République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Ecole Nationale Polytechnique

Département d'électronique

Mémoire

présenté en vue de l'obtention du diplôme de Magister en Electronique

Option : Signal et communications

Thème :

Algorithme de clustering pondéré pour sécuriser un réseau ad hoc à la base d'une infrastructure à clé publique auto organisée.

Sous la direction de :

Pr. Berkani Daoud.

Réalisé par :

Chakhrit Abderrezek.

Soutenue le 07/06/2011 devant le jury composé de :

M'hania GUERTI,	*Professeur à l'ENP :*	*Président*
Latifa HAMAMI,	*Professeur à l'ENP :*	*Examinateur*
Mohamed MEHENNI,	*Professeur à l'ENP :*	*Examinateur*
Daoud BERKANI,	*Professeur à l'ENP :*	*Rapporteur*
Rabah SADOUN,	*Chargé de Cours à l'ENP :*	*Invité*

Dédicaces

C'est grâce à Dieu, clément et miséricordieux

Tout d'abord, ce travail est dédié à l'âme de mon père que dieu l'accepte dans son vaste paradis. A ma mère, que dieu la protège.

Je dédie également ce modeste travail à mes frères, mes sœurs et leurs familles.

Ma gratitude va également à mes chers amis : Tcouketch-kebir Soufiane, Bendifallah Ramzi, Laati Abdalla, Fanouh Lhachemi.

Remerciements :

Tout d'abord je tiens à remercier ***ALLAH*** *le bon Dieu tout puissant pour la santé, la volonté et la patience qu'il m'a données afin de réaliser ce travail.*

Mes sincères remerciements s'adressent à l'honorable encadreur Monsieur ***Berkani Daoud*** *pour m'avoir accordé sa confiance, pour son soutien et ses conseils.*

Mes sincères remerciements vont également au Professeur ***M'hania Guerti*** *pour avoir accepté de présider le jury de ce mémoire, ainsi que les Professeurs* ***Mohamed Mehenni, Latifa Hamami*** *et le maître de conférence* ***Rabah Sadoun*** *qui m'ont fait l'honneur de participer au jury et examiner ce travail.*

Enfin j'adresse mes vifs remerciements aux enseignants du département d'électronique, tout le personnel de l'ENP, mes amis ainsi que toute ma famille.

ملخص :

الشبكات الاسلكية تشهد حاليا نجاحا كبيرا .ولكن نظرا لخصائصها، هي أكثر عرضة للهجمات من الشبكات السلكية.
من خلال هذه المذكرة، هدفنا هو تعزيز أمن هذه الشبكات و توفير البنية الأمنية المناسبة لها.
الحل المقترح يبدأ من خلال تقسيم الشبكة إلى مجموعات، ثم تأتي خطوة تعيين قائد لكل مجموعة وخلال كل هذه المراحل تم إدخال نظام الأمن، الذي يستند على نظام التشفير بواسطة المفتاح العام المنظم ذاتيا، لمنع الهجمات من التسلل إلى الشبكة.
و لكي يتم تطبيق هذا الحل وتقييم أداءه نلجأ إلى المحاكاة. و نلاحظ بصفة خاصة أن هذا الحل يجعل من الممكن التقليل من أثر العناصر الضارة على نحو معتبر.
كلمات المفاتيح : شبكات ad hoc، IEEE 802.11، أمن الشبكات، أنظمة التشفير، والهجمات على الشبكات.

Résumé :

Les réseaux ad hoc connaissent actuellement un grand succès. Mais en raison de leurs caractéristiques, ils sont plus vulnérables aux attaques qu'aux réseaux filaires.
Dans ce mémoire, notre objectif consiste à proposer une architecture de sécurité adaptée à ces réseaux.
La solution proposée se fait initialement par la division d'un réseau ad hoc en groupes, ensuite vient l'étape de désignation d'un chef pour chaque groupe et durant toutes ces phases, le système de sécurité, qui est basé sur une PKI auto organisée, est introduit pour empêcher les attaques de s'infiltrer au sein du réseau.
L'application de la solution est décrite et ses performances sont évaluées par la simulation. On observe notamment que la solution permet de diminuer l'impact des nœuds malveillants d'une façon notable.
Mots-clés : réseaux ad hoc, norme IEEE 802.11, sécurité, clustering, infrastructure à clé publique auto organisée, mobilité, attaques sur réseaux ad hoc.

Abstract:

Ad hoc networks are currently experiencing great success. But because of their characteristics, they are more vulnerable to attacks than wired networks.
In this paper, our goal is to provide security architecture appropriate for those networks.
The proposed solution is initially by the division of an ad hoc network in groups, and then comes the step of appointing a leader for each group and during all these phases, the security system, which is based on a PKI self organized, is introduced to prevent attacks from infiltrating the network.
The application of the solution is described and its performance is evaluated by simulation. We observe in particular that the solution reduces the impact of malicious nodes significantly.
Key words: ad hoc networks, standard IEEE 802.11, security, clustering, infrastructure with public key self organized, mobility, attacks on ad hoc networks.

Sommaire :

Chapitre 02 : Présentation des réseaux Ad Hoc

Chapitre 03 : Approche de sécurité pour les réseaux ad hoc

Chapitre 04 : Développement et performances.

Chapitre 05 : Evaluation des résultats.

Liste des figures :

Liste des acronymes

AES:	Advanced Encryption Standard
AODV:	Ad hoc On Demand Distance Vector
AP:	Access Point
BSS:	Basic Service Set
CA:	Certification Authority
CMU:	Carnegie Mellon University
DDOS:	Distribute DOS
DOS:	Denial Of Service
DES:	Data Encryption Standard
DMZ:	Demilitarized Zone
DSDV	Destination Sequenced Distance-Vector Routing
DSR	Dynamic Source Routing Protoco
ESS:	Extended Service Set
FTP :	File Transfer Protocol
GSM :	Global System for Mobile communications
IDS :	Intrusion Detection System
IEEE :	Institute of Electrical and Electronics Engineers
IP :	Internet Protocol
ISO:	International Standards Organization
LOWEST-ID:	Lowest-Identifier
MAC:	Medium Access Control
MAC:	Message Authentication Code
MANET:	Mobil Ad hoc Network
MD5:	Message Digest algorithm
MOBIC:	Lowest Relative Mobility Clustering Algorithm
MPR :	Multi Points Relays
MRDM :	Modified Random Direction Model
MRL :	Message Transmitted List
NAM :	Network Animator
NPDU :	Network Protocol Data Unit
NS:	Network Simulator

OLSR	Optimized Link State Routing
OTCL:	Object Tool Command Language
PGP:	Pretty Good Privacy
PKI:	Public Key Infrastructure
RDM:	Random Direction Model
RFC :	Requet for Comments
RSA:	Rivest Shamir Adleman
RWM:	Random Waypoint Model
SHA:	Secure Hash Algorithm
SSL/TLS:	Secure Socket Layer/Transport Layer Security
TC:	Topology Control
TCL:	Tool Command Language
TCP:	Transmission Control Protocol
TESLA:	Time Efficient Stream Loss-tolerant Authentication
TORA	Temporally Ordered Routing Algorithm
WIFI:	Wireless Fidelity
WLAN:	Wireless Local Area Network
WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

Introduction générale :

Les réseaux et les systèmes de transmission de données sans fil sont devenus en peu de temps un moyen courant pour effectuer certains types d'opérations et leur technologie n'a cessé de croître grâce aux développements technologiques dans divers domaines liés à la microélectronique. En plus, avec l'émergence des Réseaux sans fil de la norme 802.11 (WiFi : Wireless Fidelity), de nouvelles thématiques ont été ouvertes et de nouveaux défis ont vu le jour pour répondre aux besoins des personnes et aux exigences de plusieurs domaines d'application (industriel, culturel, environnemental) : Opérations de secours en cas de catastrophe, les applications militaires, les missions d'exploration, l'enseignement à distance, systèmes pour surveiller les conditions de santé et le lieu de séjour des patients, etc.

La norme IEEE 802.11 a défini deux catégories différentes pour les réseaux sans fil : Réseaux sans fil avec infrastructure et réseaux sans fil sans infrastructure appelés réseaux ad hoc mobiles (MANET).

La construction des réseaux sans fil avec infrastructure est basée sur les réseaux filaires actuels. Cette architecture consiste au minimum en un point d'accès (AP) connecté à l'infrastructure du réseau filaire et un ensemble de postes réseaux sans fil. Cette configuration est baptisée Basic Service Set (BSS, ou ensemble de services de base). Un Extended Service Set (ESS, ou ensemble de services étendu) est un ensemble d'au moins deux BSS formant un seul sous réseau.

La deuxième catégorie qui représente les réseaux ad hoc est caractérisée par un ensemble d'entités liées entre elles par des liaisons à base d'ondes radio sans infrastructure fixe ni administration centralisée. Chaque nœud joue le rôle d'hôte ou de routeur à un instant donné. L'interconnexion de tous les nœuds mobiles forme une topologie temporaire et dynamique qui se déploie aisément.

Vu de leurs caractéristiques particulières, les réseaux ad hoc font l'objet de plusieurs travaux de recherche visant à améliorer les mécanismes de communications de ces réseaux en terme de protocoles de routage, de gestion et conservation de l'énergie au niveau des entités mobiles, de qualité de service, ainsi que les problèmes liés à la sécurité des communications et le transfert des données.

En ce qui concerne la sécurité des systèmes sans fil ad hoc, cette dernière n'a jamais cessé de susciter des préoccupations du fait que ces systèmes sans fil sont exposés à des menaces supplémentaires par rapport aux systèmes filaires. En général, ces menaces viennent du fait que les communications sans fil sont transmises par ondes radios et peuvent être interceptées

clandestinement par des personnes non autorisées. Les réseaux sans fil peuvent par exemple être l'objet d'attaques de pirates à partir d'un stationnement ou d'une voiture, le pirate pénètre dans le rayon d'action du réseau et intercepte les transmissions de données.
Si les transmissions ne sont pas bien protégées, elles peuvent être lues, enregistrées et/ou modifiées. Et même si elles le sont, il existe des outils permettant de les analyser et de trouver les clés de chiffrement.

Pour ces raisons, la stratégie en matière de sécurité des systèmes sans fil a connu une évolution remarquable et des études ont été consacrées à la sécurisation des réseaux ad hoc, toutefois, les techniques proposées offrent des solutions partielles, et jusqu'à présent, les nombreux travaux traitant de la sécurité des réseaux ad hoc, s'articulent principalement selon les trois problématiques suivantes : la définition de modèles de confiance, la mise au point de mécanismes d'authentification et de gestion de clefs adaptés et la sécurisation des protocoles de routage ad hoc.

En effet, Les mécanismes de sécurité traditionnels, comme la signature digitale et le chiffrement à clé publique, reste toujours des outils essentiels pour garantir la sécurité dans les réseaux mobiles ad hoc. Ces mécanismes nécessitent un service de gestion de clés afin de garder une liaison entre une clé et un noeud (authentifier les clés utilisées), et d'établir une confiance entre les noeuds du réseau. Généralement, le service de gestion de clé était basé sur une entité digne de confiance, appelée autorité de certification CA (Certification authority) qui doit créer un certificat de clé publique à chacun des nœuds du réseau. La CA digne de confiance doit être en ligne pour traiter les cas de révocation et de renouvellement des certificats de clés publiques. Cependant il est dangereux d'installer un service de gestion de clés en utilisant une seule CA dans un réseau ad hoc. Car si cette unique autorité de certification CA est compromise, la sécurité de tout le réseau est brisée.

Pour remédier à cet inconvénient, nous sommes intéressés dans ce travail, pour sécuriser les réseaux ad hoc, à les partitionner en groupes (appelés aussi clusters), et dans chaque cluster il y a un chef (cluster-head) qui est élu suivant un ensemble de métriques. Chaque cluster-head dans le réseau est entouré par des nœuds possédants un poids le plus élevé (qui représente le degré de confiance et est égal à celui de cluster-head), à ces nœuds on attribue le rôle d'un CA pour qu'ils puissent effectuer le contrôle sur les autres nœuds qui ont un niveau de confiance bas. Et pour établir un système de gestion de clé et d'attribution des certificats nous avons fait appel à une méthode développée par [Hub01], son principe est d'offrir une infrastructure de gestion des clés publiques auto organisée dans le sens où les certificats sont délivrés par les utilisateurs eux-mêmes, en se basant sur leurs connaissances personnelles et

sans l’implication d’aucune entité centralisée. Dans cette architecture les paires de clés privée/publique sont créées localement au niveau des nœuds, en suite les nœuds sauvegardent les certificats des clés publiques qui sont délivrés par les autres noeuds CA dans un répertoire local pour les utiliser dans le processus d’authentification. Donc, avec ce mécanisme une relation de confiance peut être mise en place entre les nœuds du réseau. En général, quand deux noeuds veulent s’authentifier, ils essayent de trouver une chaîne de certificat en combinant leurs répertoires locaux. Si une chaîne de certificat existe, le noeud est authentifié.
L’avantage de cette solution est qu’elle ne nécessite aucune forme d’infrastructure centralisée. La gestion des clés et des certificats est effectuée par les nœuds eux mêmes.

Objectif et Organisation du mémoire :

L’objectif de ce mémoire est de traiter les problèmes de la sécurité dans les réseaux mobiles ad hoc. A cette fin un algorithme appelé (Algorithme de clustering Pondéré pour sécuriser un réseau ad hoc à base d’une infrastructure à clé publique auto organisée) est proposé, ce dernier permet d’organiser les nœuds d’un réseau en clusters et de confier aux nœuds dont le poids est le plus élevé la responsabilité de l’autorité de certification et de contrôle des nœuds qui possèdent un comportement malveillant. L’élection des cluster-heads se fait périodiquement selon leur poids qui est fonction de leur degré de confiance et de leur mobilité afin de supporter cette tâche. Nous avons introduit ces facteurs dans le calcul des poids pour générer des clusters stables en présence d’une topologie temporaire et dynamique. Au niveau de la gestion des clés, le rôle d’attribution des certificats à ces clés est partagé entre plusieurs nœuds dignes de confiance afin d’éviter les dangers liés à l’utilisation d’une seule CA.
Pour tester les performances de cet algorithme, une série de simulations est réalisée en utilisant le simulateur NS (Network Simulator).

L’organisation générale de ce mémoire s’articule autour de cinq chapitres.
Dans Le premier chapitre de ce manuscrit nous présenterons une vue générale sur la sécurité des systèmes d’information SI, les contraintes et les caractéristiques liées à ce domaine, ainsi que différentes solutions proposées pour les SI et nous expliquerons quels en sont les défis.

Le deuxième chapitre est une étude bibliographique sur les réseaux ad hoc. Dans cette étude, nous définirons précisément ce qu'est pour nous un réseau sans fil ad hoc, nous en donnerons les principes fondamentaux, les propriétés et les protocoles que doivent suivre de telles structures, nous allons présenter également les challenges auxquels est confrontée la

sécurité de ces réseaux, et aussi les différentes approches proposées dans la littérature pour les sécuriser.

Le troisième chapitre s'intéresse à présenter l'approche proposée. Tout d'abord nous présenterons le principe de la méthode de gestion des clés et l'attribution des certificats, en suite nous exposerons le modèle de partitionnement en cluster et présenterons les métriques d'élection des cluster-heads et enfin l'introduction de quelques types d'attaques pour tester la robustesse de l'algorithme.

Dans le quatrième chapitre plus de détails sont présentés, en ce qui concerne l'implémentation de l'algorithme au niveau du simulateur et le développement des différentes procédures de l'application, ainsi que la manière suivant laquelle les attaques sont introduites dans le réseau pour tester les performances de la solution.

Le cinquième chapitre présente une série de tests des performances notamment la résistance contre les attaques et la stabilité des groupes formés dans l'architecture proposée.

chapitre 1

Notions générales sur la sécurité des réseaux informatiques

Introduction :

La pérennité de toute Entreprise passe aujourd'hui par une disponibilité permanente de son système d'information. La continuité de l'activité de l'Entreprise appelle à la continuité de son système d'information. Cette continuité ne peut être assurée que par la mise en place de moyens de protection permettant d'apporter un niveau de sécurité adapté aux enjeux spécifiques de l'entreprise, ces derniers peuvent varier d'une entreprise à autre.

La sécurité des réseaux est donc devenue un des éléments-clés de la continuité des systèmes d'information de l'entreprise quelle que soit son activité, sa taille et sa répartition géographique. Sachant de plus que la sécurité informatique au sens large devient une problématique planétaire avec l'avènement de l'Internet, la maîtrise et la mesure de la sécurité des réseaux, devient une priorité majeure pour les opérateurs de télécommunications et leurs clients.

S’occuper de la sécurité d’un système signifie donc qu’il faut prendre en compte de nombreux facteurs, tout en poursuivant les objectifs majeurs du propriétaire de ce système.

Dans ce premier chapitre nous allons exposer les notions générales qui concernent la sécurité des données, en définissant les principaux objectifs de la sécurité des systèmes d’information, les défis ainsi que les différentes solutions proposées pour sécuriser un système d’information.

1- Définition :

La sécurité d’un système informatique a pour mission principale la protection des informations et des ressources contre toute divulgation, altération ou destruction. L’accès à ces ressources doit être également protégé et un accès autorisé à ces ressources ne doit pas être refusé [Arn08]. La sécurité informatique consiste à utiliser tous les mécanismes disponibles pour garantir les services de base qui sont :

- ***L’identification :*** L’utilisateur d’un système quelconque possède une identité, sorte de clé primaire d’une base de données qui détermine ses autorités d’usage.
- ***L’authentification :*** L’authentification a pour but de vérifier l’identité des processus communicants.
- ***La confidentialité :*** Elle désigne la garantie que les données échangées ne sont compréhensibles que par les deux entités qui partage le même secret.
- ***L’intégrité des données :*** Cette propriété consiste à prouver que les données n’ont pas été modifiées. Elles ont, éventuellement pu être copiées, mais aucun bit ne doit avoir été changé.
- ***La disponibilité :*** Est la propriété qu’une information ou un service doit être disponible quand une entité du système en a besoin [Guy04].

Du point de vue organisationnel, nous pouvons découper le domaine de la sécurité informatique de la façon suivante [Bac00]:

- La sécurité logicielle : gère la sécurité au niveau logiciel du système d'information (par exemple : l'intégration des protections logicielles comme des antivirus).
- La sécurité du personnel comprend la formation et la sensibilisation des personnes utilisant ou travaillant avec le système d'information.
- La sécurité physique regroupe : la politique d'accès aux bâtiments, la politique d'accès aux matériels informatiques, et les règles de sécurité pour la protection des équipements réseaux.
- La sécurité procédurale définit les procédures et les règles d'utilisation du système d'information.
- La sécurité réseau s'occupe de : l'architecture physique et logique du réseau, la politique d'accès aux différents services, la gestion des flux d'informations sur les réseaux, et surtout les points de contrôle et de surveillance du réseau.
- La veille technologique, souvent oubliée, permet d'évaluer la sécurité au cours du temps afin de maintenir un niveau suffisant de protection du système d'information.

2- Les dangers de manque de sécurité :

Un manque de sécurisation d'un service peut être un frein à une activité économique, et donc au développement de certaines entreprises. Par exemple, le commerce électronique se développe très lentement. La raison principale est que le paiement en ligne n'est pas encore totalement sécurisé. Même si des solutions efficaces commencent à être mises en place, l'idée qu'une transaction puisse être non sécurisée freine les consommateurs.

De nombreux autres risques guettent un système d'information non sécurisé, du simple dysfonctionnement à l'arrêt total de l'entreprise.

Sécuriser le système d'information d'une entreprise a un coût, mais deux entreprises de même taille ne réaliseront pas le même investissement dans ce secteur. Une façon de savoir quel budget une entreprise doit investir est d'estimer les pertes subies en cas d'immobilisation de son système d'information. Si le fonctionnement n'est presque pas altéré, il n'est pas nécessaire de fournir de gros efforts. Par contre, si l'entreprise est immobilisée, et si les pertes financières sont importantes, c'est que le système d'information est capital, et qu'il doit être protégé avec soin.

3- Risques pour la sécurité :

Avant d'investir le temps et l'argent dans la sécurité d'un réseau. On doit évaluer les risques pour la sécurité [Dea02]. En outre tout en examinant ces risques il faut tenir compte des effets qu'une menace peut causer à un réseau.

Toutefois, une faille de sécurité peut abîmer un réseau aussi facilement et rapidement. Pour comprendre comment gérer la sécurité d'un réseau, on doit d'abord reconnaître les types de menaces auxquelles le réseau est exploité. Tout en analysant chaque risque pour la sécurité, nous réfléchissons aux moyens de prévention possible, si cette menace s'applique, il faut évaluer les dommages qui pourraient en résulter.

Chaque organisation doit évaluer ses risques en effectuant un audit de sécurité c-à-d un examen détaillé de chaque aspect de réseau pour déterminer comment il peut être menacé.

3-1. Risques associés aux personnes :

Certaines estimations stipulent que l'erreur humaine est responsable de la moitié des failles de sécurité dans les réseaux.

Les risques associés aux personnes regroupent notamment les cas d'espèces où :

- Les intrus et les pirates obtiennent les mots de passe des utilisateurs par des tactiques de fouinage ou d'ingénierie sociale.
- Un administrateur crée ou configure incorrectement les codes d'identification d'utilisateurs, les groupes auxquels ils appartiennent ou les droit qui leurs sont alloués, causant des vulnérabilités d'accès au réseau.
- Mauvaise configuration du matériel ou logiciels du réseau par un administrateur.
- Des utilisateurs ou administrateurs choisissent des mots de passe facile à deviner.
- Des administrateurs oublient de supprimer les droits d'accès d'employés qui ont quitté l'organisation.
- Des utilisateurs notent leurs mots de passe sur des papiers et les placent à des endroits aisément accessibles.

3-2. Risques associés au matériel et à la conception du réseau :

Les risques suivants sont inhérents au matériel et à la conception d'un réseau :

- Les concentrateurs de réseaux diffusent le trafic à travers un segment entier, ce qui rend les transmissions beaucoup plus vulnérables. Par contraste, les commutateurs établissent des communications point à point, limitant ainsi la disponibilité des transmissions des données aux nœuds de transmission et de réception.

- Les ports inutilisés de concentrateurs, de routeurs ou de serveurs peuvent être exploités et interceptés par des pirates s'ils ne sont pas désactivés, un port de configuration de routeur par Telnet, peut ne pas sécuritaire.
- Si les routeurs ne sont pas configurés correctement pour masquer les sous réseaux internes, les utilisateurs des réseaux extérieurs (Internet) peuvent lire les adresses privées.
- Les modems rattachés aux périphériques d'un réseau peuvent être configurés pour accepter les appels entrants, créant ainsi des failles de sécurité s'ils ne sont pas protégés adéquatement.
- Les serveurs à accès entrant utilisés pour le télétravail ou par des employés distants peuvent être écoutés.
- Des ordinateurs traitants des données vitales peuvent coexister sur le même sous réseau que des ordinateurs ouverts au grand public.

3-3. Risques associés aux protocoles et aux logiciels :

Les systèmes d'exploitation réseaux et les logiciels d'application présentent différents risques. Dans la plupart des cas, leur sécurité et compromise suite à une mauvaise compréhension des droits d'accès aux fichiers ou par simple négligence dans la configuration du logiciel.

Voici quelques risques associés aux protocoles et aux logiciels de réseaux :

- Le protocole TCP/IP renferme plusieurs faiblesses de sécurité, les adresses IP peuvent être facilement falsifiées, UDP ne requiert aucune authentification et TCP n'emploie qu'une authentification rudimentaire.
- les implémentations du protocole TCP sont susceptibles d'être soumis à des attaques visant le déni de service (SYN flooding, Land attack, ping of death...).
- Les relations de confiance entre un serveur et un autre permettent potentiellement à un pirate d'accéder à un réseau entier par le biais d'un seul point faible.
- La plupart des applications et des systèmes d'exploitation réseaux sont vulnérables et comportent des failles en sécurité, à moins que l'administrateur ne procède à des mises à niveau régulières, un pirate peut aisément exploiter ces failles.
- Les administrateurs peuvent accepter les options de sécurité par défaut après avoirs installer un système d'exploitation ou une application. Souvent, les valeurs par défaut n'offrent pas une sécurité optimale.

- Les transactions qui s'accomplissent entre des applications, comme des bases de données et des formulaires au format WEB, sont susceptibles d'être interceptés.
- Une mauvaise configuration du système d'exploitation ou d'une application rendre le réseau plus vulnérable aux attaques, et ainsi provoque les pirates à exploiter ces faiblesses.

3-4. Risques associés à l'accès Internet :

Avec l'évolution de l'Internet et la naissance de la criminalité informatique, les menaces d'intrus extérieures sont bien réelles et ne font que croître.

En outre les entreprises et les particuliers doivent redoubler de prudence lorsqu'ils se branchent sur Internet.

Les pirates sont créatifs et ils se réjouissent de tramer de nouvelles manières aux systèmes, en conséquences de nouvelles menaces pour la sécurité sur Internet surgissent fréquemment.

Voici une liste des brèches de sécurités les plus courantes associées à l'Internet :

- Un firewall peut ne pas constituer une protection efficace, s'il n'est pas configuré adéquatement.
- Les attaques les plus courantes sont réalisées depuis l'Internet, on peut citer l'IP spoofing, le Syn-Flooding, l'espionnage….etc.
- Les navigateurs web les plus populaires contiennent par fois des bogues dans leurs versions les plus récentes, qui permettent à des scripts d'accéder à notre système pendant que nous sommes connectés à l'Internet.
- Les communications par FTP ou Telnet sur Internet, les codes d'identifications et les mots de passe sont transmis en texte clair sans aucun cryptage, quiconque surveille ce réseau peut intercepter ces identifications et mots de passe et les utiliser pour accéder au système.
- Si un utilisateur demeure connecté à une session de bavardage sur Internet, il est vulnérable aux attaques d'autres utilisateurs du Web.

4- Vulnérabilités et Types d'attaques :

La vulnérabilité peut être définie comme un défaut, connu ou suspecté, dans la conception d'un software, un hardware ou dans les opérations d'un système qui l'expose aux pénétrations ou à ce qu'il révèle certaines informations. Par contre, une menace est la possibilité de faire une tentative, non autorisée, pour accéder ou manipuler les données ou de rendre un système inutile ou non fiable.

De cet effet une attaque est l'exploitation des vulnérabilités d'un système informatique (système d'exploitation, logiciel…) à des fins non autorisées par le propriétaire du système [Arn08].

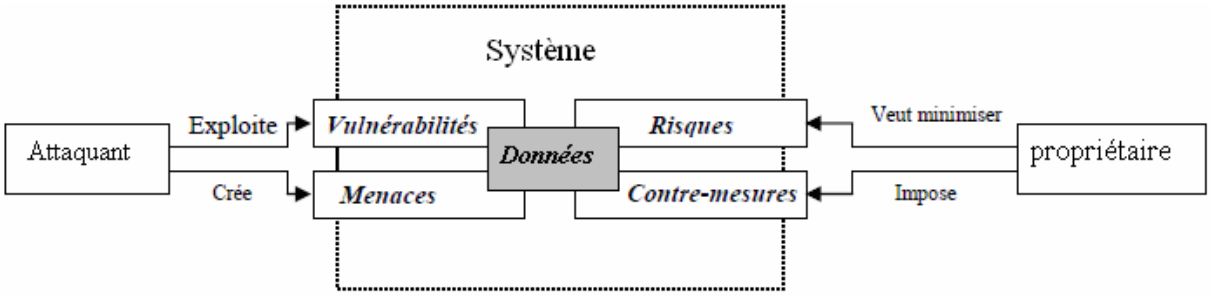


Fig.1 : Les relations entre le système, l'attaquant et le propriétaire.

Les motivations des attaques peuvent être de différentes sortes :

- Obtenir un accès au système pour en faire une machine capable d'attaquer d'autres machines (attaque par rebond) ;
- Voler des informations : secrets industriels, informations personnelles sur utilisateurs, récupérer des données bancaires, . . . ;
- empêcher le bon fonctionnement d'un service.

5- Types d'attaques :

Les attaques peuvent à première vue être classées en deux grandes catégories [55]:

- Les attaques passives : consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables mais une prévention est possible.
- Les attaques actives : consistent à modifier des données ou des messages, à s'introduire dans des équipements réseaux ou à perturber le bon fonctionnement de ce réseau. Notons qu'une attaque active peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables (permettant ainsi une réponse adéquate).

6- Profils et capacités des attaquants :

Les attaquants peuvent être classés non seulement par leurs connaissances (débutant, expert, etc.) mais également suivant leur capacité d'attaques dans une situation bien définie. Ainsi, on dénombre les capacités suivantes [Gra04] :

- Transmission de messages sans capacité d'écoute (IP spoofing);
- Interception et transmission de messages ;

- Interception et perturbation des communications (blocage de paquets, DoS (Denial of Service) et DDoS (Distribute DoS));
- Interception, perturbation et transmissions de messages;
- Interception et relais de messages (attaques type man-in-the-middle).

Une autre caractéristique des attaquants est leur contrôle unidirectionnel ou bidirectionnel sur les communications, du fait de la nature asymétrique de celles-ci. En effet, la plupart des canaux de transmissions sur Internet ou sur tout autre réseau hétérogène sont unidirectionnels et empruntent des chemins différents suivant les règles de routage. Ainsi, de nombreux protocoles de sécurité sont également unidirectionnels et il faut établir plusieurs canaux pour permettre un échange en "duplex". Ces canaux qui sont au nombre de deux minimums, sont la plupart du temps gérés de façon totalement indépendante par les protocoles de sécurité. C'est le cas pour SSL/TLS (Secure Socket Layer/Transport Layer Security) mais également pour IPSec (IP sécurisé) dont les associations de sécurité sont unidirectionnelles et indépendantes, chacune définissant son propre jeu de clés, algorithmes, etc.

7- Politique de sécurité :

Une politique de sécurité est l'ensemble des lois, des règles et pratiques qui régissent la façon dont l'information sensible et les autres ressources sont gérées, protégées et distribuées à l'intérieur d'un système spécifique [Céd05].

Les étapes suivies pour l'établissement d'une politique de sécurité sont :

- Identification des vulnérabilités
  - ✓En mode fonctionnement normal (définir tous les points faibles)
  - ✓En cas d'apparition de défaillances, un système fragilisé est en général vulnérable, c'est dans un de ces moments intermédiaires qu'une intrusion peut le plus facilement réussir.
- Évaluation des probabilités associées à chacune des menaces
- Évaluation du coût d'une intrusion réussie.
- Choix des contre mesures.
- Évaluation des coûts des contre mesure.
- Décision

8- Les Principales solutions de défense :

Actuellement, toute une série d'outils et de techniques permettent à un administrateur de sécuriser son réseau et les machines qui le composent. Chacune de ces techniques se base sur des principes fondamentalement différents, en assurant la sécurité des équipements et des

informations disponibles sur ce réseau, tout en tenant compte des contraintes de plus en plus présentes, telles que les interconnexions de réseaux, les besoins de « contacts électroniques » pour le personnel (mails, transferts de fichiers, accès Web, etc.), les systèmes d'informations complexes, et autres [Bac00].

Dans cette section, nous allons citer et expliquer brièvement quelques outils de sécurité courants :

8-1. La cryptographie :

La cryptographie a pour but d'assurer la sécurité des communications et des données stockées en présence d'un adversaire. Elle propose un ensemble de techniques permettant d'offrir des services de confidentialité, d'authentification et d'intégrité. La cryptologie, appelée aussi la Science du Secret, regroupe la cryptographie et la cryptanalyse [Fou01].

Cette discipline utilise de multiples variantes de chiffrement et de signature électronique. Par définition, un algorithme de chiffrement permet de coder des messages quelconques sous une forme inintelligible pour quiconque ne possède pas la clé de déchiffrement, Ces algorithmes de chiffrement se classifient en deux grandes familles [Mat07] :

8-1-1. Les algorithmes à clé secrète (algorithmes symétriques) :

Sont des algorithmes où la clé de chiffrement peut être calculée à partir de la clé de déchiffrement ou vice versa. Dans la plupart des cas, la clé de chiffrement et la clé de déchiffrement sont identiques. Pour de tels algorithmes, l'émetteur et le destinataire doivent se mettre d'accord sur une clé à utiliser avant d'échanger des messages. Cette clé doit être gardée secrète. La sécurité d'un algorithme à clé secrète repose sur la clé : si elle est dévoilée, alors n'importe qui peut chiffrer ou déchiffrer des messages avec celle-ci. Le schéma 2 illustre le principe de chiffrement à base d'algorithme à clé secrète. D'une manière générale, les algorithmes à clés secrètes sont très rapides car ils consistent tout simplement à réarranger les bits d'un message.

Le plus connu des crypto systèmes à clé symétrique est sans doute Data Encryption Standard (DES) introduit en 1977 avec une taille de clé de 56 bits ainsi que son successeur Advanced Encryption Standard (AES), offrant la possibilité d'utiliser des clés de 128, 192 ou 256 bits [Arn08].

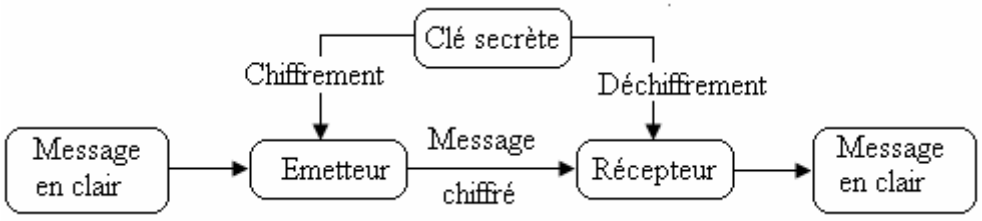


Fig.2 Chiffrement et déchiffrement avec une clé (Algorithme symétrique)

8-1-2. Les Algorithmes à clé publique (algorithmes asymétriques) :

Ils sont conçus de telle manière à ce que la clé de chiffrement soit différente de la clé de déchiffrement. De plus, la clé de déchiffrement ne peut pas être calculée (du moins en un temps raisonnable) à partir de la clé de chiffrement. De tels algorithmes sont dits “à clé publique” parce que la clé de chiffrement peut être rendue publique : n’importe qui peut utiliser la clé de chiffrement pour chiffrer un message mais seul celui qui possède la clé de déchiffrement peut déchiffrer le message chiffré résultant. Dans de tels systèmes, la clé de chiffrement est appelée clé publique et la clé de déchiffrement est appelée clé privée. Le schéma 3 illustre le principe de chiffrement à base d’algorithme à clé publique.

Les algorithmes à clé publique reposent tous sur des problèmes mathématiques qui ne peuvent, à l’heure actuelle, être résolus en temps polynomial (factorisation de nombres premiers, problème du logarithme discret,. . .). Par conséquent, ils sont beaucoup plus lents que les algorithmes symétriques. Le cryptosystème le plus couramment rencontré est RSA [Riv78].

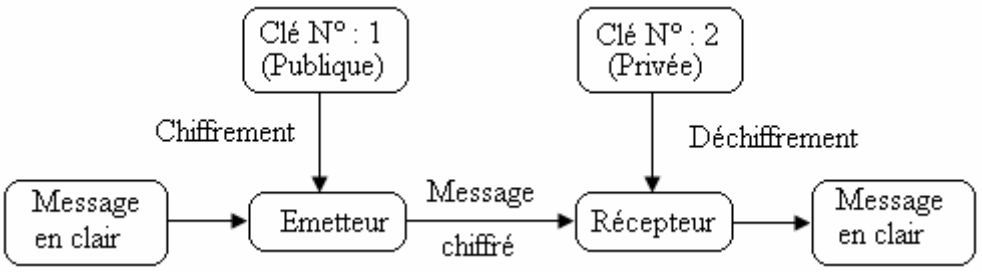


Fig.3 Chiffrement et déchiffrement avec deux clés (Algorithme asymétrique)

8-1-3. Protocole hybride :

Dans certains cas, il peut être judicieux de recourir à des techniques de chiffrement utilisant à la fois des techniques de cryptographie asymétrique et symétrique. L’un des cas le plus courant consiste à échanger une clé de session symétrique à l’aide de la cryptographie asymétrique. Ainsi seul l’établissement de la clé de session sera coûteux en temps.

En 1976, Whitfield Diffie et Martin Hellman ont proposé un algorithme hybride connu sous le nom de ses inventeurs Diffie-Hellman. Il permet à deux entités de se mettre

d'accord sur une clé secrète en communiquant à travers un canal de communication non sécurisé. Une fois cette clé secrète connue par les deux entités, elles peuvent l'utiliser et chiffrer leurs communications en utilisant les techniques cryptographiques usuelles.

8-1-4. Signature numérique :

Un des plus gros apports de la cryptographie à clés publiques est celui des méthodes de signature numérique. Les signatures numériques permettent au destinataire d'un message de vérifier l'authenticité de l'origine de ce message, et également de vérifier l'intégrité du message, c'est à dire de s'assurer qu'il n'a pas été modifié. Les signatures numériques permettent également d'assurer la non-répudiation d'un message, c'est à dire de faire en sorte que l'émetteur d'un message ne puisse pas nier l'avoir émis. Les fonctions d'intégrité, d'authenticité et de non-répudiation sont fondamentales en cryptographie. La signature numérique est nettement supérieure à la signature manuelle, car s'il est facile d'imiter une signature manuelle, il est en revanche presque impossible de contrefaire une signature numérique [Arn08], [Phi06]. L'émetteur du message utilise sa clé privée pour chiffrer le message (chaque identité dispose donc d'une clé de signature, privée, et d'une clé de vérification, publique). Le message chiffré obtenu fait alors office de signature. L'émetteur envoie à la fois le message en clair et sa signature au destinataire. L'opération de vérification de signature consiste à utiliser la clé publique de l'émetteur pour déchiffrer la signature (le message chiffré) et de comparer le résultat au message original en clair. Si les deux messages sont identiques, alors le destinataire est assuré de l'intégrité et de l'authenticité de message. De plus, l'émetteur ne peut pas nier avoir signé le message car il est le seul à posséder sa clé de signature (il s'agit d'une clé privée).

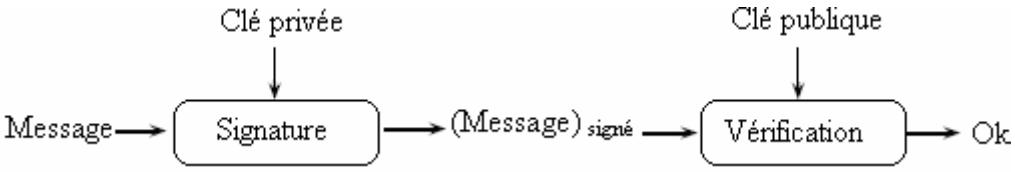


Fig.4 Principe de Signature numérique

8-1-5. Codes d'authentification de message :

L'équivalent symétrique de la signature est constitué par les codes d'authentification de message (ou encore MAC pour message authentication codes) (Fig. 5).

Les clés de vérification et de calcul d'un code d'authentification se confondent alors en une même clé privée [Mat07].

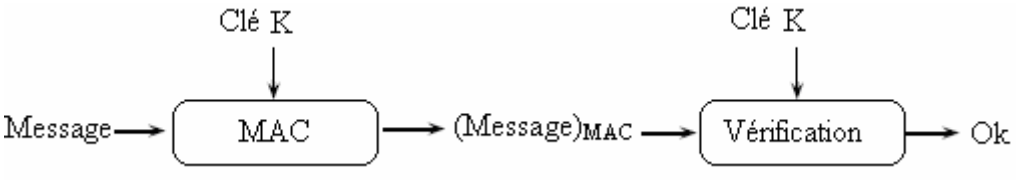


Fig.5 Code d'authentification de message (MAC)

8-1-6. Fonctions de hachage :

Le principe de signature exposé dans le paragraphe précédent présente l'inconvénient d'être coûteux en temps de calcul et en ressources. La signature obtenue est en effet relativement longue à calculer car les algorithmes de cryptographie à clés publiques utilisent des fonctions mathématiques complexes. Les fonctions de hachage apportent la solution à ce problème. Il s'agit d'une fonction mathématique à sens unique qui convertit une chaîne de caractères de longueur quelconque en une chaîne de caractères de taille fixe (souvent de taille inférieure) [Phi06]. Le résultat d'une telle fonction est appelé empreinte, somme de contrôle ou condensé. La solution consiste donc à calculer dans un premier temps l'empreinte du message, puis de ne signer uniquement que cette empreinte. Ce procédé est beaucoup plus rapide puisque l'empreinte est de taille beaucoup plus réduite. L'opération de vérification de signature consiste alors à déchiffrer l'empreinte jointe au message, puis de recalculer l'empreinte du message original et de la comparer avec celle qui vient d'être déchiffrée [Arn08]. Une fonction de hachage doit être résistante aux collisions, c'est-à-dire que deux messages distincts doivent avoir très peu de chances de produire la même signature. On considère l'utilisation d'une fonction de hachage en cryptographie si les conditions suivantes sont remplies :

- il est techniquement difficile (sur le plan matériel ou algorithmique) de trouver le contenu du message à partir de la signature (attaque sur la première préimage) ;
- à partir d'un message donné et de sa signature, il est très difficile de générer un autre message qui donne la même signature (attaque sur la seconde préimage) ;
- il est très difficile de trouver deux messages aléatoires qui donnent la même signature (résistance aux collisions).

Les algorithmes Secure Hash Algorithm 1 (SHA-1) et Message-Digest algorithm 5 (MD5), sont des fonctions de hachage utilisées fréquemment [Phi06].

8-2. Firewall :

Une définition du Firewall a été proposée par Cheswick et Bellovin [Bel00] dans leur ouvrage « Building Internet Firewalls » :

Un Firewall est un élément ou un ensemble d'éléments placé entre deux réseaux et possédant les caractéristiques suivantes :

- Tous les flux (entrant et sortant) passent au travers
- Seuls les flux autorisés par une politique locale peuvent passer
- Le système lui-même est résistant aux agressions.

Généralement un firewall (appelé aussi pare-feu, garde-barrière ou coupe-feu) est un élément matériel et/ou logiciel destiné à protéger un ordinateur ou un réseau d'ordinateurs des attaques provenant d'un réseau tiers (notamment Internet).

Sur un réseau un firewall protége à la fois les utilisateurs et les données situées sur le réseau local en deçà du monde impitoyable de l'Internet en se plaçant du point de vue du réseau interne d'une organisation, on peut qualifier d'extranet tout ce qui se situe à l'extérieur de ce réseau. Un firewall sert également à empêcher les utilisateurs du réseau local de se connecter à des sites interdits et permet de compartimenter les réseaux internes.

L'architecture des Firewalls la plus utilisée actuellement est basée sur une

«Zone démilitarisée », communément appelée DMZ (Demilitarized Zone). Elle consiste à placer un réseau intermédiaire entre l'accès Internet et le réseau interne (éventuellement plusieurs). Cette DMZ sera isolée, aussi bien vis-à-vis de l'Internet que du réseau local, par des systèmes de filtrage. Ensuite, les éventuels serveurs nécessaires à l'entreprise devant continuer à être accessibles de l'extérieur seront connectés directement sur cette DMZ, de manière à les séparer du réseau interne.

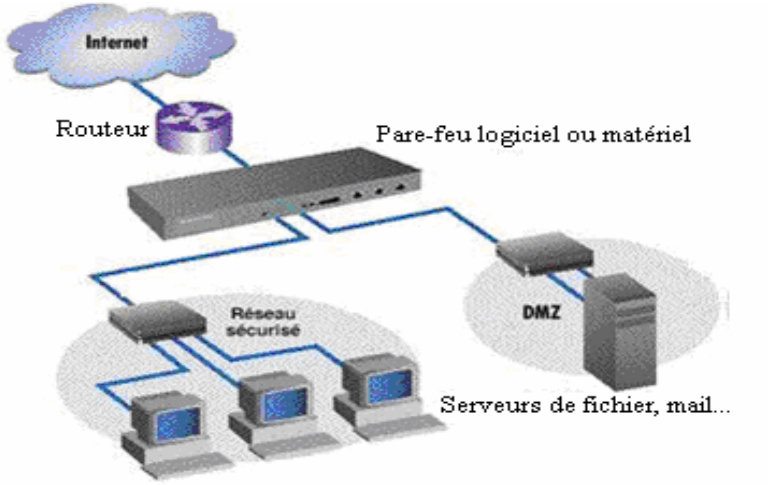


Fig.6 un réseau utilisant un firewall avec une zone DMZ [Blo07]

8-2-1. Fonctionnement de firewall :

Le filtrage est la fonction principale du firewall, il analyse les entêtes des paquets qui passent au travers et donc d'autoriser ou d'interdire l'accès de tel ou tel paquet entrant ou

sortant et ainsi de filtrer le trafic entre les différentes zones auxquelles il est connecté [Gou06]. Le Filtrage IP permet d'interdire l'accès non autorisés au réseau sans gêner les accès autorisés. Le Filtrage est fait à l'aide des règles définies par l'administrateur du firewall selon la politique de sécurité adoptée qui dépend des besoins de sécurité de l'entreprise mais aussi des capacités du firewall. De cette manière les données ne peuvent passer que si les règles du firewall les autorisent.

Deux politiques peuvent être appliquées aux firewalls [Den07] :

- Tout ce qui n'est pas explicitement interdit est autorisé : cette stratégie est dangereuse à utiliser d'un point de vue de la sécurité. En effet, par défaut, tous les services TCP/IP sont autorisés. De la sorte, il suffit qu'à l'installation d'un nouveau service, l'officier de sécurité oublie de modifier la configuration du pare-feu pour que le pare-feu devienne une vraie passoire.
- Tout ce qui n'est pas explicitement permis est interdit : pour une sécurité stricte cette stratégie est très recommandée que la première et est souvent implémentée par défaut.

Le filtrage peut être effectué à plusieurs niveaux : le filtrage simple de paquet (stateless), le filtrage à état (stateful) ou le filtrage applicatif.

8-3. Systèmes de détection d'intrusions :

La détection d'intrusion est le processus qui consiste à surveiller les évènements se produisant dans un ordinateur ou dans un réseau informatique, et de les analyser pour découvrir des signes d'intrusions, définies comme des tentatives de compromission de la confidentialité, l'intégrité, la disponibilité et la responsabilité, ou pour dévier des mécanismes de sécurité [Int89].

Les intrusions sont provoquées par : l'accès d'attaquants externes aux systèmes via des réseaux ouverts comme Internet, des utilisateurs autorisés qui essayent de gagner des privilèges additionnels pour lesquels ils ne sont pas autorisés, ou des utilisateurs autorisés qui abusent de leurs privilèges [Bac00].

Les systèmes de détection d'intrusions (IDS : Intrusion Detection System) sont les systèmes logiciels ou matériels qui automatisent cette tâche de surveillance et d'analyse.

L'auteur de [Wes00] simplifie le système de détection d'intrusion dans un détecteur qui analyse les informations en provenance du système surveillé (voir la Fig.7).

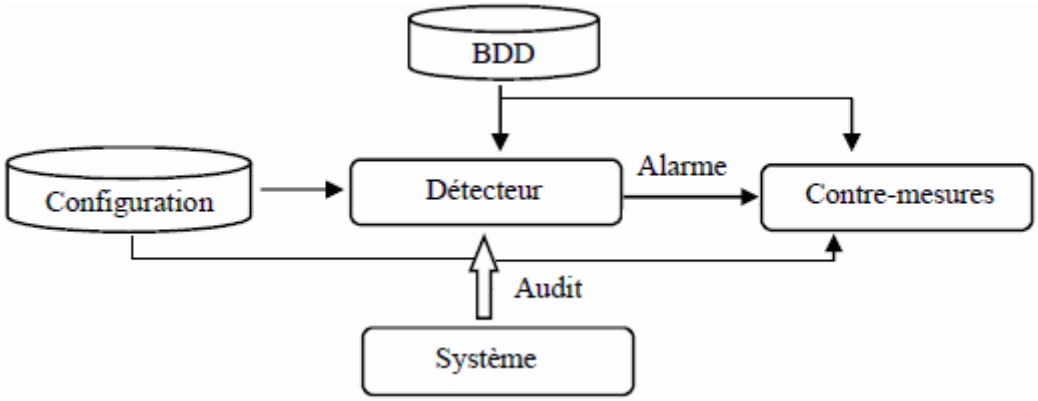


Fig.7 Modèle simplifié d'un système de détection d'intrusions.

8-3-1. Classification des systèmes de détection d'intrusion :

Les systèmes de détection d'intrusion se divisent en deux catégories de base : un système de détection basé sur la signature et un système de détection d'anomalie.

La première méthode utilisée par la plupart des systèmes commerciaux. Elle se base sur la recherche de motifs (chaînes de caractères ou suite d'octets) au sein du flux de données. L'IDS comporte une base de signatures où chaque signature contient le protocole et le port utilisés par l'attaque ainsi que le motif qui permettra de reconnaître les paquets suspects. L'Avantage de cette méthode qu'elle est très efficace pour détecter des attaques sans produire un grand nombre de fausses alarmes, et son principal inconvénient, est que seules les attaques reconnues par les signatures seront détectées. Il est donc nécessaire de mettre à jour régulièrement la base donnée des signatures.

La détection par anomalie consiste à considérer comme hostile tout ce qui n'est pas normal. Cette méthode est puissante car elle permet d'identifier des attaques inhabituelles. Cependant, il est très difficile de distinguer ce qui est normal de ce qui ne l'est pas, car les schémas d'activités varient largement d'un système réel à un autre. La détection d'anomalies à donc tendance de générer de nombreuses fausses alarmes.

8-3-2. Le comportement de la détection (la réponse) :

Le comportement de la détection décrit la réponse du système de détection d'intrusion à une attaque. Elle est qualifiée d'active, si le détecteur réagit activement par des actions correctives, ou proactives (changer les règles de filtrage de Firewall, arrêter des connexions TCP, ou encore attaquer l'attaquant, etc.). Si le système de détection d'intrusion génère simplement des alarmes (afficher un message sur l'écran, générer un son spécifique, l'envoi d'un email, archivage dans un fichier ou dans une base de données, etc.), la réponse est qualifiée de passive.

8-4. Logiciels anti-virus :

La plupart des ordinateurs sont dotés de programmes qui permettent de détecter la présence des menaces virales s'ils sont régulièrement mis à jour, appelés Logiciel anti-virus.
La politique de sécurité du réseau doit mentionner que tous les ordinateurs du réseau doivent être tenus à jour et théoriquement qu'ils doivent tous être protégés par le même système d'antivirus (entre autres, afin de réduire au maximum les frais de maintenance et de mise à jour).

Conclusion :

La sécurité constitue un problème essentiel dans les systèmes des technologies de l'information modernes. Il se posera certainement de manière encore plus importante dans les technologies du futur (réseau des capteurs, Internet du futur, réseaux autonomes, 4G, etc.). La démocratisation des technologies de l'information et des communications, matérialisée par l'interconnexion de divers systèmes (filaires, sans fil, autonomes ou autres), rend la protection des données et des infrastructures considérablement plus complexe.

Incapables de trouver un bon compromis entre la sécurité et son coût pour les services requis par leurs propres moyens, les entreprises et les particuliers deviennent plus exigeants sur les garanties de sécurité que leur apportent les fournisseurs des services. La sécurité apparaît donc comme l'un des enjeux majeurs pour la commercialisation des services et des produits dans le domaine des systèmes d'information et dépasse les dimensions purement techniques.

Aujourd'hui, la sécurité concerne tous les acteurs impliqués (opérateurs des réseaux, fournisseurs des services, intégrateurs des systèmes, utilisateurs, Etat, etc.). La législation, les industriels, les académiques et les utilisateurs sont appelés à collaborer pour développer des meilleures méthodologies pour les processus de protection.
Une sensibilité accrue aux problématiques de sécurité se reflète aujourd'hui dans les débats politiques, le marketing des produits et les demandes des clients. En même temps, une panoplie de travaux est menée par les académiques, les industriels et dans le cadre de consortiums et d'organismes de normalisation afin d'apporter des améliorations et de définir des solutions plus robustes.

En revanche, il faut comprendre qu'il ne peut pas y voir de sécurité standardisée, suffisante pour tout le monde. Cela est dû à l'appréciation très différente des risques autour d'un système dans un environnement donné mais surtout à cause de la complexité croissante des systèmes d'information.

De plus le développement constant des nouveaux services et de nouveaux produits amène des nouvelles vulnérabilités, dont la gravité ne peut être mesurée à l'avance, car elle dépend parmi d'autre de l'échelle du déploiement. La sécurité reste un processus qui doit accompagner le développement d'un système d'information. Le monde devenant plus connecté et plus communicant, les problèmes de sécurité vont probablement s'aggraver dans le futur.

Enfin, il faut noter que la sécurité ne peut être assurée à cent pour cent, et que les outils ne sont pas parfaits. Ils possèdent toujours des failles. Cependant, avec une bonne politique de sécurité, et un bon déploiement des outils, la sécurité peut être très proche des niveaux acceptés.

chapitre 2

Présentation des réseaux Ad Hoc

Introduction :

Le développement technologique au cours de ces dernières décennies a révolutionné un certain nombre de domaines, notamment celui des réseaux informatiques et plus spécialement les réseaux sans fil. Les performances ne cessent d'augmenter et les prix de chuter. Cette avancée les a rendus abordables par le grand public. Ce marché de l'équipement sans fil est actuellement en plein essor.

Les environnements sans fil offrent aujourd'hui une grande flexibilité d'emploi. En particulier, ils permettent la mise en réseau des sites dont le câblage serait trop onéreux à réaliser dans leur totalité, voire même impossible.

Plusieurs systèmes utilisent déjà le modèle cellulaire de réseaux sans fil, et connaissent une très forte expansion à l'heure actuelle : exemple les réseaux GSM. L'inconvénient majeur du modèle cellulaire et qu'il requière une importante infrastructure logistique et matérielle fixe. La contrepartie des réseaux cellulaires sont les réseaux mobiles ad hoc. Ces réseaux ad hoc sont de plus en plus populaires à cause de leur facilité de déploiement. Ce type de réseaux a un rôle crucial à jouer au sein des réseaux informatiques. En offrant des solutions ouvertes pour fournir la mobilité ainsi que des services essentiels là où l'installation d'infrastructures n'est pas possible.

Dans cette partie nous allons donner une présentation générale des réseaux ad hoc à savoir leurs caractéristiques, les protocoles de routage utilisés, les menaces ainsi que les différentes solutions proposées pour assurer la sécurité de ces réseaux.

1- La norme WiFi :

La norme IEEE 802.11 (ISO/IEC 802-11) est un standard international décrivant les caractéristiques d'un réseau local sans fil (WLAN). Le nom WiFi (contraction de Wireless Fidelity, parfois notée Wi-Fi) correspond initialement au nom donné à la certification délivrée par la Wi-Fi Alliance, anciennement WECA (Wireless Ethernet Compatibility Alliance), l'organisme chargé de maintenir l'interopérabilité entre les matériels répondant à la norme 802.11. Par abus de langage (et pour des raisons de marketing) le nom de la norme se confond aujourd'hui avec le nom de la certification. Ainsi un réseau Wifi est en réalité un réseau répondant à la norme 802.11. Des révisions ont été apportées à la norme originale 802.11, afin d'optimiser le débit et d'assurer une meilleure sécurité ou une meilleure interopérabilité comme 802.11a, 802.11b et 802.11g.

Les réseaux mobiles ou sans fil, peuvent être classés en deux classes : les réseaux avec infrastructure et les réseaux sans infrastructure [Ron09], le réseau sans fil consiste au

minimum en un point d'accès (AP, Access point) connecté à l'infrastructure du réseau filaire et un ensemble de postes réseaux sans fil. Cette configuration est baptisée Basic Service Set (BSS, ou ensemble de services de base). Un Extended Service Set (ESS, ou ensemble de services étendu) est un ensemble d'au moins deux BSS formant un seul sous réseau.

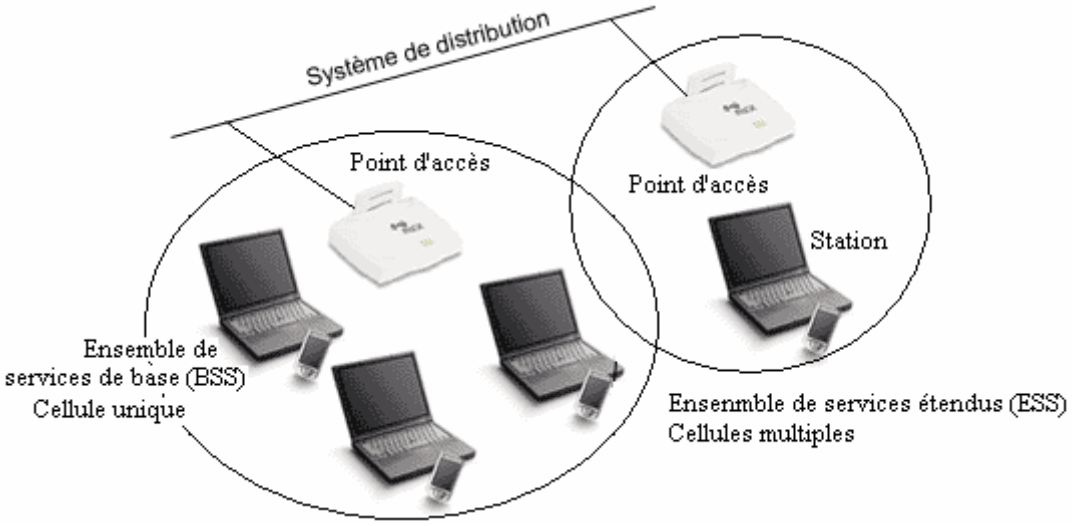


Fig.1 Réseau sans fil : Mode infrastructure [Ron09].

En Mode Ad hoc, également baptisé point à point, ou ensemble de services de base indépendants soit IBSS (Indépendant Basic Service Set), représente simplement un ensemble de stations sans fil 802.11 qui communiquent directement entre elles sans point d'accès ni connexion à un réseau filaire. Cette architecture a comme avantage de se passer de point d'accès et surtout d'être flexible et dynamique.

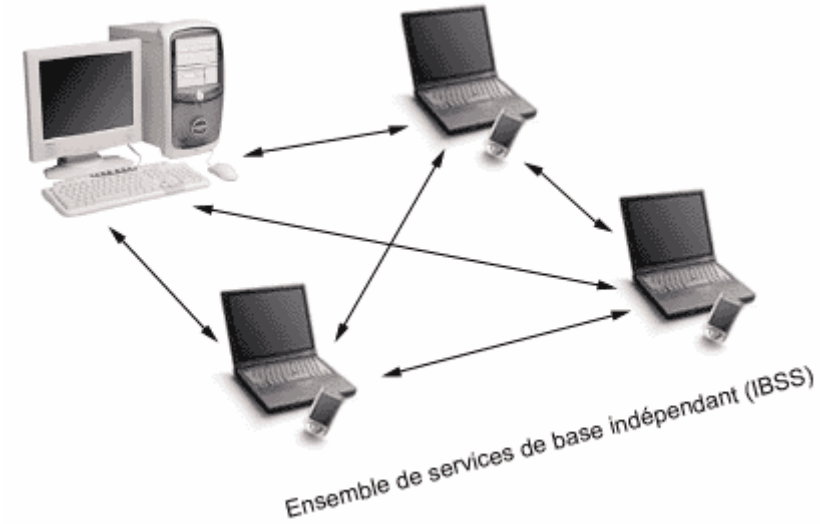


Fig.2 Réseau sans fil : Mode Ad hoc (sans infrastructure) [Ron09].

2- Définition des réseaux ad hoc :

Un réseau sans fil ad hoc est un réseau sans fil d'entités mobiles liées entre elles par des liaisons à base d'ondes radio sans infrastructure fixe ni administration centralisée. Chaque nœud joue le rôle d'hôte ou de routeur à un instant donné. L'interconnexion de tous les nœuds mobiles forme une topologie temporaire dynamique qui se déploie aisément. Une définition de ces réseaux est donnée formellement dans la RFC 2501 [Cor99].

Le terme « ad hoc » est une locution d'origine latine qui signifie « qui convient au sujet, à la situation » On parle donc de réseaux auto-adaptatifs (capables de s'organiser par eux-mêmes).

Les réseaux Ad hoc peuvent également être connectés au monde filaire (voir Fig.3) par l'intermédiaire d'une ou plusieurs passerelles, qui sont appelées, des points d'accès (AP). De tels réseaux sont communément appelés réseaux hybrides [The06]. Chaque terminal du réseau Ad hoc, s'il possède une double interface filaire et sans fil peut donc agir en tant que passerelle pour les autres clients de la bulle ad hoc.

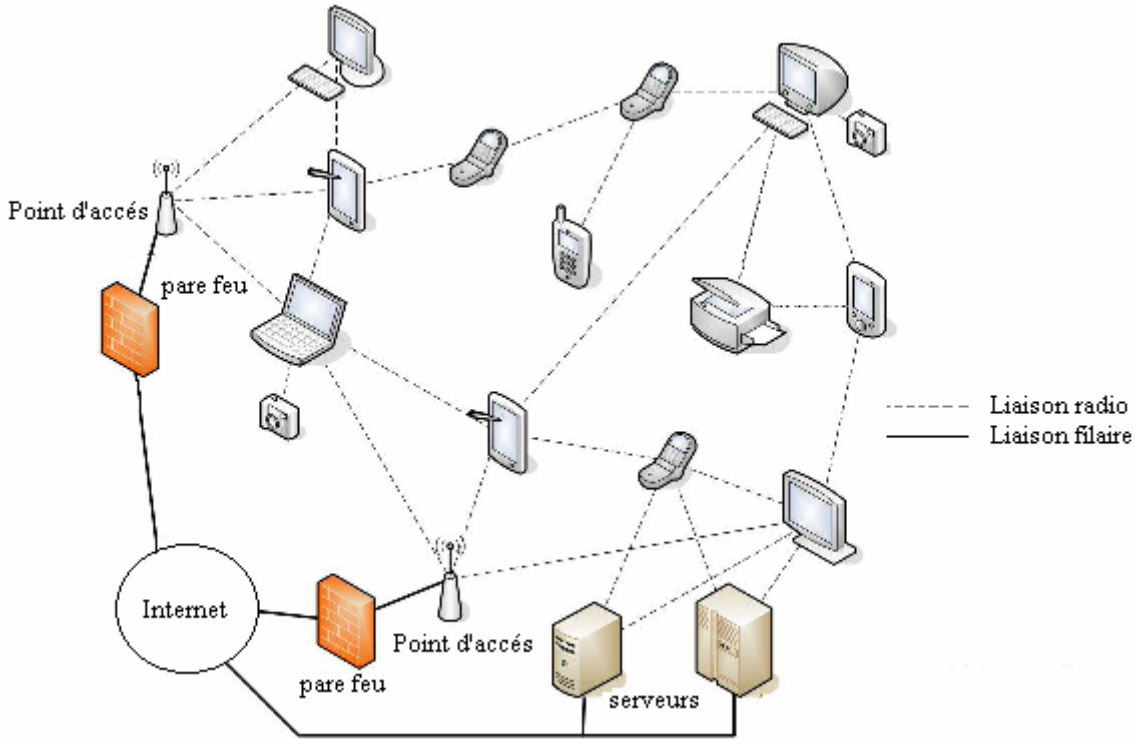


Fig.3 Réseau Ad hoc connecté à l'Internet.

3- Caractéristiques des réseaux ad hoc :

Avant toute étude d'un quiconque système, la connaissance de ses caractéristiques présente une étape importante, dans ce paragraphe on va présenter les différentes propriétés

des réseaux ad hoc qui doivent être prises en compte lors de la conception des algorithmes pour le bon fonctionnement de ces réseaux, à savoir :

Une topologie dynamique : en raison de la mobilité des noeuds, la topologie du réseau est à chaque instant définie par les positions des nœuds qui se déplacent arbitrairement formant ainsi un graphe d'interconnexion composé de liaisons unidirectionnelles et bidirectionnelles.

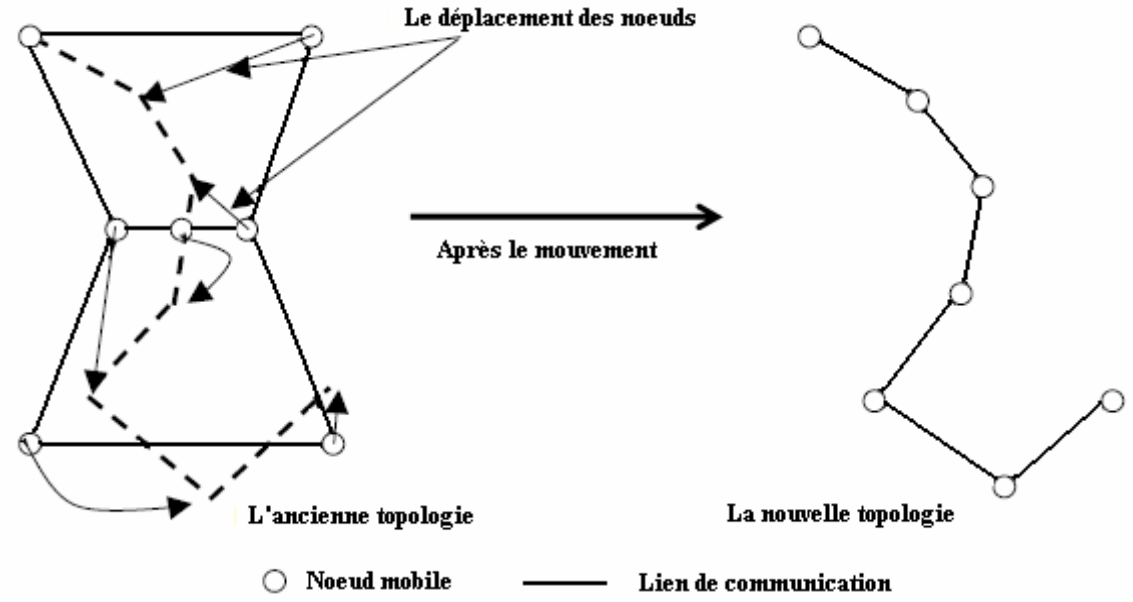


Fig.4 Le changement de la topologie des réseaux ad hoc [Meh07].

L'absence d'infrastructure : Les réseaux ad hoc se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructures préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue.

La taille des réseaux Ad hoc : Elle est souvent de petite ou moyenne taille (une centaine de noeuds), le réseau est utilisé pour étendre temporairement un réseau filaire, comme pour une conférence ou des situations où le déploiement du réseau fixe n'est pas approprié (ex : catastrophes naturelles). Cependant, quelques applications des réseaux Ad hoc nécessitent une utilisation allant jusqu'à des dizaines de milliers de noeuds, comme dans les réseaux de senseurs [Fre01]. Des problèmes liés au passage à l'échelle tels que : l'adressage, le routage, la gestion de la localisation des senseurs et la configuration du réseau, la sécurité, etc., doivent être résolus pour une meilleure gestion du réseau.

Des contraintes d'énergie : Les hôtes mobiles sont alimentés par des sources d'énergie autonomes comme les batteries. De ce fait le paramètre d'énergie doit être pris en considération dans tout contrôle fait par le système.

Une sécurité limitée : Compte tenu de la souplesse de déploiement des réseaux ad hoc, n'importe quel nœud peut faire partie du réseau juste en se plaçant dans une zone de propagation, où il pourra écouter tout ce qui passe par le médium physique; ce qui réduit énormément la sécurité du réseau. Dans les réseaux ad hoc, non seulement les données sont vulnérables comme dans les réseaux filaires mais, il en est de même pour le trafic de contrôle et de gestion du routage. Les problématiques de la sécurité dans les réseaux ad hoc sont donc très complexes, puisque l'on cherche à autoriser de nouveaux mobiles à participer au réseau, tout en évitant des noeuds "malins" qui détourneraient ou perturberaient le fonctionnement même du routage. De plus, du fait de la nature du canal radio (interférences et taux d'erreur élevés), les protocoles de qualité de service habituels (par exemple IntServ / RSVP ou Diff-Serv) ne sont pas utilisables directement dans le monde ad hoc et des solutions spécifiques doivent être proposées [Che99].

4- Domaines d'utilisation des réseaux ad hoc :

Les applications ayant recours aux réseaux ad hoc couvrent un très large spectre, incluant les applications militaires et de tactique comme les opérations de secours et les missions d'exploration, l'enseignement à distance, les systèmes de fichiers répartis, les applications de calcul distribué ou méta-computing ou plus simplement, une réunion de travail peut demander la création ponctuelle d'un tel réseau entre ses participants. D'une façon générale, les réseaux ad hoc sont utilisés dans toute application où le déploiement d'une infrastructure réseau filaire est trop contraignant, soit parce que difficile à mettre en place, soit parce que la durée d'installation du réseau ne justifie pas de câblage à demeure.

4-1. Les réseaux ad hoc et applications :

L'architecture très flexible et aisément déployable des réseaux ad hoc permet le développement de diverses applications. Les télécommunications tactiques comme les opérations de secours et les missions d'exploration et les réseaux militaires sont les demandes les plus évidentes de réseaux hoc. Indépendamment des télécommunications tactiques, il y a beaucoup d'autres domaines d'application qui exigent le déploiement rapide et mobile de communications.

On peut citer à titre d'exemple :

- Déploiement de réseau provisoire : des réseaux ad hoc peuvent être déployés quand il n'est pas viable ou rentable pour construire une infrastructure. Par exemple, ils peuvent être employés en tant que provisoire solution dans les secteurs de conférences, dans des zones peuplées et clairsemés ce qu'il est trop difficile d'installer une infrastructure.

- Opérations de secours en cas de catastrophe : les possibilités rapides de déploiement des réseaux ad hoc font ils une technologie éminente à employer pour la gestion des opérations de soulagement après désastres à grande échelle tels que des tremblements de terre, des tsunamis et des inondations.
- Bâtiments : un grand nombre de sondes et de déclencheurs peuvent être déployés en dehors installation de toute infrastructure pour créer des milieux intelligents et un calcul sensible d'environnement.
- Soins de santé : systèmes pour surveiller les conditions de santé et le lieu de séjour des patients et les personnes âgées forment un autre domaine d'application évident pour les réseaux ad hoc.

5- Le routage dans les réseaux ad hoc :

Le routage s'occupe de l'acheminement des paquets vers les destinations désirées. Il offre des services aux différentes applications qui désirent envoyer des données à d'autres applications se trouvant dans d'autres nœuds ou réseaux distants. De nombreux protocoles de routage ont été proposés pour les réseaux ad hoc. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance. Le problème consiste à trouver l'investissement de moindre coût en capacités nominales et de réserves qui assure le routage du trafic nominal et garantit sa continuité en cas de n'importe quelle panne de noeuds.

Si on suppose que les coûts des liens sont identiques, le chemin indiqué dans la figure suivante est le chemin optimal reliant la station source et la station destination. Une bonne stratégie de routage utilise ce chemin dans le transfert des données entres les deux stations.

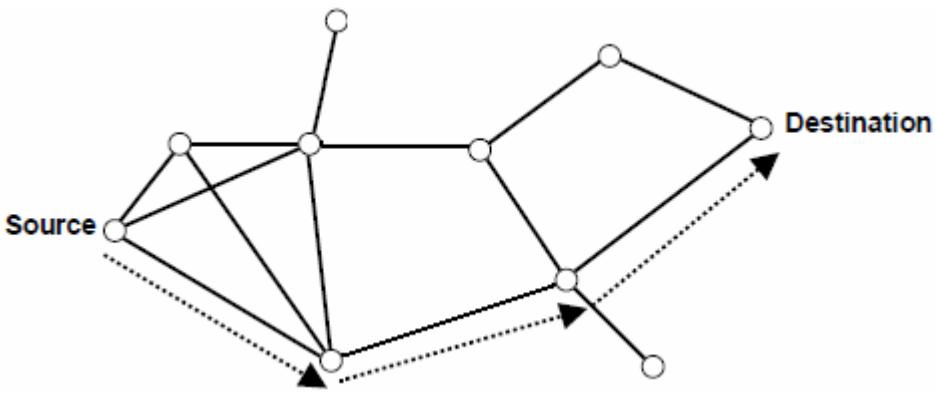


Fig.5 Le chemin optimal utilisé dan le routage entre la source et la destination.

Le problème qui se pose dans le contexte des réseaux ad hoc est l'adaptation de la méthode d'acheminement utilisée avec le grand nombre d'unités existant dans un environnement caractérisé par de modestes capacités de calcul et de sauvegarde et de changements rapides de topologies.

Il semble donc important que toute conception de protocole de routage doive étudier les problèmes suivants :

- La minimisation de la charge du réseau
- Offrir un support pour pouvoir effectuer des communications multipoints fiables
- Assurer un routage optimal
- Offrir une bonne qualité concernant le temps de latence

5-1. Classification des protocoles de routage :

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en deux catégories, les protocoles proactifs et les protocoles réactifs. Les protocoles proactifs établissent les routes à l'avance en se basant sur l'échange périodique des tables de routage, alors que les protocoles réactifs cherchent les routes à la demande.

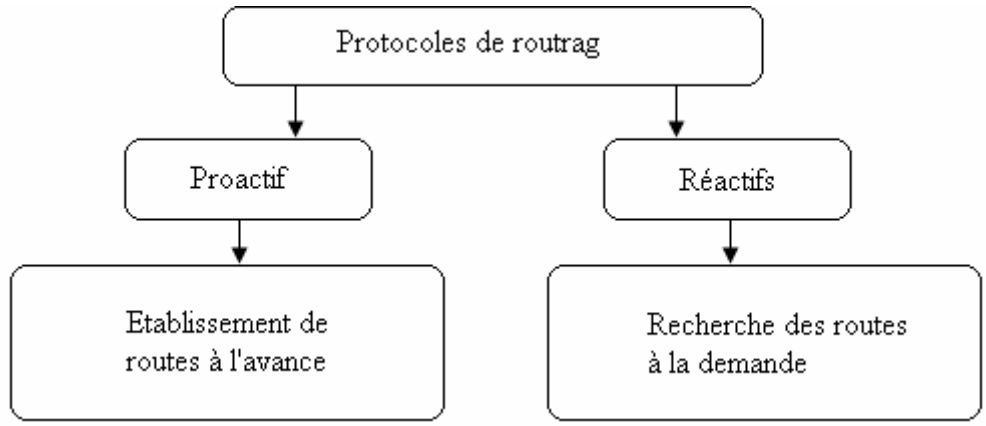


Fig.6 Classification des protocoles de routage.

5-1-1.Les protocoles de routage proactifs :

Ce type d'algorithmes est principalement basé sur la garde des tables de routage au niveau de chaque nœud du réseau. Chaque émetteur consulte sa table pour trouver une entrée vers la destination voulue. Plusieurs algorithmes ont été proposés et même implémentés et testés. Exemple DSDV, WRP et OLSR.

5-1-1-1. DSDV (Destination Sequenced Distance-Vector Routing) :

Ce protocole est basé sur le mécanisme de routage classique de Bellman-Ford. Chaque nœud du réseau maintient une table de routage vers toute destination possible. La mise à jour des tables de routage est faite périodiquement Pour alléger la charge du réseau, il existe deux méthodes de mise à jour [Per94]:

- Full dump : Des paquets transportant toute information disponible du routage et peut demander plusieurs NPDUs, leur transmission est occasionnelle.
- Incremental : Des paquets portent juste l'information de changement sur les tables de routage depuis la dernière mise à jour. Les nœuds maintiennent aussi une autre table où ils enregistrent les paquets envoyés.

5-1-1-2. WRP (Wireless Routing Protocol) :

Chaque nœud dispose de 4 tables

- Table de destination
- Table de routage
- Table des coûts des liens
- Table de la liste des messages à retransmettre (MRL)

Chaque entrée de la table MRL contient :

- Numéro de séquence du message de mise à jour.
- Un compteur de retransmission.
- Un vecteur de flag des demandes d'acquittement avec une entrée par voisin.
- Une liste de mise à jour qui sera envoyée dans les messages de mise à jour

Les enregistrements de la MRL mis à jour depuis la réception d'un message de mise à jour doivent être retransmis aux voisins qui doivent accuser réception. Les messages sont envoyés seulement entre les nœuds voisins et contenant une liste de mise à jour (destination, distance à la destination, prédécesseur de la destination), aussi bien une liste de réponses indiquant quels nœuds devrait reconnaître la mise à jour. Les nœuds envoient des mises à jour après traitement de mise à jour reçue ou bien après avoir détecter des changements dans les liens voisins. En cas de rupture de liens, un message de mise à jour est envoyé aux voisins qui à leurs tours mettent à jour leurs tables des distances et cherchent une nouvelle route par d'autres nœuds. Chaque changement est signalé par un message de mise à jour aux voisins. Sans les messages de mise à jour, les nœuds s'envoient mutuellement des paquets « hello » pour confirmer l'existence et la validité des liaisons sinon nous pourrions déduire qu'une liaison vient d'être perdue. Dans le cas d'une réception d'un nouveau « hello » envoyé par un

nouveau nœud, nous l'insérons dans la table de routage et une copie de cette table sera envoyée au nouveau nœud. L'exception majeure de cet algorithme est qu'il évite le problème du « compte à l'infini » en obligeant chaque nœud d'effectuer des tests consistants sur les informations disponibles depuis les voisins. Ainsi il n'y aura plus de boucle et un routage plus rapide est garanti.

5-1-1-3. OLSR (Optimized Link State Routing) :

Protocole proactif présente une optimisation de « link state » dont:

- La réduction de l'impact de l'inondation sur le réseau, par la réduction du nombre de nœuds participants juste aux Multi-Points Relais [MPR], ce qui économise la bande passante.

- La minimisation de la taille des messages de contrôle, qui ne contiendront que l'information du voisinage de l'expéditeur mais pas de tout le réseau.

- En plus des routes sans cycle qui sont garanties, OLSR offre des routes symétriques de plus court chemin. Des paquets (TC) « Topology Control » sont périodiquement diffusés dans le réseau, ne transitent que par les (MPR) et ne contiennent que la liste des relais multipoints (MPR). Un système d'élection des (MPR) est mis en place et chaque nœud élu reçoit l'information dans un message « hello » [Cla03].

OLSR supporte l'adressage IP, où à chaque nœud est associé une adresse IP régulière, de plus il ne demande aucun changement sur le format des paquets IP. Le protocole n'intervient que sur la gestion de la table de routage.

5-1-2. Les protocoles de routage Réactifs :

Ce type d'algorithmes ne nécessite pas le maintien permanent de tables de routage, ni une connaissance préalable de la topologie du réseau au moment de l'envoi. Par contre une phase précédant l'envoi consiste à rechercher le chemin vers la destination voulue. Cette phase est initialisée par la diffusion d'un paquet « découverte de route » depuis la source sur tout le réseau et redirigé par chaque nœud intermédiaire jusqu'à atteindre la destination, où ce paquet sera retransmis vers la source sous forme d'une réponse « réponse de route » traçant ainsi un chemin vers la destination, des algorithmes ont été proposés, des améliorations et des évaluations sont en cours. Nous citons à titre d'exemple AODV, DSR et TORA.

5-1-2-1. AODV (Ad hoc On Demand Distance Vector):

AODV est un algorithme de routage à la demande, c'est-à-dire qu'il ne construise de routes entre noeuds que lorsqu'elles sont demandées par les noeuds sources, et ce pour réduire le nombre de diffusions de messages. AODV utilise les principes des numéros de séquence afin de maintenir la consistance des informations de routage. Les numéros de séquence

permettent d'utiliser les routes les plus récentes. Il utilise une requête de route dans le but de créer un chemin vers une destination. La route peut ne pas exister si la destination n'est pas connue au préalable, ou si le chemin existant vers la destination a expiré ou il est devenu défaillant. Cependant, AODV maintient les chemins d'une façon distribuée en gardant une table de routage, au niveau de chaque noeud de transit appartenant au chemin cherché. Afin de maintenir des routes cohérentes, une transmission périodique du message "Hello" est effectuée. Si au bout d'un certain temps aucun message "Hello" n'est reçu à partir d'un noeud voisin, le lien en question est considéré défaillant. Le protocole AODV ne présente pas de boucle de routage, et offre une convergence rapide quand la topologie du réseau Ad hoc change [Per01]. Le protocole AODV est un protocole uniforme, de type Distance Vector.

5-1-2-2. DSR (Dynamic Source Routing Protocol):

Le protocole DSR est basé sur le principe de diffusion à la demande pour calculer une route vers une destination. Il utilise un routage par la source, et se base principalement sur deux mécanismes coopératifs : la découverte de route et la maintenance de route. Il permet aussi l'existence de plusieurs routes vers la destination. A partir des informations de routage qui sont incluses dans les paquets de données, les noeuds appartenant à la route, ainsi que leurs noeuds voisins, peuvent les collecter et les mettre dans leurs caches pour une utilisation ultérieure. Chaque noeud dans le réseau envoyant ou relayant un paquet est responsable de confirmer son acheminement vers le prochain noeud en recevant un acquittement. Si un nœud détecte une cassure de route, un message d'erreur de route est retourné à la source. Lors de la réception d'un message d'erreur de route, la source supprime la route défaillante de son cache. Si un chemin alternatif est disponible, il peut être employé pour des données restantes à la destination, autrement, une nouvelle découverte de route est lancée. Comme AODV, DSR bufférise les paquets IP dans le noeud de source quand la découverte de route est effectuée. Ce protocole est un protocole réactif, uniforme, de type link state [Joh04].

5-1-2-3. TORA (Temporally Ordered Routing Algorithm) :

Ce protocole a été conçu principalement pour minimiser l'effet des changements de la topologie qui sont fréquents dans les réseaux ad hoc.

Afin de s'adapter à la mobilité des réseaux ad hoc, le protocole stocke plusieurs chemins vers une même destination, ce qui fait que beaucoup de changements de topologie n'auront pas d'effets sur le routage des données, à moins que tous les chemins qui mènent vers la destination soient perdus (rompus).

TORA est caractérisé essentiellement par le fait que les messages de contrôle sont limités à l'ensemble des noeuds proches du lieu de l'occurrence du changement de la topologie. Dans ce protocole, l'utilisation des meilleurs chemins a une importance secondaire, les longs chemins peuvent être utilisés afin d'éviter le contrôle induit par le processus de découverte de nouveaux chemins. Ce protocole est basé sur l'utilisation de la propriété appelée "orientation destination" des graphes acycliques orientés. Un graphe acyclique orienté (DAG) est orienté destination s'il y a toujours un chemin possible vers une destination spécifiée. Le graphe devient non orienté destination, si un lien (ou plus) devient défaillant. Dans ce cas, les algorithmes utilisent le concept d'inversement de liens. Ce concept assure la transformation du graphe précédent, en un graphe orienté destination durant un temps fini [Par01].
Afin de maintenir le DAG orienté destination, l'algorithme TORA utilise la notion de taille de noeud. Chaque noeud possède une taille qui l'échange avec l'ensemble de ses voisins directs. Un lien est toujours orienté du noeud qui a la plus grande taille, vers le noeud qui la plus petite taille.

5-1-3. Protocoles hybrides :

Les protocoles hybrides combinent les approches réactive et proactive. Le principe est de connaître notre voisinage de manière proactive jusqu'à une certaine distance (par exemple trois ou quatre sauts), et si jamais une application cherche à envoyer quelque chose à un nœud qui n'est pas dans cette zone, d'effectuer une recherche réactive à l'extérieur. Avec ce système, on dispose immédiatement des routes dans notre voisinage proche, et lorsque la recherche doit être étendue plus loin, elle en est optimisée (un noeud qui reçoit un paquet de recherche de route réactive va tout de suite savoir si la destination est dans son propre voisinage. Si c'est le cas, il va pouvoir répondre, et sinon il va propager de manière optimisée la demande hors de sa zone proactive). Selon le type de trafic et les routes demandées, ce type de protocole hybride peut cependant combiner les désavantages des deux méthodes échange de paquets de contrôle réguliers et inondation de l'ensemble de réseau pour chercher une route vers un noeud éloigné. ZRP [Haa02] est un exemple de ce protocole qui combine des approches proactives et réactives, essayant de ce fait de rassembler les avantages des deux approches. Il définit autour de chaque noeud une zone qui contient les noeuds voisins à un nombre donné de sauts du noeud. Des algorithmes proactifs et réactifs sont employés par le noeud pour acheminer les paquets, respectivement, dans et en dehors de la zone.

6- Les attaques contre les réseaux Ad hoc :

Un réseau sans fil est davantage versatile mais plus vulnérable aux attaques qu'un réseau filaire [Kar02], car les transmissions radio sont effectuées dans l'air.

Sur un réseau filaire, un intrus nécessiterait d'avoir un accès physique à une machine du réseau, ou bien de se connecter aux câbles. Dans le cas d'un réseau sans fil, l'intrus peut écouter passivement tous les messages échangés pourvu qu'il se trouve dans l'aire d'émission. Donc l'adversaire a accès au réseau et peut intercepter aisément les données transmises, sans même que l'émetteur ait connaissance de l'intrusion (par exemple, au moyen d'un ordinateur portable dans un véhicule stationné dans une rue on peut intercepter les communications échangées à l'intérieur d'un immeuble voisin). L'intrus, en étant potentiellement invisible, peut enregistrer, modifier, et ensuite retransmettre les paquets comme s'ils avaient été envoyés par un utilisateur légitime.

En outre, à cause des limitations du support, les communications peuvent facilement être perturbées; l'intrus peut effectuer cette attaque en occupant le support avec ses propres messages, ou tout simplement en perturbant les communications avec du bruit.

6-1. Classification des attaques :

Les attaques sur les réseaux ad hoc peuvent se produire de différentes manières. La classification de ces attaques dépend de plusieurs paramètres :

- Attaques internes ou externes.
- Attaques passives ou actives.
- Attaques sur des protocoles ou par consommation de ressources.
- Attaques individuelles ou attaques distribuées.
- Attaques directes ou attaques indirectes.

Les attaques internes se posent dans le cas d'un noeud compromis. Dans ce cas il est relativement difficile de détecter une telle attaque puisque l'attaquant a l'accès simple au réseau en utilisant le noeud compromis qu'il possède [Gha02]. Ce type d'attaque pose le problème de confiance entre les noeuds d'un réseau ad hoc, une station ne peut pas par suite avoir toujours confiance en ses stations voisines. Par contre dans les attaques externes, l'attaquant ne possède pas un terminal du réseau mais peut se connecter au réseau de l'extérieur c'est-à-dire à partir d'un autre réseau comme l'Internet par exemple. Dans ce type d'attaques, il est difficile de déterminer la source d'une attaque puisqu'il n'y a pas un point de concentration de trafic dans le réseau et l'attaquant peut rejoindre le réseau à partir de différents points d'accès.

Les attaques dites actives permettent de récupérer des informations à partir des postes ou des paquets transmis entre les différents terminaux. Ces attaques peuvent être très graves si les informations récupérées sont sensibles. Les attaques passives [Pie04] ne permettent pas de récupérer des données mais ils peuvent par exemple empêcher le réseau de bien fonctionner et ceci en exploitant les failles dans les programmes ou les protocoles utilisés. Ces attaques ont un aspect désastreux et sont aussi très dangereuses [Bou03].

Les attaques sur protocoles [San02, Qin03] sont des attaques qui visent les protocoles qui nécessitent un travail collectif entre les différents noeuds. Le routage surtout pose des problèmes spécifiques: chaque station du réseau peut servir de relais et a donc la possibilité de capturer ou bien de détourner le trafic en transit. Des attaques en déni de service sont également possibles. Donc ce cas, l'intérêt de l'attaquant est essentiellement de nuire au bon déroulement des processus de routage. Un autre protocole qui est aussi vulnérable que les protocoles de routage est le protocole d'accès au médium. Une attaque dans ce sens peut par exemple consister à occuper la bande passante pour une longue durée empêchant les autres stations d'utiliser le lien radio pour communiquer.

Les attaques peuvent enfin être classées en attaques individuelles ou attaques distribuées. Les attaques individuelles sont simples et ils sont issus d'une seule source et par un chemin simple sans utiliser des stations intermédiaires.

Par contre, une attaque distribuée est une attaque évoluée invoquant plusieurs stations ou provenant de plusieurs sources [Sch02]. Les attaques distribuées sont plus dangereuses et difficiles à détecter puisqu'ils utilisent plusieurs stations intermédiaires, ce qui a pour effet la difficulté de déterminer la source d'une telle attaque.

Les attaques par déni de service (DoS), affectent en général la couche physique et la couche liaison [Jar02], ont souvent l'aspect d'attaques distribuées. Ces attaques peuvent faire un brouillage du canal radio pour empêcher toute communication, comme ils peuvent faire une tentative de gaspillage de l'énergie de noeuds.

Un autre type d'attaques, appelées spoofing, sont des attaques où le noeud malveillant usurpe l'identité d'un autre noeud dans les réseaux [Bur03], où les attaquants peuvent recevoir les messages de cheminement qui sont dirigés vers les noeuds qu'ils ont usurpés.

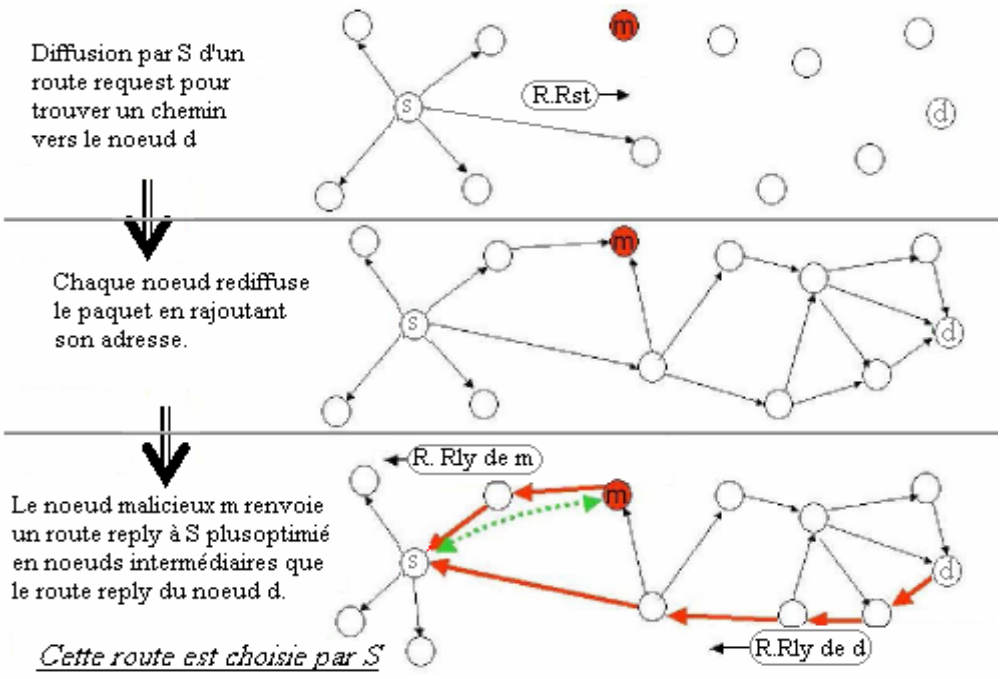


Fig.7 Exemple d'attaque : Attaque black hole : Le nœud malicieux m capte le trafic dédié au nœud d

7- Etat de l'Art des Solutions de sécurité pour les réseaux ad hoc :

Vu de leur vulnérabilité, les réseaux ad hoc font l'objet de plusieurs travaux de recherche visant d'améliorer les mécanismes de sécurité de ces réseaux. Dans cette section, nous donnons un aperçu de différentes approches proposées dans la littérature ainsi qu'une analyse de chaque approche.

7-1. Solutions pour l'Authentification :

Pour introduire un système d'authentification basé sur la cryptographie à clé publique pour les réseaux ad hoc, il faut être très attentif, car l'absence d'infrastructure centralisée dans ces réseaux compromet l'utilisation directe de ces systèmes. En effet, les systèmes d'authentification supposent l'utilisation de certificats établis par une autorité centrale. Le certificat, signé par l'autorité centrale, permet de garantir qu'une clé publique appartient bien à son propriétaire et non à un usurpateur. Les solutions proposées s'appuient essentiellement sur un modèle de confiance. On a donc :

7-1-1. La cryptographie à seuil :

Dans [Zho99], les auteurs ont proposé d'utiliser la cryptographie à seuil (threshold cryptography) pour distribuer les fonctionnalités d'une autorité de certification parmi les noeuds du réseau. Dans cet article, la clé privée de l'autorité de certification qui sert à signer les clés publiques des nœuds est partagée parmi un ensemble de n nœuds du réseau.

Un schéma à seuil (k, n) permet de distribuer les services de l'autorité de certification à un groupe de n nœuds serveurs, de façon à ce que k nœuds puissent exécuter ensemble les

services de l'autorité de certification, tandis qu'il est impossible d'exécuter ces services par moins de k nœuds.

Cette solution exige que tous les nœuds soient capables de faire les calculs nécessaires, du fait que cette solution est basée sur le chiffrement à clé publique qui nécessite une grande puissance de calcul. D'autre part, cette solution suppose que certains nœuds doivent jouer le rôle des serveurs, ce qui n'est pas toujours réaliste, au moins dans les applications civiles. Un autre problème est comment rétablir le secret partagé ; car les k nœuds choisis peuvent être changés à cause de la topologie dynamique du réseau ou à cause des nœuds compromis. Cela est difficile à réaliser à cause de l'absence d'une autorité centralisée qui gère les nœuds serveurs.

7-1-2. Le Modèle PGP (Pretty Good Privacy) :

PGP a été crée dans un but précis [Abd97], offrir à tout le monde un moyen de préserver la confidentialité des informations. Celles ci, peuvent être des messages de courriers électroniques (L'usage le plus fréquent de PGP), des fichiers que l'on souhaite archiver, ou encore des documents dont on souhaite garantir. Il a été créé par P. Zimmermann [Zim96]. En effet, plusieurs modèles ont été proposés basés sur PGP. J. Hubaux est parmi qui ont beaucoup contribués pour assurer la confiance pour les réseaux ad hoc en basant sur le modèle PGP. Le PGP est une étape importante dans l'histoire de cryptographie, avec la cryptographie à clé publique, le PGP d'abord produit une clé de session aléatoire et chiffre le texte avec cette clé. La clé de session et le texte chiffré sont alors chiffrés avec la clé publique du destinataire et expédiés au destinataire. De plus, l'architecture de PGP est une architecture complètement décentralisée, cette décentralisation intervient au niveau de l'autorité de certificat (AC). Le niveau de confiance d'un certificat est associé directement au niveau de confiance accordé à l'autorité de certification qui l'a généré, où on peut trouver différents niveaux de confiance dans les certificats des utilisateurs. Ce modèle de confiance de PGP stipule que le niveau de confiance accordé à un certificat dépendra du niveau de confiance accordé au signataire du certificat. Mais, l'AC fait appel à des annuaires pour attribuer des paires de clés et des certificats aux utilisateurs.

Un des problèmes majeurs du PGP réside dans la distribution et la gestion des clés. Le manque de chemins fixes ou formels de certification signifie que l'authenticité incertaine de n'importe quel certificat de clé de PGP devient une question plutôt significative. Ceci, doit être contrôlée et faite clairement, de sorte qu'un utilisateur puisse pouvoir utiliser cet outil puissant convenablement, et d'une manière efficace.

7-1-3. Accord de clé secrète commune (Key agreement) :

Les recherches en matière de Key agreement dans les réseaux ad hoc se focalisent sur la manière d'établir une clé commune entre plusieurs participants qui ne se connaissent pas à priori. Les participants établissent entre eux une clé secrète leur permettant de s'authentifier afin de communiquer de manière sécurisée. La mise en place de cette clé peut se faire de manière distribuée. Dans ce cas, la clé secrète est fournie aux participants du réseau ad hoc via un canal supposé sûr. C'est le cas lorsque des collègues souhaitant établir une communication sûre entre eux à l'occasion d'une réunion dans une salle de conférence close, ils distribuent un mot de passe inscrit sur un morceau de papier qui fait le tour de la salle. Seules les personnes présentes dans la salle en ont connaissance. Une clé forte peut être dérivée du mot de passe à l'aide d'une fonction de hachage. La difficulté de ce mode de fonctionnement est de trouver un canal sécurisé pour distribuer la clé. Une autre manière d'établir une clé secrète commune est de faire en sorte que chaque participant apporte sa contribution à la clé finale. Lorsqu'il n'y a que deux noeuds, le protocole de Diffie-Hellman peut être utilisé.

Méthode de Diffie-Hellman :

Alice et Bob se mettent d'accord sur un entier N et un générateur α du groupe cyclique fini d'ordre N (ce groupe est constitué de tous les entiers positifs ou nul strictement inférieur à N, les calculs dans le groupe cyclique se font modulo N). Alice et Bob choisissent chacun un nombre secret utilisé comme exposant.

Le secret d'Alice est a, et celui de Bob est b. Alice envoie alors α^a modulo (N) à Bob et Bob envoie α^b modulo (N) à Alice. Une fois que Bob a reçu α^a modulo (N) de la part d'Alice, il peut utiliser son nombre secret b pour calculer :

(α^a modulo (N))^b modulo(N) => (α^a)^b modulo(N) => ($\alpha^{a.b}$) modulo(N).

De son côté, Alice peut calculer :

(α^b modulo (N))^a modulo(N) => (α^b)^a modulo(N) => ($\alpha^{b.a}$) modulo(N).

La clé résultante, secret partagé par Alice et Bob, sera $\alpha^{a.b}$ modulo (N). Un attaquant qui a la possibilité d'écouter les échanges entre Alice et Bob, ne pourra pas deviner la clé car il est très difficile (en terme de puissance de calcul), lorsque N, α et exposants sont suffisamment grands, de calculer le nombre secret a connaissant N et α. L'opération revient à calculer un logarithme discret dans le groupe cyclique fini d'ordre N.

La généralisation de cet algorithme à n nœuds est que chaque noeud i possède son exposant secret e_i, la clé résultante sera $\alpha^{e1.e2.e3...en}$. La mise en pratique du protocole de Diffie-Hellman

à de multiples participants n'est pas si simple et fait l'objet de nombreuses recherches. En effet, il ne suffit pas que chacun envoie la valeur α^{ei} aux autres pour que tous les noeuds aient la possibilité de déterminer la clé. Le protocole de Diffie-Hellman s'appuie sur le fait que les participants calculent la clé à l'aide de la valeur reçue des autres participants et leur exposant secret. Il faudrait donc que le noeud i reçoive la valeur $\alpha^{e1.e2...ei-1.ei+1...en}$ pour être capable de calculer la clé.

7-1-4. Le resurrecting duckling :

Dans une volonté de permettre une distribution facile des clés dans un réseau Ad hoc, Franck Stajano et Ross Anderson ont proposé dans [Sta99] un mécanisme pour échanger une clé secrète entre deux noeuds. Ce modèle, appelé “The resurrecting duckling”, repose sur la relation de maître/esclave et sur le concept d’imprégnation. Ainsi, pendant une phase d’initialisation (avant son introduction au sein du réseau), un noeud esclave doit être ‘imprégné’ par son noeud maître (éventuellement, le propriétaire) par le biais d’un contact physique (par exemple électrique). Lors de ce contact, une clé secrète est échangée en toute confidentialité. Par la suite, cette clé peut être utilisée pour chiffrer et authentifier des informations, comme une liste d’autres clés partagées par exemple. Bien qu’innovante, cette approche laisse plusieurs questions en suspens. La première concerne la phase d’imprégnation. Si un contact physique est possible dans le cadre d’un petit réseau (un piconet [Ben97] par exemple) avec un leader désigné, il devient moins envisageable dans le cadre d’un grand réseau ouvert. Le deuxième problème porte sur la gestion de clés. En effet, l’approche ne propose pas comment faire pour échanger une clé secrète entre chaque paire de noeuds du réseau. Par ailleurs, si l’un des noeuds est corrompu, toutes les autres clés liées à ce nœud peuvent se trouver menacées et rien n’est mentionné quant à la répudiation d’une clé. Une réinitialisation systématique paraît difficile à mettre en place.

7-1-5. TESLA (Time Efficient Stream Loss-tolerant Authentication) :

Cette méthode a été proposée par Perrig et al. [Per02]. Il s'agit d'une extension au protocole décrit en [And98], dit Guy Fawkes' protocol. TESLA permet d'authentifier les messages avec un MAC dépendant d'une clé secrète qui n'est divulguée par l'émetteur du message qu'après un délai d'attente β. La valeur β est calculée de manière à ce qu'on soit sûr que le destinataire a reçu le message avant la divulgation de la clé, cette condition garantie l'intégrité du message. Le temps β ne doit pas être trop important pour limiter les latences dans le réseau, en effet un destinataire doit attendre la divulgation de la clé secrète avant de pouvoir effectivement traiter un message.

La clé secrète utilisée pour le MAC est issue d'une chaîne de clés. Un élément de la chaîne ki est calculé de la manière suivante : ki = H (ki+1) où H est une fonction de hachage.
L'élément initial kn est choisi par l'émetteur. Celui-ci va utiliser ces clés par ordre croissant c'est à dire en commençant par k1. En réception, le destinataire pourra vérifier la relation suivante : ki-1 = H(ki) où ki est la clé dernièrement reçue et ki-1 correspond à la clé précédente. Cette condition assure que la clé ki fasse bien partie de la chaîne de clé de l'émetteur, ce qui assure, en plus de l'intégrité, la propriété d'authentification du paquet. Il est à noter que ce processus doit être initialisé par l'authentification du premier paquet émis à l'aide d'une signature numérique.

7-2. Solutions pour l'intégrité des données :

Les chaînes de hachage sont un outil très efficace et permettent d'offrir une protection très suffisante à bien moindre coût par rapport aux approches cryptographiques détaillées précédemment. Ainsi le protocole SEAD (Secure Efficient distance vector Routing for mobile Ad hoc networks) propose de renforcer la sécurité du protocole DSDV en utilisant les chaînes de hachage à sens unique. Celles-ci permettent de prévenir d'éventuels attaquants d'incrémenter artificiellement le nombre de sauts dans l'en-tête des paquets de signalisation. Un noeud génère une chaîne de hachage et la décompose en plusieurs segments de m éléments
(h_0, h_1, ..., h_{m-1}),..., (h_{km}, h_{km+1},…, h_{km+m-1}) avec k = m/n-i, m correspondant au diamètre maximal du réseau et i étant le numéro de séquence (voir figure 8).

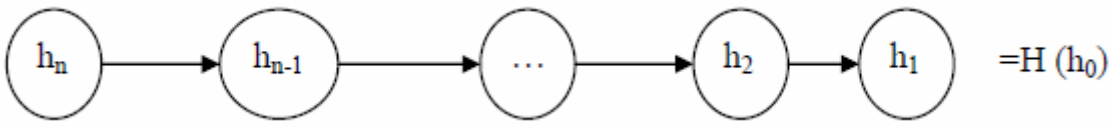


Fig.8 : Les chaînes de hachage dans SEAD

Puisque hi = H (hi-1), connaissant hi, il est facile de vérifier l'authenticité de hj, tant que j reste inférieur à i. De plus, comme des différentes fonctions de hachage sont utilisées pour des diamètres et des métriques différentes, un attaquant ne peut jamais forger une valeur de métrique inférieure ou un plus grand numéro de séquence. Enfin, le protocole DSDV spécifie que lorsqu'un noeud reçoit un message de signalisation, il met à jour sa table de routage si le numéro de séquence est plus grand ou identique avec une métrique inférieure. Donc, SEAD permet d'empêcher un attaquant potentiel de décrémenter artificiellement le nombre de sauts ou d'incrémenter le numéro de séquence des paquets. A la base de ses fonctions beaucoup de travaux sont développés pour garantir l'intégrité des données comme dans [Pap02].

7-3. Solutions pour la Confidentialité :

La confidentialité dans les réseaux sans fil ad hoc est d'abord traitée par l'utilisation de transmission par saut de fréquences. Les données sont transmises sur une séquence de fréquences définies pseudo aléatoirement. L'attaquant doit connaître cette séquence pour pouvoir se synchroniser en réception.

Une fois l'authentification des participants clairement établie, les outils cryptographiques permettent de rendre les communications confidentielles. Toutefois, étant donné qu'une des contraintes des réseaux ad hoc est de devoir être adaptable à des noeuds ayant de faibles capacités de calcul, la cryptographie symétrique sera préférée à la cryptographie à clé publique, cette dernière nécessitant beaucoup plus de puissance de calcul.

7-4. Les Cartes à puce :

Les cartes à puces sont généralement considérées comme le système informatique le plus sécurisé, c'est-à-dire qu'il est très difficile (voire impossible) de déduire par quelque méthode (physique ou logique) que ce soit, les clés utilisées lors de l'exécution d'un algorithme cryptographique (RSA, DES, AES, …) par le processeur d'une puce sécurisée. Les composants actuels réalisent un calcul RSA (2048 bits) en une demi seconde, et intègrent une capacité mémoire de l'ordre de 128Ko; cette valeur atteint environ un mégaoctet grâce à la technologie FLASH [Uri00]. Ces dispositifs offrent des performances satisfaisantes (temps d'authentification de l'ordre de la seconde) et des capacités de stockage confortables pour la gestion de plusieurs réseaux.

L'ajout de cartes à puce dans les architectures sans fil [Uri02] introduit un niveau de sécurité supplémentaire puisque l'utilisateur n'a pas accès aux clés cryptographiques requises pour son authentification. De manière analogue au réseau GSM la puce est la propriété d'un fournisseur de service réseau (entreprise, administration, opérateur, …), donc il est difficile de cloner un tel composant.

7-5. Les systèmes de détection d'intrusions IDS :

L'utilisation de détecteurs d'intrusion dans les réseaux Ad hoc est une solution complémentaire faisant l'objet de recherches intensives [Alb02].

Les systèmes de détection d'intrusions proposés pour un réseau Ad Hoc sont en général de trois catégories différentes : les IDS individuels, les IDS coopératifs et les IDS hiérarchiques.

Pour les IDS individuels chaque noeud a son propre système et détecte des attaques indépendamment dans cette architecture. Il n'y a aucune coopération entre les noeuds et toutes les décisions sont basées sur des informations collectées par différents nœuds.

Les IDS coopératifs interviennent puisque les réseaux ad hoc sans fil sont distribués et basés sur la coopération des noeuds, la détection d'intrusion et le système de réponse devraient naturellement être répartis également et coopérative [Put04]. Dans cette architecture, chaque noeud a un agent d'identification et fait décisions de détection locale par lui-même. En même temps, tous les noeuds participent à un procédé global de détection.

Dans les réseaux ad hoc sans fil multicouche les noeuds sont divisés en groupes. Pour répondre aux conditions de ce type d'architecture les IDS hiérarchiques sont proposés, où chaque noeud a son propre agent IDS responsable de la détection locale d'intrusion. En même temps, l'agent IDS du chef de groupe est responsable de la détection d'intrusion locale et globale. La surveillance totale de réseau est assurée en activant les agents globaux dans chaque chef de groupe.

Conclusion :

Dans ce chapitre nous avons présenté qu'est ce qu'un réseau ad hoc, ces caractéristiques, la norme qui suive, le problème de routage et les différents protocoles permettant l'acheminement des données dans cet environnement. Pour avoir une idée sur les concepts de ces réseaux et permettre ensuite de comprendre leurs mécanismes de fonctionnement afin de savoir les problèmes que les réseaux ad hoc peuvent rencontrer, en termes de déploiement, de routage, de sécurité…

Nous avons abordé également le problème de sécurité au niveau des réseaux ad hoc, les types d'attaques ainsi que les solutions proposées dans la littérature, ces approches résolvent certaines parties des besoins en termes de sécurité, mais aucune d'entre elles ne propose une solution satisfaisant l'ensemble de toutes les spécificités exigées par de tels réseaux, celles-ci étant délicates à satisfaire en totalité.

Dans ce mémoire nous allons proposer une méthode pour la sécurité des réseaux ad hoc, en essayant de participer à résoudre le problème de vulnérabilité de ce type de réseaux, l'explication de l'approche proposée, ses détails ainsi que ses performances sont exposées dans les chapitres qui suivent.

chapitre 3

Approche de sécurité pour les réseaux ad hoc

Introduction :

Plusieurs solutions sont proposées pour sécuriser les réseaux de type ad hoc, parmi celles ces on trouve les approches basées sur le principe de groupement (clustering). En effet, un réseau ad hoc (également connu sous le nom de MANET) a commencé à attirer l'attention ces dernières années, il se compose des noeuds mobiles qui se déplacent librement et communiquent les uns avec les autres avec des liens sans fil, ce type de réseaux n'exige aucune infrastructure fixe et peut être installé n'importe où. L'approche de clustering consiste à partitionner le réseau en un certain nombre de groupes (clusters), plus homogènes selon une métrique spécifique ou une combinaison de métriques, et former une topologie virtuelle. Les clusters sont généralement identifiés par un noeud particulier appelé cluster-head (Chef de groupe), et est élu pour jouer ce rôle selon une métrique bien définie ou une combinaison de métriques [Leh09]. Ce dernier permet de coordonner entre les membres de son cluster on en lui offrant des tâches particulières à réaliser comme la gestion de la sécurité dans le réseau.
Dans ce travail, nous envisageons un algorithme de sécurité basé sur le principe de clustering pour les réseaux ad hoc qui adapte une approche auto organisée d'attribution de certificats aux clés publiques.

1- L'organisation dans les systèmes distribués :

Le principal problème dans les systèmes dont l'administration est centralisée, est que si le serveur ou la station qui gère le réseau est « conquis », tout le réseau sera à la disposition entière du tiers qui arrive à accéder au système. Pour cela, intervient la nécessité de former des systèmes dont la gestion des différentes tâches est distribuée aux éléments du système.
Donc, un système distribué peut être défini comme étant un ensemble de processus, machines, objets, ou de matériels pouvant communiquer via un réseau par échange de messages. Un tel système n'a ni horloge globale, ni mémoire commune, et est utilisé pour accomplir un travail coopératif [Abd00].

D'une manière générale, l'organisation est un modèle permettant aux composants d'un système distribué de coordonner leurs actions au cours de la résolution d'une ou de plusieurs tâches [Ben07]. Elle définit d'une part une structure hiérarchique comprenant un ensemble de rôles qui doivent être attribués aux composants et un ensemble de chemins de communication entre ces rôles. Elle définit d'autre part un régime de contrôle (ex: une relation maître esclave) qui dicte le comportement social des composants. Enfin, elle définit des processus de coordination qui déterminent la décomposition des tâches en sous tâches, l'allocation des sous tâches aux composants, et la réalisation des tâches de façon cohérente.

2- L’avantage d'organiser un système en groupes :

Le concept de groupes nous permet de définir un groupe d'entités comme une seule entité virtuelle, et cela permet d'attribuer le même nom à chaque membre d'un groupe particulier, et de communiquer avec eux en utilisant la même adresse [Ben07]. Dans un système distribué en général, la communication entre les noeuds, le transfert d'objets ou de processus, la prise de décisions sont des problèmes qui ne peuvent pas être résolus pour tous les noeuds. Pour remédier à ces problèmes, on a recourt à la formation de sous ensembles d'entités appelés groupes, clusters ou partitions. Ces groupes sont composés de membres. Dans chaque groupe, un des membres joue un rôle particulier, et est appelé leader, head, manager ou chef. Parmi les différentes tâches qui peuvent être accordées à un chef de groupe, on peut citer :

- Il est responsable de la communication entre les différents membres en recevant l'information et en la renvoyant aux autres membres.
- Il contrôle l’organisation interne du groupe.
- IL gère la sécurité au niveau du groupe, et participe, avec les autres chefs de groupes, à la sécurité du système entier.

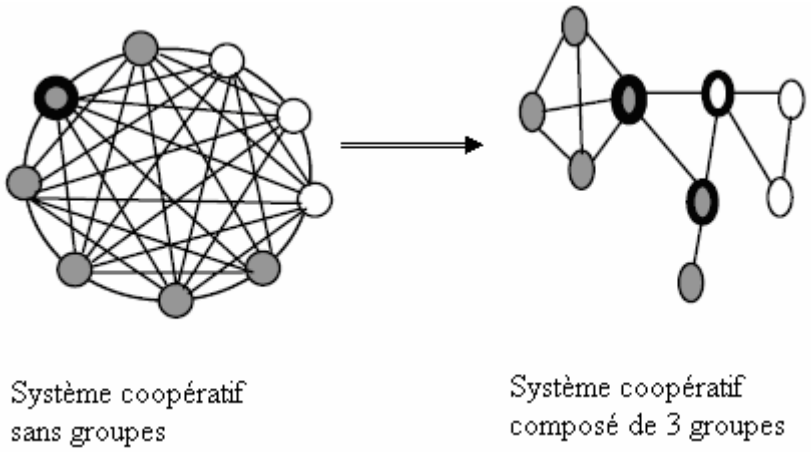


Fig.1 Formation de groupes [Ben07].

3- Quelques approches de clustering :

Dans la littérature, il existe de nombreuses propositions pour «clusteriser» les réseaux ad hoc. Toutes ces solutions sont destinées à identifier un sous ensemble des noeuds du réseau (appelé cluster). Les clusters sont identifiés par leur Cluster-head. Si le cluster-head disparaît, le cluster n’est plus valide. Les différents algorithmes se distinguent sur le critère de sélection

des cluster-heads. Les premiers algorithmes sont Lowest-ID et Mobic, dans [Eph87], les auteurs ont proposé un algorithme "Plus Petit ID" (Lowest-Identifier ou Lowest-ID) pour la construction des clusters, où chaque noeud se déclare cluster-head ou non en se basant sur son identifiant et ceux de ses voisins, ainsi que sur le statut de ses voisins comme il est illustré sur la figure 2. Dans l'algorithme "Plus Petit ID", un noeud peut avoir quatre statuts : ordinaire, cluster-head, membre, ou passerelle. Au début, tous les noeuds ont un statut de noeud ordinaire. Si un noeud *u* possède le plus petit identifiant dans son 1-voisinage, il se déclarera comme cluster-head et ses 1-voisins dont les identifiants sont supérieurs à celui du cluster-head le joignent et deviennent des noeuds membres. Sinon, il attendra que tous ses 1-voisins déclareront leurs statuts. Ainsi si un parmi eux se déclare cluster-head alors le nœud *u* déclare à son 1-voisinage son statut de noeud membre. Si tous les voisins du noeud *u* ayant un identifiant plus petit que celui de *u* qui a le statut de noeud membre, alors le noeud *u* se déclarera cluster-head. Une fois que tous les noeuds ont soit le statut de membre ou de cluster-head, alors si un nœud, parmi ses 1-voisins, a plus d'un cluster-head, il se déclarera passerelle.

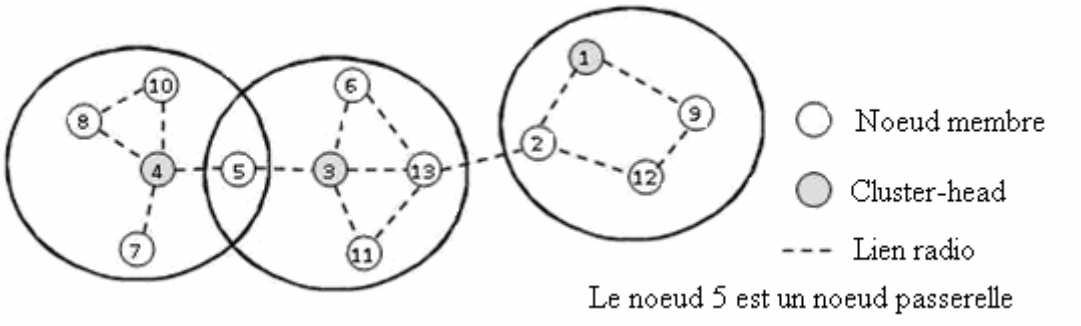


Fig.2 Formation de clusters basée sur ID.

Basu, Khan et Little dans [Kha01] ont proposé un algorithme de clustering distribué appelé MOBIC (Lowest Relative Mobility Clustering Algorithm). Cet algorithme implique la mobilité relative des noeuds pour structurer le réseau en clusters. La mobilité relative d'un noeud représente le rapport des niveaux de puissance des transmissions successives reçues par un noeud de ses voisins. Ainsi, le noeud ayant la plus faible mobilité dans son voisinage sera un bon candidat pour être cluster-head puisqu'il gardera un voisinage plus stable au cours du temps et favorisera la stabilité des clusters.

Dans [Lin97], les auteurs présentent un mécanisme de Clustering qui permet de réduire l'over-head de Clustering. Chaque noeud ne diffuse qu'un seul message pendant la phase de

formation des domaines, toutefois, l'hypothèse d'absence de mobilité pendant la formation des Clusters doit être vérifiée. En outre, le mécanisme de clustering proposé s'affranchit de la notion des Cluster-heads et ne traite pas le cas où ces derniers quittent le Cluster.

L'algorithme "Distributed and Mobility Adaptive Clustering", présenté dans [Bas99] et [Sid02] ont proposé un mécanisme de Clustering qui permet de réagir aux changements de topologie. L'algorithme ne nécessite aucune synchronisation entre les noeuds. Pour améliorer la stabilité des Clusters formés, deux nouveaux facteurs de performances ont été définis. Le premier, *K*, autorise au maximum *K* Cluster-heads à être voisins directs. Le deuxième, *H*, permet de limiter les réaffiliations entre les Clusters. Les noeuds ne se ré-affilient à un nouveau Cluster-head que lorsque le poids de ce dernier est supérieur d'un certain facteur *H* au poids de leur Cluster-head courant. Toutefois, cette solution ne permet la formation que de Clusters à un saut et le facteur de performance *H* est difficile à spécifier de façon judicieuse.

Dans [Tur02], les auteurs ont présenté une formule multicritères pour les choix des Cluster-heads. Elle prend en considération la mobilité, la connectivité et l'énergie disponible. Ce mécanisme de Clustering nécessite, toutefois, une synchronisation globale et un échange de voisinage entre tous les noeuds du réseau.

Dans d'autres travaux [Noc02] et [Ami00], les auteurs ont essayé de présenter des algorithmes adéquats à la formation de Clusters à *K* sauts. Toutefois, [Ami00] gère la mobilité par réexécution périodique de tout l'algorithme. [Noc02] nécessite d'une part des informations sur le k-voisinage et d'autre part que les noeuds vérifient l'hypothèse de non mobilité pendant la phase de Clustering.

[Siv04] présente un mécanisme de Clustering basé d'une part sur la connaissance préalable de l'aire de déploiement du réseau et sur la capacité de se positionner et d'autre part sur la prédiction des mouvements des noeuds en considérant leur historique.

4- Mise en place de l'architecture proposée :

Pour la mise en place du mécanisme de sécurité envisagé, nous avons établi une architecture hiérarchique basée sur la division du réseau sous forme de clusters. De plus, pour assurer l'identification et l'authentification des nœuds du réseau, nous avons besoin d'un mécanisme qui assure ces exigences tout en respectant les caractéristiques des réseaux ad hoc. Donc l'utilisation d'une entité centralisée pour l'authentification et l'identification n'est pas adéquate à la dynamique du réseau, les techniques de la cryptographie symétrique ne présentent pas une bonne protection contre les vulnérabilités des réseaux ad hoc. Pour cela,

cette architecture va introduire la notion d'une infrastructure à base de clé publique auto organisée pour assurer les communications entre les nœuds du réseau.

4-1. Modèle d'Infrastructure à Clé Publique Auto Organisée :

Le principe de cette idée, développée dans [Hub01], se base sur des infrastructures à clé publique (public key infrastructure PKI), autogérées au sein du réseau ad hoc. Chaque noeud établit des certificats pour les noeuds en qui il a confiance. Lorsque deux éléments d'un réseau veulent communiquer sans connaissance au préalable l'un de l'autre, ils s'échangent leur liste de certificats et vont essayer de créer une chaîne de confiance entre eux. Supposons qu'un noeud *u* veuille communiquer avec un noeud *v*, si *u* fait confiance en un troisième élément *z* et *v* fait aussi confiance en *z*, alors une chaîne de confiance entre *u* et *v* pourra être établie via *z*. Dans ce cas, *u* peut donner sa clé publique à *z* pour la signer, puis il redonne la clé signée à *u* et en garde une copie. Quand *u* veut communiquer avec *v*, il lui envoie une copie de la clé que *z* a signée. Le noeud *v* qui a déjà la clé publique signée de *z* et qui fait confiance à *z* pour certifier les clés d'autres nœuds, vérifie sa signature sur la clé de *u* et l'accepte. De ce fait *z* a recommandé *u* à *v*.

4-1-1. Principe du modèle :

Pour gérer l'authentification et la confidentialité de la communication entre les nœuds et assurer ainsi la sécurité du réseau, un rôle pour l'autorité de certification CA doit être défini.
Dans ce but plusieurs solutions sont proposées, notre algorithme va introduire l'idée d'une infrastructure à clé publique auto organisée. En effet, le principe de cette technique est que les certificats sont délivrés par les utilisateurs eux-mêmes. Ceci sans participation de n'importe quelle autorité de certification centrale. A la différence des solutions à base de clé publique «classique», celle-ci est conçue pour les réseaux ad hoc, où les noeuds n'ont aucun rapport antérieur, la clé publique et la clé privée correspondante de chaque noeud sont créées localement par le noeud lui-même. Chaque noeud a les possibilités de certifier des clés publiques à d'autres noeuds. Si un noeud *u* croit qu'une clé publique donnée *kv* appartient à un nœud donné *v*, alors *u* peut délivrer un certificat de clé publique dans lequel la clé *kv* est liée à *v* par la signature de *u*.

Chaque nœud maintient un dépôt local de certificats, qui contient un nombre limité de certificats choisis selon un algorithme donné. Dans ce modèle, les clés publiques et les certificats du système sont représentés par un graphe orienté G (V, E), appelé graphe de confiance (trust graph). V et E représentent, respectivement, les sommets et les arcs du graphe, où les sommets représentent des clés publiques et les arcs représentent des certificats.

Une chaîne de certificats de noeud *u* vers un noeud *v*, est représentée par un chemin du sommet *u* vers le sommet *v* dans G. Ainsi, l'existence d'une chaîne de certificat de *u* au *v* signifie que le sommet *v* est accessible du sommet *u* dans G et on note ça par : $ku \rightarrow G.kv$
Le dépôt local de certificats de chaque noeud contient des certificats crées par le noeud lui-même, et des certificats sélectionnés crées par d'autres noeuds dans le système, les certificats sont publiés avec une période de validité limitée T_V, et chaque certificat contient ses temps de publication et d'expiration.

Chaque noeud possède un graphe local (représente le dépôt local). Quand *u* désir vérifier l'authenticité de la clé publique de *v*, *u* et *v* fusionnent leurs dépôts locaux (leurs graphes locaux), et *u* essaye de trouver une chaîne de certificats de *u* vers *v* dans le dépôt fusionné, si un tel chemin est trouvé, *u* vérifie l'exactitude des certificats et effectue l'authentification. S'il n'y a aucun chemin de *v* vers *u*, *u* n'authentifie pas la clé de *v*.

Si, un nœud détecte un comportement «malhonnête» d'un autre nœud du réseau, il peut retirer le certificat qu'il l'a délivré et publie ainsi un rapport aux autres nœuds du réseau de la révocation du certificat qui a fait, ce qui permet à chaque noeud de réagir à n'importe quelle mauvaise conduite détectée et d'effectuer l'authentification avec une confiance plus élevée.

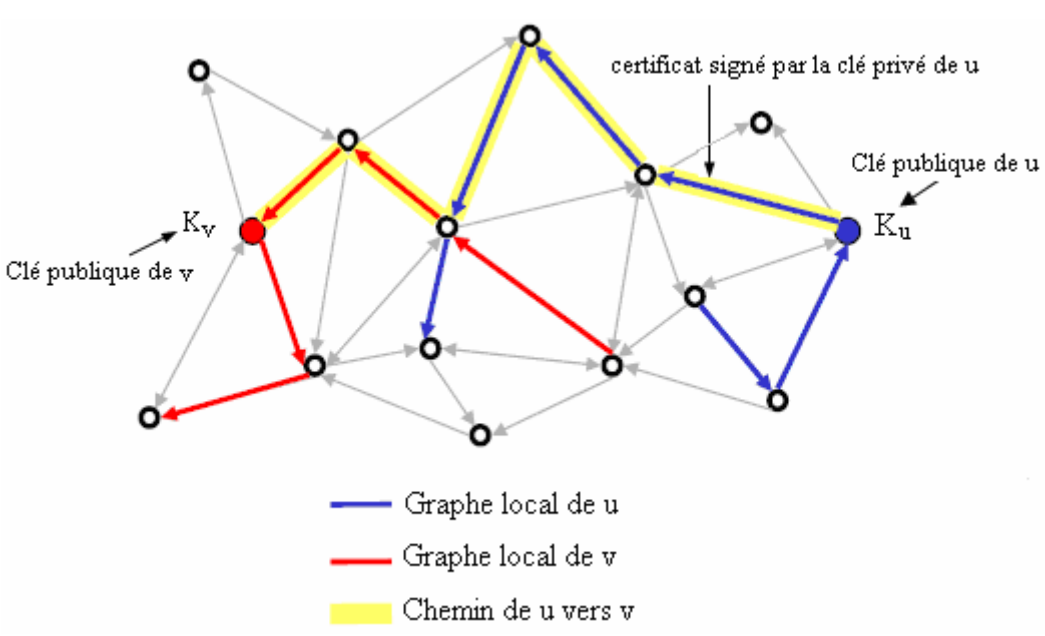


Fig.3 Chemins des certificats entre les noeuds *u* et *v* dans leurs dépôts locaux fusionnés [Hub01].

4-1-2. Procédure d'échange de certificats :

Dans la section précédente nous avons mentionné que chaque nœud maintient un dépôt local de certificats, qui contient un nombre limité de certificats choisis selon un algorithme

donné. En ce qui suit nous allons donner une méthode avec laquelle les nœuds produisent leur dépôt local.

Dans une première phase du système, chaque noeud tient dans son dépôt local seulement les certificats qu'il a publiés et les certificats que d'autres noeuds ont lui fournis. Ici, nous supposons qu'à chaque fois qu'un noeud *u* délivre un certificat qui lie une clé publique *kv* au noeud *v*, *u* envoie le certificat au *v*, de cette façon, chaque certificat est stockés au moins deux fois, par son émetteur et par le noeud à qui il est fourni.

La procédure d'échange de certificat consiste en échange périodique des certificats entre les noeuds. Chaque noeud a un compteur de temps local et informe périodiquement les autres noeuds pour des certificats. Pour chaque noeud, il y a une fréquence prédéfinie F_E à laquelle il effectue l'échange de certificat. Pour la simplicité, nous supposons que chaque noeud échange des certificats avec la même fréquence d'échange F_E. Nous notons également que les noeuds ne courent pas l'échange synchroniquement. L'échange de certificat est effectué de la façon suivante.

Chaque noeud *u* diffuse son graphe local aux autres nœuds. Dans ce message, *u* n'envoie pas les certificats réels mais seulement les identités appropriées (par exemple, leurs valeurs de Hachage), Les nœuds qui reçoivent le message de *u* répondent par les valeurs de Hachage des certificats dans leurs dépôts, en suite le noeud *u* vérifie les valeurs reçues avec les certificats qu'il tient et demande aux autres seulement les certificats qu'il n'a pas dans son dépôt.

Pour libérer l'espace de stockage, les noeuds suppriment les certificats expirés basés sur leur temps d'expiration. De plus, pour minimiser la charge de communication du réseau, les noeuds échangent seulement les certificats créés récemment, et non pas les certificats qui sont déjà stockés.

4-2. L'algorithme de clustering pondéré :

Dans cette section nous décrivons l'algorithme de clustering pondéré pour la formation et la stabilité des clusters en présence de la mobilité des noeuds, de telle sorte que les propriétés de clustering multi saut suivantes sont satisfaites :

- L'algorithme est dynamiquement adapté à tout changement de topologie.
- Chaque cluster-head est le chef d'un seul cluster.
- Pour chaque Cluster, il existe un seul cluster-head.
- Seulement les noeuds dont le poids W=1 (nœuds de confiance) qui peuvent être candidats d'être cluster-head et ont le rôle de contrôler le comportement de chaque noeud ayant un poids faible au sein du cluster.

- Les noeuds qui appartiennent au cluster doivent être à une distance maximum de k sauts du noeud cluster-head.

Pour la sélection des cluster-head on suppose que le réseau est pondéré, c.-à-d., un poids W_v (tel que $W_v \in]\,0, 1]$) est attribué à chaque nœud *v*, et que chaque noeud *v* dans le réseau ad hoc possède une identité et une paire de clés (Privée/ publique).

Le choix d'un cluster-head est basé sur le poids associé à chaque noeud : Plus le poids d'un noeud est grand, plus ce noeud a de chances pour jouer le rôle d'un cluster-head.

Initialement, les nœuds se connaissent entre eux (par l'identité et la clé publique) et ils sont considérés comme des nœuds honnêtes qui ne doivent pas générer des faux certificats (les noeuds de confiance).

Seuls les nœuds qui possèdent le poids le plus élevé ($W_v = 1$) qui peuvent être candidats à être cluster-head.

Un nœud *v* possède un poids le plus élevé, s'il est connu par d'autres nœuds de confiance et a échangé les clés via un canal sécurisé (par exemple, au-dessus d'un canal infrarouge à l'heure d'une rencontre physique [Cap03] avec un ou plusieurs noeuds de confiance).

Chaque nœud inconnu commence avec le poids le plus bas ($W_v = 0.1$), l'idée de ce principe consiste à obliger les nœuds inconnus à coopérer et bien de se comporter.

Si un nouveau nœud est ajouté à la liste des nœuds de confiance par un ou plusieurs nœuds de confiance, les autres nœuds doivent mettre à jour leurs listes des noeuds de confiance.

4-2-1. Calcul du poids *W :*

Comme on a cité plus haut que les nœuds de plus grand poids contrôlent ceux de faible poids, dans cette section, on va présenter les critères avec lesquels les cluster-heads attribuent des poids aux noeuds.

Nous définissons une valeur de recommandation *R* (*u, v)* d'un nœud *v* effectue par le nœud *u* sur le graphe de certificat G comme le rapport de nombre des paires de clé (*Ku, kv*) pour lequel il y a un chemin dirigé de *Ku* au *kv* dans leur graphe local fusionnés, sur le nombre des paires de clé (*Ku, kv*) pour lequel il y a un chemin dirigé de *Ku* au *kv* dans le graphe de certificat G, tel que *u* est un nœud de confiance.

$$R(u,v) = \frac{\left|\left\{ (Ku, Kv) \in V \times V : Ku \to Gu \cup Gv.Kv \right\}\right|}{\left|\left\{ (Ku, Kv) \in V \times V : Ku \to G.Kv \right\}\right|} \qquad \text{(Eq 3.1)}$$

Par conséquent, la recommandation *R* (*u, v)* exprime le rapport de nombre des noeuds *u* qui certifient le nœud *v* dans leur graphe local sur le nombre total des noeuds de confiance dans *G* comme s'ils sont tous certifient le nœud *v*.

Outre du critère basé sur la recommandation des nœuds dans le graphe de certificat G, on va introduire un critère de réputation qui évalue la coopération et le bon comportement des nœuds dans le réseau et détecte les nœuds malveillants.

Afin d'empêcher certains nœuds d'occuper le canal de communication (pour régler le problème des nœuds égoïstes), les nœuds chargés du contrôle génèrent un coefficient N1 $\in$ [0, 1] sur ses voisins qui ont un plus faible poids, en calculant la durée de l'occupation du canal par les nœuds, si cette durée dépasse un certain seuil, le nœud en question sera puni en lui diminue le coefficient N1.

Dans le but de juger la coopération et le bon comportement des nœuds dans le réseau, les nœuds chargés du contrôle observent le comportement de ses voisins par rapport à une fonction de transfert de paquets (packet forwarding), et collecte des informations sur l'exécution de cette fonction. L'idée consiste à calculer pour chaque nœud la proportion de paquets bien retransmis par rapport au nombre total de paquets devant être transmis sur une certaine période. Soient deux noeuds *u* et *v* avec $W_u > W_v$, dans ce cas, le noeud *u* peut contrôler le noeud *v* en calculant le rapport N_2 de paquets acheminés sur le nombre total des paquets devrant être transmis de *v* vers *u*.

En suite une moyenne N (*u*, *v*) des deux rapports N1 et N2 est calculé pour évaluer la réputation globale d'un nœud *v* effectue par le nœud contrôleur *u*.

Chaque nœud de contrôle *u* envoie les deux valeurs d'évaluation N (*u*, *v*) et R (*u*, *v*), qui à effectuer sur le nœud *v*, au cluster-head, ce dernier puisse donc affecter un poids au nœud *v* par la formule suivante :

$$W(v) = \frac{1}{2k} \sum_{i=1}^{k} \left(N(u,v) + R(u,v) \right) \qquad \text{(Eq 3.2)}$$

Suivant cette valeur le cluster-head peut effectuer un classement des nœuds (nœud membre, passerelle…) et ceux qui ont un poids élevé seront classés comme nœuds de confiance et auront la possibilité d'être élus comme cluster-heads selon les conditions de sélection.

4-2-2. Calcul de la mobilité :

Dans la littérature, certains algorithmes de clustering ne gèrent pas la mobilité des nœuds dans la phase de formation des clusters. Pour remédier à cet inconvénient, on va introduire le

critère de la mobilité relative des nœuds comme paramètre de sélection des cluster-heads pour assurer la stabilité des clusters (et augmenter ainsi leur durée de vie).

A cette fin on a recours au critère utilisé par [Bas01] qui consiste à mesurer la mobilité relative entre deux nœuds voisins par le calcul d'un rapport M_v (u) de puissance du signal de deux messages « Hello» reçus successivement entre les noeuds *u* et *v*.

$$M_v(u) = 10 \log_{10} \frac{P^{new}_{u \to v}}{P^{old}_{u \to v}} \qquad \text{(Eq 3.3)}$$

Si $P^{new}_{u \to v} < P^{old}_{u \to v}$, donc $M_v(u) < 0$, et une valeur négative de la mobilité relative métrique entre deux noeuds quelconques indiqueront que les deux noeuds s'écartent l'un de l'autre. D'autre part si $P^{new}_{u \to v} > P^{old}_{u \to v}$, alors $M_v(u) > 0$, et celui indique que les noeuds se rapprochent l'un de l'autre. Pour un noeud avec n voisins, on a n valeurs pour le M_v.

Donc, le calcul de la mobilité relative M_v de *v*, par apport à tous ses voisins, est déduit en calculant la variance de l'ensemble entier des valeurs de la mobilité relative $M_v(u)$ où u_i est un voisin de *v* :

$$M_v = Var(M_v(u_1), M_v(u_2), M_v(u_3), ..., M_v(u_n)) \qquad \text{(Eq 3.4)}$$

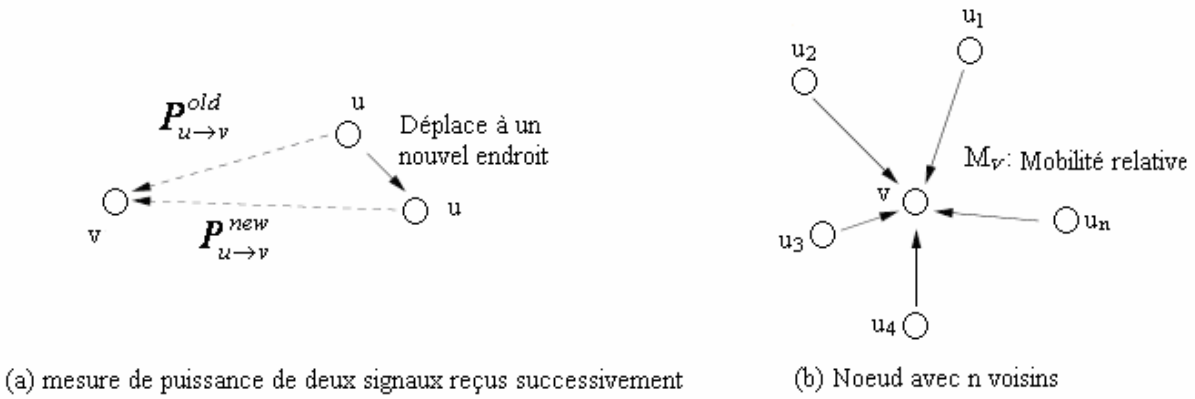


Fig.4 Calcul de la mobilité relative globale [Bas01].

Ici, M_v dénote la variance qui est la valeur prévue des places des n mobilités relatives provenant des voisins du *v*.

De cette manière, n'importe quel nœud *v* mesures les niveaux de puissance dans les transmissions successives de tous ses voisins et une variance de ces valeurs est une valeur représentative pour la mobilité relative globale M_v pour ce noeud. Le principe

précédant calcule la variance des valeurs des mobilités relatives de chaque voisin est qu'une valeur basse de M_v indique que *v* est relativement moins mobile par apport à ses voisins. Au contraire, des valeurs élevées de M_v indiquent que *v* est fortement mobile par apport à ses noeuds voisins. A l'aide de ce principe, on devrait favoriser les noeuds qui ont des petites variances des mobilités relatives par apport à ses voisins, pour le choix des cluster-heads.

Noter que la variation de la puissance du signal reçu est un meilleur indicateur de la mobilité que la distance ou la vitesse des nœuds à cause des caractéristiques de l'environnement. Par exemple, la communication entre deux nœuds voisins dans un milieu hostile peut souffrir d'une atténuation élevée de la puissance du signal que celle entre deux nœuds qui sont plus loin, mais dans un milieu propre.

Après le calcul des poids et des mobilités relatives, les nœuds de confiance (seuls qui ont le droit d'être élus comme cluster-head) entrent en compétition pour la sélection des cluster-heads. En formant les différents clusters, le réseau est maintenant capable de gérer des communications entre ses entités avec un minimum de risque d'avoir des intrus qui gênent son fonctionnement normal. Cette caractéristique de sécurité est assurée par le cluster-head et les autres nœuds de confiance qui jouent le rôle des contrôleurs en s'appuyant sur la vérification des certificats octroyés aux nœuds pour les identifier et par conséquent luter contre des nœuds malveillants.

5- Génération des certificats :

Pour assurer l'authentification des nœuds dans le réseau, un service de gestion de clés doit être mis en place. L'approche la plus connue au problème de gestion des clés publiques est basée sur la génération des certificats aux nœuds porteurs des clés. Un certificat de clé publique est une structure de données en laquelle une clé publique est liée à une entité, afin d'en assurer sa validité par la signature numérique de l'émetteur du certificat (Autorité de certification CA). Ce mécanisme permet aussi de garantir l'intégrité et la non répudiation.

Initialement, la clé publique K_p et la clé privée (ou secrète) K_s correspondante de chaque utilisateur sont créées localement par l'utilisateur lui-même. Et à l'installation du réseau les nœuds peuvent échanger leur clé publique à travers un canal sécurisé (par exemple par une connexion infra rouge).

La délivrance de certificats se fait comme suit :

1- Lorsqu'un utilisateur *v* veut obtenir un certificat, il envoie une requête incluant sa clé publique (K_p) à l'autorité de certification CA.

2- La CA génère un certificat pour *v* et en applique une fonction de hachage (c'est la méthode SHA qui est utilisée dans notre travail) pour obtenir l'empreinte digitale du certificat, en suite, la CA signe le résultat obtenu par sa clé privée, K_{SCA}. La structure de données résultante s'appelle le certificat signé de la clé publique de l'utilisateur *v*, en suite CA renvoie le certificat à l'utilisateur *v* et garde une copie pour elle.

3- Pour que l'utilisateur s'authentifie auprès d'un autre nœud de confiance, il l'envoie son certificat pour prouver et vérifier son identité.

L'opération de vérification se fait de la façon inverse. Le nœud vérificateur applique une fonction de hachage au certificat, reçu en claire d'une part et déchiffre la signature avec la clé publique de CA générateur de ce certificat d'autre part, si les deux résultats sont identiques, alors le porteur de la clé incluse dans le certificat est authentique.

5-1. Format du certificat :

Les principales informations inclues dans un certificat sont :

- Numéro de série du certificat : C'est une valeur assignée par CA qui a émis ce certificat pour assurer l'unicité de la valeur de chaque certificat émis.
- Désignation du générateur de certificat : Ce champ identifie l'autorité de certification qui a délivré le certificat.
- Période de validité : identifie la date de début et la date de fin de validité d'un certificat. En dehors de cet intervalle, le certificat n'est plus considéré comme valide.
- Identité du porteur de la clé publique : Ce champ identifie l'identité du propriétaire de la clé publique à certifier.
- Algorithme de chiffrement et valeur de la clé publique : Ce champ contient la clé publique de l'entité à authentifier. Il identifie aussi l'algorithme asymétrique à utiliser, ainsi que tout autre paramètre utile à cet algorithme.
- Identification de l'algorithme de signature et valeur de la signature : Ce champ contient l'identifiant de l'algorithme (fonction de hachage) utilisé par CA pour signer le certificat, ainsi que la valeur de la signature numérique.
- Statut de certificat : Ce champ indique si le certificat est révoque ou non.

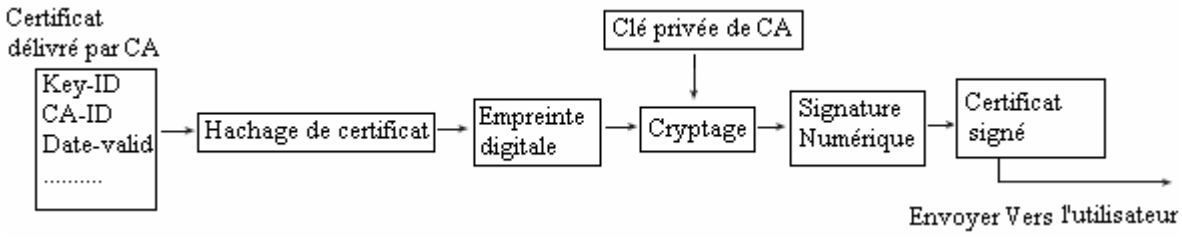


Fig.5 Génération de certificat.

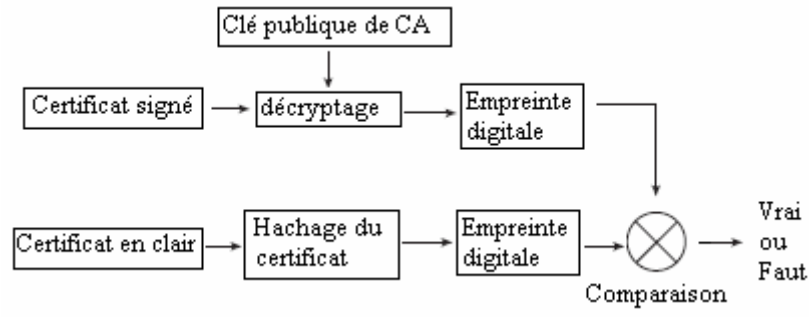


Fig.6 Vérification d'un certificat.

6- Tester la robustesse de l'algorithme contre les attaques :

L'algorithme de clustering proposé a pour but de garantir un niveau de sécurité acceptable contre certaines attaques destinées à perturber le bon fonctionnement des réseaux ad hoc, donc pour vérifier son efficacité et évaluer sa robustesse, différents scénarios d'attaques sont introduits dans le réseau simulé, ces types de scénarios sont mentionnés comme suit :

6-1. Attaques visant à falsifier les clés et les certificats :

Un nœud malhonnête peut essayer de tromper d'autres noeuds du réseau en les fait croire qu'un nœud du réseau ressemble à un faux nœud, en lui délivrant des faux certificats, il peut publier plusieurs types de certificats faux. D'abord, il peut délivrer un certificat qui lie une clé k_v à un nœud *w* au lieu du nœud *v*, de cette façon, un nœud malhonnête peut tromper d'autres nœuds pour croire que k_v est la clé publique de l'utilisateur *w*, quand en réalité c'est la clé publique du nœud *v*.

En second lieu, il peut délivrer un certificat qui lie le nœud *v* à une fausse clé k'_v, qui peut alors faire croire d'autres nœuds que k'_v est en effet la clé du nœud *v*. troisièmement, un utilisateur malveillant peut inventer un certain nombre de noms d'utilisateurs et de clés publiques et les lier par les certificats appropriés et dans ce cas, le nœud malveillant peut employer ces clés publiques pour délivrer des faux certificats et essayer de convaincre un nœud donné que les certificats sont corrects, comme si, ils ont été signés par d'autres nœuds de confiance.

6-2. Les attaques provenant de l'intérieur du réseau lui-même :

Un autre problème peut être envisagé, dans le cas où un nœud dans le réseau qui est déjà certifié, mais il présente maintenant un comportement « soupçonneux ». Lorsqu'un nœud du réseau essaye d'épuiser les ressources des autres nœuds (bande de transmission et batteries) par émission d'un grand nombre de requêtes, ou bien d'absorber le trafic de données et ne répondre pas aux requêtes des autres nœuds (problème des nœuds égoïstes dans le réseau).

6-3. Les attaques de déni de service DOS :

Dans ce type d'attaque, l'attaquant inonde le réseau d'un grand nombre de messages afin d'occuper (ou de saturer) la bande de transmission et de gaspiller l'énergie des nœuds récepteurs dans le but d'empêcher les nœuds légitimes du réseau de réaliser leurs communications utiles. L'intérêt essentiel de cette action malveillante est de faire tomber le réseau en panne.

Conclusion :

Dans ce chapitre, nous avons présenté le principe de l'approche proposée pour la sécurité des réseaux ad hoc, cette approche permettant de rendre un réseau ad hoc mobile résistant à la présence de nœuds malveillants. La méthode est basée sur la vérification de l'identité des nœuds participants à la communication dans le réseau, la vérification repose essentiellement sur l'affirmation de l'exactitude des certificats attribués aux nœuds. Le principal apport de cette solution est de mettre en oeuvre la formation des groupes ainsi que la vérification des certificats dans le réseau ad hoc mobile, sans pour autant nécessiter de synchronisation entre les nœuds ni d'ignorer l'état mobile des nœuds. Ces caractéristiques font que la méthode est aisément intégrable et que la stabilité des groupes formés est assurée.

De plus, le moyen de prouver l'efficacité de cette approche est de la tester dans un environnement hostile en injectant des nœuds qui se comportent comme des attaquants et vérifier ce que cette solution va bien gérer une telle situation ou non.

chapitre 4

Développement et performances

Introduction :

Le caractère dynamique de la topologie, la nature distribuée des algorithmes et les nombreux paramètres influant sur les conditions de fonctionnement du réseau font que l'étude analytique de l'impact d'une solution est souvent trop compliquée. Dans ces situations, nous avons recours à la simulation qui, loin de fournir une quelconque preuve de correction ni même une mesure fiable des performances, permet d'avoir un ordre d'idée du comportement du système par comparaison à un système similaire dans des conditions sinon identiques.

Dans le cas de notre travail, outil d'évaluation et de perfectionnement c'est bien la simulation. En effet, ce chapitre présente le modèle utilisé, les différents aspects des performances sur lesquels s'est portée notre attention.

Nous commençons par une introduction de l'outil de simulation. Ensuite, les étapes d'implémentation des différentes parties de la solution, les contres mesure, la capacité d'identifier les noeuds malveillants et la gestion des attaques injectées dans les simulations seront présentés à la fin du chapitre.

1- Environnement de Simulation :

1-1. Le simulateur de réseau :

Pour évaluer les performances de l'algorithme de sécurité présenté dans le chapitre précédant, nous avons utilisé comme plate-forme d'implémentation le simulateur NS (la version ns-allinone-2.30) [Ber09] pour générer les différents scénarios de simulation. En effet, le simulateur NS-2 est très sollicité et le plus populaire dans le domaine de la simulation des réseaux. NS-2 (Network Simulator) est un simulateur à événements discrets disponible gratuitement sur le site http://www.isi.edu/nsnam/.

Il est développé en C++ avec une interface textuelle utilisant le langage OTcl (Object Tool Command Language) qui est une extension objet du langage de commande TCL (Tool Command Language).

À travers ce langage, l'utilisateur décrit les conditions de la simulation : topologie du réseau, caractéristiques des liens physiques, protocoles utilisés, communications... La simulation doit d'abord être saisie sous forme de fichier texte que NS utilise pour produire un fichier trace contenant les résultats. NS est fourni avec différents utilitaires dont des générateurs aléatoires et un programme de visualisation : NAM.

1-2. L'outil d'animation :

Le Nam (Network Animator) est un outil d'animation basé sur Tcl/TK, utilisé dans NS afin de visualiser le tracé de simulation des réseaux, ainsi que les tracés de données échangées. L'outil d'animation Nam a été non seulement créer pour lire un large ensemble de données d'animation, mais il est aussi suffisamment extensible pour être utilisé quelque soit le type de réseau simulé (*fixe, mobile ou mixte*). Ce qui permet de visualiser tout type de situation possible.

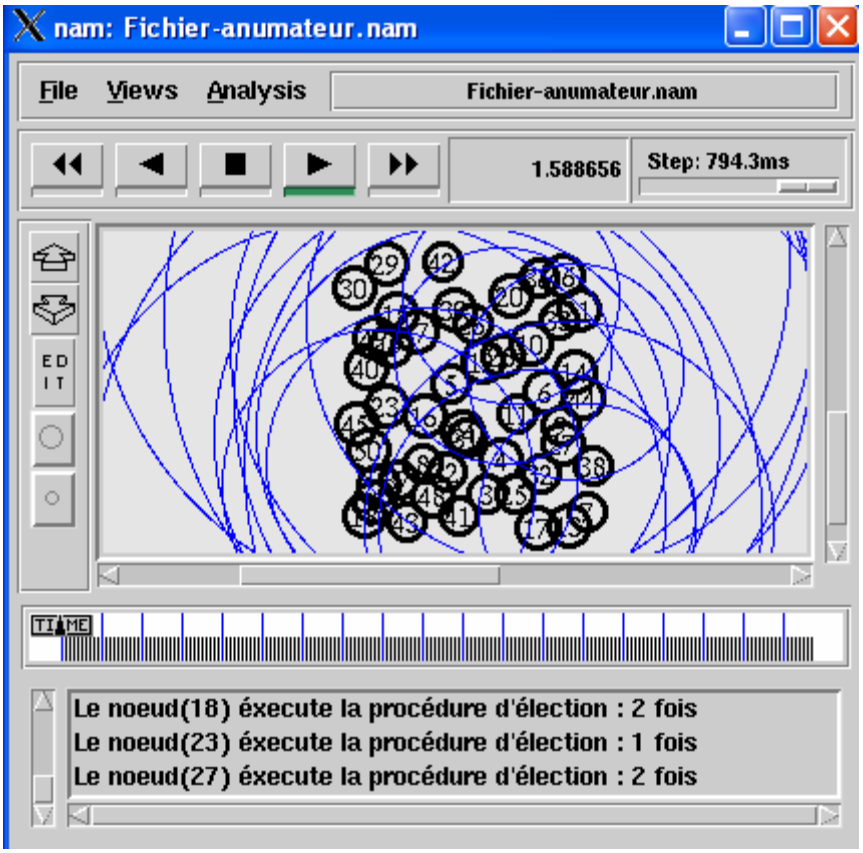


Fig.1 Visualisation du réseau par l'outil Nam

1-3. Le fichier Trace et obtention des résultats :

Nous obtenons les résultats de simulation à partir du fichier Trace (.tr) résultant du script TCL. Les fichiers de Trace incluent tous les événements dans la simulation telle que le temps d'envoie des paquets, le noeud qui les génère, le noeud qui les reçoit, le type des paquets envoyés, est ce qu'ils sont ignorés et pourquoi, l'énergie consommée etc.

2- Description de la simulation :

Le processus de simulation est composé de trois phases essentielles :

- Phase de préparation : s'occupe de la génération des fichiers d'entrées
- Phase de simulation : lance les simulations et génère les traces
- Phase d'analyse : Analyse les traces et génération des courbes

La figure 2 présente le processus général de la simulation sous ns.

Dans la première phase de préparation, les fichiers d'entrées de la simulation sont générés pour qu'ils soient utilisables par le programme principal (script de lancement implémenté en Tcl). Ces fichiers sont classés en trois catégories :

1. Fichiers de scénario qui décrivent les nœuds, leurs positions ainsi que leurs mouvements.
2. Fichiers de communication dans le réseau.
3. Quelques procédures et modifications au niveau de NS.

Une fois la simulation lancée en deuxième phase, elle prend comme entrée les trois types de fichiers et interprète le script de lancement écrit en Tcl, en suite le NS génère en sortie deux rapports principaux d'analyse simultanément. L'un d'entre eux est un fichier généré par outil NAM (animateur de réseau) qui montre l'animation visuelle de la simulation, et l'autre est un fichier journal appelé aussi le fichier trace qui comprend toutes les opérations et les transactions effectuées dans le temps au sein du réseau.

A ce moment, la phase d'analyse peut être débuté en prenant le fichier trace comme entrée pour le programme d'analyse.

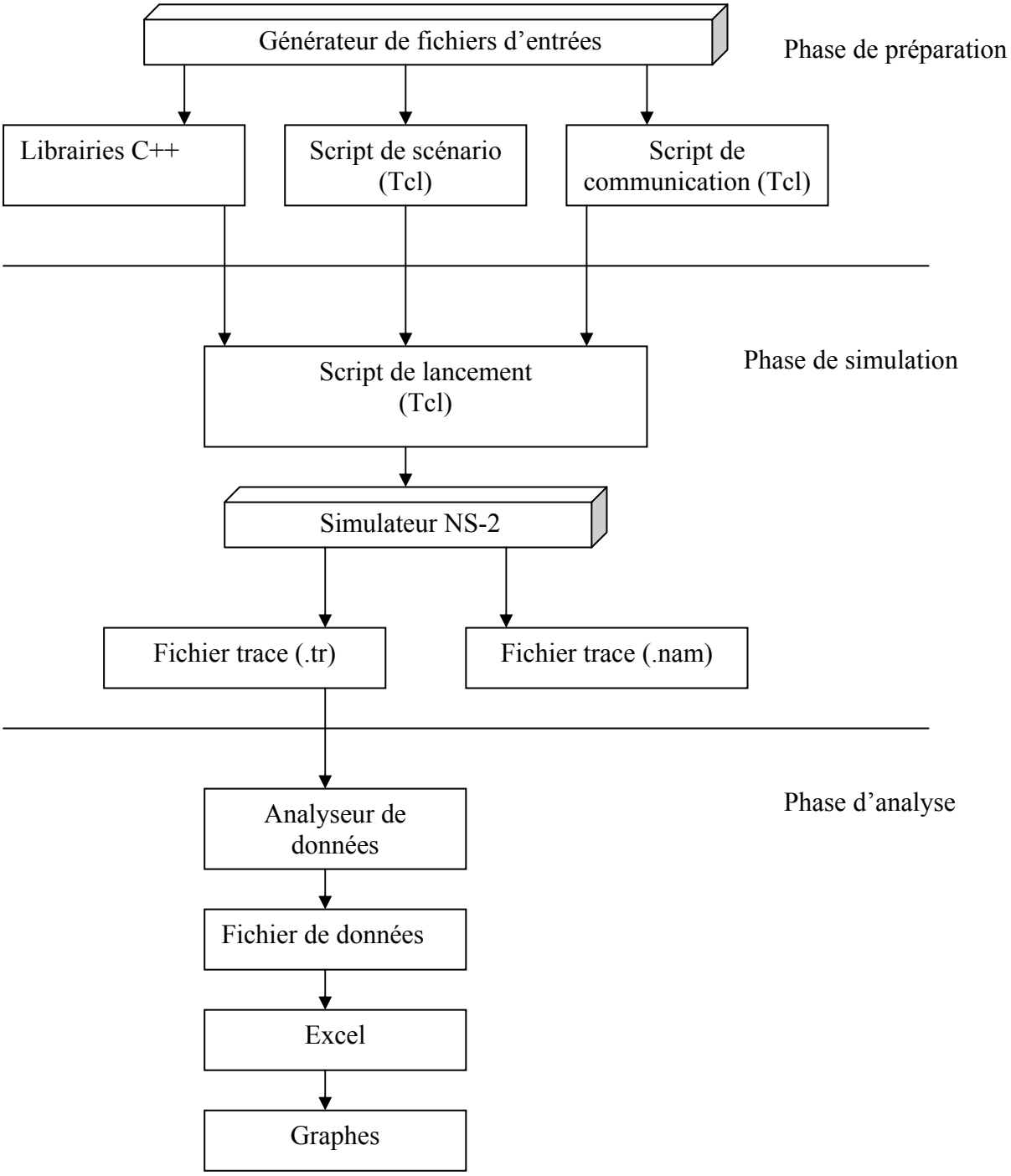


Fig.2 Processus général de simulation.

3- Paramètres de la simulation :

Dans cette partie, nous allons présenter les paramètres des différentes simulations, le modèle de topologie, la propagation du signal radio, le trafic des données, la mobilité des nœuds et le modèle de l'énergie utilisé par chacun des nœuds.

3-1. Le modèle de mobilité :

Dans l'environnement de simulation NS-2, nous retrouvons plusieurs types de modèles de simulation. Grâce à ces modèles, nous pourrons mesurer ce facteur important, qui est la mobilité. Parmi ces modèles de mobilités nous retrouvons

3-1-1. Random Waypoint Model (RWM) :

Dans ce modèle, la mobilité des nœuds est typiquement aléatoire. En effet, la destination et la vitesse de chaque noeud mobile, désirant se déplacer, est aléatoire, et est limité à un intervalle bien déterminé. Après son déplacement le noeud mobile s'immobilise pour un temps fini (c'est le temps de pause), puis se déplace à nouveau de la même manière que la première fois, et cela jusqu'à la fin de la simulation [Nav04].

3-1-2. Random Direction Model (RDM) :

De la même façon que le modèle précédent (RWM), la destination ainsi que la vitesse du noeud sont choisies aléatoirement. Mais dans le modèle RDM, le noeud en déplacement doit atteindre les bornes de la surface de simulation, puis s'immobilise. Une fois le noeud immobile, et dans un intervalle de 180° par rapport à la position d'arrêt (borne atteinte), le noeud mobile peut entreprendre à nouveau son mouvement aléatoire.

3-1-3. Modified Random Direction Model (MRDM) :

Cette version modifier du RDM permet aux nœuds mobiles, en déplacement, de ne pas forcement atteindre les bornes de la surface de simulation [Lar98].

Notons que le modèle RWM reflète bien les caractéristiques des réseaux ad hoc. Car il offre une mobilité aléatoire aux noeuds mobiles appartenant au réseau ad hoc, contrairement aux modèles RDM et MRDM qui, d'une manière indirecte, conditionne le mouvement des noeuds. Cette constatation nous a poussés à choisir ce modèle (le RWM) pour notre simulation avec une vitesse aléatoire dans l'intervalle [0.2, 10] en mètres par seconde.

3-2. Le modèle de propagation radio :

Ce modèle est employé pour prévoir la puissance reçue de signal de chaque paquet. À la couche physique de chaque nœud sans fil, il y a un seuil de réception. Quand un paquet est reçu, si sa puissance de signal est au-dessous du seuil de réception, il est marqué comme erreur et est rejeté. Il existe dans NS le modèle de propagation radio « Friss-space » avec une

atténuation 1/r2 pour les petites distances. Nous avons pris l'autre modèle appelé « Two Ray Ground » qui emploie une atténuation de 1/r4 pour les grandes distances, Leur implémentation peut être trouvée dans ns-2.30 \ mobile/propagation.{cc, h}, et ns-2.30 \ mobile/tworayground.{cc, h}

Le type d'antenne radio NS choisie est « OmniAntenna » qui représente des antennes omnidirectionnelles où l'émission est en 360°. Tous les nœuds que nous avons pris ont la même configuration et les mêmes antennes d'où les mêmes rayons de propagation.

Le protocole de l'accès au médium MAC (Medium Access Control [Gum00]) que nous avons utilisé est le standard IEEE 802.11 développé par CMU. Le support de transmission a été configuré pour fonctionner comme l'interface radio du Lucent WaveLAN DSSS implémenté dans NS, avec une modification au niveau de la puissance de transmission pour donner un rayon de connexion radio approximatif à 250 mètres.

3-3. Le modèle d'énergie :

Le modèle d'énergie utilisé est celui implémenté dans NS, ce modèle représente le niveau de l'énergie dans un nœud mobile. Le modèle d'énergie dans un nœud a une valeur initiale qui est le niveau de l'énergie que le nœud possède au début de la simulation. Ceci est connu comme initialEnergy. Il a également une énergie d'utilisation pour chaque paquet en transmission ou en réception. Les fichiers où le modèle d'énergie est défini sont ns-2.30 \energy-model\mobile [.cc et .h].

3-4. Le modèle de topologie :

La topologie est déterminée par la manière de déplacement des nœuds et le nombre total de noeuds qui détermine la densité du réseau, Nous avons lancé des simulations pour un nombre total de 50 nœuds qui se déplacent aléatoirement dans une surface de 150x150 m².

4- Description générale de l'algorithme :

4-1. Repérage des voisins :

Afin de repérer ses voisins, chaque nœud *v* du réseau commence à construire une liste à partir de la réception de messages « Hello » provenant des voisins. La réception d'un message « Hello » d'un nœud *u* permet de détecter des liens en suivant les chaînes de confiance entre eux, ce qui mène à une modification de l'état de *v* à propos de sa confiance en *u*, c à d *v* connaît *u* mais ne lui fait pas encore confiance, car il n'est pas sûr que *u* fonctionne honnêtement. En vérifiant la certification de *u* par les nœuds de confiance de son groupe, *v* l'accepte comme un voisin et l'ajoute dans la liste des voisins ou le refuse s'il n'est pas certifié.

A la réception d'un message « Hello» d'un nœud *u*, la procédure suivante permet à un nœud *v* de repérer ses voisins et de les mettre dans une liste :

Algorithme de repérage des voisins :

Début
  Si (Dis (*u, v*) ≤ R) Alors
    Si (*u* est authentifié) Alors
      Ajouter *u* comme voisin ;
      Saisir son poids ;
      Calculer sa Mobilité ;
    Sinon (*u* n'est pas authentifié)
      Envoyer un refus ;
      Alerter le Cluster-head ;
  Sinon (Dis (*u, v*) > R)
    Envoyer refus ;
Fin

4-2. Calcul de la mobilité :

Pour calculer la mobilité relative par rapport aux autres nœuds voisins, tous les noeuds envoient (et reçoivent) des messages "Hello" à leurs voisins. Chaque noeud mesure les niveaux de puissance de deux transmissions successives de messages "Hello" reçus de chaque voisin, puis calcule la mobilité en utilisant l'équation suivante :

$$M_v(u) = 10 \log_{10} \frac{P_{u \to v}^{new}}{P_{u \to v}^{old}} \qquad \text{(Eq 4.1)}$$

Où

$P_{u \to v}^{new}$: représente la puissance du premier message envoyé de u vers v.

$P_{u \to v}^{old}$: représente la puissance du deuxième message envoyé de u vers v.

Ensuite il calcule la mobilité relative globale M utilisant l'équation :

$$M_v = Var(M_v(u_1), M_v(u_2), M_v(u_3), ..., M_v(u_n)) . \qquad \text{(Eq 4.2)}$$

Noter que seulement les transmissions reçues avec succès par la couche MAC qui sont considérées.

De plus, l'intervalle de temps T_0 après quoi la mobilité globale M_v est calculée devrait inclure deux transmissions successives de messages "Hello" de tous les voisins de *v*. Il est possible que pendant cet intervalle de temps, certains noeuds quittent (ou rejoignent) le voisinage du

noeud *v*, dans ce cas les noeuds qui ne participent pas à deux transmissions successives au noeud *v* sont exclus du calcul. Ainsi, seulement les noeuds qui ont été dans le voisinage de *v* durant cet intervalle de temps sont considérés pour le calcul de la métrique de mobilité.

Algorithme de calcul de la mobilité :

```
Début
  Tant que (T ≤ T0) faire
        Pour (i=1 ; Nvoisins)
            Si (Pas de réception d'un Hello) Alors
                  Ignorer ce nœud ;
            Sinon
                  Calculer Mv(ui) ;
            Fin Si
        Fin Pour
     Mv = Var (Mv(u1), Mv(u2),…, Mv(uNvoisins)) ;
  Fin
Fin
```

Durant la phase de repérage des voisins chaque nœud de confiance peut repérer le nombre des nœuds de confiances qui l'entourent et en ajoutant la mobilité de chacun de ces nœuds calculée précédemment, les nœuds de confiance forment une liste des nœuds qui peuvent être des candidats pour la sélection des cluster-head pendant la phase de sélection.

4-3. Calcul du poids :

Pour qu'il puisse attribue un poids à un nœud membre *u* de son cluster, le cluster-head reçoit périodiquement des valeurs de réputation et recommandation calculées par les autres nœuds de confiance, il calcule le poids pour le nœud *u*, il l'envoie au nœud concerné et informe les autres nœuds de confiance.

Dans chaque cluster-head l'algorithme suivant est exécuté :

Algorithme de calcul du poids :

```
Début
w (u) = 0 ;
    Pour i = 1 à k faire ;
        Reçoit N (i, u) ;
        Reçoit R (i, u) ;
        w (u) = w (u) + 1/2k (N (i, u) + R (i, u)) ;
    Fin pour ;
    Si w (w) = 1 Alors ;
        Marquer u comme noeud de confiance ;
    Sinon
        u est un nœud membre ;
Envoie w (u) à u ;
Envoie w (u) aux noeuds de confiance ;
Fin ;
```

4-4. Phase d'élection :

L'algorithme de clustering pondéré est exécuté à chaque nœud, ne détectant pas dans son k-voisinage un Cluster-head à qui s'affilier, de telle manière qu'à un certain temps un nœud *v* décide pour changer son rôle. Cette décision est entièrement basée sur la décision des nœuds de confiance voisins de *v*.

Dans cet algorithme nous employons trois types de messages qui sont échangés entre les nœuds :

CH (*u*) utilisé par un nœud de confiance pour mettre au courant ses voisins qu'il est (ou qu'il va être) un cluster-head.

JOINDRE (*v*, *u) :* avec lequel un nœud *v* communique à ses voisins qu'il fera partie du cluster dont le cluster-head est le nœud *u.*

DÉMISSIONNER (*u*) : qui informe un cluster-head, sous certaines conditions, qu'il doit démissionner son rôle.

Le déroulement de l'algorithme se fait comme suit :

Après la phase d'initialisation expliquée plus haut, dans laquelle chaque nœud devrait envoyer des messages «Hello» afin de repérer ses voisins et de calculer sa relative mobilité par rapport aux autres nœuds, on passera à une phase d'élection de cluster-head. En effet, chaque nœud *u* candidat d'être un cluster-head émit dans des périodes bien définies un paquet d'élection CH (*u*) contenant les informations suivantes :

- L'identité du nœud candidat au rôle de cluster-head.
- Le nombre des nœuds de confiance voisins (D)
- Mobilité relative (M) par rapport à ses voisins.
- Nombre de saut vers le cluster-head (d)
- Numéro de séquence du paquet d'élection.
- L'empreinte digitale du paquet

Chaque nœud *v* du cluster recevant le message CH (*u*), vérifie si *u* est son cluster-head, dans ce cas, rien faire, sinon *v* va vérifier la mobilité M_u de ce dernier ainsi que le nombre des nœuds de confiance qui l'entourent avec les mêmes paramètres de son cluster-head courant pour choisir son nouveau cluster-head. Si le courant cluster-head perd la compétition, on lui envoie un message DEMISSIONNER (*u*) et il joindra le nouveau cluster-head.

Quand un nouveau nœud veut joindre (ou partir) le cluster il diffuse à ses voisins un message JOINDRE (*v*, *u*), chaque cluster-head *u* doit vérifier si *v* veut joindre son cluster, dans ce cas, *v* est ajouté au cluster (*u*) ou bien, si *v* appartenait à son cluster et joint maintenant un autre cluster dans ce cas, *v* est retiré du cluster (*u*). Dans le cas où *v* soit un nœud hors du réseau et veut le joindre, le cluster-head le plus proche exécute le procédé expliqué dans le chapitre précédent de graphe de confiance pour vérifier s'il peut lui accorder un certificat ou non.

Avant d'arriver à l'étape d'implémentation de l'algorithme, on doit définir quelques paramètres à savoir :

K : Nombre de sauts (la taille du groupe) à ne pas dépasser par le nœud qui désire changer son rôle à un cluster-head.

d : Nombre de sauts actuels vers le cluster-head.

D : le nombre des nœuds de confiance voisins au nœud candidat qui veut jouer le rôle d'un cluster-head.

NM : nœud membre, qui appartient au groupe.

NP : nœud passerelle, sera sélectionné parmi l'ensemble des nœuds de confiance directement voisins (d=1) au cluster-head.

Finalement l'exécution asynchrone de l'algorithme de formation des clusters par chaque nœud *v* qui reçoit un message CH (*u*) de nœud *u*, sera de la manière suivante, en considérant bien sur que les mobilités relatives et les poids sont déjà calculés :

Algorithme de sélection d'un cluster-head :

```
Début
  Si (W <1) Alors ;
     Ignorer ce nœud ;
     Aller à fin ;
  Sinon          // c à d W = 1
    Si (d >K) Alors ;
        Dépassement de la taille du cluster ;
        Aller à fin ;
    Sinon              // c à d  d ≤ k
       Si ((Mv < Mu) ou ((Mv = Mu) et (D(v) > D(u)))) Alors ;
             Etat (v) = cluster-head ;
             v envoie DÉMISSIONNER(u) ;
             u envoie JOINDRE (u ,v)
          Si (d =1) Alors ;
               Etat (u) = NP ;
               d (v) =1 ;
          Sinon (d > 1)
               Etat (u) = NM ;
       Sinon
         Si ((Mv > Mu) ou ((Mv = Mu) et (D(v) = D(u)))) Alors ;
              Garder le courant cluster-head ;
              v reste encore candidat pour la prochaine sélection ;
         Fin Si.
       Fin Si.
    Fin Si.
  Fin Si.
Fin.
```

Si un nœud *v* ne reçoit pas le message CH (*u*) de son cluster-head après un certain temps prédéfini; l'algorithme suivant lui permet de détecter son absence et savoir ainsi le changement de la topologie.

Algorithme de détection de changement de la topologie :

```
Début
  Si  (il peut joindre son cluster-head avec un autre noeud de confiance) Alors ;
      Garde son courant cluster-head ;
      Mettre à jour le nœud NP et d ;
  Sinon
    Si (il peut trouver un autre cluster-head z) Alors ;
      Si (W = 1) Alors ;
           Il diffuse JOINDRE (v, z) ;
        Si (d = 1) Alors ;
              Etat (v) = NP;
              d (nouveau Cluster-head) = 1;
        Sinon          // c à d d >1
              Etat (v) = NM;
              d (nouveau Cluster-head) = d + 1;
        Fin Si.
      Sinon
          Demande de certification d'un noeud de confiance;
      Fin Si.
    Fin Si.
  Fin Si.
Fin.
```

4-5. Attribution et vérification de certificats :

Cette solution empêche des attaques de s'infiltrer en permettant à des nœuds de détecter les certificats contradictoires et de déterminer quelles sont les clés qui sont corrects. La procédure d'échange de certificats permet aux nœuds de recueillir pratiquement tous les certificats du graphe G. Ceci permet aux nœuds de contre-vérifier les porteurs de clé dans les

certificats qu'ils maintiennent et de détecter toutes les contradictions (c.-à-d. certificats contradictoires).

Deux certificats sont considérés contradictoires s'ils contiennent des porteurs de clé contradictoires (c.-à-d. si les deux certificats contiennent le même nom d'utilisateur mais différentes clés publique, ou s'ils contiennent la même clé publique, mais sont liés à différents noms d'utilisateurs).

Si un certificat reçu par un nœud *u* contient un porteur de clé (v, k_v) non contenue dans n'importe quel certificat dans le dépôt de certificats de *u*, alors (v, k_v) et les certificats (nœuds) qui le certifient sont marqués par *u* comme *non spécifiés*.

Un certificat marqué non spécifié signifie que le nœud *u* n'a pas assez d'information pour évaluer si le porteur de clé dans le certificat est correct. Quand (v, k_v) est reçu, *u* attend une période prédéfinie T. Si au cours de cette période *u* ne reçoit aucun certificat contradictoire concernant (v, k_v), le statut, de ce nœud et du certificat qui le certifie, change en *non-contradictoire*.

Cependant, ce mécanisme n'empêche pas les nœuds du réseau de créer des identités virtuelles ou de voler l'identité des personnes qui ne participent pas au réseau.

Lorsqu'un nœud *v* veut s'authentifier auprès d'un *u*, *v* envoie une demande de certification contenant sa liste des certificats L_{certv} à *u*, ce dernier exécute l'algorithme suivant :

Algorithme d'attribution et de vérification de certificats :

```
Début
u reçoit L_certv ;
 Si (L_certu ∩ L_certv ≠ 0) Alors ;
    Vérifier l'exactitude des certificats dans L_certv ;
       Si (certificats dans L_certv sont correctes) Alors ;
          Si (certificats dans L_certv sont valides) Alors ;
             Attribue un certificat à v ;
             H (cert)   // hachage de certificat ;
             Sign (cert)  // signature de certificat avec la clé privée de u
             Envoie certificat signé à v ;
          Sinon (certificat expiré) Alors ;
             Signaler à v que le certificat est expiré pour revoir son émetteur ;
```

Sinon (certificat non correcte)
Si (certificat en conflit) Alors ;
Appliquer la procédure de résolution d'un conflit ;
Sinon rejette la demande de certification ;
Sinon ($L_{certu} \cap L_{certv} = 0$) Alors ;
Marquer *v* et les nœuds qui le certifient comme non spécifiés ;
Demande aux autres nœuds de confiance des informations sur *v* ;
Si (*u* reçoit une confirmation pendant un temps T) Alors ;
Marquer *v* et les nœuds qui le certifient comme spécifiés ;
Générer un certificat signé à *v* ;
Sinon // *u* ne reçoit aucune confirmation pendant un temps T.
Rejette la demande de certification ;
Fin ;

4-6. Procédure de résolution d'un conflit :

Si un certificat reçu par un nœud *u* contient un porteur de clé (*v*, k_v) qui est en conflit avec un porteur de clé (*v*, k'_v) contenue dans un autre certificat tenu par *u*, les deux porteurs (*v*, k_v) et (*v*, k'_v) et les certificats qui les ont certifiés sont marqués *conflit*. Quand un nœud *u* détecte un conflit, il vérifie la validité des certificats contradictoires avec leurs émetteurs, s'ils sont encore valides, et il essaye de résoudre le conflit, pour résoudre le conflit, *u* essaye de trouver des chaînes de certificats *non-contradictoires* et valides pour les clés publique k_v et k'_v, basé sur les deux principes de recommandation et réputation (expliquées dans le chapitre précédent), deux valeurs de confiance (c-à-d de poids) qui prouvent l'exactitude de la confiance des deux porteurs sont calculées. Les deux valeurs sont alors comparées, et un porteur de clé est marqué *non-contradictoire* et l'autre est marqué faux. Si, basé sur les valeurs de confiance calculées, aucune décision ne peut être prise en ce qui concerne les porteurs de clé en question, ces derniers sont marqués en tant que conflit et le nœud attend jusqu'à ce que plus d'informations soient recueillies de sorte que le conflit puisse être résolu (par exemple, si le nœud reçoit un ensemble additionnel de certificats qui résolvent le conflit, ou si deux nœuds se rencontrent physiquement, qui garantit que le conflit sera résolu).

Le mécanisme de résolution du conflit est encore employé pour évaluer la confiance dans les nœuds pour délivrer des certificats corrects et pour détecter les nœuds malveillants.
Lorsqu'un nœud *u* détecte un conflit entre deux certificats $cert_1$ et $cert_2$, il exécute la procédure suivante :

Algorithme de résolution d'un conflit :

Début
u détecte un conflit entre $cert_1$ et $cert_2$;
Vérifier validité de $cert_1$ et $cert_2$ avec ses émetteurs ;
  Si ($cert_1$ et $cert_2$ sont valides) Alors ;
    Si (w ($cert_1$) > w ($cert_2$)) Alors ;
      $cert_1$ est non-contradictoire ;
      $cert_2$ est faut ;
    Sinon (w ($cert_1$) ≅ w($cert_2$)) Alors ;
      Marquer $cert_1$ et $cert_2$ comme conflit ;
      Attendre plus d'informations ou un rencontre physique ;
  Sinon ($cert_1$ et $cert_2$ ne sont pas valides) Alors ;
    Rejeter $cert_1$ et $cert_2$;
Fin ;

4-7. Procédure de révocation :

Si un nœud de confiance détecte un comportement « soupçonneux » d'un autre nœud du réseau qui est déjà certifié, dans ce cas il peut retirer le certificat qu'il l'a délivré et publie ainsi un rapport aux autres nœuds du réseau pour les informer de la révocation du certificat qui a fait.

Cet algorithme présente deux types de révocations de certificats, une directe et une autre indirecte.

Pour la révocation directe, un nœud de confiance peut retirer un certificat qu'il a délivré s'il croit que le nœud de la clé en ce certificat n'est plus « honnête», le nœud publie ainsi un rapport explicite de révocation.

La révocation indirecte est basée sur la période d'expiration des certificats. Spécifiquement, chaque certificat est implicitement retiré après son temps d'expiration. Car nous avons déjà expliqué que chaque certificat contient son temps de publication et une période T_v de validité. Après que cette période s'écoule, le certificat n'est plus considéré valide.

Le principe de la révocation des clés permet aux nœuds de confiance d'effectuer l'authentification avec une confiance plus élevée pour la validité des certificats, mais également dans l'exactitude des porteurs des clés contenues dans les certificats, en raison de leur période limitée de validité.

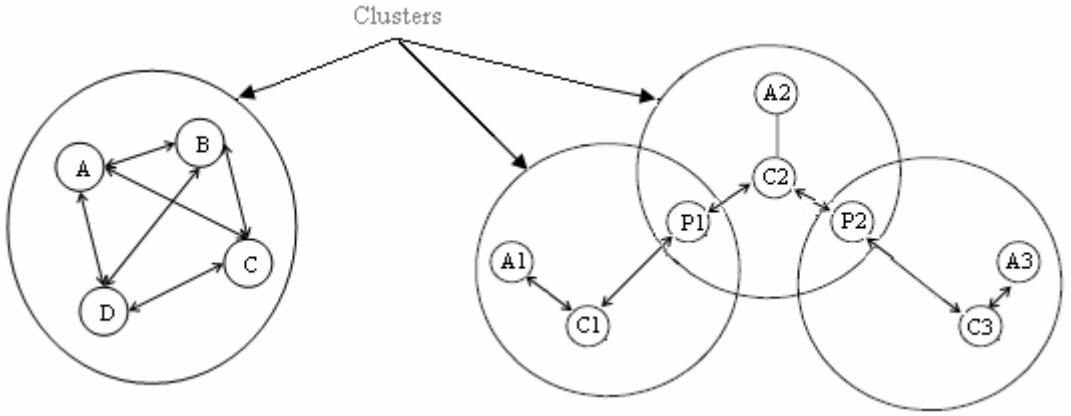
Algorithme de révocation :

```
Début
  Si (u détecte un comportement malhonnête d'un nœud v) Alors ;
     Retire le certificat délivré pour v ;
     Envoie un rapport de révocation aux autres nœuds de confiance ;
  Si (Tv de certificat de v est dépassé) Alors ;
    Signaler à v que le certificat est invalide pour revoir son émetteur ;
        Si (émetteur revalide le certificat à v) Alors ;
            v est encore authentifié ;
        Sinon (pas de renouvellement du certificat) Alors ;
            v est retiré de la liste des nœuds authentifiés ;
  Fin si
  Fin si
Fin
```

4-8. Routage :

Les étapes d'exécution de l'algorithme, tel que le repérage des voisins, permet à tous les éléments du réseau d'avoir une liste des voisins (nœuds appartenants au cluster) et de savoir le rôle de chacun d'eux, donc les nœuds ont déjà une bonne connaissance préalable en ce qui concerne la topologie du réseau et les chemins adéquats, pour cette raison l'utilisation d'un protocole de routage de type proactif semble plus commode afin de minimiser la charge dans le réseaux.

Pour la mise en application de la solution, l'idée est d'apporter quelques modifications sur un protocole proactif (DSDV dans notre cas) pour l'adapter au fonctionnement de l'algorithme. La première idée consiste à introduire dans les messages Hello de repérage des voisins un champ de découverte de route qui permet de calculer toutes les routes possibles dans un cluster donné. La deuxième étape est d'obliger les nœuds membres dans un cluster, qui veulent communiquer à d'autres nœuds dans un autre cluster, de passer leurs communications par le nœud passerelle en suite par le cluster-head qui est à son tour passe cette communication à l'autre cluster-head, nœud passerelle et finalement au nœud destination.



(a) - Routage dans un cluster : A, B, C, D : nœuds quelconques

(b) - Routage hors cluster : Ai nœuds membres, Pi nœud passerelle, Ci cluster-head

Fig.3 Principe de routage employé

5- Génération des attaques pour les testes :

5-1. Les attaques qui s'appuient sur la falsification des certificats et clés :

Dans les simulations effectuées, les réseaux ad hoc sont composés de plusieurs nœuds placés aléatoirement. Les attaquants sont introduits de telle manière que chaque attaquant choisit aléatoirement un ensemble de nœuds cibles selon le scénario d'attaque retenu. Le scénario d'attaque s'effectue selon les étapes suivantes :

1- Un attaquant identifie un nœud cible *v*.

2- L'attaquant détecte ses voisins communs avec la cible *v*, et modifie son message Hello en déclarant leurs voisins.

3- L'attaquant a trois possibilités (Scénarios) à faire :

- Soit il délivre un certificat qui lie la cible *v* à une fausse clé k'*v* (faux certificat) pour faire croire ses voisins que *v* est un intrus qui n'appartient pas au réseau pour l'isoler en suite.
- Il peut aussi délivrer un certificat qui lie une clé k_w un nœud *v* au lieu du nœud *w*, pour tromper d'autres nœuds pour croire que k_w est la clé publique de l'utilisateur *v*, quand en réalité c'est la clé publique du nœud *w*.
- troisièmement, l'attaquant peut inventer un certain nombre d'identités et de clés publiques et les lier par les certificats appropriés et essayer de convaincre un nœud donné que les certificats sont corrects, comme si, ils ont été signés par d'autres nœuds de confiance.

4- Lorsque cet attaquant réalise une de ces actions, il délivre un rapport de mise à jour aux nœuds voisins pour les informer des modifications effectuées.

5-2. Les attaques provenant d'un nœud du réseau lui-même :

Dans ce type on considère le cas où un nœud du réseau, qui est déjà certifié présente un comportement suspect. Deux cas sont envisagés :

- le nœud suspect envoie un grand nombre de message à ses voisins dans une période très étroite.
- Ou bien, ce nœud ne répond pas aux demandes de ses voisins lorsqu'ils veulent des informations pendant leurs communications.

Pour remédier à ce problème on définie deux paramètres N_{paquet} (c'est le nombre maximum de paquets reçu de la part d'un nœud donné pendant un temps défini T_{paquet}) et $T_{réponse}$ (c'est la durée maximale durant laquelle un nœud doit répondre à une requête d'un autre nœud).

- Pour le premier cas un compteur et un temporisateur sont initialisés. Le compteur est incrémenté pour chaque paquet reçu, si les demandes d'un nœud sont identiques et dépassent le seuil N_{paquet}, une attaque est signalée. Le compteur est remis à zéro si le nœud reçois un nombre de paquets $N < N_{paquet}$ dans un temps $T > T_{paquet}$.
- Une attaque est signalée aussi quand deux nœuds se communiquent mais l'un d'eux ne répond pas à une des requêtes pendant un temps $T > T_{réponse}$.

Dans les deux cas si un nœud de confiance détecte ce mauvais comportement, ce dernier réduira sa confiance sur le nœud suspect. Mais dans le cas où ce comportement sera davantage remarquable, le nœud de confiance peut retirer le certificat qu'il l'a délivré et publie ainsi un rapport aux autres nœuds du réseau de la révocation du certificat qui a fait, et les nœuds dans le réseau qui reçoivent le message de révocation enlèvent complètement leur confiance pour le nœud accusé.

5-3. Les attaques de déni de service DOS :

La simulation de ce type d'attaques est faite de telle manière qu'un attaquant (un nœud qui n'appartient pas au réseau) envoie un grand nombre de messages à un nœud (ou à un ensemble de nœuds) du réseau afin d'essayer de consommer ses (leurs) ressources, comme la largeur de bande de transmission (pour qu'elle soit occupée uniquement par l'attaquant) ou affecter l'énergie des batteries (pour qu'elle soit consommée par le traitement des requêtes de l'attaquant), dans le but de gêner le fonctionnement normal du réseau.

Pour résoudre ce type de problèmes provenant des nœuds hors du réseau, les mêmes procédures sont faites comme dans le cas des attaques qui s'appuient sur la falsification des

certificats et clés (il faut vérifier l'exactitude des certificats), mais pour ne pas gaspiller l'énergie des batteries et n'augmenter pas la charge du réseau, les nœuds responsables du contrôle de la sécurité dans le réseau, une fois détectent qu'un nœud à l'extérieur du réseau (qui possède de faux certificats) qui essaie d'inonder le réseau d'un grand nombre de requêtes pendant une durée de temps courte, les nœuds de confiance ne gèrent pas ce type de requête et publient une seule fois un rapport informant le reste des nœuds de cette action malveillante.

6- Changements nécessaires pour l'implémentation :

Pour intégrer les attaques citées plus haut, la mis en place de quelques modifications au niveau du protocole utilisé est nécessaire pour donner aux nœuds leur comportement malveillant. Ces changements sont effectués sur le même protocole utilisé par le réseau qui est le protocole DSDV, puisque ces attaques doivent employer le nouveau protocole de routage pour transmettre les mêmes types de paquets que les nœuds légitimes dans le réseau.

Tous les protocoles de routage dans NS sont installés dans le répertoire "ns-allinone-2.30\ns-2.30". Nous commençons le travail par reproduire le protocole dsdv dans ce répertoire et changeons le nom en "Attaqueroute".

Renommer tous les fichiers qui sont marqués par "dsdv" dans le répertoire par "Attaqueroute" comme Attaqueroute.cc, Attaqueroute.h, etc. Ainsi nous avons changé tous les classes, fonctions, structs, variables et noms de constantes dans tous les fichiers dans le répertoire.

Les deux protocoles dsdv et Attaqueroute sont conçus pour envoyer l'un à l'autre des paquets dsdv. Ces deux protocoles sont en effet identiques.

Après les changements cités ci-dessus, nous avons changé des fichiers communs qui sont employés dans NS-2 pour intégrer le nouveau protocole Attaqueroute au simulateur. Les changements sont expliqués comme suit :

Le premier fichier à modifier est "ns-2.30\common/packet.h", où on ajoute dans l'énumération *paquet_t* une constante pour indiquer le nouveau type de paquet, *PT_Attaqueroute*, où tous les types de paquets sont énumérés. Nous ajouterons *PT_Attaqueroute* à cette liste comme suit :

ns-2.30\common\packet.h

```
enum packet_t {
    PT_TCP,
```

```
    PT_UDP,
    PT_CBR,
// insert new packet types here
    PT_Attaqueroute,
    PT_NTYPE // This MUST be the LAST one
};
```

Juste après l'énumération *paquet_t* il y a la classe p_info. À l'intérieur du constructeur nous donnerons un nom à notre type de paquet :

ns-2.30\common\packet.h

```
p_info() {
    name_[PT_TCP]= "tcp";
    name_[PT_UDP]= "udp";
    name_[PT_CBR]= "cbr";
// insert new names
    name_[PT_Attaqueroute]= "Attaqueroute";
    name_[PT_NTYPE]= "undefined";
}
```

Le deuxième fichier modifié est "ns-2.30\tcl\lib\ns-packet.tcl " où nous devons ajouter le nouveau protocole Attaqueroute à la liste des protocoles :

ns-2.30\tcl\lib\ns-packet.tcl

```
foreach prot {
   Attaqueroute
   AODV
   # ...
} {
  add-packet-header $prot
   }
```

Un autre fichier doit être modifié, c'est "ns-2.30/tcl/lib/ns-default.tcl" dont on indique les valeurs par défaut à l'intérieur du fichier et on fixe la valeur par défaut de la taille des paquets Attaqueroute, à l'extrémité du fichier on ajoute :

ns-2.30/tcl/lib/ns-default.tcl

```
# ...
# Defaults defined for Attaqueroute
Agent/Attaqueroute set packetSize_ 128
Agent/Attaqueroute set accessible_var_ true
```

Enfin, nous devons modifier le "ns-2.30/tcl/lib/ns-lib.tcl". Nous devons ajouter des agents de protocole comme procédures pour créer un nœud. Le but est de créer un noeud sans fil avec Attaqueroute comme protocole de routage.
La procédure de *create-wireless-node*, avec d'autres, sert de fournir l'agent de routage à un nœud. Nous devons modifier cette procédure pour créer le protocole Attaqueroute.
Les modifications sont faites comme suit :

ns-2.30/tcl/lib/ns-lib.tcl

```
Simulator instproc create-wireless-node args {
    # ...
 switch -exact $routingAgent_ {
    Attaqueroute {
        set ragent [$self create-Attaqueroute-agent $node]
                    }
    # ...
  }
    # ...
 }
```

Ensuite l'agent create-Attaqueroute-agent sera codé comme suit :

ns-2.30/tcl/lib/ns-lib.tcl

```
Simulator instproc create-Attaqueroute-agent { node } {
```

```
# Create Attaqueroute routing agent
  set ragent [new Agent/Attaqueroute [$node node-addr]]
  $self at 0.0 "$ragent start"
  $node set ragent_ $ragent
  return $ragent
}
```

La ligne 3 crée un nouvel agent Attaqueroute avec l'adresse du nœud. Cet agent est programmé pour commencer au début de la simulation (ligne 4), et est assignée comme agent de routage du nœud dans la ligne 5.

A ce point toutes les modifications sont mises en place, il nous reste que de recompiler le simulateur. Pour le faire, il faut modifier le fichier "ns-2.30\ *makefile* " en ajoutant les lignes suivantes :

ns-2.30\ makefile

```
OBJ_CC = \
  tools/random.o tools/rng.o tools/ranvar.o common/misc.o common/timer-handler.o \
  # ...
  Attaqueroute/Attaqueroute.o Attaqueroute/Attaqueroute_rtable.o \
  # ...
$(OBJ_STL)
```

Comme nous avons modifié le fichier *common/packet.h* mais pas le fichier *common/packet.cc* nous devrions recompiler ce dernier par la commande "*touch*". Ensuite nous pouvons nous exécuter la commande " *Make* " pour introduire le protocole Attaqueroute.

```
[ns-2.30]$ touch common/packet.cc
[ns-2.30]$ make
```

Conclusion :

Dans ce chapitre, nous avons concrétisé l'implémentation des différents composants de l'algorithme proposé pour sécuriser un réseau ad hoc (exposé dans le chapitre précèdent) dans le but d'évaluer ces performances, pour cela, nous avons présenté le modèle de simulation de l'algorithme par une description détaillée qui consiste à définir les différentes phases d'exécution. Celles-ci décrivent le comportement des nœuds quand un événement se produit, que se soit la formation des groupes ou la détection des mauvaises actions.

chapitre 5

Evaluation des resultats

Introduction :

L'idée de la gestion décentralisée dans les réseaux ad hoc est devenu une notion importante, puisque un tel réseau ne présente aucune infrastructure fixe, ainsi qu'un changement de topologie se produit d'une façon entièrement arbitraire. Pour remédier à ce problème, les chercheurs ont recommandé de partitionner le réseau à des clusters. Un principal problème lié à ces réseaux est comment maintenir la stabilité des clusters en augmentant leur durée de vie dans la présence de la mobilité des nœuds, de plus comment garantir la sécurité de la communication dans un tel réseau soumis à des contraintes de vulnérabilité accrue.

Dans les chapitres précédents, nous avons expliqué le fonctionnement de l'algorithme de sécurité qui est basé sur le partitionnement des réseaux ad hoc en groupes pour bénéficier des avantages d'une architecture distribuée, ensuite d'accorder le rôle d'autorité de certification à différentes entités dignes de confiance pour renforcer la surveillance contre les actions malicieuses. Cependant, une simulation avec la présence d'attaques est nécessaire pour prouver la capacité et l'efficacité de l'approche. Ce chapitre a donc pour objectif de démontrer les performances de la solution et évaluer les résultats obtenus à travers son implémentation, cette partie consiste à simuler le comportement des nœuds légitimes face à quelques attaques injectées dans le réseau ad hoc d'une part, et d'autre part de tester la stabilité de l'architecture en vérifiant la durée de vie des clusters formés et le taux de réaffiliation des nœuds.

1- Etude et évaluation des résultats obtenus :

Pour visualiser les résultats de simulation, nous avons considéré une topologie de réseau de 50 nœuds mobiles, où chacun d'entre eux a une portée de transmission allant jusqu'à 250 mètres. Les nœuds sont aléatoirement déployés sur une zone de superficie 150x150 m². La vitesse de déplacement des nœuds est variée entre 0.2m/s et 10m/s, et la valeur k des sauts de 1, 2, 3, un temps de pause de 20s. De plus, pour tester la capacité de système de sécurité, nous avons implémenté des nœuds malveillants dont les différents scénarios d'attaques sont expliqués dans le chapitre précédent. Les simulations se déroulent en une durée de 2100s, et afin de garantir des mesures les plus précises possibles, nous allons exécuter cinq fois chacune des simulations et faire la moyenne des mesures.

2. Evaluation de l'algorithme en terme de résistance contre les attaques :

Cette évaluation consiste à introduire les différents types d'attaques présentés dans le chapitre précédent et d'analyser la robustesse de l'algorithme pour éviter de telles actions.

2-1. Evaluation de l'algorithme en présence des attaques de type 1 :

Les deux figures ci-dessous, Fig.1 et Fig.2 présentent l'efficacité de l'algorithme en présence des nœuds malicieux. En effet, dans un réseau non protégé (dont l'algorithme de sécurité n'est pas intégré) un ou plusieurs attaquants ont la possibilité de s'intégrer dedans et peuvent facilement échanger des données avec les autres nœuds légitimes du réseau, cet échange de données est exprimé par la croissance dans le temps du graphe (en rouge) qui présente la quantité des données échangée avec les nœuds attaquants. Par contre en présence de l'algorithme de sécurité, on remarque que l'échange de données des attaquants avec le reste du réseau est nul, à l'exception d'une étape initiale pendant laquelle l'attaquant doit forcement s'identifier auprès des nœuds de confiance dans le réseau, par l'échange des demandes d'authentification, mais après la vérification de l'identité de l'attaquant, ce dernier sera isolé complètement du réseau et une alerte est annoncée dans le réseau de cette menace.

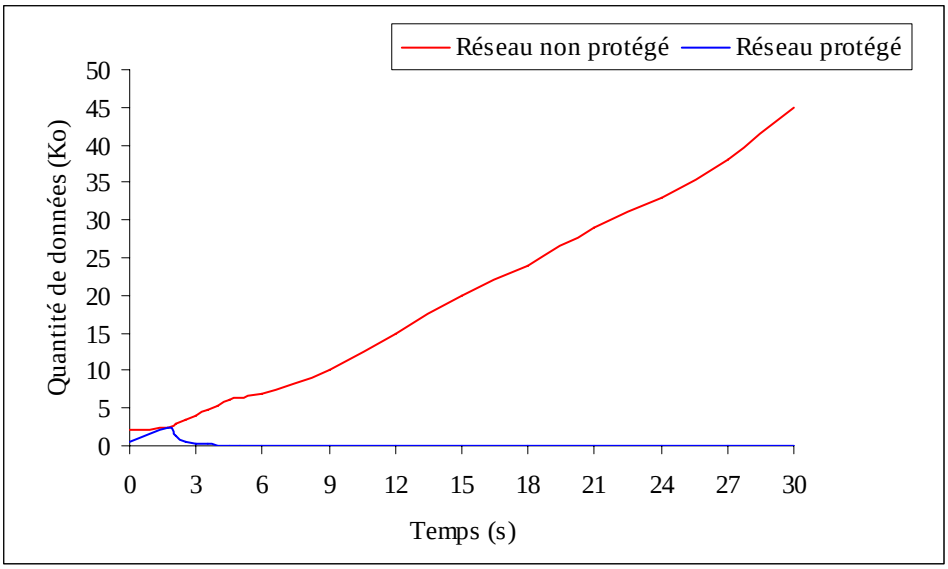


Fig.1 Quantité de données échangée par un nœud malicieux dans le réseau.

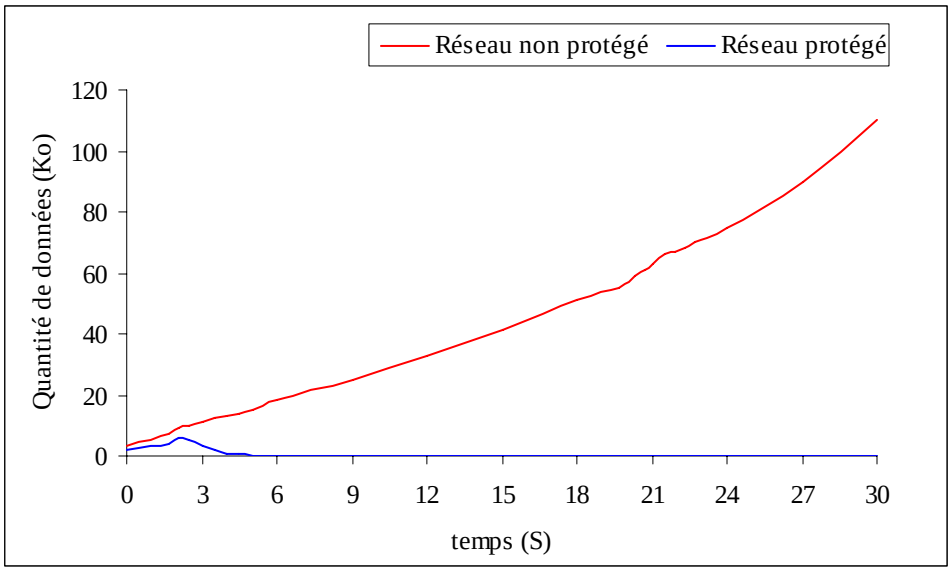


Fig.2 Quantité de données échangée par 5 nœuds malicieux dans le réseau.

2-2. Evaluation de l'algorithme en présence des attaques de type 2 et 3 :

Dans cette expérience on va essayer de voir l'effet des nœuds égoïstes dans le réseau sur l'échange de données. Sur la figure 3, le taux de trafic représente le nombre de paquets envoyés par la source sur le nombre de paquets reçus par les nœuds de destination. En absence du système de protection l'existence de ce type de nœuds dans le réseau défavorise la circulation des données entre les entités, parce qu'un nœud égoïste ne participe pas dans le processus d'échange de données et de services, on remarque que la dégradation de l'échange de trafic peut atteindre jusqu'à 0.25, cela signifie que 3/4 des données envoyées sont perdues (absorbées par les nœuds égoïstes) en présence de 20 nœuds malicieux dans le réseau. Donc la présence des nœuds égoïstes a un effet négatif sur le taux de trafic et par conséquent sur la qualité de service dans le réseau. En revanche, par application du système de confiance la performance du réseau peut être récupérée, parce qu'il permet au réseau d'isoler les nœuds présentant un tel comportement et de faire passer le trafic par les nœuds digne de confiance. Néanmoins une perte de donnée peut être remarquée dans le réseau malgré l'utilisation de l'algorithme de sécurité, cela est due aux problèmes rencontrés dans le protocole de routage lui-même comme : les paquets perdus (dropped packet), aux échecs d'itinéraire, à la congestion et aux pertes sans fil du canal.

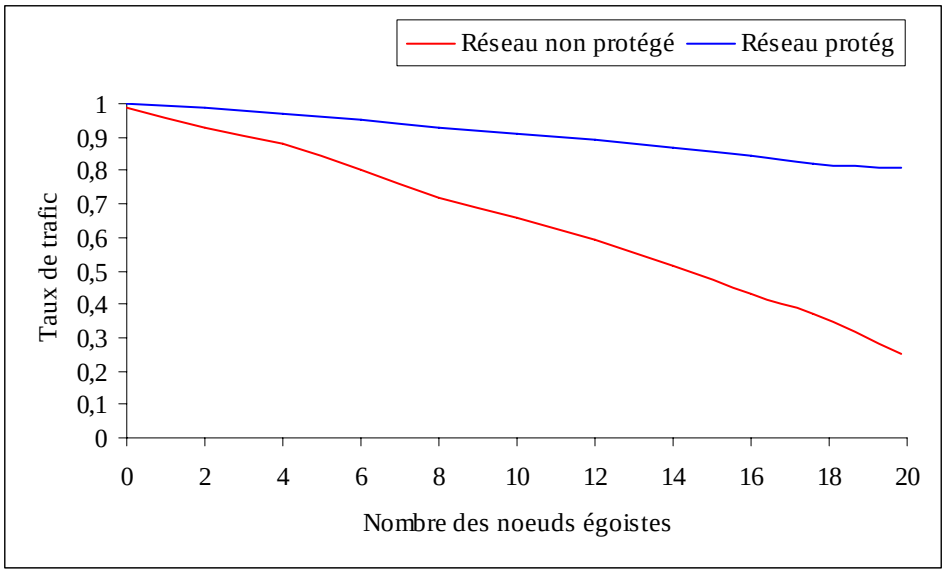


Fig.3 Taux de trafic échangé dans le réseau en présence des nœuds égoïstes.

Il est évident que le paramètre d'énergie a une importance élevée dans les réseaux mobiles sans fil, parce que les entités mobiles sont alimentées par des batteries limitées en capacité énergétique et que la consommation non rationalisée d'énergie de ces batteries n'affecte non seulement le nœud lui-même mais affecte aussi le bon fonctionnement du réseau entier. La figure suivante montre l'effet d'un nœud attaquant sur la consommation d'énergie dans un réseau mobile lorsqu'il est soumis à des attaques de type déni de service qui essaient de gaspiller les ressources des nœuds légitimes. Le graphe montre qu'en présence d'un système de protection contre ce type d'attaque, le taux moyen d'énergie consommée est relativement petit et la partie consommée est celle utilisée dans les processus de communication ordinaires. Tandis qu'en cas d'un réseau non protégé, le taux moyen d'énergie consommée est plus élevé en présence des nœuds malicieux travaillants pour faire tomber le réseau en panne.

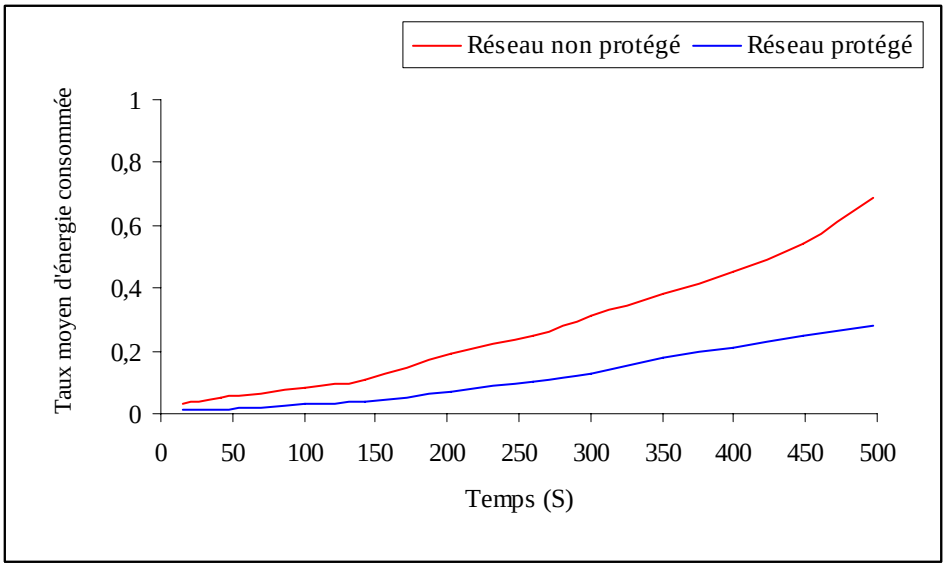


Fig.4 Taux moyen d'énergie consommée en présence des nœuds malicieux.

3- Evaluation de l'algorithme en termes de stabilité du réseau :

Dans la phase de formation des clusters, certains algorithmes de clustering ne prennent pas en considération la mobilité des nœuds, une conséquence directe est que les clusters formés ne sont pas assez stable et que la réaffiliation des nœuds et la réélection des cluster-heads ont un taux élevé.

Dans cet algorithme, nous avons introduit la métrique de la mobilité relative des nœuds pour le choix des cluster-heads et nous lui donnons un avantage en plus que le critère de nombre des nœuds de confiance entourant le nœud candidat à être cluster-head. En effet, lorsque deux nœuds entrent en compétition pour le choix de cluster-head, nous comparons d'abord leur mobilité, et le plus stable (celui qui a une mobilité faible Mv < Mu) sera le vainqueur, mais quand ils ont des mobilités égales, dans ce cas nous faisons appel au critère de nombre des nœuds de confiance qui l'entourent. Parce que les deux candidats sont déjà des nœuds de confiance, c'est pour cette raison que nous avons donné beaucoup d'importance à la stabilité qu'au dernier critère.

Pour tester la stabilité des groupes formés, nous évaluons en termes de clusters formés en fonction de la portée de transmission et de la vitesse de déplacement des nœuds.

Evaluation de l'algorithme en fonction de la portée de transmission :

Au début, nous avons mené des simulations pour calculer le nombre de clusters formés par rapport au rayon de transmission. Et nous avons comparé ces résultats à ceux obtenus par les algorithmes de Lowest-ID et MOBIC.

La comparaison entre les trois algorithmes est représentée par la figure 05.

Nous remarquons une grande différence au niveau de la portée de transmission à 50 m, cela est dû à la condition imposée pour la formation de groupes (Clusters), un nœud de confiance tout seul ne peut pas former son propre groupe, il doit avoir au moins un nœud voisin de confiance. Dans cette simulation, le nombre de groupes formés ne dépasse pas 25 clusters. Cependant, avec la portée de transmission entre 50 et 125 m, le nombre de groupes diminue rapidement, mais lorsque la portée de transmission dépasse les 150 m, le réseau devient plus stable et le nombre de groupes devient plus ou moins stable, parce que, pour des petits rayons de transmission il y a plus de possibilités qu'un nœud quitte son cluster pour communiquer avec d'autres nœuds éloignés s'il ne trouve pas un chemin libre, mais pour des rayons étendus, le nœud peut atteindre plusieurs d'autres nœuds sans qu'il soit obligé de quitter son cluster.

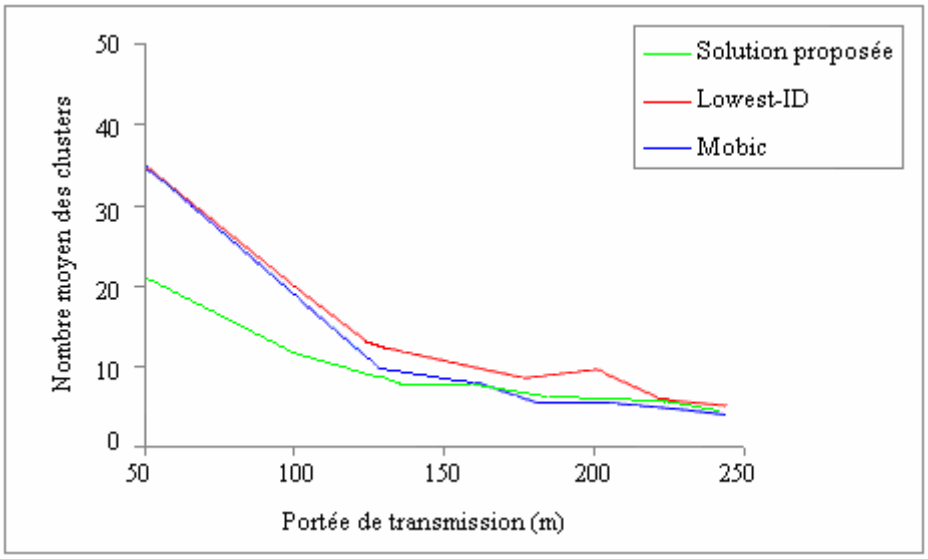


Fig.05 Comparaison du nombre moyen de clusters formés avec les algorithmes de Lowest-ID et Mobic.

Evaluation de l'algorithme en fonction de la vitesse de déplacement des nœuds :

La figure 06 représente le nombre moyen de Clusters générés en fonction de la vitesse de déplacement des nœuds. On remarque que ce nombre diminue faiblement en fonction de la vitesse. Ce résultat montre qu'en cas de mobilité, les nœuds sont capables de se réorganiser et de joindre des Clusters déjà existants.

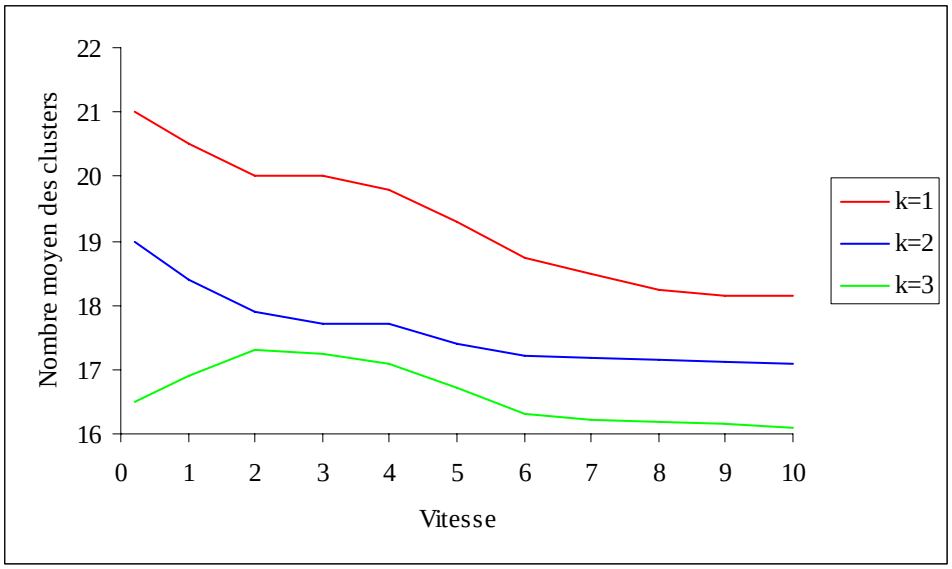


Fig.06 Nombre moyen de clusters en fonction de la vitesse.

Evaluation de l'algorithme en fonction des changements de la topologie :

Il est généralement difficile de gérer le clustering dans les réseaux mobiles parce que le changement de la topologie est fréquent à cause des déplacements de nœuds.

Evaluation de l'algorithme en fonction des changements de cluster-heads :

Un algorithme de clustering est efficace si les cluster-heads ne changent pas fréquemment leurs statuts parce que l'occurrence de ces changements cause une restructuration locale ou générale du réseau. De ce fait, pour évaluer la robustesse de l'algorithme face à la mobilité des nœuds, nous proposons d'estimer la durée de vie (le nombre moyen de changements des cluster-heads) en fonction de la vitesse de déplacement des nœuds.

Les durées de vie représentées dans la figure 07 diminuent en fonction de la mobilité puisque la mobilité des nœuds introduit plus d'instabilité dans le réseau. Toutefois, on remarque que cette diminution n'est pas trop importante surtout pour la valeur de $k = 1$.

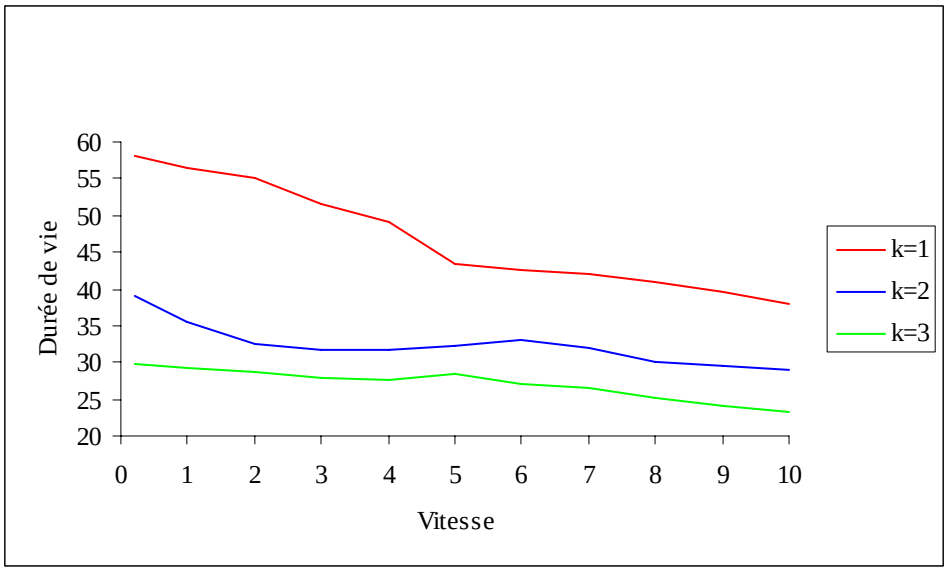


Fig.07 Durée de vie des cluster-heads en fonction de la vitesse.

Evaluation de l'algorithme en fonction des réaffiliations des nœuds :

Les figures 08 et 09 représentent respectivement les taux des réaffiliations et d'élections en fonction de la vitesse, ces deux taux augmentent en fonction de la vitesse des nœuds. En effet, plus la vitesse est grande, plus la probabilité qu'un nœud se trouve hors de son Cluster suite à un mouvement est grande. En revanche, nous remarquons pour cette configuration, que les réaffiliations deviennent presque constantes à partir de la vitesse 8 m/s. On remarque aussi que les valeurs de k (1, 2 et 3) influent peu sur les réaffiliations et les élections ce qui laisse plus de flexibilité à l'utilisateur pour le choix du rayon des Clusters.

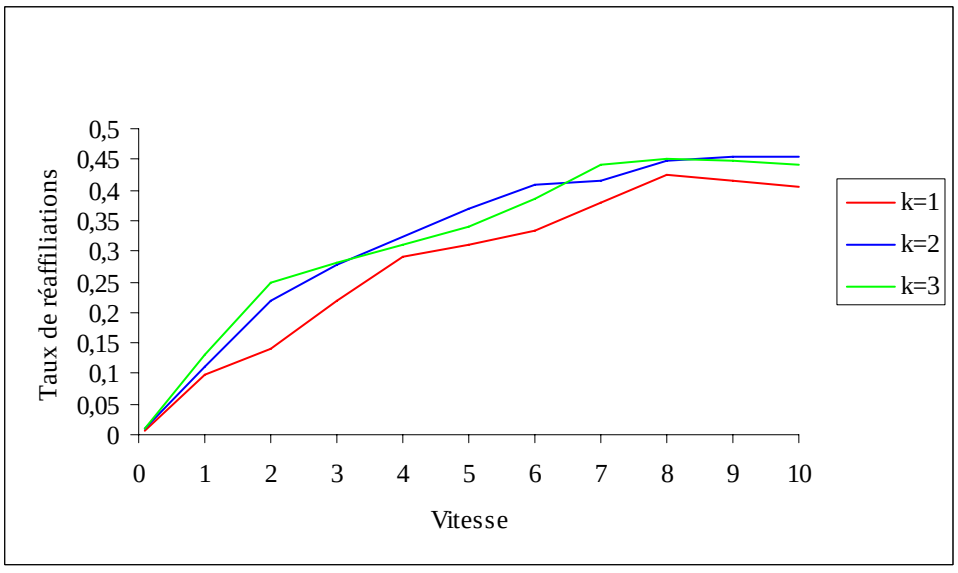


Fig.08 taux de réaffiliation en fonction de la vitesse.

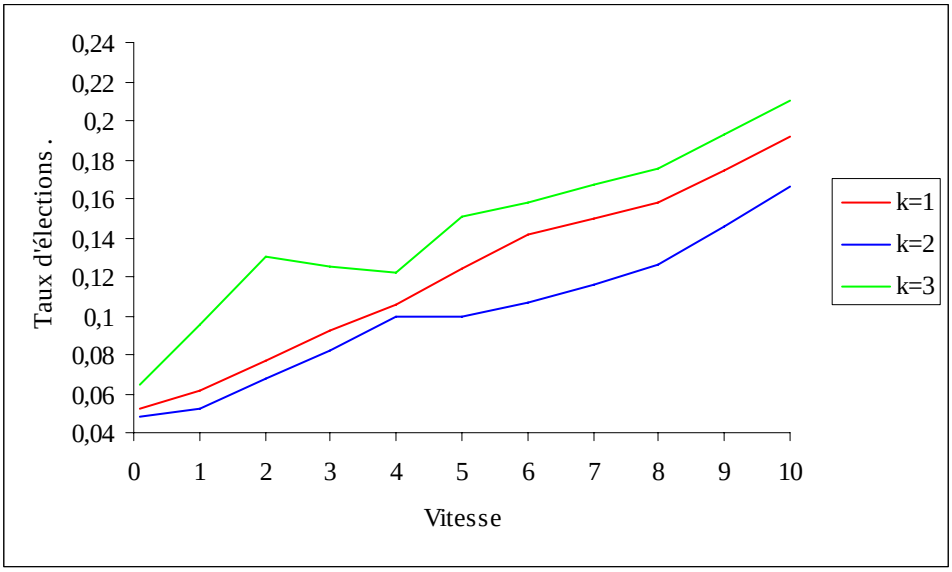


Fig.09 Taux d'élection des cluster-heads en fonction de la vitesse.

Conclusion :

Dans ce chapitre, nous avons présenté les résultats obtenus lors de la simulation de l'algorithme proposé, ce dernier prend en considération les spécificités et les contraintes imposées par les réseaux ad hoc. Nous avons visé avec cet algorithme la création d'une topologie stable pour minimiser la réélection fréquente et éviter la restructuration globale de tout le réseau. Pour cela, nous avons impliqué une métrique de stabilité pour élire les cluster-heads. En effet, les résultats obtenus ont prouvé que l'algorithme était robuste face à la mobilité des nœuds.

De plus, la sécurité de l'architecture qu'on a proposée dépend principalement du modèle de confiance assuré par la gestion auto organisée des clés publiques dans les réseaux mobiles ad hoc. L'idée consiste à ce que la certification de chaque nœud soit délivrée par les nœuds de confiance et que chaque nœud inconnu doit commencer avec un statut dont le niveau de confiance est le plus bas. Avec ce procédé on est sûr qu'aucun attaquant ne pourra se procurer le statut d'un membre légitime, donc la présence d'un grand nombre de nœuds de confiance augmente le niveau de sécurité du réseau.

Un avantage de cet algorithme est qu'avec seulement un nombre limité de nœuds (de confiance) dans un réseau on peut effectuer l'authentification, basée seulement sur leur information locale. Une conséquence importante de ce fait, est que l'authentification est

encore possible même lorsque le réseau est divisé et les nœuds peuvent communiquer avec seulement un sous-ensemble d'autres nœuds.

Il est également important de noter que la solution proposée exige la participation consciente des utilisateurs quand leurs paires de clés publiques/privées sont créées et pour délivrer et retirer des certificats ; toutes autres opérations (y compris l'échange de certificats et la construction des dépôts de certificats) sont complètement automatiques.

Conclusion générale :

La sécurité des réseaux mobiles et spécialement des réseaux mobiles ad hoc n'a jamais cessé de susciter des préoccupations du fait qu'ils sont exposés à des menaces supplémentaires par rapport aux réseaux filaires. En général, ces menaces viennent du fait que les communications sans fil sont transmises par des ondes radios et peuvent être écoutées par des personnes non autorisées, des topologies dynamiques, de l'absence d'autorité centrale, etc. Toutes ces contraintes concourent à rendre la sécurité des réseaux sans fil ad hoc difficile et complexe à appréhender. Ce sujet va devenir d'autant plus critique que le développement de tels réseaux va rapidement s'étendre. De plus, Les mécanismes de sécurité traditionnels ne répondent pas aux exigences de tels réseaux. Il s'agit donc de concevoir de nouveaux mécanismes afin de garantir la sécurité de ces réseaux.

Dans ce mémoire, après avoir étudié les caractéristiques des réseaux ad hoc et analysé leurs vulnérabilités, nous avons proposé un système de sécurité destiné aux réseaux ad hoc basé sur la gestion des clés publiques et le principe de clustering.

Le premier objectif de ce travail est d'établir une architecture de clustering plus stable pour éviter la mise à jour fréquente du système et réduire le rétablissement des clusters formés. Afin de garantir cette condition de stabilité adaptable aux environnements mobiles, nous avons introduit la mobilité relative des nœuds comme paramètre dans la phase de sélection des cluster-heads.

Le deuxième objectif est de renforcer le mécanisme de sécurité par un modèle de confiance multicritères, distribué et coopératif basé sur le principe d'une infrastructure à clé publique auto organisée. Ce modèle combine à la fois les éléments classiques de la sécurité (principe de la cryptographie asymétrique) et de nouveaux éléments conformes aux réseaux ad hoc, et qui sont nourris par les interactions d'un nœud avec son environnement.

Le troisième objectif vient compléter le deuxième en introduisant des comportements malveillants dans le réseau simulé pour tester la capacité de la solution de détecter et isoler les nœuds se comportant mal.

Ensuite, nous avons présenté les résultats de simulations afin de montrer les performances de l'algorithme en termes de résistance contre les attaques et de stabilité des clusters formés.

Selon les résultats obtenus et l'évaluation des performances on peut conclure que cette architecture est adaptée aux changements dynamiques de la topologie des réseaux et elle est robuste face aux actions présentant des malveillances.

Finalement, nous pouvons dire que l'architecture proposée n'a pas la prétention de correspondre à toutes les situations que l'on peut rencontrer des réseaux mobiles ad hoc, cependant elle propose une architecture globale de sécurité, dynamique et auto organisée, permettant de gérer la mise en relation d'objets communicants.

Perspectives :

Le travail réalisé peut être étendu selon plusieurs chemins d'amélioration, notamment : Premièrement, nous envisageons d'introduire d'autres paramètres pour la sélection des cluster-heads comme le niveau d'énergie restant dans les batteries.

Deuxièmement, nous souhaitons évaluer la résistance de ce modèle de confiance face à d'autres types d'attaques.

Bibliographie :

[Abd97] : A. Abdul-Rahman : «The PGP trust model », EDI-Forum: the Journal of Electronic Commerce, 1997.

[Abd00] : Abdelhafid Abouaissa : « Synchronisation multimédia dans les systèmes de communication de groupes », Thèse de Doctorat, Université de Franche Comté, 2000.

[Alb02] : P.Albers, O. Camp, J-M. Percher, B. Jouga, L. Mé, R. Puttini : «Security in Ad hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches », WIS, Ciudad Real Spain, 2002.

[Ami00] : A. Amis, R. Prakash, T.Vuong, D. Huong : « Max-Min D-cluster formation in wireless Ad hoc networks », in Proceeding of IEEE Infocom, 2000.

[And98] : R. Anderson, F. Bergadano, B. Crispo, J.H. Lee C. Manifavas, R. Needham : « A new family of authentication protocols », ACM SIGOPS Operating Systems Review, Vol. 32, 1998.

[Arn08] : Arnaud contes : «Une architecture de sécurité hiérarchique, adaptable et dynamique pour la grille», Thèse de doctorat, Université de Nice - Sophia Antipolis, 2008.

[Bac00] : R. Bace, P. Mell : « Intrusion Detection Systems», NIST Special Publication on Intrusion Detection Systems.

[Bas99] : S. Basagni : « Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks», Proceeding of IEEE VTS 50th Vehicular Technology Conference, 1999.

[Bas01] : P. Basu, N. Khan, T.D.C. Little: «A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks », in Proc. IEEE ICDCS 2001, Workshop on Wireless Networks and Mobile Computing, Phoenix, AZ, 2001.

[Bel00] : Cheswick, Bellovin : «Building Internet Firewalls».

[Ben97] : F. Bennett, D.clarke, J. Evans, A. Hopper, D. Leask : « Piconet Embedded Mobile Networking », IEEE Personnel Communications, 1997.

[Ben07] : N. Benaouda, H. Guyennet, A.Hammad, A.H. Benaouda, M. Mostefai : « Modélisation par les groupes d'un réseau sans fil Ad-Hoc », 4th International Conference on Computer Integrated Manufacturing CIP, 2007.

[Ber09] : UC Berkeley, LBL, USC/ISI, and Xerox PARC : «The ns Manual», 2009
Url: http://www.isi.edu/nsnam/ns/ns-documentation.html

[Blo07] : L. Bloch, C. Wolfhugel : « Sécurité informatique, Principes et méthode », éditions Eyrolles, 2007.

[Bou03] : S. Bouam, J. B. Othman : « Data Security in Ad hoc Networks using MultiPath Routing», in Proc. of the 14th IEEE PIMRC, 2003.

[Bur03] : A. Burg : « Ad hoc networks specific attacks », Technische Universität München, Institut für Informatik, Seminar Paper, Seminar Ad Hoc Networking: concept, applications, and security, 2003.

[Cap03] : S. Capkun, L. Buttyan, J.P. Hubaux : « Self–ogranized public key management for mobile Ad hoc networks », IEEE Transactions on mobile computing, 2003.

[Céd05] : Cédric Llorens : « Mesure de la sécurité "logique" d'un réseau d'un opérateur de télécommunications», Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications, France, 2005.

[Che99] : S. Chen, K. Nahrstedt : «A distributed quality-of-service routing in ad-hoc networks», IEEE Journal on Selected Areas in Communications, 1999.

[Cla03] : T. Clausen, P. Jacquet : « Optimized Link State Routing Protocol (OLSR) », Request for Comments 3626, Octobre 2003.

[Cor99] : S. Corson : « Mobile Ad hoc Networking (MANET), Routing Protocol Performance Issues and Evaluation Considerations», Requet for Comments (Informational) RFC 2501, IETF, 1999.

[Dea02] : Dean, Piette et Villeneuve : « Réseaux informatiques 2ème édition », édition, 2002.

[Den07] : Z. Deng-yin, W. Qian-qian, X. Jian : «Design and analysis of firewall-penetrated scheme based on trusted host», The journal of China Universities of posts and telecommunications, 2007.

[Dif76] : W. Diffie, M.E. Hellman : « New directions in cryptography», IEEE Transactions on Information Theory, 1976.

[Eph87] : A. Ephremides, J. E. Wieselthier, D.J. Baker : « A design concept for reliable mobile radio networks with frequency hopping signalling », In Proceedings of the IEEE, 1987.

[Fou01] : Pierre-Alain Fouque : « Le partage de clés cryptographiques : théorie et Pratique», Thèse de Doctorat, Université Paris 7, 2001.

[Fre01] : J.A. Freebersyser, B. Leiner : «A DoD perspective on mobile Ad hoc networks», Ad Hoc Networking, Addison Wesley, pp. 29-51, 2001.

[Gha02] : S. Ghazizadeh, O. Ilghami, E. Sirin, F. Yaman : « Security-Aware Adaptive Dynamic Source Routing Protocol», In Proc. of 27th Conference on Local Computer Networks, 2002.

[Gou06] : M.G. Gouda, A.X. Liu : « Structured firewall design», scienceDirect, Computer Networks, 2006.

[Gra04] : Le Grand Livre de Securiteinfo.com Url : http://www.securiteinfo.com , 9 février 2004.

[Gum00] : V. Gummalla, A. Chandra, J. O. Limb : « Wireless Medium Access Control Protocols», IEEE Communications Surveys, Second Quarter 2000.
[Guy04] : Guy pujolle : « Sécurité Wi-Fi », Editions Eyrolles, 2004.
[Haa02] : Z. Haas, M. Pearlman, P. Samar : « The Zone Routing Protocol (ZRP) for Ad Hoc Networks», Internet Draft, 2002.
http: //www.draft-ietf-manet-zone-zrp-02.txt.
[Hub01] : J. P Hubaux, L. Buttyan, S. Capkun : « The Quest for Security in Mobile Ad Hoc Networks », Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Long Beach, CA, 2001.
[Int89] : International Standards Organization. Information Processing Systems- OSI – Basic Reference Model - Part 2: Security Architecture. ISO 7498-2, February 1989.
[Jar02] : J. Al-Jaroodi : « Security Issues in Wireless Mobile Ad Hoc Networks at the Network Layer», University of Nebraska-Lincoln, Dept. of Computer Science and Engineering, Technical Report TR02-10-07, 2002.
[Jea99] : Jean Pierre Tual : « MASSC, A Generic Architecture for Multiapplication Smart Cards », IEEE Micro, September-October 1999..
[Joh04] : D. Johnson, D.A. Maltz, Y. Hu : « The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) », Internet Draft, 2004.
http: //www.draft-ietf-manetdsr-10.txt.
[Kar02] : T. Karygiannis, L. Owens : « Wireless Network Security, 802.11, Bluetooth and Handheld Devices», NIST Publication, p. 800(48), 2002.
[Ken06] : Kenaza Tayeb : « Détection d'intrusion coopérative basée sur la fusion de données», Thèse de magister, Institut National de formation en Informatique INI, 2006.
[Kha01] : N. Khan, P. Basu, T.D.C. Little : « A mobility Based Metric for Clustering in Mobile and Ad Hoc Networks», In Proceedings of International Conference of Distributed Computing Systems Workshop (ICDCSW’01), Arizona, USA, 2001.
[Lar98] : Larson, N. Hedman : « Routing Protocols in Wireless Ad-hoc Networks –
A Simulation Study », Mater’s Thesis, Lulea University of Technology Stockholm, 1998.
[Leh09] : LEHSAINI Mohamed : «Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique », Thèse de Doctorat, Université A.B Tlemcen, 2009.
[Lin97] : C. Lin, M. Gerla : « Adaptive Clustering for Mobile Wireless Networks», IEEE Journal on Selected Areas in Communications, 1997.
[Mat07] : Mathieu Baudet : « Sécurité des protocoles cryptographiques : aspects logiques et calculatoires », Thèse de Doctorat, Ecole Normale Supérieure de Cachan, 2007.

[Meh07]:M.Mehdi, A.Anou, S.Zair, M.Bensebti, M.Djebari « La Sécurité dans les Réseaux Ad Hoc», 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, Tunisia, 2007.

[Nav04] : W. Navidi, T. Camp : « Stationary distributions for the random waypoint mobility model », IEEE Transactions on Mobile Computing, 2004.

[Noc02] : F. Nocetti, G. Chen, J.S. Gonzalez, T. Stojmenovic : « Connectivity-based K_hop clustering in wireless networks », in Proceeding of the 35th international Conference on System Sciences(HTCSS-35) , Hawaii, 2002.

[Pap02] : P. Papadimitratos, Z.J. Haas : « Secure Routing For Mobile Ad Hoc Networks», SCS Communication Networks and Distributed Systems Modelling and Simulation Conference, San Antonio, Etats Unis, 2002.

[Par01] : V. Park, S. Corson : « Temporally-Ordered Routing Algorithm (TORA) Version 1», Internet Draft, 20 July 2001.

[Per94] : C. Perkins, P. Bhagwat : « Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers », 1994.

[Per01] : C.E. Perkins, E.M. Royer, S.R. Das : « Ad hoc On-Demand Distance Vector (AODV) Routing», Internet Draft, 2001.

[Per02] : A. Perrig, R. Canetti, J.D. Tygar, D. Song : « Efficient Authentication and Signing Multicasts Streams over Lossy Channels», IEEE Symposium on Security and Privacy, 2002.

[Phi06] : Philippe Guillot : « Introduction `a la cryptographie », 3ième Ecole Informatique de Printemps – EIP'06 Sécurité Informatique : Tendances et Applications, INI, Alger, 2006.

[Pie04] : Pietro Michiardi : « Mécanismes de sécurité et de coopération entre nœuds d'un réseau mobile ad hoc», thèse de doctorat, école nationale supérieure des télécommunications, Paris, 2004.

[Put04] : R. Puttini, J-M. Percher, L. Mé, O. Camp, B. Jouga, P. Albers : « Un système de détection d'intrusions distribué pour réseaux ad hoc », RSTI – TSI, Sécurité informatique, 2004.

[Qin03] : X. Qin, H. Li, Z. Chen : « Secure Routing in Wired Networks and Wireless Ad Hoc Networks», Univ. of Kentucky, Department of Computer Science, Term-paper, 2003.

[Riv78] : R. L. Rivest, A. Shamir, and L. Adleman : «A method for obtaining digital signatures and public-key cryptosystems», Communication of the ACM, 1978.

[Ron09] : C. Rong, E. Çayirci : « Security inWireless Ad Hoc and Sensor Networks», editions Wiley, 2009.

[San02] : K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, E. Belding-Royer : « A Secure Routing Protocol for Ad Hoc Networks», in Proc. of IEEE International Conference on Network Protocols (ICNP), 2002.

[Sch02] : G. Schäfer : « Research Challenge in Security for Next Generation Mobile Networks», Position Papers PAMPAS '02 - Workshop on Requirements for Mobile Privacy & Security, 2002.

[Sid02] : A. Siddiqui, R. Prakash : « Effect of availability factor threshold and clustering gap on performance of clustering mechanisms for multi-cluster mobile Ad hoc networks », IEEE international Conference on Communications, ICC 2002.

[Siv04] : S. Sivavakeesar, G. Pavlou, C. Bohoris, A. Liotta : « Effective Management through Prediction-Based-Clustering Approach in the Next-Generation Ad hoc Networks », in the Proceeding of the IEEE international Conference on Communications (ICC'04), France, 2004.

[Sta99] : F. Stajano, R. Anderson : « The Resurrecting Duckling: Security Issues for Ad-hocWireless Networks », 7th International Workshop on Security Protocols, 1999.

[The06] : F. Theoleyre: « Une auto-organisation et ses applications pour les réseaux Ad hoc et hybrides», Thèse de doctorat, Institut national des sciences appliquées de Lyon, 2006.

[Tur02] : D. Turgut, M. Chatterjee, K. Das : « WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks », Journal of Cluster Computing, 2002.

[Uri00] : P. Urien, M. Loutrel : « La carte à puce EAP, un passeport pour la sécurité des réseaux émergents Wi-Fi »,

[Uri02] : P.Urien, M.Loutrel, K.Lu : « Introducing Smartcard in Wireless LAN Security», 10th International Conference on Telecommunication Systems, Monterey, California. 2002.

[Wes00] : A. Wespi, H. Debar, M. Dacier : « A revised taxonomy for intrusion detection systems», Annales des télécommunications. July–August 2000.

[Zho99] : L. Zhou, Z. J. Haas, «Securing Ad Hoc Networks», IEEE Networks, Volume 13, Issue 6, 1999.

[Zim96] : P. Zimmermann : « The Official PGP User's Guide. », MIT Press, 1995.