelle l'attaquant possède une preuve valide.

Remove(u,link) Selon la procédure de vérification d'un état de lien dans AdvSig, la rétention d'une information valide par attaquant, c'est-à-dire une information présente dans sa connaissance réelle du réseau, n'est pas prise en considération. Les cibles pour les actions $Remove(u, link)$ avec $(link = asym \vee sym \vee mrp)$ sont donc identiques à celles identifiées sur OLSR.

3.3.2.4 Usurpation d'une identité

Seul un attaquant de type A_3 ou A_6 peut exploiter les identités de nœuds pour lesquelles il est en possession de clés cryptographiques afin de simuler à fin niveau de granularité leurs liens auprès de ses nœuds voisins.

En revanche, en comparaison avec OLSR sans mécanisme de sécurité, l'attaquant est limité dans ses actions d'ajout unitaire d'un état de lien symétrique. Ceci est dû au fait qu'il maitrise finement uniquement les liens pour lesquels il peut soit directement créer des preuves soit collecter indirectement des preuves par l'intermédiaire des autres nœuds physiques. Par conséquent, les potentielles combinaisons d'actions non conformes $Usurpate(v)$ et $Add/Remove(u, link)$ sont également réduites.

3.3.2.5 Réplication

Un attaquant A est toujours en mesure de capturer le message d'un nœud u (tel que $A \in p_u$), puis de le réémettre tel quel vers un nœud v appartenant à une zone distante de u (tel que $v \in p_A$ et $v \notin p_u$). En d'autres termes, à l'exception du fait que le nœud u a émis à un instant donné un message de contrôle signé, aucune garantie n'est apportée au niveau de la couche physique de communication. Cette nouvelle action est dénotée $Replicate(M_u)$ où M_u représente un message de contrôle initialement généré par le nœud identifié u. Elle conduit à un effet de simulation de l'état d'un lien d'un nœud u auprès d'un nœud cible v.

Néanmoins, certaines contraintes sont imputées à l'attaquant. Premièrement, un attaquant réduit à répliquer un message de contrôle n'est pas en mesure de court-circuiter la procédure d'établissement d'état de lien entre deux victimes distantes. Il doit réaliser plusieurs actions de réplications consécutives afin de respecter l'ordre logique de leurs échanges de messages HELLO et ainsi faire passer leur état de lien d'inconnu à asymétrique puis d'asymétrique à symétrique, voire éventuellement à symétrique MPR. Toute interruption dans l'échange entraînera une perte du lien.

En outre, l'attaquant ne manipule plus une information précise contenue dans un message de contrôle, mais un ensemble d'informations. Elle peut être formulée comme étant une agglomération indivisible d'une action non conforme d'ajout d'une identité et de plusieurs actions non conformes d'ajout et/ou de suppression d'une information d'état de lien entre deux nœuds. Par conséquent, un niveau de granularité plus élevé dans les effets occasionnés se dégage de cette classe d'action non conforme. Elle conduit implicitement à un ensemble d'effets secondaires directs qui ne sont pas maîtrisables par l'attaquant. Ces effets secondaires sont des créations et des suppressions de liens symétriques impliquant le nœud à l'origine du message répliqué sur la connaissance de la victime.

En termes de cibles, l'attaquant de type A_1, A_2 ou A_3 est réduit à simuler les états de liens d'un nœud présent dans son voisinage direct. L'attaquant de type A_4, A_5 ou A_6 gouverne plusieurs nœuds physiques. Il peut alors simuler des états de liens soit entre lui-même et tout nœud dans le voisinage de ses nœuds physiques distants, soit entre tout nœud dans son voisinage et tout nœud dans le voisinage de ses nœuds physiques distants.

Le tableau 3.4 décrit, pour chacun des effets unitaires que nous avons précédemment identifiés sur la version non sécurisée d'OLSR, quelles sont les actions non conformes per-

mettant de les atteindre et quelles sont les cibles restantes selon les différents types d'attaquants. Par exemple, la première ligne de ce tableau montre que l'effet unitaire d'ajout d'un état de lien symétrique entre la victime u et l'attaquant A est atteint uniquement à partir d'une action non conforme de type $Add(u, asym)$. Ce type d'action est réalisable par des attaquants en possession d'au moins une identité de nœud valide dans le réseau, et ces cibles sont identiques à celles identifiées sur OLSR. En comparaison avec la version d'OLSR de référence, l'action non conforme de type $Add(u, sym)$ n'est plus présente car quel que soit l'attaquant, aucune preuve de lien symétrique valide entre lui-même et une cible de son voisinage ne peut être fabriquée. En prenant en considération la dernière ligne de ce tableau, l'effet de simulation des états de liens du nœud v auprès d'une victime u est atteint à partir de deux actions non conformes : l'usurpation (couplée à un ou plusieurs actions de type $Add/Remove(u, link)$ ou la réplication. En comparaison avec OLSR, le nombre de cibles pour l'action d'usurpation est fortement réduit et dépend du nombre d'identités valides possédées par l'attaquant. Quant à l'action de réplication, elle est réalisable par n'importe quel type d'attaquant, et le nombre de cibles potentielles dépend du nombre de nœuds physiques distants gouvernés par l'attaquant.

3.3.2.6 Discussion

Les aspects temporels ne sont pas pris en considération dans notre analyse ce qui constitue une limitation importante. Dans AdvSig, une synchronisation des horloges entre les nœuds est présupposée. Ainsi, avant de valider une information d'état de lien, tout nœud récepteur vérifie l'estampille temporelle de la preuve correspondante. Si la preuve est suffisamment récente, alors l'information de lien est traitée, sinon elle est rejetée. Ce mécanisme de vérification rend certaines actions non conformes inopérantes. En particulier, dès lors que l'attaquant exploite les ressources fournies par le réseau pour récupérer des preuves d'états de liens des nœuds physiques distants qu'il gouverne, des retards sont introduits. La durée de ces retards dépend directement de la longueur du chemin les séparant. Si elle est trop importante, alors les preuves d'états de liens seront immédiatement rejetées lors de la vérification locale des estampilles temporelles par les victimes.

Il est également important de préciser que de l'aveu même des auteurs, AdvSig n'est pas résistant contre les attaquants en collusion. Dans notre analyse, il était par conséquent prévisible que certains des effets décrits en section 3.2.3.3 soient atteignables par ce type d'attaquants. Néanmoins, notre approche met en relief les limitations de l'attaquant pour les atteindre, ce qui permet d'identifier précisément quels sont les apports de cette version sécurisée en comparaison avec son équivalent non sécurisé. Nous observons en particulier :

- Une réduction du nombre de cibles pour la plupart des actions non conformes. Cette réduction est d'autant plus substantielle que les capacités de l'attaquant sont limitées. Seules les cibles pour les actions non conformes de retrait d'une information sont identiques à celles identifiées sur OLSR.
- Une complexification des actions non conformes, car l'attaquant doit soit constamment échanger des informations avec les membres de sa collusion, soit fabriquer de fausses preuves.
- Une perte de contrôle dans les effets obtenus, car à défaut de pouvoir agir sur une information unitaire relative à un état de lien ou une identité, l'attaquant est réduit

Effet unitaire	Action non conforme	Cible de l'action selon le type d'attaquant						Victime
		$Active - n - m, m = 1$			$Active - n - m, m > 1$			
		$n = 0$ (A_1)	$n = 1$ (A_2)	$n > 1$ (A_3)	$n = 0$ (A_4)	$n = 1$ (A_5)	$n > 1$ (A_6)	
$C_u : Create(A, sym)$	$Add(u, asym)$	$\emptyset$	OLSR[a]		$\emptyset$	OLSR		OLSR
$C_u : Create(A, mpr)$	$Add(u, mpr)$	$\emptyset$	$\forall u \in NS_A - MPRS_A$		$\emptyset$	$\forall u \in NS_A - MPRS_A$		OLSR
$C_u : Create(v, sym/A)$	$Add(v, sym)$	$\emptyset$		$\forall v \in Y$	$\emptyset$	$\forall v \in \bigcup_{x \in X} NS_A(x)$	$\forall v \in \bigcup_{x \in X} NS_A(x) \cup Y$	OLSR
$C_u : Delete(A, sym)$	$Remove(u, sym)$	$\emptyset$	OLSR		$\emptyset$	OLSR		OLSR
$C_u : Delete(A, mpr)$	$Remove(u, mpr)$	$\emptyset$	OLSR		$\emptyset$	OLSR		OLSR
$C_u : Delete(v, sym/A)$	$Remove(v, sym)$	$\emptyset$	OLSR		$\emptyset$	OLSR		OLSR
$C_u : Create(v, asym)$	$Usurpate(v)$	$\emptyset$		$\forall u \in Y$	$\emptyset$		$\forall u \in Y$	OLSR
	$Replicate(M_v)$	$\forall v \in p_A$			$\forall v \in \cup p_x$ $x \in X$			OLSR
$C_u : Simulate(v)$	$Usurpate(v)$ *et* $Add(u, link)$ *ou* $Remove(u, link)$	$\emptyset$		$\forall v \in Y$	$\emptyset$		$\forall v \in Y$	OLSR
	$Replicate(M_v)$	$\forall v \in p_A$			$\forall v \in \cup p_x$ $x \in X$			$v, u \vert u \in p_A$

Tab. 3.4 – Analyse de sécurité d'OLSR couplé avec AdvSig ; voir l'analyse d'OLSR en comparaison (tableau 3.1)

[a] Mêmes cibles ou victimes que pour le OLSR de référence (voir le tableau 3.1)

à manipuler des messages complets.

3.3.3 Analyse d'OLSR résistant à la manipulation

Nous décrivons puis analysons notre solution de sécurité où un maximum de moyens pour assurer le bon déroulement des opérations d'OLSR est mis en œuvre. Ici, un équipement matériel résistant à la manipulation telle qu'une carte à puce ou un coprocesseur sécurisé, embarqué sur chacun des nœuds faisant partie du réseau, est responsable des opérations de routage. Parmi ces opérations figurent la génération et le traitement des messages de contrôle, et le stockage des données collectées (ensembles de connaissance du réseau) et calculées (ensemble MPR et table de routage).

3.3.3.1 Description

Dans notre solution, le format des messages OLSR n'est pas modifié. Néanmoins, afin de garantir qu'un nœud participant aux opérations de routage OLSR exécute une version valide de l'implantation d'OLSR, l'en-tête de sécurité décrit en section 2.2.3.3 est attaché à tout message OLSR généré par le processus de routage avant de sortir du TRD. Cet en-tête de sécurité est ensuite vérifié par le TRD d'un nœud récepteur d'un tel message afin d'en authentifier l'origine. Tout message présentant un en-tête correspondant valide est transmis puis traité par le processus de routage, sinon il est rejeté.

Comme nous l'avons vu au chapitre 2 (section 2.2.2), le TRD n'est pas considéré comme étant un nœud réseau communicant autonome. Il requiert les ressources de communication de son hôte. Ce dernier a pour responsabilité de relayer les messages, à savoir d'émettre sur le réseau les messages de contrôle formés par son TRD, puis de retransmettre les messages de contrôle reçus sur son interface de communication à son TRD.

Des nœuds non sécurisés, à savoir les nœuds ne passant pas par l'intermédiaire d'un tel équipement, peuvent interpréter et traiter les messages OLSR en provenance de nœuds sécurisés. Pour cela, il leur suffit de ne pas tenir compte de l'en-tête de sécurité inclus dans les messages qu'ils reçoivent. Cependant, les messages non signés qu'ils généreront seront immédiatement rejetés par les nœuds sécurisés. Par conséquent, aucune relation durable ne pourra être établie entre un nœud non sécurisé et un nœud sécurisé.

3.3.3.2 Connaissance utile du réseau

Dans cette version renforcée d'OLSR, seuls les messages présentant un en-tête valide sont traités par les TRD de nœuds OLSR distants. De plus, les ensembles de connaissance du réseau d'un nœud OLSR sont maintenus, selon les règles définies par le protocole, dans une enceinte sécurisée. Etant données ces caractéristiques, un attaquant ne peut ni intervenir directement sur ces ensembles de connaissance, ni contrôler la génération des messages de contrôle. Ainsi, même s'il peut prendre connaissance de ses voisins asymétriques, symétriques ou mpr à partir des messages qu'il capture, ces informations sont inexploitables pour la conduite de ses actions. Toutes les actions non conformes relatives à l'ajout unitaire

d'une identité et à l'ajout ou la suppression unitaire d'un état de lien avec un nœud dans un message de contrôle sont donc naturellement éliminées.

En revanche, dans la mesure où l'attaquant demeure un intermédiaire incontournable entre son propre TRD et les TRD de ses nœuds OLSR voisins, la principale vulnérabilité réside dans le flux des messages de contrôle en entrée et en sortie du TRD. Les messages de contrôle complets qu'il intercepte représenteront donc sa principale connaissance utile pour mener ses actions. Ainsi, en plus de l'action de réplication (décrite en section 3.3.2.5), nous introduisons trois nouvelles classes d'actions non conformes relatives au flux d'un message de contrôle :

- Le rejeu local. Un nœud attaquant rejoue localement à son TRD un message déjà traité par ce dernier. Cette action est dénotée $Replay(M_u)$.
- La suppression en réception. Un nœud attaquant omet de transmettre à son TRD un message en provenance d'un nœud directement à portée de communication. Cette action est dénotée $DropRecv(M_u)$.
- La suppression en émission (totale ou sélective). Un nœud attaquant omet d'émettre sur le réseau un message de contrôle généré par son TRD. Cette action est dénotée $DropSent()$.

Sur la base de ces nouvelles actions non conformes, nous mettons en exergue dans les sections suivantes les contraintes imputées à l'attaquant pour atteindre les effets unitaires observés sur la version non sécurisée d'OLSR. Ces résultats sont résumés dans le tableau 3.5.

3.3.3.3 Réplication

Comme nous l'avons observé en section 3.2.1.1, les ensembles de connaissance du réseau maintenus par un nœud OLSR représentent les points critiques du routage OLSR, car ils conditionnent le contenu des messages de contrôle. Ainsi, un moyen détourné pour l'attaquant d'ajouter ou de supprimer une information d'état de lien entre lui-même et un de ses voisins directs, sur la connaissance de ce dernier, est d'intervenir sur le flux d'un message afin de corrompre sa propre connaissance.

Pour y parvenir, l'attaquant de type A_4, A_5, ou A_6 peut accomplir plusieurs actions consécutives de réplication de ses propres messages vers un nœud v distant (tel que $v \in \cup_{x \in X} p_x$, avec X l'ensemble des nœuds physiques gouvernés par l'attaquant). Ces actions se concrétisent par la création d'un état de lien avec v sur la connaissance du réseau l'attaquant. Les effets directs et secondaires de ce type d'action ont précédemment été étudiés (section 3.3.2.5) et s'appliquent de la même façon que pour AdvSig. Indirectement, cette information d'état de lien non valide sera ajoutée dans ses générations futures de messages, puis traitée par ses voisins symétriques directs, soit $\forall u \in NS_A, C_u : Create(v, sym/A)$ (voir la quatrième ligne du tableau 3.5).

3.3.3.4 Rejeu local

Dans OLSR, le numéro de séquence sert à écarter les messages de contrôle obsolètes reçus. Il est représenté sur 16 bits, c'est-à-dire laissant des possibilités allant de 0 à 65535

messages générés par nœud avant une remise à zéro du compteur. Un attaquant peut exploiter cette particularité du protocole pour rejouer des messages de contrôle précédemment traités (et annonçant un état de lien le concernant) à son TRD. Cette action conduit à la création d'un état de lien (potentiellement durable de par des actions de rejeu consécutives) entre l'attaquant et l'identité du nœud à l'origine du message rejoué sur la connaissance de l'attaquant. L'état de lien créé est dépendant de l'information contenue dans le message rejoué. Comme dans le cas de la réplication, cette information non valide d'état de lien sera ajoutée dans ses générations futures de messages.

3.3.3.5 Suppression en réception

Un attaquant peut sélectivement supprimer des messages en provenance de n'importe quel nœud cible présent dans son voisinage direct. La victime directe de cette action est l'attaquant qui retirera de sa connaissance toute information d'état de lien avec la cible (soit $C_A : Delete(cible, asym),\ Delete(cible, sym),\ et\ Delete(cible, mpr)$), ainsi que toute information d'état de lien avec l'ensemble de ses nœuds voisins (soit $\forall u \in NS_{cible}, C_A : Delete(u, sym/cible)$).

De manière indirecte, un attaquant peut parvenir à retirer toute information d'état de lien symétrique et mpr (soit $C_{cible} : Delete(A, sym)\ et\ Delete(A, mpr)$) le concernant sur la connaissance de la cible/victime puisque dans ses émissions futures de messages de contrôle, aucune information à son égard ne sera annoncée. Remarquons que l'attaquant ne maitrise pas finement l'état de lien qu'il supprime avec la victime, et que d'autre part sa propre connaissance est diminuée.

3.3.3.6 Suppression en émission

La victime directe d'une action de suppression en émission est l'ensemble des nœuds voisins de l'attaquant. Elle entraîne directement le retrait sur la connaissance du réseau de tous les nœuds voisins de l'attaquant, toute information d'état de lien le concernant (soit $\forall u \in LS_A, C_u : Delete(A, asym),\ Delete(A, sym),\ et\ Delete(A, mpr)$).

Les effets secondaires sont pour l'attaquant un ensemble de retrait d'états de liens avec son voisinage dans sa propre connaissance. Alors que cette action menée par un attaquant indépendant ne présente pas d'intérêt puisqu'elle équivaut à une sortie du nœud du réseau, un attaquant gouvernant plusieurs nœuds physiques distants peut exclure tous les nœuds de son voisinage de la réception d'un message tout en restant connecté au réseau à travers des nœuds distants.

3.3.4 Comparaison des résultats

Les résultats de l'application de notre méthode d'analyse à AdvSig et à notre solution d'OLSR intégré à un TRD caractérisent les situations de base où la sécurité du protocole peut être compromise.

Concernant AdvSig, nos résultats révèlent que contrairement à ce qui est annoncé par ses auteurs, la connaissance de la topologie du réseau des nœuds peut être compromise par un attaquant indépendant. Nous avons notamment décelé deux cas :

Effet unitaire	Action non conforme	Cible de l'action selon le type d'attaquant						Victime
		$Active - n - m, m = 1$			$Active - n - m, m > 1$			
		$n = 0$ (A_1)	$n = 1$ (A_2)	$n > 1$ (A_3)	$n = 0$ (A_4)	$n = 1$ (A_5)	$n > 1$ (A_6)	
$C_u : Create(A, sym)$	$\emptyset$	$\emptyset$			$\emptyset$			$\emptyset$
$C_u : Create(A, mpr)$	$\emptyset$	$\emptyset$			$\emptyset$			$\emptyset$
$C_u : Create(v, sym/A)$	$Replay(M_v)$	$\emptyset$	v		$\emptyset$	v		$\forall u \vert A \in NS_A, A$
	$Replicate(M_A)$ (indirect)	$\emptyset$			$\emptyset$	$A, \forall v \in \cup p_x$ $x \in X$		$\forall u \vert A \in NS_A, A, v$
$C_u : Delete(A, sym)$	$DropSent(M_A)$	$\emptyset$	A		$\emptyset$	A		$A, \forall u \vert u \in p_A$
	$DropRecv(M_u)$(indirect)	$\emptyset$	$\forall u \vert A \in LS_u \vee NS_u$		$\emptyset$	$\forall u \vert A \in LS_u \vee NS_u$		u, A
$C_u : Delete(A, mpr)$	$DropSent(M_A)$	$\emptyset$	A		$\emptyset$	A		$A, \forall u \vert u \in p_A$
	$DropRecv(M_u)$(indirect)	$\emptyset$	$\forall u \vert A \in MPRS_u$		$\emptyset$	$\forall u \vert A \in MPRS_u$		u, A
$C_u : Delete(v, sym/A)$	$DropSent(M_A)$	$\emptyset$	A		$\emptyset$	A		$A, \forall u \vert u \in p_A$
	$DropRecv(M_v)$(indirect)	$\emptyset$	$\forall v \vert v \in NS_A$		$\emptyset$	$\forall v \vert v \in NS_A$		$A, \forall u \vert u \in NS_A, v$
$C_u : Create(v, asym)$	$Replicate(M_v)$	$\forall v \in p_A$			$\forall v \in \cup p_x$ $x \in X$			OLSR[a]
$C_u : Simulate(v)$	$Replicate(M_v)$	$\forall v \in p_A$			$\forall v \in \cup p_x$ $x \in X$			$v, u \vert u \in p_A$

Tab. 3.5 – Analyse de sécurité d'OLSR intégré à un matériel résistant à la manipulation ; voir l'analyse d'OLSR en comparaison (tableau 3.1).

[a] Mêmes victimes que pour le protocole OLSR de référence (voir le tableau 3.1)

- La **création d'un état de lien symétrique** entre l'attaquant et une cible/victime présente dans son voisinage. AdvSig n'apporte pas de solution à ce problème puisque les cibles et les actions non conformes qui conduisent à cette altération sont identiques à la version non sécurisée d'OLSR. Notons cependant que ce défaut de sécurité a été découvert indépendamment par Chen *et al.* [CXL06]. Pour le corriger, ils ont proposé l'ajout d'une nouvelle preuve d'état de lien.
- La **suppression de n'importe quel type d'état de lien** impliquant l'attaquant sur la connaissance d'une victime présente dans son voisinage. Comme précédemment, les cibles et victimes des actions non conformes de retrait d'une information sont strictement identiques à la version non sécurisée d'OLSR. Ce type d'altération de la topologie peut porter atteinte à l'efficacité du routage, car il conduit éventuellement à une moins bonne répartition du trafic ou à des partitionnements du réseau. Ceci montre qu'AdvSig peut alors s'avérer inadapté aux situations de déploiement pour lesquelles la qualité et la disponibilité des services de communication importent.

Quant aux autres effets unitaires que nous avons identifiés sur la version non sécurisée d'OLSR, ils sont toujours envisageables, mais seulement par : (1) des attaquants plus puissants ou (2) des actions non conformes de manipulation des messages complets. Dans le premier cas, nos résultats montrent qu'AdvSig permet une réduction du nombre de cibles pour la plupart des actions non conformes, et que cette réduction est d'autant plus importante que les capacités de l'attaquant sont limitées (voir le tableau 3.4). Dans le second cas, le niveau de granularité plus élevée de l'action non conforme entraine implicitement un sous-ensemble d'effets secondaires non maitrisables par l'attaquant.

Notre solution à base de TRD contribue à exclure toutes les actions non conformes relatives à la modification unitaire d'un état d'un lien, et ce quel que soit le type de l'attaquant. En comparaison avec AdvSig, nous observons que pour corrompre la connaissance du voisinage, un attaquant :

- A un ensemble de cibles potentielles encore plus restreint, et cet ensemble est indépendant de sa puissance (à l'exception des cibles pour l'action non conforme de réplication qui dépendent du nombre de nœuds physiques gouvernés par l'attaquant).
- Mène des actions sur le flux des messages qui ont des répercutions sur sa propre connaissance du réseau.

Néanmoins, nos résultats ne sont pas parfaits car ils révèlent l'existence de failles causées par des actions non conformes sur le flux des messages - problème que nous n'avons pas traité.

3.4 Conclusion

Dans ce chapitre, nous avons proposé un cadre d'analyse de la sécurité d'OLSR. Appliqué à deux de ses versions sécurisées, ce cadre a permis de formaliser la difficulté croissante pour l'attaquant d'atteindre un effet unitaire nuisible sur les données de routage. A défaut d'un réel classement, notre approche permet une certaine comparaison entre différentes versions renforcées d'un même protocole.

Les tableaux 3.4 (pour AdvSig) et 3.5 (OLSR sur TRD), résument nos résultats d'analyse de façon synthétique. Ils permettent de visualiser les actions non conformes nécessaires

pour générer les mêmes effets unitaires, ainsi que leurs cibles/victimes.

Pour permettre cette comparaison, nous avons d'abord décrit l'ensemble des actions non conformes sur le contenu des messages HELLO. Ensuite, nous avons analysé leurs cibles et leurs effets unitaires directs sur la connaissance du réseau des nœuds victimes. Les résultats de cette analyse nous ont servi de référence pour comparer les apports et les limitations d'AdvSig avec notre version basée sur un TRD.

Bien que ces deux solutions contribuent soit à une réduction, soit à une suppression des instances d'actions non conformes sur le contenu des messages, elles n'apportent toutefois pas de solution complète aux problèmes de la manipulation du flux des messages. En effet, malgré de fortes hypothèses de protection telles qu'un équipement résistant à la manipulation, des actions non conformes sur le flux des messages demeurent et permettent à un attaquant d'atteindre certains des effets néfastes identifiés sur le protocole OLSR de référence. Toutefois, nous montrons que ces actions non conformes sont de complexité et de granularité plus importante, et que le choix des cibles se trouve fortement réduit, ce qui renforce le gain net apporté par ces solutions.

Conclusion et perspectives

Travaux réalisés

Dans les réseaux ad hoc, la fonction de routage est primordiale dans la mesure où l'objet du réseau est de mettre en relation des nœuds communicants et d'acheminer leurs données. Il est donc d'une première nécessité de savoir la protéger contre les altérations, volontaires ou non. Cependant, les contraintes liées à l'absence d'infrastructure de gestion centralisée, à l'utilisation de canaux de communication sans fil et à la protection limitée des nœuds, ainsi que la nécessaire coopération entre les nœuds, rendent cette tâche notablement plus difficile pour les réseaux ad hoc que pour les réseaux traditionnels. Alors que dans les réseaux filaires, la fonction de routage est accomplie par des nœuds dédiés et physiquement protégés, dans les réseaux ad hoc, elle doit être réalisée par tous les nœuds qui y participent. Ainsi, dès lors qu'un nœud altère les opérations qui lui sont confiées, le fonctionnement de base de la totalité du réseau peut être compromis.

De nombreux travaux sur la sécurité des protocoles de routage ont été entrepris. Parmi les solutions existantes, certaines font appel à des mécanismes basés sur la cryptographie ou la détection d'intrusions pour protéger les messages de contrôle de ces protocoles contre l'inoculation de fausses informations. D'autres reposent sur des mécanismes d'observation des comportements des nœuds ou d'acquittement explicite des messages pour sécuriser spécifiquement l'opération de retransmission des paquets de données. Dans l'état de l'art, nous avons présenté les principales caractéristiques de ces solutions, puis nous en avons discuté leurs avantages et inconvénients.

Nos travaux ont tout d'abord porté sur la définition d'une couche de communication sécurisée maintenue à l'intérieur d'un équipement résistant à la manipulation tel qu'un coprocesseur sécurisé ou une carte à puce. En effet, grâce à leurs propriétés d'inviolabilité, l'utilisation de tels systèmes nous avait semblé attrayante pour sécuriser les protocoles de routage.

Nous avons donc décrit les fonctionnalités de routage et d'authentification de cette couche qui permet d'empêcher les attaques par fabrication, modification, ou rediffusion de messages de contrôle anciens. Cependant, les attaques par suppression des messages étant toujours possibles, nous avons ajouté une seconde couche de contrôle, de niveau inférieur, permettant l'évaluation de la capacité d'un nœud à émettre ou recevoir des messages.

Pour cela, nous avons tiré profit de la confiance accordée à ce matériel distribué sur chacun des nœuds du réseau pour définir un schéma de contrôle purement local et indépendant du protocole de routage sous-jacent. Contrairement aux solutions existantes, notre schéma présente l'avantage d'être fiable et d'accélérer la détection et l'identification des nœuds à l'origine de la suppression d'un message. De plus, il est adapté aussi bien à la détection des suppressions pour les messages émis par diffusion locale (utilisés dans la phase de signalisation) que ceux émis en point à point (utilisés dans la phase de signalisation et d'acheminement des données).

Bien que nous n'ayons pas validé ce schéma de contrôle par la simulation, nous l'avons néanmoins étudié de façon théorique. Nous avons pris le cas de la transmission d'un message par diffusion à partir d'un nœud vers l'ensemble de ses voisins, car ce mode de fonctionnement sert de base à la plupart des protocoles de routage. Nous avons alors observé que malgré les nets progrès en terme de fiabilité et de vitesse de détection, nous ne pouvions déterminer avec précision le nœud responsable des attaques par suppression.

Nous avons ensuite complété l'étude de notre schéma dans le cas concret de la retransmission des messages de contrôle dans les protocoles réactifs. Nous avons alors découvert que nous pouvions détecter facilement les nœuds attaquants seuls ou en collusion ne suivant pas les règles de réception et d'émission (et par extension de retransmission) définies par le protocole. Nous avons donc formalisé cette propriété sous la forme d'une méthode simple, fondée sur les délais de transmission à un saut des messages, pour réduire les possibilités d'attaques par encapsulation. Cette méthode exploite directement les informations de fautes sur les liens collectés et nécessite peu de modifications des messages de contrôle du protocole de routage ; néanmoins, elle n'est pas adaptée au cas où les deux attaquants seraient relativement proches l'un de l'autre.

D'un point de vue général, les performances de notre nouveau schéma de contrôle viennent au coût d'une augmentation importante du nombre des messages. Pour réduire ces coûts, nous avons suggéré plusieurs alternatives qui demandent à être vérifiées par la simulation.

Dans une seconde partie, nous avons survolé les différentes méthodes d'évaluation de la sécurité des protocoles de routage dans les réseaux ad hoc. Nous avons vu que seules les méthodes de spécification et de vérification formelle offrent une certaine complétude dans les résultats de validation qu'elles produisent. En revanche, à cause de la complexité des protocoles de routage ad hoc, ce type de méthode s'avère particulièrement difficile à appliquer.

Nous avons donc proposé un cadre d'analyse systématique de la sécurité de différentes versions renforcées du protocole de routage OLSR (*Optimized Link State Routing*) [CJ03]. Nous avons créé un modèle permettant de représenter les attaques en termes : d'actions non conformes sur les messages de contrôle HELLO, de cibles de ces actions et d'effets directs sur les données de routage maintenues par les nœuds appartenant à un même voisinage. Cette représentation permet de construire une base de référence substantielle sur laquelle

il est possible de raisonner pour comparer les versions renforcées. Notre approche ne prétend pas affirmer qu'une version renforcée d'OLSR est exempte d'attaques complexes ou classifier complètement plusieurs versions renforcées, mais met en évidence la potentialité de failles en se basant sur la possibilité ou non de réaliser certaines actions non conformes.

Perspectives

Dans le premier chapitre, nous avons analysé en détail un schéma d'acquittement pour détecter et identifier les comportements de suppression des messages. La suite de ces travaux est de toute évidence la validation de ce schéma par la simulation. En particulier, il serait intéressant d'étudier sa précision par une analyse du taux de détection, dans un environnement statique pour commencer, puis dans un environnement mobile.

Par ailleurs, l'architecture de sécurité que nous avons proposée impose des contraintes relativement fortes dans la mesure où toutes les fonctionnalités des protocoles de routage sont gérées à l'intérieur d'un équipement matériel résistant à la manipulation. Au vu de leurs capacités de calcul relativement limitées, cette architecture risque de dégrader les performances du réseau. Pour alléger ces traitements, une alternative intéressante à étudier serait une architecture où une partie « non critique » - à définir - des opérations de routage seraient sous la responsabilité du système hôte. Le principe de cette répartition consisterait à confier la réalisation des fonctions qui nécessitent des traitements conséquents au niveau logiciel de l'hôte, et de confier les fonctions les plus simples, mais suffisantes pour assurer les services de sécurité requis, à l'équipement matériel résistant à la manipulation.

Concernant notre méthode complète d'analyse de la sécurité, elle peut être appliquée à d'autres protocoles du même type qu'OLSR (*i.e.* des protocoles proactifs). En revanche, la notion d'effet unitaire direct utilisée dans notre approche pour faire apparaître les situations de base à partir de laquelle les données de routage peuvent être compromises n'est pas adaptée aux protocoles de routage réactifs. Ceci est du au fait que le traitement d'un message de contrôle par un nœud n'entraine pas forcément de modification de sa connaissance du réseau. Des travaux supplémentaires sont donc nécessaires pour l'adapter et la généraliser.

Bibliographie

[ABB+05] Alessandro Armando, David A. Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuéllar, Paul Hankes Drielsma, Pierre-Cyrille Héam, Olga Kouchnarenko, Jacopo Mantovani, Sebastian Mödersheim, David von Oheimb, Michaël Rusinowitch, Judson Santiago, Mathieu Turuani, Luca Viganò, and Laurent Vigneron. The AVISPA tool for the automated validation of internet security protocols and applications. In Kousha Etessami and Sriram Rajamani, editors, *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05)*, volume 3576 of *Lecture Notes in Computer Science*, pages 281–285, Edinburgh, Scotland, UK, July 2005. Springer.

[ABV06] Gergely Ács, Levente Buttyán, and István Vajda. Provably secure on-demand source routing in mobile ad hoc networks. *IEEE Trans. Mob. Comput.*, 5(11) :1533–1546, 2006.

[ACC+04] Baptiste Alcalde, Ana R. Cavalli, Dongluo Chen, Davy Khuu, and David Lee. Network protocol system passive testing for fault management : A backward checking approach. In David de Frutos-Escrig and Manuel Nú nez, editors, *FORTE*, volume 3235 of *Lecture Notes in Computer Science*, pages 150–166. Springer, 2004.

[ACH+08] Baruch Awerbuch, Reza Curtmola, David Holmer, Cristina Nita-Rotaru, and Herbert Rubens. Odsbr : An on-demand secure byzantine resilient routing protocol for wireless ad hoc networks. *ACM Trans. Inf. Syst. Secur.*, 10(4) :1–35, 2008.

[ACJ+03] Cédric Adjih, Thomas Clausen, Philippe Jacquet, Anis Laouiti, Paul Mühlethaler, and Daniele Raffo. Securing the olsr protocol. In *Proceedings of the 2nd IFIP Annual Mediterranean Ad Hoc Networking Workshop*, Mahdia, Tunisia, June 25–27 2003.

[ACL+05] Cédric Adjih, Thomas Clausen, Anis Laouiti, Paul Mühlethaler, and Daniele Raffo. Securing the OLSR routing protocol with or without compromised nodes in the network. Technical Report INRIA RR-5494, HIPERCOM Project, INRIA Rocquencourt, February 2005.

[AdSJBM07] Asmaa Adnane, Rafael T. de Sousa Jr., Christophe Bidan, and Ludovic Mé. *Analysis of the implicit trust within the OLSR protocol.* Moncton Canada, November 2007.

[AdSJBM08] Asmaa Adnane, Rafael T. de Sousa Jr., Christophe Bidan, and Ludovic Mé. Autonomic trust reasoning enables misbehavior detection in olsr. In *SAC '08 : Proceedings of the 2008 ACM symposium on Applied computing*, pages 2006–2013, New York, NY, USA, 2008. ACM.

[aHL04] Yi an Huang and Wenke Lee. Attack analysis and detection for ad hoc routing protocols. In Erland Jonsson, Alfonso Valdes, and Magnus Almgren, editors, *RAID*, volume 3224 of *Lecture Notes in Computer Science*, pages 125–145. Springer, 2004.

[AHNRR02] Baruch Awerbuch, David Holmer, Cristina Nita-Rotaru, and Herbert Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *WiSE '02 : Proceedings of the 1st ACM workshop on Wireless security*, pages 21–30, New York, NY, USA, 2002. ACM.

[AJV02] Cédric Adjih, Philippe Jacquet, and Laurent Viennot. Computing connected dominated sets with multipoint relays. Research Report RR-4597, INRIA, 2002.

[BB02] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol. In *MobiHoc*, pages 226–236. ACM, 2002.

[BB03] Sorav Bansal and Mary Baker. Observation-based cooperation enforcement in ad hoc networks. *CoRR*, cs.NI/0307012, 2003.

[BCP+98] Kirk A. Bradley, Steven Cheung, Nicholas J. Puketza, Biswanath Mukherjee, and Ronald A. Olsson. Detecting disruptive routers : A distributed network monitoring approach. In *IEEE Symposium on Security and Privacy*, pages 115–124. IEEE Computer Society, 1998.

[BDV05a] Kashyap Balakrishnan, Jing Deng, and Pramod K. Varshney. TWOACK : Preventing selfishness in mobile ad hoc networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC '05)*, volume 4, pages 2137–2142, New Orleans, LA, USA, March 13-17 2005.

[BDV05b] Levente Buttyán, László Dóra, and István Vajda. Statistical wormhole detection in sensor networks. In Refik Molva, Gene Tsudik, and Dirk Westhoff, editors, *ESAS*, volume 3813 of *Lecture Notes in Computer Science*, pages 128–141. Springer, 2005.

[BH03] Levente Buttyán and Jean-Pierre Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *MONET*, 8(5) :579–592, 2003.

[BLB04] Sonja Buchegger and Jean-Yves Le Boudec. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In *Workshop on the Economics of Peer-to-Peer Systems (P2PEcon'04)*, June 2004.

[blu] Norme IEEE 802.15. `http://grouper.ieee.org/groups/802/15/`.

[BV04] Levente Buttyán and István Vajda. Towards provable security for ad hoc routing protocols. In Sanjeev Setia and Vipin Swarup, editors, *SASN*, pages 94–105. ACM, 2004.

[CBH03] Srdjan Capkun, Levente Buttyán, and Jean-Pierre Hubaux. Sector : secure tracking of node encounters in multi-hop wireless networks. In Sanjeev Setia and Vipin Swarup, editors, *SASN*, pages 21–32. ACM, 2003.

[CCBNR07] Frédéric Cuppens, Nora Cuppens-Boulahia, Seila Nuon, and Tony Ramard. Property based intrusion detection to secure olsr. In *ICWMC '07 : Proceedings of the Third International Conference on Wireless and Mobile Communications*, page 52, Washington, DC, USA, 2007. IEEE Computer Society.

[CCBRT07] Frédéric Cuppens, Nora Cuppens-Boulahia, Tony Ramard, and Julien A. Thomas. Misbehaviors detection to ensure availability in olsr. In Hongke Zhang, Stephan Olariu, Jiannong Cao, and David B. Johnson, editors, *MSN*, volume 4864 of *Lecture Notes in Computer Science*, pages 799–813. Springer, 2007.

[CGKO04] Jon Crowcroft, Richard J. Gibbens, Frank P. Kelly, and Sven Östring. Modelling incentives for collaboration in mobile ad hoc networks. *Perform. Eval.*, 57(4) :427–439, 2004.

[CJ03] Thomas Clausen and Philippe Jacquet. IETF RFC3626 : Optimized link state routing protocol (OLSR). October 2003.

[CLM08] Hakima Chaouchi and Maryline Laurent-Maknavicius. Toward a new ad hoc node design for secure service deployment over ad hoc network. In Maryline Laurent-Maknavicius and Hakima Chaouchi, editors, *MWNS '08 : Proceedings of the workshop on Mobile and Wireless Networks Security*, pages 1–11, Singapore, May 2008. World Scientific Publishing.

[CM99] S. Corson and J. Macker. IETF RFC 2501 : Mobile Ad Hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations. 1999.

[CXL06] Lin Chen, Xiaoyun Xue, and Jean Leneutre. A lightweight mechanism to secure olsr. In Sio Iong Ao, Jeong-A. Lee, Oscar Castillo, Pranay Chaudhuri, and David Dagan Feng, editors, *IMECS*, Lecture Notes in Engineering and Computer Science, pages 887–895. Newswood Limited, 2006.

[Dou02] John R. Douceur. The sybil attack. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *IPTPS '01 : Revised Papers from the First International Workshop on Peer-to-Peer Systems*, Lecture Notes in Computer Science, pages 251–260, London, UK, 2002. Springer-Verlag.

[DRWT97] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi. Signal stability-based adaptive routing (ssa) for ad hoc mobile networks. *Personal Communications, IEEE*, 4(1) :36–45, Feb 1997.

[EGHK99] Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar. Next century challenges : Scalable coordination in sensor networks. In *MOBICOM*, pages 263–270, 1999.

[GHP07] Oscar F. Gonzalez, Michael P. Howarth, and George Pavlou. Detection of packet forwarding misbehavior in mobile ad-hoc networks. In Fernando Boavida, Edmundo Monteiro, Saverio Mascolo, and Yevgeni Koucheryavy, editors, *WWIC*, volume 4517 of *Lecture Notes in Computer Science*, pages 302–314. Springer, 2007.

[GKF02] Vikram Gupta, Srikanth Krishnamurthy, and Michalis Faloutsos. Denial of service attacks at the mac layer in wireless ad hoc networks. volume 2, pages 1118–1123 vol.2, Oct. 2002.

[Gor06] Maria Gorlatova. Review of existing wormhole attack discovery techniques. Technical Report ADA462894, OTTAWA University, 2006.

[Gri00] Gilles Grimaud. *CAMILLE : un système d'exploitation ouvert pour carte à microprocesseur*. PhD thesis, Laboratoire d'Informatique Fondamentale de Lille (LIFL), 2000.

[HAB00] John R. Hughes, Tuomas Aura, and Matt Bishop. Using conservation of flow as a security mechanism in network protocols. In *IEEE Symposium on Security and Privacy*, pages 132–131, 2000.

[HE04] Lingxuan Hu and David Evans. Using directional antennas to prevent wormhole attacks. In *NDSS*. The Internet Society, 2004.

[HGLBV01] Jean-Pierre Hubaux, Thomas Gross, Jean-Yves Le Boudec, and Martin Vetterli. Towards self-organized mobile ad hoc networks : The terminodes project. *IEEE Communications Magazine*, 39(1) :118–124, 2001.

[HJP02] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. Sead : Secure efficient distance vector routing for mobile wireless ad hoc networks. In *WMCSA*, pages 3–13. IEEE Computer Society, 2002.

[HPJ02] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne : a secure on-demand routing protocol for ad hoc networks. In Ian F. Akyildiz, Jason Yi-Bing Lin, Ravi Jain, Vaduvur Bharghavan, and Andrew T. Campbell, editors, *MOBICOM*, pages 12–23. ACM, 2002.

[HPJ03] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes : A defense against wormhole attacks in wireless networks. In *INFOCOM*, 2003.

[HPJ05] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne : a secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.*, 11(1-2) :21–38, 2005.

[HPJ06] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Wormhole attacks in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(2) :370–380, 2006.

[HSR06] Michaël Hauspie and Isabelle Simplot-Ryl. Cooperation in ad hoc networks : enhancing the virtual currency based models. In Imrich Chlamtac, editor, *InterSense*, volume 138 of *ACM International Conference Proceeding Series*, page 24. ACM, 2006.

[HTR+04] Andreas Hafslund, Andreas Tønnesen, Roar B. Rotvik, Jon Andersson, and Øivind Kure. Secure extension to the olsr protocol. In *OLSR Interop and Workshop*, August 2004.

[ibm06] IBM 4758 PCI Cryptographic Coprocessor. Online at `http://www-03.ibm.com/security/cryptocards/pcicc/overview.shtml`, 2006.

[ibm07] IBM 4764 PCI-X Cryptographic Coprocessor. Online at `http://www-03.ibm.com/security/cryptocards/pcixcc/overview.shtml`, 2007.

[IEE07] IEEE. IEEE Standard for Information technology-Telecommunications and information exchange between systems-local and metropolitan area networks-Specific requirements - Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages C1–1184, December 2007.

[IETa] IETF. Internet Engineering Task Force. `http://www.ietf.org/home.html`.

[IETb] IETF. Mobile Ad-hoc Networks (manet) working group. `http://www.ietf.org/html.charters/manet-charter.html`.

[Int94] International Organization for Standardization. Information Processing Systems — Open Systems Interconnection (OSI) — Basic Reference Model. ISO/IEC 7498 (Second Edition), 1994.

[ISO87] ISO. ISO/IEC 7816-1 : Identification cards – Integrated circuit(s) cards with contacts – Part 1 : Physical characteristics, 1987.

[JHB03] Markus Jakobsson, J. P. Hubaux, and L. Buttyan. A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks. 2003.

[JLMV02] Philippe Jacquet, Anis Laouiti, Pascale Minet, and Laurent Viennot. Performance of multipoint relaying in ad hoc mobile routing protocols. In Enrico Gregori, Marco Conti, Andrew T. Campbell, Cambyse Guy Omidyar, and Moshe Zukerman, editors, *NETWORKING*, volume 2345 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 2002.

[JM96] David Johnson and David Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, pages 153–181, 1996.

[JMC+01] Philippe Jacquet, Paul Mühlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol. In *IEEE INMIC*, pages 62–68, Lahore, Pakistan, 28–30 December 2001.

[JMH07] David B. Johnson, David A. Maltz, and Yih C. Hu. IETF RFC4728 : The dynamic source routing protocol (DSR) for mobile ad hoc networks. February 2007.

[Kah75] Robert E. Kahn. The organization of computer resources into a packet radio network. In *AFIPS National Computer Conference*, volume 44 of *AFIPS Conference Proceedings*, pages 177–186. AFIPS Press, 1975.

[KBS05] Issa Khalil, Saurabh Bagchi, and Ness B. Shroff. Liteworp : A lightweight countermeasure for the wormhole attack in multihop wireless networks. In *DSN*, pages 612–621. IEEE Computer Society, 2005.

[KMD05] Panayiotis Kotzanikolaou, Rosa Mavropodi, and Christos Douligeris. Secure multipath routing for mobile ad hoc networks. In *WONS*, pages 89–96. IEEE Computer Society, 2005.

[KNK+06] Bounpadith Kannhavong, Hidehisa Nakayama, Nei Kato, Yoshiaki Nemoto, and Abbas Jamalipour. A collusion attack against olsr-based mobile ad hoc networks. In *GLOBECOM*. IEEE, 2006.

[Kor05] Turgay Korkmaz. Verifying physical presence of neighbors against replay-based attacks in wireless ad hoc networks. In *ITCC '05 : Proceedings of the*

International Conference on Information Technology : Coding and Computing (ITCC'05) - Volume II, pages 704–709, Washington, DC, USA, 2005. IEEE Computer Society.

[KV03] Pradeep Kyasanur and Nitin H. Vaidya. Detection and handling of mac layer misbehavior in wireless networks. In *DSN*, pages 173–182. IEEE Computer Society, 2003.

[LDVB07] Kejun Liu, Jing Deng, Pramod K. Varshney, and Kashyap Balakrishnan. An acknowledgment-based approach for the detection of routing misbehavior in manets. *IEEE Trans. Mob. Comput.*, 6(5) :536–550, 2007.

[Low95] Gavin Lowe. An attack on the needham-schroeder public-key authentication protocol. *Inf. Process. Lett.*, 56(3) :131–133, 1995.

[Low96] Gavin Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In Tiziana Margaria and Bernhard Steffen, editors, *TACAS*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer, 1996.

[Low98] Gavin Lowe. Casper : a compiler for the analysis of security protocols. *J. Comput. Secur.*, 6(1-2) :53–84, 1998.

[LP04] Loukas Lazos and Radha Poovendran. SeRLoc : secure range-independent localization for wireless sensor networks. In Markus Jakobsson and Adrian Perrig, editors, *Workshop on Wireless Security*, pages 21–30. ACM, 2004.

[LSF] Association Lille Sans Fil. `http://www.lillesansfil.org`.

[Man96] Steve Mann. Smart clothing : The shift to wearable computing. *Commun. ACM*, 39(8) :23–24, 1996.

[Mar03] J. Marshall. An analysis of the secure routing protocol for mobile ad hoc network route discovery : using intuitive reasoning and formal verification to identify flaws. Master's thesis, Departement of Computer Science, Florida State University, Tallahassee, FL, April 2003.

[MGD07] Ritesh Maheshwari, Jie Gao, and Samir R. Das. Detecting wormhole attacks in wireless networks using connectivity information. In *INFOCOM*, pages 107–115. IEEE, 2007.

[MGLB00] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00 : Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, New York, NY, USA, 2000. ACM.

[MJK+00] Robert Morris, John Jannotti, M. Frans Kaashoek, Jinyang Li, and Douglas Decouto. Carnet : a scalable ad hoc wireless network system. In *ACM SIGOPS European Workshop*, pages 61–65. ACM, 2000.

[MKD07] Rosa Mavropodi, Panayiotis Kotzanikolaou, and Christos Douligeris. Secmr - a secure multipath routing protocol for ad hoc networks. *Ad Hoc Networks*, 5(1) :87–99, 2007.

[MM02] Pietro Michiardi and Refik Molva. Core : a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In Borka

Jerman-Blazic and Tomaz Klobucar, editors, *Communications and Multimedia Security*, volume 228 of *IFIP Conference Proceedings*, pages 107–121. Kluwer, 2002.

[NH06] Sebastian Nanz and Chris Hankin. A framework for security analysis of mobile wireless networks. *Theor. Comput. Sci.*, 367(1-2) :203–227, 2006.

[ns] The Network Simulator NS-2. `http://www.isi.edu/nsnam/ns/`.

[NS78] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12) :993–999, 1978.

[NS03] Peng Ning and Kun Sun. How to misuse aodv : A case study of insider attacks against mobile ad-hoc routing protocols. In *IAW*, pages 60–67. IEEE, 2003.

[NS05] Peng Ning and Kun Sun. How to misuse aodv : a case study of insider attacks against mobile ad-hoc routing protocols. *Ad Hoc Networks*, 3(6) :795–819, 2005.

[OAC05] Jean-Marie Orset, Baptiste Alcalde, and Ana R. Cavalli. An efsm-based intrusion detection system for ad hoc networks. In Doron Peled and Yih-Kuen Tsay, editors, *ATVA*, volume 3707 of *Lecture Notes in Computer Science*, pages 400–413. Springer, 2005.

[OC06] Jean-Marie Orset and Ana R. Cavalli. A security model for olsr manet protocol. In *MDM*, page 122. IEEE Computer Society, 2006.

[omn] Omnet++. `http://www.omnetpp.org/`.

[OTL04] R. Ogier, F. Templin, and M. Lewis. IETF RFC3684 : Topology dissemination based on reverse-path forwarding (TBRPF). February 2004.

[PB94] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM*, pages 234–244, 1994.

[PBRD03] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. IETF RFC3561 :ad hoc on-demand distance vector (AODV) routing. July 2003.

[PH02] Panos Papadimitratos and Zygmunt J. Haas. Secure Routing for Mobile Ad hoc Networks. In *Proceedings of the SCS Commnication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, pages 193–204, San Antonio, TX, USA, January 2002.

[PH03] Panos Papadimitratos and Zygmunt J. Haas. Secure message transmission in mobile ad hoc networks. *Ad Hoc Networks*, 1(1) :193–209, 2003.

[PR99] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, 1999.

[PS03] Venkata N. Padmanabhan and Daniel R. Simon. Secure traceroute to detect faulty or malicious routing. *Computer Communication Review*, 33(1) :77–82, 2003.

[PTSC00] Adrian Perrig, J. D. Tygar, Dawn Song, and Ran Canetti. Efficient authentication and signing of multicast streams over lossy channels. In *SP '00 : Proceedings of the 2000 IEEE Symposium on Security and Privacy*, page 56, Washington, DC, USA, 2000. IEEE Computer Society.

[QVL02] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *HICSS '02 : Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, page 298. IEEE Computer Society, 2002.

[Rab89] Michael O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *J. ACM*, 36(2) :335–348, 1989.

[RACM04] Daniele Raffo, Cédric Adjih, Thomas Clausen, and Paul Mühlethaler. An advanced signature system for olsr. In *SASN '04 : Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 10–16, New York, NY, USA, 2004. ACM.

[RC07] Kasper Rasmussen and Srdjan Capkun. Implications of radio fingerprinting on the security of sensor networks. In *Proceedings of IEEE SecureComm*, 2007.

[RE03] W. Rankl and W. Effing. *Smart Card Handbook*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[Res] ReseauCitoyen.be. `http://reseaucitoyen.be/wiki`.

[SBP01] Dawn Xiaodong Song, Sergey Berezin, and Adrian Perrig. Athena : a novel approach to efficient automatic security protocol analysis. *J. Comput. Secur.*, 9(1-2) :47–74, 2001.

[SDL+02] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. A secure routing protocol for ad hoc networks. In *ICNP*, pages 78–89. IEEE Computer Society, 2002.

[SW99] Sean W. Smith and Steve Weingart. Building a high-performance, programmable secure coprocessor. *Computer Networks*, 31(8) :831–860, 1999.

[TBK+03] Chin-Yang Tseng, Poornima Balasubramanyam, Calvin Ko, Rattapon Limprasittiporn, Jeff Rowe, and Karl N. Levitt. A specification-based intrusion detection system for aodv. In Sanjeev Setia and Vipin Swarup, editors, *SASN*, pages 125–134. ACM, 2003.

[Tec] Opnet Technologies. OPNET Modeler. `http://www.opnet.com/`.

[Toh96] Chai-Keong Toh. A novel distributed routing protocol to support ad hoc mobile computing. In *Proc. IEEE 15th Annual International Phoenix Conference on Computers and Communications, IEEE IPCCC 1996, March 27-29, Phoenix, AZ, USA*, pages 480–486. IEEE, IEEE, March 1996.

[TSF] Association Toulouse Sans Fil. `http://www.toulouse-sans-fil.net`.

[Wei93] Mark Weiser. Some computer science issues in ubiquitous computing. *Commun. ACM*, 36(7) :74–84, 1993.

[wim04] IEEE standard for local and metropolitan area networks part 16 : Air interface for fixed broadband wireless access systems. Technical report, 2004.

[WLMG05] M. Wang, L. Lamont, P. Mason, and M. Gorlatova. An effective intrusion detection approach for olsr manet protocol. pages 55–60, 2005.

[WTSL04] Chris Wullems, Kevin Tham, Jason Smith, and Mark Looi. A trivial denial of service attack on ieee 802.11 direct sequence spread spectrum wireless lans. *Wireless Telecommunications Symposium, 2004*, pages 129–136, May 2004.

[YB03] Shahan Yang and John S. Baras. Modeling vulnerabilities of ad hoc routing protocols. In Sanjeev Setia and Vipin Swarup, editors, *SASN*, pages 12–20. ACM, 2003.

[YKT03] Zhenqiang Ye, Srikanth V. Krishnamurthy, and Satish K. Tripathi. A framework for reliable routing in mobile ad hoc networks. In *INFOCOM*, 2003.

[YW99] Alec Yasinsac and William A. Wulf. A framework for a cryptographic protocol evaluation workbench. In *HASE '99 : The 4th IEEE International Symposium on High-Assurance Systems Engineering*, pages 197–206, Washington, DC, USA, 1999. IEEE Computer Society.

[ZA02] Manel Guerrero Zapata and N. Asokan. Securing ad hoc routing protocols. In W. Douglas Maughan and Nitin H. Vaidya, editors, *Workshop on Wireless Security*, pages 1–10. ACM, 2002.

[ZCY03] Sheng Zhong, Jiang Chen, and Yang Richard Yang. Sprite : A simple, cheat-proof, credit-based system for mobile ad-hoc networks. 2003.