

*Détection de la malveillance et
réactions dans les réseaux
ad hoc - Bibliographie*

Julien Thomas

31 Janvier 2007

Encadrants: Frédéric Cuppens, Nora Cuppens et
Tony Ramard

**École Nationale Supérieure des
Télécommunications**
2 rue de la Châtaigneraie
CS 17607 35576 CESSON SEVIGNE CEDEX, France
Téléphone : +33(0)2.99.12.70.00
Télécopieur : +33(0)2.99.12.70.19

Détection de la malveillance et réactions dans les réseaux ad hoc - Bibliographie

Julien Thomas

31 Janvier 2007

Résumé

De nos jours, les réseaux sans fil connaissent un grand succès, avec notamment l'existence du WiFi. Parmi ces types de réseaux, les réseaux ad hoc, actuellement peu rencontrés dans le grand public, peuvent être adaptés à de nombreuses situations, par exemple professionnelles.

Différentes techniques ont été proposées pour sécuriser certains des protocoles de routage envisagés pour les réseaux ad hoc, souvent au niveau du chiffrement des messages ou encore de l'authentification des nœuds. Toutefois, l'aspect sécurité de ces réseaux sans infrastructure est très complexe et encore nouveau. De plus, les solutions sont limitées à peu de protocoles (principalement les protocoles OLSR et AODV).

Mots-clés : sécurité, réseaux ad hoc, OLSR, NOMAD, confiance, authentification

Table des matières

1	Introduction	2
2	Généralités sur les réseaux ad hoc	2
2.1	Les différences entre les réseaux filaires et sans fils	2
2.2	Les différences entre les réseaux sans fil avec infrastructure et ceux sans infrastructure	3
2.3	Les différents protocoles de routage dans les réseaux ad hoc	3
2.3.1	Les protocoles non-uniformes	3
2.3.2	Les protocoles uniformes	4
2.3.3	Comparaison des catégories de protocoles	4
2.4	Présentation du protocole OLSR	5
2.4.1	Notions de base	5
2.4.2	Structure des messages	5
2.4.3	Sélection des nœuds MPR	5
3	La sécurité dans les réseaux ad hoc	6
3.1	Les besoins	6
3.2	Les attaques existantes	7
3.2.1	Présentation	7
3.2.2	Exemples d'attaques	7
3.3	Les solutions existantes	9
3.3.1	Établissement de la confiance, évaluation des réputations	9
3.3.2	Différentes techniques pour l'authentification	10
3.3.3	ADVSIG : sécurisation d'OLSR	11
3.3.4	MATA	13
3.3.5	Nomad : spécification formelle de la sécurité	15
3.3.6	Utilisation de Nomad dans OLSR	16
4	Conclusion	16
4.1	Bilan	16
4.2	Sujets du Stage	17

1 Introduction

De nos jours, la possibilité d'accéder rapidement et facilement aux réseaux informatiques est devenue une nécessité. Les solutions sans fil actuelles offrent ces services, mais il est nécessaire de prévoir une architecture de gestion.

Les réseaux ad hoc ont une architecture de graphe dans lequel les nœuds forment un réseau sans l'aide d'une infrastructure ou d'une administration centralisée. Si les protocoles pour ces réseaux ont fait l'objet de nombreuses études, il reste à améliorer les solutions existantes pour notamment prendre en compte la sécurité.

Des études ont été consacrées à la sécurisation des réseaux ad hoc. Toutefois, les techniques proposées offrent des solutions partielles et adaptées principalement aux protocoles les plus connus.

Dans la section 2, je ferai une brève introduction sur les différents types de réseaux, ainsi que sur les protocoles de routage existant pour les réseaux ad hoc.

Dans la section 3, je présenterai différents aspects sur la sécurité : les besoins, les attaques existantes, sur tous types de réseaux et les moyens de protection existants ou proposés.

Je terminerai avec la section 4.2 par un aperçu des différents sujets d'études envisagés pour mon stage. Parmi ceux-ci, on trouvera deux grands thèmes : renforcement de la sécurisation sur les protocoles connus (par exemple OLSR) et étude des protocoles hybrides, auxquels peu d'études sont actuellement consacrées.

2 Généralités sur les réseaux ad hoc

2.1 Les différences entre les réseaux filaires et sans fils

Ces dernières années, les réseaux sans fil ont connu une forte croissance sous l'impulsion de technologies telles que Bluetooth¹ [IEEE 802.15] et le WiFi [IEEE 802.11].

A la différence des réseaux filaires, ces réseaux ne sont pas reliés physiquement. Ils peuvent ainsi fournir de nouveaux services, avec notamment la possibilité d'accéder simplement à Internet (via les bornes *Tipi*²) ou encore d'accéder à un réseau privé (domestique ou professionnel) sans avoir à installer de câble. Toutefois, ce moyen de communication remet en cause plusieurs notions assimilées pour les réseaux filaires. L'absence d'infrastructure remet par exemple en question la gestion des accès, vu que n'importe qui peut tenter d'accéder au réseau (ou l'écouter). De plus, certains de ces réseaux ne possèdent pas d'architecture logique, ce qui complique encore plus le problème, notamment pour la gestion des services, ce qui est présenté dans la prochaine section.

La suite du rapport sera consacrée aux problèmes engendrés par l'absence d'infrastructure, cadre du sujet de mon stage (voir la section 4.2)

¹Le Bluetooth est une technologie visant à relier les appareils numériques, généralement des périphériques à un centralisateur (par exemple un poste informatique).

²le mot *Tipi* désignant traditionnellement une tente de forme conique utilisée par certaines tribus d'indiens d'amérique est utilisé pour définir les points d'accès à des réseaux Internet

2.2 Les différences entre les réseaux sans fil avec infrastructure et ceux sans infrastructure

Les réseaux sans fil les plus répandus et les plus connus sont ceux se basant sur une infrastructure. Cela est notamment dû au fait qu'ils sont utilisés pour fournir des services de haut niveau, par exemple une connexion à Internet. Comme leur nom l'indique, les réseaux sans fil avec infrastructures sont construits de manière logique, en respectant une infrastructure globale stable, notamment au niveau de la gestion du réseau. Ainsi, toutes les fonctionnalités d'administration du réseau, du routage, sont gérées par exemple par un fournisseur d'accès à Internet, ce qui offre un service fiable et simple.

Les réseaux sans fil sans infrastructure sont des réseaux qui se construisent d'eux mêmes, à la différence des réseaux utilisant actuellement la technologie WiFi et pour lesquels il existe un réseau d'administration (pour notamment la gestion des services) pré-établi. Ces réseaux (appelées MANET [11], Mobile Ad hoc NETwork) se basent sur des stations faisant partie du réseau pour assurer tous les besoins nécessaires au bon fonctionnement du réseau, notamment les politiques de routage. La gestion est donc dynamique et dépendante du réseau lui-même.

Ces réseaux offrent de nouveaux services, comme la possibilité de déployer rapidement des systèmes, sans avoir à prévoir une structure d'administration proprement dite (on pourrait par exemple penser au déploiement d'un réseau de capteurs communicants, l'administration étant dans ce cas répartie entre les différents capteurs). Mais, bien entendu, l'absence d'infrastructure engendre des faiblesses, lesquelles peuvent être exploitées pour attaquer la disponibilité du réseau (voir la section 3). Un des problèmes évidents est que les autres nœuds sont également responsables d'une partie de l'administration. Or, un nœud n'a pas forcément de connaissance a priori sur les autres, la confiance n'est donc pas aussi simple que dans le cas des réseaux classiques.

2.3 Les différents protocoles de routage dans les réseaux ad hoc

Les différents problèmes posés par les réseaux ad hoc (forte mobilité des nœuds, faible espace mémoire, batterie limitée, ...) font que plusieurs grandes catégories de protocoles ont été étudiées. Ces catégories sont résumées dans la figure 1.

Les deux grandes catégories de protocoles de routage sont les protocoles *uniformes* et les protocoles *non-uniformes*, lesquelles peuvent être également divisés en sous-catégories (présentées dans les sections suivantes). Les protocoles se distinguent par le fait que les nœuds jouent tous (ou non) le même rôle. Il est à noter qu'il existe une classification transversale des protocoles en fonction de la périodicité de calcul des routes : les protocoles *proactifs* calculent les routes à différents intervalles, tandis que les protocoles *réactifs* calculent les routes à la demande, c'est-à-dire lorsqu'un message doit être propagé par le nœud (les protocoles intermédiaires étant appelés *hybrides*).

FIG. 1 – Les protocoles de routage dans les réseaux ad hoc

2.3.1 Les protocoles non-uniformes

Les protocoles non-uniformes se distinguent par le fait que tous les nœuds n'ont pas le même rôle au sein du réseau. Classiquement, certains interviendront directement dans la gestion du routage, et d'autres non. Parmi ces protocoles, on distingue ceux dits à partitionnement (avec par exemple CBRP [7]), dans lesquels un nœud maître est choisi pour chaque sous-partie du réseau, de ceux dits à sélection de voisins. Les protocoles à sélection de voisins sont les plus utilisés, avec en autres le protocole OLSR [13] qui possède plusieurs implémentations³ et fait l'objet de nombreuses études. Cette deuxième catégorie de protocoles repose sur la désignation de nœuds voisins responsables du routage.

2.3.2 Les protocoles uniformes

Parmi ces protocoles, on distingue deux sous-catégories : les protocoles orientés topologie (*link-state protocols*) avec par exemple FSR [10], lesquels reposent sur l'état des connexions avec les voisins, et les protocoles orientés destination (*distant-vector protocols*). Cette deuxième catégorie de protocoles est proche des protocoles de routage utilisés sur les réseaux filaires. Dans ce cas, les nœuds maintiennent des informations sur la distance (en nombre de sauts) les séparant des nœuds destination, ainsi que le prochain nœud à atteindre. L'un des protocoles les plus connus, AODV [2], fait partie de cette catégorie.

2.3.3 Comparaison des catégories de protocoles

Bien entendu, chaque catégorie possède des points forts et des points faibles, lesquels dépendent de la topologie du réseau déployé.

Les protocoles proactifs permettent l'envoi rapide de messages, mais nécessitent une gestion périodique des routes, ce qui est coûteux en énergie. Les protocoles réactifs uniformes ne sont pas efficaces sur des réseaux à grande échelle ou fortement mobiles.

Le protocole OLSR est celui sur lequel de nombreuses études ont été menées et est également l'un des protocoles les plus prometteurs (il est par exemple déjà défini comme RFC par l'IETF, ce qui n'est pas le cas de tous). Le but de la section suivante est de présenter ce protocole, mais également d'introduire les spécificités des réseaux ad hoc à travers cet exemple.

2.4 Présentation du protocole OLSR

2.4.1 Notions de base

Le protocole Optimized Link State Routing est une extension pour les réseaux ad hoc des protocoles du type Link State Routing (avec notamment le protocole OSPF [14], Open Shortest Path First) déployé sur les réseaux classiques. Il s'agit d'un protocole proactif non uniforme dont les nœuds assurant la diffusion des messages sont appelés MPRs (MultiPoint Relay, relai multipoint), les autres nœuds ayant pour fonction de choisir ces MPRs.

³les sites <http://www.olsr.org> et <http://hipercom.inria.fr/OOLSR/> proposent des informations et des implémentations de ce protocole

Il se base sur deux types de messages, Hello et TC (que nous décrivons dans les sections suivantes). L'état d'un lien entre deux nœuds peut être Inexistant, Unidirectionnel ou Bidirectionnel. Les états varient lors des échanges de messages entre les nœuds (Inexistant → Unidirectionnel → Bidirectionnel).

2.4.2 Structure des messages

Les messages HELLO

Les messages HELLO sont échangés périodiquement entre les voisins, mais non propagés. Ils contiennent des informations sur l'état des liens entre les nœuds voisins. Un message HELLO contient trois listes, définissant les nœuds à liens bidirectionnels, unidirectionnels, et les nœuds MPRs du nœud à l'origine du message.

Les messages TC

Les messages TC (Topology Control) sont propagés par inondation sur tout le réseau. Ils ont pour but de signaler l'état global du réseau.

A noter que ces deux types de messages contiennent des informations redondantes utilisées par certaines versions sécurisées du protocole pour effectuer des contrôles.

2.4.3 Sélection des nœuds MPR

Le but d'un nœud MPR est de propager les données émises par un de nœuds MPR l'ayant sélectionné comme nœud MPR (on parle de *MPR Selector*), pour assurer la propagation des messages dans le réseau. Afin d'avoir un fonctionnement optimal, la sélection des nœuds MPR par un nœud se fait de la manière suivante : d'après l'ensemble des messages HELLO reçus, chaque nœud connaît tous les autres nœuds accessibles en deux sauts. En utilisant un algorithme d'optimisation, chaque nœud choisit un ensemble minimal de nœuds à un saut, lequel permet de couvrir tous les nœuds accessibles en deux sauts. Par récurrence, on arrive à couvrir tout le réseau.

Cet algorithme n'est pas forcément optimal, notamment s'il y a des choix arbitraires, mais il permet d'assurer un recouvrement total pour chaque nœud. La figure 2 illustre le fonctionnement de ce protocole sur un réseau en étoile. Les nœuds MPRs indiqués sont ceux choisis par le nœud central.

FIG. 2 – Fonctionnement du protocole OLSR. Les nœuds MPRs sont en noir

3 La sécurité dans les réseaux ad hoc

3.1 Les besoins

D'un point de vue sécurité, les besoins pour les réseaux ad hoc peuvent être traduits par cinq propriétés.

- La confidentialité est nécessaire en raison de plusieurs caractéristiques des réseaux ad hoc. Parmi celles-ci, les plus importantes sont :
 - l’aspect sans fil fait que n’importe qui peut écouter les conversations ;
 - l’aspect sans infrastructure (logique ou physique) fait qu’un nœud ne peut pas faire des suppositions sur le chemin emprunté par des données, il peut donc ne pas faire confiance aux nœuds intermédiaires.
- L’intégrité des données dans les réseaux ad hoc peut être remise en cause par de nombreux événements. Parmi ceux-ci, les attaques visant à modifier le contenu des messages et la faible fiabilité des liaisons sans fil (elles sont beaucoup moins fiables que les liaisons filaires) sont les deux plus connus et les plus importants.
- L’authentification dans un réseau ad hoc est tout aussi importante que dans les autres types de réseaux. Toutefois, l’aspect sans fil (n’importe qui peut envoyer des messages) rend ce besoin plus important.

Ces différentes propriétés peuvent être assurées par des techniques de chiffrement et de signature. Mais comme nous le verrons dans la section 3.3.2, le problème n’est pas aussi trivial que cela.

Deux autres propriétés peuvent être mises en évidence pour ce type de réseaux.

- La fiabilité, qui vise à avoir un réseau robuste, capable de gérer des problèmes d’engorgement. Des techniques telles que des procédures de secours sont généralement employées pour renforcer cette propriété.
- La disponibilité, qui vise à assurer la persistance d’un service. De nombreuses attaques dans les réseaux classiques ont pour but de remettre en cause cette propriété. Parmi ces attaques, on trouve notamment les dénis de services (les DoS, décrits dans la section suivante).

3.2 Les attaques existantes

3.2.1 Présentation

Le but de cette section est de présenter les attaques les plus connues et contre lesquelles les protocoles actuels proposent des solutions partielles.

Les attaques décrites ci-dessous ne sont pas spécifiques aux réseaux ad hoc mais l’absence de structure dans ces réseaux fait qu’il n’existe pas une autorité centrale gérant l’aspect sécurité. Ce problème, associé avec l’aspect sans fil dans ces réseaux, aide à la concrétisation d’attaques.

3.2.2 Exemples d’attaques

Wormhole

Cette attaque est réalisée lorsque plusieurs nœuds sont compromis. Le but est alors de simuler un nœud dans le voisinage. La technique de base utilisée par les nœuds compromis est l’encapsulation des messages, pour former un lien de voisinage virtuel, comme illustré dans la figure 3. Cette technique permet à B de fournir les preuves demandées par le protocole, comme si elles venaient de A.

FIG. 3 – Principe du wormhole

Blackhole / Grayhole

Les techniques *blackhole* et *grayhole* ont pour but de ne retransmettre aucun ou une partie seulement des paquets reçus. Pour le cas des *grayholes*, le choix des paquets retransmis n'est généralement pas lié aux hasard, le but étant par exemple de favoriser une partie du trafic. Toutefois, il est à noter que cette attaque peut être également confondue avec le fait qu'un nœud est soit surchargé, soit incapable (nœud à faible capacité) de jouer le rôle d'un routeur, ce qui peut compliquer la détection de ce genre d'attaques.

Eavesdropping

L'*eavesdropping* consiste à écouter les messages circulant sur le réseau. Le but classique est souvent de récupérer différentes informations non chiffrées (par exemple dans le cas de protocoles non sécurisés) ou encore d'obtenir un volume de données chiffrées assez important pour effectuer des analyses afin de retrouver les clés de chiffrement.

Cette technique est notamment employée pour casser le chiffrement WEP [21], lequel est généralement utilisé pour chiffrer les connexions WiFi.

Masquerade attack (attaque par imposture)

L'attaque par imposture consiste à usurper l'identité d'une autre personne. Cette attaque est efficace s'il est possible d'usurper l'identité d'une entité (par exemple un nœud) ayant accès au réseau. Dans les réseaux, les différentes techniques sont des variantes d'attaques consistant à modifier ses adresses MAC ou IP (on parle de spoofing). Pour cette raison, les *masquerade attacks* sont également appelées *Spoofing attacks*.

Toutefois, la notion d'imposture n'est pas toujours utilisée par des procédés ayant des buts illégaux. On a par exemple la gestion du NAT⁴ des firewalls qui est qualifiée de (IP-)masquerade

Denial Of Service - DoS

Le but du DoS est de rendre le service indisponible en s'attaquant aux structures fournissant le service lui-même (indisponibilité du serveur, ...). Sur les réseaux ad hoc, l'absence de structure fait que le service (de routage, par exemple) est généralement réparti entre les entités. Les attaques du type DoS peuvent donc être appliquées sur les nœuds les plus faibles, parmi ceux effectuant le service.

Ces attaques sont généralement prises en compte dans les protocoles et peuvent être contrées, par exemple, soit en ayant peu de nœuds associés au service, et dans ce cas ces nœuds peuvent être renforcés, soit un grand nombre de nœud, mais avec une technique de redondance matérielle. Ce genre d'attaque est étudié dans l'authentification du type *Threshold Cryptography*, cf. section 3.3.2.

⁴le Network Address Translation (NAT) consiste en une translation d'adresse (IP). Une description est disponible à l'adresse http://en.wikipedia.org/wiki/Network_address_translation

Traffic jamming

Le but de cette attaque est de paralyser totalement le réseau en générant un signal radio sur les mêmes fréquences que le réseau. Cet aspect radio fait qu'aucun autre signal ne peut être émis, sinon il serait totalement bruité par l'attaque, et donc inutile.

La solution étant dans ce cas de supprimer physiquement l'attaque, des moyens logiciels n'étant pas adaptés.

3.3 Les solutions existantes

3.3.1 Établissement de la confiance, évaluation des réputations

Présentation

Dans une étude [9] faite par Jinshan Liu et Valérie Issarny, un système de recommandation et de réputation a été proposé. Il s'agit de fournir à chaque nœud différents paramètres (résumés dans le tableau 1) pour évaluer la qualité des autres nœuds. Parmi ces paramètres, on trouve SRep, qui est la réputation d'un nœud déduite par toutes les données disponibles et SExp qui est la réputation obtenue par les seules expériences entre ce nœud et soi. Chaque nœud possède également des informations sur la qualité de la recommandation des autres nœuds (RRep), ce qui indique la valeur à associer aux recommandations faites par ses pairs. Il est à noter que seule la recommandation faite par un nœud au sujet d'un autre (Rec) est distribuée aux autres nœuds.

Le fait d'avoir dissocié ces différents paramètres va permettre à chaque nœud d'évaluer au mieux deux notions importantes : la qualité du service d'un nœud et la qualité de ses recommandations. En effet, lorsque le nœud possède un bon service, on peut l'inclure dans le routage, même si ses recommandations sont peu fiables, et donc peuvent être ignorées.

$SRep_a(o)^t$	Qualité du nœud o , d'après a , à l'instant t
$RRep_a(o)^t$	Qualité de la recommandation de o , d'après a , à l'instant t
$SExp_a(o)^t$	Réputation de o par a à l'instant t , déduite des expériences entre les deux nœuds
$Rec_a(o)^t$	Recommandation faite par a au sujet de o , à l'instant t . Normalement, $Rec_a(o) = SRep_a(o)$
ρ_e, ρ_c	facteur de vieillissement des informations

TAB. 1 – Éléments de base du système de recommandation et de réputation

Évolution de la réputation

Bien entendu, la qualité de l'évaluation dépend de la méthode utilisée pour faire évoluer les différents paramètres de cette évaluation. Pour la réputation, le mécanisme est simple : la réputation d'un nœud est fonction de l'évaluation faite par le nœud courant a , ainsi que celles des autres (nœuds p) pondérées par la confiance envers ces tiers. La formule suivante résume l'évaluation de la qualité d'un nœud o , perçue par le nœud courant a .

$$SRep_a(o)^t = \rho_e \cdot SExp_a(o)^t + (1 - \rho_e) \cdot \sum_p (RRep_a(p) \cdot Rec_p(o)) / \sum_p RRep_a(p).$$

Évolution de la qualité de la recommandation

L'évolution de la qualité de la recommandation d'un nœud p se base sur les différences (entre p et a) sur l'évaluation de la qualité (Rec_p et SEXP_a) du service d'un nœud o . On obtient ainsi la formule suivante : $\text{diff}_1 = | \text{Rec}_p(o) - \text{SEXP}_a(o) |$.

Toutefois, la différence entre les évaluations de deux nœuds peut se justifier par des analyses différentes sur des données différentes (soit un contexte différent). De ce fait, deux évaluations différentes ne veulent pas forcément dire qu'un nœud évalue mieux que l'autre. Ce deuxième nœud ne doit donc pas forcément voir sa qualité de recommandation diminuer. De ce fait, il a été proposé d'ajouter un seuil de tolérance δa afin d'assurer une flexibilité des évaluations. Ceci fait que la différence de recommandation est définie ainsi : $\text{diff} = (1 - \text{diff}_1) / \delta a$.

L'évolution de la recommandation respecte alors le principe de base suivant : la réputation (d'après le nœud a) d'un nœud p à l'instant t dépend de sa réputation précédente (temps t') et des différences d'évaluation des autres nœuds perçues entre ces deux évaluations de la réputation de p .

$$\text{RRep}_a(p)^t = \text{RRep}_a(p)^{t'} \cdot \rho_c^{(t-t')} + \text{diff} \cdot (1 - \rho_c^{(t-t')}) .$$

3.3.2 Différentes techniques pour l'authentification

Dans les premiers paragraphes de cette section, je vais m'attacher à présenter différents systèmes d'authentification, basés sur des systèmes de clés asymétriques. Je détaillerai ensuite une technique de clé symétrique globale.

Cryptographie à seuil - threshold cryptography

Considérons un réseau comportant N nœuds. Cette méthode d'authentification repose sur 3 éléments :

- n nœuds, $n \leq N$, sont chargés de jouer le rôle d'autorité de certification partielle (partial CA). Par certification partielle, on entend le fait qu'un nœud n'est pas suffisant pour fournir un certificat valide. Il est nécessaire d'avoir la certification d'au moins k nœuds pour obtenir un certificat (Dans notre cas, on choisit $k = t + 1$, comme indiqué juste après),
- l'authentification se base sur un ensemble de certificats partiels,
- $t + 1$ certificats sont nécessaires, t étant le seuil de stabilité de l'authentification.

Le but des protocoles est alors de s'assurer qu'il n'y a jamais plus de t nœuds compromis parmi les n assurant le rôle d'autorité partielle.

Il existe différents types de protocoles à seuil, jouant sur les paramètres n et t , ainsi que sur l'élaboration des certificats. Cette technique permet également la révocation de certificats, ce qui fait qu'un nœud, une fois qu'il est compromis, peut être rejeté sans problème. Différents algorithmes ont été proposés pour fournir un système de cryptographie à seuil. Parmi ceux-ci, on trouve notamment celui de S. Jarecki et A. Lysyanskaya [5].

Cette méthode rend le réseau beaucoup plus robuste, notamment face aux problèmes causés par le départ des nœuds. Tant qu'au moins $t + 1$ nœuds sont des autorités de certifications partielles, le réseau résiste. De plus, cette technique rend le réseau plus

résistant à différentes attaques, par exemple le problème des dénis de service ou encore de la collusion de nœuds (le nombre de nœuds malveillants doit être supérieur à t pour pouvoir gêner le réseau).

Self-organized PKI⁵

En se basant sur le système PGP (*Pretty Good Privacy*), un système de PKI distribuée [17] a été proposé pour les réseaux ad hoc. Il se base sur une notion de confiance transitive : si A croit B et B croit C, alors A croit C (mais C ne croit pas forcément A, la notion n'étant pas symétrique). Cette technique permet de former des chaînes de certifications, pour avoir au final un graphe de certificats.

Différentes techniques ont été proposées pour simplifier le graphe, afin par exemple d'avoir des chaînes de certificats les plus courtes possibles, pour limiter les surcharges au niveau des certifications.

Cette technique permet également à deux entités ne se connaissant pas de se faire confiance, en croisant les certifications, ce qui n'est pas évident dans les autres techniques d'authentification.

ID-based cryptography

Le but de cette technique est d'éliminer les certificats proposés dans les méthodes précédentes, afin d'alléger la surcharge globale. La suppression des certificats se base sur le fait que chaque nœud possède un identifiant, par exemple l'adresse MAC ou IP⁶. En utilisant une technique de hachage (pour avoir des clés publiques similaires), on peut obtenir une clé publique, donc trouvable par tous.

Chaque nœud obtient sa clé privée en consultant le serveur PKG (*Private Key Generator*). À noter que la sécurité repose sur le fait que seul le PKG peut faire la conversion public vers privé. Le PKG peut également être distribué (par exemple avec la technique *threshold cryptography*, présentée précédemment), pour améliorer la fiabilité de la technique et être aussi plus résistant aux attaques du type DoS.

Cryptography-based address

Cette méthode repose sur le principe inverse de la cryptographie basée sur les identifiants. Chaque nœud génère un couple de clés privée/publique. Différentes méthodes peuvent ensuite être appliquées pour générer l'adresse de l'hôte à partir d'une (ou des deux) clés. Il existe par exemple la méthode SUCV addresses (Statistically Unique Cryptographically Verifiable), utilisée dans le protocole MATA, présenté dans la section 3.3.4.

Cette technique utilise les adresses IPv6 (codées sur 128 bits) et non IPv4 (32 bits), afin d'être résistant aux attaques sur la découverte des clés privés à partir des clés publiques, ou de l'adresse (par exemple les attaques dites brute force, qui testent toutes les possibilités).

⁵PKI : Public Key Infrastructure, structure de partage des clés publiques

⁶les adresses MAC et IP sont les identifiants de base d'un hôte sur un réseau

3.3.3 ADVSIG : sécurisation d'OLSR

Présentation

Le système ADVance SIGNature [19], proposé par Raffo, a pour but de renforcer la sécurité du protocole OLSR. Chaque message ADVSIG, associé à un message OLSR classique, contient des certificats (ou preuves) ainsi qu'une estampille temporelle. L'estampille temporelle se base sur le principe suivant : l'état d'un réseau, au niveau des liens, dépend de l'état précédent. Ainsi, un nœud A peut propager l'information qu'il a un lien avec B à l'instant t_i , si cette information lui a été également fournie (sous forme de certificat) par B, à l'instant t_{i-1} .

Le principe est simple : chaque nœud stocke les certificats fournis à t_{i-1} par ses voisins. Il réémet ensuite ces certificats, comme preuves, lors de l'envoi de ses messages à t_i , pour valider l'état de ses liens.

Bien entendu, les preuves requises dépendent du type du message. Par exemple, il faut prouver

- que le paquet a été entendu, si A veut déclarer un lien de type ASYM_LINK avec B (i.e. qu'il a reçu un message de B);
- que l'on a reçu une déclaration de SYM_NEIGH (i.e. qu'il a reçu un message de B et B a reçu un message de A) ou MPR_NEIGH (B est un nœud MPR), si A veut déclarer B comme voisin.

Une présentation détaillée est disponible dans [19].

Critiques du système

Ce système repose sur deux hypothèses fortes : la synchronisation entre tous les nœuds, et l'existence d'une PKI. L'hypothèse de synchronisation dans les réseaux ad hoc peut être validée par des algorithmes tels que TSF/ATSP [8]⁷ ou encore un protocole en cours d'étude, prévu pour les réseaux ad hoc [12] (informations obtenues d'après les spécifications de MATA, voir la section 3.3.4). Pour les PKI, il en est de même.

Toutefois, ces deux hypothèses font que la mise en place d'un tel système rend le protocole OLSR beaucoup plus lourd. Ce problème est même signalé sur le site de l'auteur (une clé de 128 bits est conseillée pour limiter la surcharge des entêtes).

Dans [18], un problème plus important a été mis en évidence : dans ADVSIG, les certificats sont envoyés pour la déclaration des liens, non pour l'élaboration de ceux-ci. Cela fait qu'une attaque sur l'aspect symétrique des liens peut être effectuée.

Considérons le schéma suivant :

source → *destinataire* : {message, certificats, date} *signataire*, les certificats étant représenté par {message, date} *signataire*.

Une attaque sur l'aspect symétrique des liens est la suivante : A va simuler avoir reçu un message de B du type {0,0,T1'}SB. L'absence de vérification à ce niveau du protocole va

⁷Adaptive Timing Synchronization Procedure est une version modifiée de Timing Synchronization Function, proposée par Huang and Lai

permettre à A de faire comme s'il l'avait reçu (étape 3).

$A \rightarrow B : \{0,0,T1\}SA$

pour tester, B envoie le message `ASYM_LINK` vers A, qui ne le recevra pas.

$A \rightarrow B : \{\{B : ASYM_LINK,T2\}SA,0, T2\}SA$ (étape 3)

$B \rightarrow C : \{\{A : SYM_LINK,T3\}SB, \{B : ASYM_LINK,T2\}SA, T2\}SB$

Une amélioration de ADVISG, notée ADVSIG⁺, propose de signer les messages d'élaboration des liens. Toutefois, Cette version présente également des lacunes, à nouveau au niveau de la phase de confirmation des liens symétriques (cf. section 3.4.5 de [18]).

Aspects intéressants

Bien que ce protocole présente plusieurs problèmes, il est intéressant de remarquer que c'est l'un des premiers protocoles ad hoc mettant en place un système de validation des messages par des preuves. En l'absence des problèmes d'usurpation des identités, on aurait une identification fiable des nœuds, ainsi que des messages. La seule attaque persistante étant dans ce cas celle de la collusion de plusieurs attaquants sur le réseau.

3.3.4 MATA

Présentation

La méthode MATA [18] (*Message Authentication protocol based on Temporary Addresses*) se base sur l'authentification des stations à partir de leur adresse. Un réseau sous MATA est également reconnaissable par un ensemble de variables qui le définissent :

- préfixe du réseau IPv6 (le choix d'un réseau IPv6 est motivé par le mécanisme SUCV, voir plus loin),
- T_{max}, la durée maximale d'existence du réseau,
- T_{int}, la durée de chaque intervalle. Chaque intervalle permet d'associer une clé à un nœud. On peut ainsi réduire le temps d'utilisation de chaque clé, et donc augmenter la sécurité.

Au niveau de chaque nœud, le calcul des adresses fixes exploite le mécanisme SUCV. Celui-ci permet d'identifier de manière unique chaque nœud du réseau, sans avoir à utiliser une PKI. Chaque nœud effectue donc les opérations suivantes :

- calcul d'une clé secrète s_i ,
- calcul du nombre de clés nécessaires pour la station : $n = E(T_{max} / T_{int}) + 1$.
Chaque clé k étant $h^k(s_i)$,
- l'adresse AD_i de la station i étant dans ce cas $\langle \text{prefixeIPv6}, \text{hash64}(h^n(s_i)) \rangle$

L'utilisation des adresses IPv6 permet d'avoir un hachage de la clé sur 64bits, ce qui est nécessaire pour la robustesse du système.

L'envoi de messages dans MATA exploite le fait que certains champs resteront fixes lors du trajet (par exemple les données), tandis que d'autres varieront (par exemple le TTL⁸). On peut décomposer le message M en M_{fix} et M_{var} . Cela nous permet de former un message au final $\langle M, h^{n-k+1}(h(M_{fix}), M_{var}) \rangle$, où $h^{n-k-1}(A) = \text{HMAC}(h^{n-k+1}(s_i), A)$.

⁸le TimeToLive est un champ contenu dans les paquets IP, et diminuant à chaque passage par un routeur

Supervision

La politique de supervision utilisée par défaut est SURO⁹, la technique étant la suivante : lors de l'envoi d'un message M vers B , la station A stocke $h(M_{fix})$ dans sa table. Ensuite, lorsque B propage M , sous la forme $\langle M', h^{n-k+1}(h(M'_{fix}), M'_{var}) \rangle$, A vérifie (à partir de M') que $h(M_{fix}) = h(M'_{fix})$. Il stocke également les identifiants de B : AD_B , $h^{n-k+1}(h(M'_{fix}), M'_{var})$ et $h(M'_{var})$.

Dans d'un intervalle temporel, chaque station possède donc les informations suivantes sur ses voisins :

A	$AD_B, h_i^{n-k+1}(h(M'_{fix}), h(M'_{var}))$ et $h(M'_{var})$
B	$AD_A, h_i^{n-k+1}(h(M_{fix}), h(M_{var}))$ et $h(M_{var})$

Authentications

La première authentification consiste à vérifier l'identité de l'émetteur, notamment lors du changement de clé (i.e. lors du changement d'intervalle). A chaque transition $k+1$, les nœuds émettent en broadcast un message Key Disclosure (KD) structuré de la manière suivante : $\langle KD, AD_i, h^{n-k+1}(s_i) \rangle$.

Lorsqu'un nœud j reçoit ce message, il a deux possibilités : ignorer le message si aucun message de i n'est à authentifier, ou changer sa clé pour authentifier ces données. Dans ce cas, il faut d'abord authentifier la nouvelle clé.

Pour ce faire, le test suivant est appliqué : $AD_i = \langle IPv6_prefixe, Hash_{64}(h^{k-1}(h^{n-k+1}(s_i))) \rangle$.

Pour l'authentification des messages, les données sont déjà connues de j (voir la section 3.3.4). Le test à effectuer est $h_i^{n-k+1}(h(M_{fix}), h(M_{var}))_{stocke} = h_i^{n-k+1}(h(M_{fix})_{stocke}, h(M_{var})_{stocke})_{calcule}$.

Idées et problèmes

Le protocole MATA permet d'authentifier simplement les messages et les entités. Il est à noter que la génération de n clés par nœud permet de diminuer l'utilisation d'une même clé dans le temps. Cela augmente donc la sécurité du protocole.

Toutefois, deux éléments sont assez contraignants et font que ce protocole ne peut pas être adapté à des réseaux grandes échelles.

Le premier problème est le besoin de synchronisation entre les différents nœuds. En effet, la gestion de clés, ou encore de l'authentification se base sur la détection de l'intervalle courant. Or, sur un réseau ad hoc grande échelle, la synchronisation est assez complexe (quelques exemples sont cités en 3.3.3) et coûteuse.

Le deuxième problème est la détermination d'un *max*, définissant la durée maximale d'existence du réseau. Pour des réseaux visant à offrir des services limités ou temporaires, cela convient parfaitement. Mais on voit ici que différentes utilisations (par exemple les réseaux de capteurs) ne pourront pas supporter ce protocole.

Un problème commun à de nombreux protocoles est que, premièrement, ils ne se basent

⁹Supervision on ROute, mécanisme de supervision utilisé dans MATA, avec également SUNE, Supervision on NEighborhood

pas sur une modélisation du protocole de base à sécuriser et que, deuxièmement, ils ne sont pas eux-mêmes modélisés de manière formelle (du moins pas tous) avec des mécanismes comme BAN¹⁰. La méthode NOMAD, présentée dans la section suivante, permet de spécifier de manière formelle les politiques de sécurité et peut résoudre ces problèmes de manque de formalisme.

3.3.5 Nomad : spécification formelle de la sécurité

Présentation succincte

La méthode Nomad [15, 6] permet de décrire des politiques de sécurité. En se basant sur des notions d'actions et de deadlines, il est possible de décrire différents protocoles. Cela permet d'avoir une possibilité d'expression importante au niveau des protocoles réseau, notamment par rapport à différentes méthodes se basant sur les notions de structures, d'enchaînements et d'objets. En effet, dans les réseaux, différentes notions se basent sur le temps. Un exemple très simple est la gestion des timers (par exemple dans TCP, pour la réémission de paquets).

Afin d'introduire les possibilités d'expression offertes par Nomad, et son utilisation dans OLSR, voici quelques rappels sur les bases de Nomad.

Nomad se base premièrement sur les actions temporisées.

$\bigcirc^d A$ se lit A sera vrai dans d unités de temps.

$\bigcirc^{<=d} A$ se lit A sera vrai avant d unités de temps.

Viennent ensuite les notions de privilèges :

$O.A$, $F.A$ et $P.A$ définissent respectivement les aspects Obligatoire, Interdit (Forbidden) et autorisé (Permitted).

$\square.A$ signifie que A est nécessaire, $\diamond.A$ signifie que A est possible.

Ces données permettent de spécifier différentes notions propres à l'aspect sécurité d'un protocole. Comme indiqué dans ce même rapport [6], la notion de *Maximum Waiting Time* (MWT) est essentielle pour éviter des cas de famine, ce qui peut engendrer par la suite des atteintes à la disponibilité. MWT peut être spécifié de la manière suivante : $O(\bigcirc^{<=d} \text{start}(\alpha) \mid \text{req}(\alpha))$, qui se traduit par l'obligation d'exécuter α dans moins de d unités de temps après la requête.

Différents exemples sont disponibles dans [16], avec notamment une spécification formelle des propriétés de TCP.

Intégration dans un protocole

La spécification avec Nomad permet d'exprimer des propriétés de sécurité, sans se soucier de la partie programmation proprement dite du protocole. Dans ce cadre, Nomad peut être utilisé de deux manières : identifier les besoins d'un nouveau protocole pour assurer l'aspect sécurité ou augmenter la sécurité d'un protocole existant.

Il existe une approche permettant de relier ces aspects sécurité à l'aspect programmation. Il s'agit du weaving, à rapprocher de la méthode Aspect-Oriented Programming (AOP).

¹⁰Une description de la méthode BAN est disponible sur http://en.wikipedia.org/wiki/BAN_logic

Cela permet d'élaborer la partie aspect et la partie fonctionnelle de manières séparées, pour ensuite les fusionner. On trouve facilement différents outils pour exploiter le weaving, adaptés à des langages de programmation (par exemple AspectJ, pour la programmation en Java). Une explication détaillée de l'AOP et du weaving¹¹ est donné dans [4].

3.3.6 Utilisation de Nomad dans OLSR

Au niveau de l'équipe SERES de l'ENST Bretagne, les prédicats de Nomad appliqués à OLSR ont déjà été définis. Ceux-ci ont été spécifiés en fonction des différents types d'attaques envisagées sur un réseau supportant ce protocole.

Supposons les prédicats de base suivants :

RECEIVE(na, nb, m) : le nœud nb reçoit de na le message m

NEIGHBOR(nb, nc) : nb peut émettre directement à nc

MPR_NEIGHBOR(na, nb) : nb est le nœud MPR de na

Dans le protocole OLSR, un nœud coopératif au niveau du routage est un nœud jouant correctement son rôle de routeur, autrement dit de MPR_NEIGHBOR. Cela veut dire que lorsqu'un nœud nb reçoit un message d'un nœud na , na étant l'un de ses MPR_SELECTOR (na a choisi nb comme MPR_NODE), il doit le réémettre à ses voisins dans un délai inférieur à t_{max} . Cela est décrit de la manière suivante :

$$\text{COOPERATIVE}(nb) \Leftrightarrow \Box(\text{RECEIVE}(na, nb, m) \wedge \text{NEIGHBOR}(nb, nc) \wedge \text{MPR_NEIGHBOR}(na, nb) \Rightarrow \bigcirc^{<=t_{max}} \text{PROPAGATE}(nb, nc, m)).$$

Un nœud égoïste, n'assure jamais les techniques de routage, et donc ne propage jamais les messages alors qu'il devrait le faire. L'expression EGOIST permet de spécifier ce cas (à noter que ce n'est pas non plus le contraire de COOPERATIVE, EGOIST ne transmet jamais, alors qu'un nœud non COOPERATIVE ne transmet pas tout le temps dans le pire des cas).

$$\text{EGOIST}(nb) \Leftrightarrow \Box(\text{RECEIVE}(na, nb, m) \wedge \text{NEIGHBOR}(nb, nc) \wedge \text{MPR_NEIGHBOR}(na, nb) \Rightarrow \neg \text{PROPAGATE}(nb, nc, m)).$$

Un troisième exemple va nous permettre de décrire les nœuds menteurs : un nœud menteur génère des informations incorrectes et les transmet sur le réseau. Il s'agit par exemple de la propagation des messages autres que TC.

$$\text{SLANDERER}(nb) \Leftrightarrow \neg \text{TC}(na, nb, m) \wedge \text{NEIGHBOR}(nb, nc)$$

$$\wedge \text{MPR_NEIGHBOR}(na, nb) \wedge \text{PROPAGATE}(nb, nc, m)$$

$$\vee$$

$$\neg \text{HELLO}(na, nb, m) \wedge \text{NEIGHBOR}(nb, na) \wedge \text{SEND}(nb, nc, m)$$

A partir de ce modèle formel, il est donc possible de décrire différentes actions malveillantes (nœuds malveillants, égoïstes), comme indiqué dans [3].

¹¹ voir également http://en.wikipedia.org/wiki/Aspect-oriented_programming

4 Conclusion

4.1 Bilan

Comme nous l'avons dit précédemment, les techniques proposées actuellement pour les réseaux ad hoc sont principalement dédiées aux deux protocoles retenus comme RFC, à savoir OLSR et AODV. Ces méthodes résolvent certaines parties des besoins en terme de sécurité, mais aucune d'entre elles ne propose une solution satisfaisant toutes les propriétés introduites dans la section 3.1, celles-ci étant délicates à satisfaire en totalité.

Toutefois, ces solutions proposent différentes méthodes à caractère générique (authentification par preuve, mécanismes de réputation, ...), lesquelles pourront être réutilisées dans d'autres propositions, soit pour ces mêmes protocoles, soit pour d'autres protocoles peu présents dans la littérature actuelle.

4.2 Sujets du Stage

Actuellement, nous avons identifié plusieurs sujets d'étude envisageables pour mon stage. Les trois premiers axes ne sont pas totalement indépendants et les résultats obtenus par chacun des axes permettra d'apporter une bonne sécurisation des réseaux de ce type. Selon l'avancement du stage, on choisira d'explorer tout ou partie de ces axes ainsi que le niveau d'investigation à leur accorder.

Authentification et réputation

Comme nous l'avons vu, il existe actuellement différentes approches pour évaluer la réputation des nœuds au sein d'un réseau. De ce fait, il serait possible d'utiliser les autres nœuds pour pouvoir analyser plus finement le caractère éventuellement malhonnête d'un nœud.

Toutefois, il est évident qu'il faut dans ce cas mettre en place un mécanisme d'authentification, pour éviter des usurpations d'identités.

L'utilisation et l'analyse des informations recueillies par les différents nœuds du réseau nous permettra d'avoir une vue globale du réseau, ce qui est évidemment le mieux pour pouvoir prendre des bonnes décisions au niveau d'un nœud.

Gestion de groupes de confiance et cryptographie

Les mécanismes de cryptographie permettent d'assurer la confidentialité des informations. Il est ainsi possible de chiffrer toutes les communications au sein d'un groupe, pour s'assurer que seuls les nœuds du groupe participent aux activités de base (routage, ...) du réseau.

De plus, dans ce cas, l'authentification est simple : les nœuds "malhonnêtes" sont ceux ne faisant pas parti du groupe.

Pour des soucis de passage à l'échelle (ou encore *scalability*), il faudra envisager la création de sous groupes avec des nœuds modérateurs. Dans ce cas, il faudra également étudier un

protocole fiable de désignation du modérateur, et de changement de modérateur quand ce sera nécessaire.

A noter que les problèmes actuels sont les surcharges provoquées par les mécanismes de chiffrement, notamment lors de l'utilisation de chaînes de certification.

Isolation des nœuds malveillants

La création de groupes, les mécanismes d'authentification et de réputation ont pour but de détecter au mieux les nœuds malveillants. Toutefois, après cette détection, il est nécessaire de réagir. Les actions évidentes sont l'isolation du nœud au niveau des différents mécanismes de gestion de services (routage, autorités de certification partielles, ...). Un problème est alors : comment propager l'information sur le fait qu'un nœud est malveillant ? Comment assurer la fiabilité de ce message ? Et, finalement, comment effectuer la réaction (et laquelle ?) proprement dite ?

On peut noter que ce troisième sujet exploite les deux précédentes études envisagées pour mon stage.

Analyse des protocoles hybrides

Actuellement, les différents travaux sur les réseaux ad hoc se basent sur la cryptographie ou sur les mécanismes de signature. De plus, les principaux protocoles étudiés sont OLSR et AODV, les 2 protocoles les plus stables. L'étude d'autres protocoles, en particulier ceux classifiés d'hybrides, pourra nous permettre de trouver une solution plus générique aux problèmes liés aux réseaux ad hoc en général, et non seulement à une catégorie de protocoles.

Références

- [1] Xiaoyun XUE et Jean Leneutre, *DRAFT : ad hoc network security state of the art*, 14 avril 2005
- [2] C. Perkins, E. Belding-Royer et S. Das, *ad hoc On-Demand Distance Vector (AODV) Routing*, juillet 2003
- [3] Seila Nuon, «Analyse de la disponibilité dans les réseaux ad hoc», *Rapport de master de recherche*, ENST Bretagne, juin 2006
- [4] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J-M. Longitier et J. Irwin, «Aspect-Oriented Programming», *European Conference on Object Oriented Programming*, 6-13 juin 1997
- [5] Stanislaw Jarecki et Anna Lysyanskaya, «Adaptively secure threshold cryptosystems without erasures», *Lecture Notes in Computer Science*, vol 1807, 2000, p221
- [6] Frederic Cuppens, Nora Cuppens-Boulahia et Tony Ramard, «Availability Enforcement by Obligations and Aspects Identification», *First International Conference on Availability, Reliability and Security (ARES'06)*, 20 avril 2006, pp. 229-239
- [7] M. Jiang, J. Li et Y. C. Tay, «Cluster Based Routing Protocol (CBRP)», *IETF Internet-Draft*, août 1999
- [8] Jungmin So et Nitin H. Vaidya, *A Distributed Self Stabilizing Time Synchronization Protocol for Multi-hop Wireless Networks*, rapport technique, UIUC, janvier 2004
- [9] Jinshan Liu et Valérie Issarny, «Enhanced Reputation Mechanism for Mobile ad hoc Networks», *Second International Conference on Trust Management (iTrust'2004)*, 1 avril 2004
- [10] Mario Gerla, UCLA, Xiaoyan Hong, UCLA et Guangyu Pei, «Fisheye State Routing Protocol (FSR) for ad hoc Networks», *IETF Internet-Draft*, 17 décembre 2001
- [11] IETF Secretariat, *Mobile Ad-hoc Networks (manet)*
- [12] Carlos H. Rentel et Thomas Kunz, *Network Synchronization in Wireless ad hoc Networks*, Department of Systems and Computer Engineering, Thèse de doctorat, Carleton University, Canada, juillet 2004,
- [13] T. Clausen et P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, RFC 3626, IETF, Project Hipercom, INRIA, octobre 2003,
- [14] J. Moy, *OSPF Version 2*, RFC 2328, IETF, avril 1998.
- [15] Frederic Cuppens, Nora Cuppens-Boulahia et Thierry Sans, «Nomad : A Security Model with Non Atomic Actions and Deadlines», *CSFW '05 : Proceedings of the 18th IEEE Computer Security Foundations Workshop*, 20 juin 2005, pp. 186-196

- [16] Frederic Cuppens, Nora Cuppens-Boulahia et Tony Ramard, «Property Based Intrusion Detection to Secure OLSR», *The Third International Conference on Wireless and Mobile Communications (ICWMC 2007)*, Guadeloupe, French Caribbean, 4-9 mars 2007, GET/ENST Bretagne
- [17] Jean-Pierre Hubaux, Levente Butty et Srdan Capkun, «The Quest for Security in Mobile Ad Hoc Networks», *MobiHoc '01 : Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, 2001, pp. 146–155
- [18] Lin Chen et Jean Leneutre, «Security of Routing Protocols in ad hoc Networks», *rapport technique interne*, Université de Pierre et Marie Curie, Août 2006,
- [19] Daniele Raffo, *Security Schemes for the OLSR Protocol for ad hoc Networks*, Thèse de doctorat de l'Université de Paris 6 - Pierre et Marie Curie, 15 Septembre 2005,
<http://perso.crans.org/raffo/papers/phdthesis/thesisap1.html>
- [20] Ghassan Oreiby, *Spécification et vérification du protocole OLSR*, ENST Bretagne et LaBRI, 22 juin 2006
- [21] Adam Stubblefield, John Ioannidis et Aviel D. Rubin, «A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP)», *ACM Transactions on Information and System Security (TISSEC)*, Vol 7, numéro 2, Mai 2004, p 319-332,